*Statistical shape analysis in a Bayesian framework; The geometric classification of fluvial sand bodies.*

THOMAI TSIFTSI

# Statistical shape analysis in a Bayesian framework.

## The geometric classification of fluvial sand bodies

**Thomai Tsiftsi**

A Thesis presented for the degree of
Doctor of Philosophy


Statistics and Probability Group
Department of Mathematical Sciences
University of Durham
England

September 2015

# *Dedicated to*

the beloved memory of Σωτήρης,

το πρόδηλο εγώ του άδηλου απείρου,

the man who never doubted my mathematical abilities and always thought that the way I do maths is just like myself: a bit chaotic.

# *Also to*

my family for always being there

and

James, my very own Atticus.

**19 November 1957**

Dear Monsieur Germain,

I let the commotion around me these days subside a bit before speaking to you from the bottom of my heart. I have just been given far too great an honour, one I neither sought nor solicited. But when I heard the news, my first thought, after my mother, was of you. Without you, without the affectionate hand you extended to the small poor child that I was, without your teaching and example, none of all this would have happened. I don't make too much of this sort of honour. But at least it gives me the opportunity to tell you what you have been and still are for me, and to assure you that your efforts, your work, and the generous heart you put into it still live in one of your little schoolboys who, despite the years, has never stopped being your grateful pupil. I embrace you with all my heart.

***Albert Camus***

# Statistical shape analysis in a Bayesian framework.

# The geometric classification of fluvial sand bodies

**Thomai Tsiftsi**

Submitted for the degree of Doctor of Philosophy
September 2015

## Abstract

We present a novel shape classification method which is embedded in the Bayesian paradigm. We focus on the statistical classification of planar shapes by using methods which replace some previous approximate results by analytic calculations in a closed form. This gives rise to a new Bayesian shape classification algorithm and we evaluate its efficiency and efficacy on available shape databases. In addition we apply our results to the statistical classification of geological sand bodies. We suggest that our proposed classification method, that utilises the unique geometrical information of the sand bodies, is more substantial and can replace ad-hoc and simplistic methods that have been used in the past. Finally, we conclude this work by extending the proposed classification algorithm for shapes in three-dimensions.

# Declaration

The work in this thesis is based on research carried out at the Statistics and Probability Group, the Department of Mathematical Sciences, England. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text. The work that makes up chapters 2, 3 and 5 was carried out in collaboration with Dr. Ian Jermyn and Dr. Jochen Einbeck. For chapter 4, I would like to thank Dr. Jochen Einbeck and Dr. James Edwards for very fruitful discussions and advice.

**Ιθάκη**

Σα βγείς στον πηγαιμό για την Ιθάκη,
να εύχεσαι νάναι μακρύς ο δρόμος,
γεμάτος περιπέτειες, γεμάτος γνώσεις.
Τους Λαιστρυγόνας και τους Κύκλωπας,
τον θυμωμένο Ποσειδώνα μη φοβάσαι,
τέτοια στον δρόμο σου ποτέ σου δεν θα βρείς,
αν μέν΄ η σκέψις σου υψηλή, αν εκλεκτή
συγκίνησις το πνεύμα και το σώμα σου αγγίζει.
Τους Λαιστρυγόνας και τους Κύκλωπας,
τον άγριο Ποσειδώνα δεν θα συναντήσεις,
αν δεν τους κουβανείς μες στην ψυχή σου,
αν η ψυχή σου δεν τους στήνει εμπρός σου.

Να εύχεσαι νάναι μακρύς ο δρόμος.
Πολλά τα καλοκαιρινά πρωιά να είναι
που με τι ευχαρίστησι, με τι χαρά
θα μπαίνεις σε λιμένας πρωτοειδωμένους
·
να σταματήσεις σ΄ εμπορεία Φοινικικά,
και τες καλές πραγμάτειες ν΄ αποκτήσεις,
σεντέφια και κοράλλια, κεχριμπάρια κ΄ έβενους,
και ηδονικά μυρωδικά κάθε λογής,
όσο μπορείς πιο άφθονα ηδονικά μυρωδικά·

**Ithaka**

As you set out for Ithaka
hope that your journey is a long one,
full of adventure, full of discovery.
Laistrygonians and Cyclops,
angry Poseidon - don't be afraid of them:
you'll never find things like that on your way
as long as you keep your thoughts raised high,
as long as a rare sensation
stirs your spirit and your body.
Laistrygonians and Cyclops,
wild Poseidon- you won't encounter them
unless you bring them along inside your soul,
unless your soul sets them up in front of you.

Hope that your journey is a long one.
May there be many a summer morning when,
with what pleasure, what joy,
you come into harbours seen for the first time;
may you stop at Phoenician trading stations
to buy fine things,
mother of pearl and coral, amber and ebony,

σε πόλεις Αιγυπτιακές πολλές να πας,
να μάθεις και να μάθεις απ΄ τους σπουδασμένους.

Πάντα στον νου σου νάχεις την Ιθάκη.
Το φθάσιμον εκεί είν΄ ο προορισμός σου.
Αλλά μη βιάζεις το ταξείδι διόλου.
Καλλίτερα χρόνια πολλά να διαρκέσει·
και γέρος πια ν΄ αράξεις στο νησί,
πλούσιος με όσα κέρδισες στον δρόμο,
μη προσδοκώντας πλούτη να σε δώσει η Ιθάκη.

Η Ιθάκη σ΄ έδωσε τ΄ ωραίο ταξείδι.
Χωρίς αυτήν δεν θάβγαινες στον δρόμο.
Άλλα δεν έχει να σε δώσει πια.

Κι αν πτωχική την βρείς, η Ιθάκη δεν σε γέλασε.
Έτσι σοφός που έγινες, με τόση πείρα,
ήδη θα το κατάλαβες η Ιθάκες τι σημαίνουν.

sensual perfume of every kind -
as many sensual perfumes as you can;
and may you visit many Egyptian cities
to learn and learn again from their scholars.
Keep Ithaka always in your mind.
Arriving there is what you are destined for.
But do not hurry the journey at all.
Better if it lasts for years,
so you are old by the time you reach the island,
wealthy with all you have gained on the way,
not expecting Ithaka to make you rich.

Ithaka gave you the marvellous journey.
Without her you would not have set out.
She has nothing left to give you now.

And if you find her poor, Ithaka won't have fooled you.
Wise as you will have become, so full of experience,
you will have understood by then what these Ithakas mean.

*Κωνσταντίνος Καβάφης- 1911*

# Acknowledgements

"As you set out for Ithaka, hope that your journey is a long one [...], arriving there is what you are destined for." This long, difficult journey, full of obstacles, would have never been accomplished and hence I would never have arrived at my destination without the assistance, the love, the affection and the support of people that have "paddled" in these angry seas at my side. The least I can do to return the gratitude is to thank them all from the bottom of my heart.

It doesn't feel like it has been a long time since I was a secondary school student; my classics teacher κ. Μαρία Εξηντάρη said something, quoting Alexander the Great, that has been imprinted in my mind and I have been carrying in my heart ever since: Στους γονείς μου οφείλω το ζην και στον δάσκαλό μου το εύ ζην (I am indebted to my parents for living, but to my teacher for living well). Having this in mind, allow me to pay dues to these people first.

Firstly, I must thank my first supervisor Dr. Ian Jermyn. Your help, your ideas and the stir you gave me during the past four years have been more than useful. You introduced me to the wonderful world of geometrical statistics and you woke up this little physicist I kept well hidden inside me. For that I thank you.

Secondly, I would like to thank my second supervisor Dr. Jochen Einbeck. Jochen, without you I do not know how I could cope in this academic world. I could write pages and pages about all the wonderful things you have done for me and sometimes I feel I didn't even deserve them. Thank you for never saying no to me however little or big was what I asked for, for supporting me always as no one ever did. But mostly Jochen I want to thank you for being the one and only person who has taught me the academic ethos. I couldn't have asked for a better role model and I hope that one day I will be like you: an inspiration to my own

students.

Another person that I am grateful to is Yiannis Platis. I still remember that first visit I paid you in your office. Every time I have a student visiting me, I feel that I am returning the good that you have done for me. You were the person who truly believed that I can make it even if I only count on my own strength. Your guidance and your positive stance really made me reconsider lots of things that were happening in a difficult time of my life. I hope our scientific paths cross in the future so that I can really show you what I am made of. I will never forget what you have done for me. Γιάννη βαθιά σε ευχαριστώ!

I am also grateful to people who have been helpful and have stood next to me for various academic reasons. I would like to thank Dr. Eleanor Loughlin who has helped me make my dream to make maths accessible come true by instigating "Mathlab." Thank you Eleanor for believing in me, for being so nice and supportive for giving me space and time to learn and put in practice everything that I have learnt from you. I would also like to thank Dr. Lowry McComb and Dr. Robert Matthew for their assistance and advice throughout my studies and for making me evaluate how important the role of a teacher is. It has certainly been a very "constructive" experience!

My family has shown me more than endless love and support, not only throughout these past four years but ever since I was little. I am deeply thankful to my Mum and Dad who have done more than they could ever do to contribute to my education. They never put a limit, they never declined whatever would add to my advance as a person or as a scientist and there is nothing more that I could have asked and they have not already done. Thank you for always being there and for allowing me to open my wings and fly "away" but not away from you. I cannot apologise enough for not finding time to communicate with you or for the times that I just didn't have the patience to deal with things that I didn't find important at the time. I am sure that once again you understand and you forgive me for my stubbornness. I love you both and I will never forget what you have done for me and I want you to know that I really appreciate the fact that I would never be here without your help and patience. I hope to be able one day to return the love you have shown me and

I wish I could really show my appreciation other than in these lines - σας αγαπώ! I hope I made you proud and continue making you proud throughout my life! I would also like to thank my brother Τάσος for giving a nice twist to our boredom during his two months' visit to Durham. Thank you for teaching your squash buddy Greek and thank you for keeping us company!

I am grateful from the bottom of my heart to my sister Καρολίνα and my brother Χρίστος for the unconditional love and help they have shown me. I want to say an extra thank you to my sister who, years ago, has set the path and has set high targets so that I can surpass myself and reach them. You have been a role model and I certainly wanted to follow in your footsteps - look where I am now! Καρολίνα και Χρίστο I don't know what I have done to deserve everything you have done for me in so many ways and so many times. Thank you for all the times you have given me everything that I haven't even asked for! I wouldn't have done it without you and I wouldn't have reached the "heights" I have reached without your continuous and unlimited moral and psychological support. Thank you for trusting me, for believing that I can actually make it and that I can actually reach the top! However, I would mostly like to thank you for offering me the best present of my life, my Δάφνη who has put a smile on my face and has been the sun to shine in the darkness and greyness that everyday life has offered me in Durham. Σας αγαπώ πολύ and I hope I repay you one day when Δάφνη becomes a great mathematician just like her aunt!

My thanks also go to Julie, Paul, Rob, Granddad John and Grandma Teresa who have been my second family here in the UK and have acted as one the times that I needed it. I have been blessed to come to know you all and thank you so very much for being more than nice to me. I am sorry I haven't been able to repay in the slightest the love and affection you have given me all these years. I can reassure you that I loved the UK through you and because of you and I hope I made you love Greece a tiny bit more!

I am very lucky to have met so many great and intelligent people here in Durham that made the gloominess of my Ph.D. life brighter and happier. Thank you to my friend and office mate Lewis Paton who has been extremely patient with me and has answered every single stupid question I had about statistics or the way Latex works!

My thanks go to my friend Alex Cockburn who has given me lots of moments of laughter. I hope your attempts to teach me English didn't feel as hard as "flogging a dead horse." Special thanks go to Michela who has been a friend, has laughed with my silly jokes and taught me how to swim! Thank you for the carefree moments in the swimming pool! Also, thanks go to Michelle with whom I have had a great time at the Vic on Fridays and who has been the most loyal fan of my seminars. Thank you for all those times we tried to pretend we could stretch enough for our yoga lessons. I will miss you lots! I would like to thank Dr. Gabor Kiss who has been the most pleasant company in the department and has taught me how to keep my head high when facing struggles in my life. I am very grateful to my friend Daniel Bonetti who has continuously made me smile and I have shared with him one of the most amazing experiences in my life: a crazy week in Göttingen! I enjoyed teaching you "estatistics" so much! Thank you Daniel for always being there for me and for sharing my happiness all the time. You are an amazing person and thank you for being my friend. I would also like to thank you for sharing your code with me - it has been extremely helpful! Special thanks to Jessica Turner who, although I met briefly the past year, has been extremely positive and supportive for my final goal. Thank you for all the coffees you drank with me and the advice you gave me on various matters. I will miss you! I would like to thank my cousin Έλενα for keeping me company on difficult days with her chatting and her carefree discussions (especially about our dieting!), taking my mind away from things that I was struggling with. Lastly, I would like to thank my two childhood friends, Κατερίνα και Μαρίνα, the former because she showed me that when one sets goals one has to find the strength to achieve them and the latter for showing me how much perseverance one needs to follow their dreams! Thank you both for thinking that I am your smart friend! Also thanks go to Ανέστης for proofreading this thesis.

It is also a nice surprise to meet people out of the blue and for these people to stay in your life where the memories with them are carved in your mind and make an impact like no one else. I am referring to Daniele Galloni and Catherine Banner who have been an oasis of comfortness and laughter during the dull weeks that have gone by during our short stay in Durham. You have been nothing else

but encouraging and inspiring and filled me with hope that I can actually cross the finish line. You have understood all aspects of my Ph.D. life and it has been really helpful to be able to talk it through with you. Thank you for all the nights of pool and cider at the Queen's Head, the cooking at your house, Daniele's crazy stories, Cat's interesting facts of an author's life and her contribution to our sanity since she was kept away from all this madness. I have been very lucky to meet you, I love you both and I will miss you so much!

Finally I would like to thank James; your support and your guidance make everything that I do possible. I would not have completed this work without you being constantly by my side answering my stupid questions, helping me with my coding, listening to my complaints about pretty much everything. This Ph.D. and this thesis do not belong to me but to us because it has been done with your help, with your guidance and supervision when I had been abandoned by everyone. I don't think I can ever give you enough credit for everything that you have done for me. You have no idea how much I admire you and what an inspiration you are to me as a person, as a researcher and as a mathematician and I can certainly only be better next to you. I do not know how to thank you enough for everything! Thank you so very much for investing not 100 but 1000% in me; thank you for pushing me harder and harder when I felt lonely and weary. Thank you for showing me what it means to remain faithful to your goals and your dreams and thank you for being my pride and joy. Not many people have the chance in life to be next to such amazingly beautiful people and my life's path is different just because I was blessed to have the opportunity to walk it by your side. Without your love I would be a single point particle without any contact interactions, unable to be integrated over a fixed gauged slice and full of Weyl anomalies in the vastness of this universe. I want you to remember that your love is an "ever-fixed mark." Αλλά εγώ, σ΄ αγαπώ κι αγαπώντας σε, σε περιέχω, σε έχω αφού είμαι, είμαι απο σένα και μαζί σου κι όπου κι αν είμαι έρχεσαι. Είμαστε στο παντού και στο πάντα τώρα που σ΄ αγάπησα κι η αγάπη μου μας κάνει αδιαίρετους. I believe that no words can capture the pureness of your soul and mind and I thank you for, like my very own Atticus, continually reminding me what is truly important. But above all, thank you for helping me make my

dream come true and arrive to my own Ithaka: to become a better mathematician - I will be eternally grateful.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

*Whence and what art thou, execrable shape?*
John Milton (1667), *Paradise Lost II, 631*

Our everyday lives and most of our daily activities include interaction with objects which we are called to recognise through our visual system. Through the evolution of time, human nature and the advance of our visual systems, we are able to recognise objects through their boundaries and their variations. Indeed, it is the efficacy of our visual systems that led us to recognise complicated geometries and invent the definition of "shape." The word shape has many meanings; sometimes its meaning is embedded in the word that describes the object, for example by hearing the word wheel one can immediately picture a circular object with concentric spokes. Shape is an important feature of objects we see and one could argue that humans are the best object and shape identifiers since they are trained to interact with shapes in nature from their very first days of life. This intellectual ability allows us to also recognise objects that are connected through complicated mathematical operations. For instance, were we given two identical objects differing only by a rigid rotation we would recognise them as the same. The human eye is trained to identify objects that are related under such transformations. It is this powerful ability that led humans to make progress in the field of collection, processing, interpretation and analysis of geometrical information and in particular geometrical information in conjunction with the concept of shapes.

For the past 20 years a very powerful subject has been established in the area of mathematics and statistics: shape analysis. The technological bloom and the combination of computer science, mathematics, physics and statistics has helped make shape analysis become an integral part of many branches of science such as computer vision, pattern recognition, shape representation and shape classification. This has enabled other sciences to also embrace and adopt these developments so that shape analysis can now be applied from biology to archaeology and many more. Examples of such developments can be found in medical image analysis [1, 2], bioinformatics [3], morphometrics [4, 5], text recognition [6, 7], archaeology [8, 9], anthropology [10] and others.

However, one might ask how do we describe all this using mathematical formalisms? Before we proceed and discuss shapes more abstractly, it is imperative to give the mathematical definition of shape. We follow Kendall [11], one of the pioneers of statistical shape analysis:

"We here define *shape* informally to be what is left when the differences which can be attributed to translations, rotations, and dilatations have been quotiented out."

Following Kendall's definition in a mathematical setting, we expect shapes to be invariant under the transformations of rotations, scale and translations. This means that any two realisations of shapes that can be generated from one another by applying a series of rotations, translations and scalings to be regarded as the same shape.

Statistical shape analysis is a powerful tool that can be used to solve many problems such as the shape classification of objects. Shape classification addresses the following question: given objects that come from a priori known categories, how can we classify them in their own class? How can we automate the procedure of the classification of objects into pre-determined classes? Can we give a confidence level to quantify the probability our classification is correct? This is where shape classification starts becoming important.

An important aspect of classification is the representation of the observed shapes. Although most mathematical methods that have been applied to shape classification in the past were based on landmark points (see section [1.3]), these have proved to be limited. In recent years great progress has been made towards the study of continuous planar shapes. In this work we focus on the statistical classification of continuous planar shapes and in particular on their geometric statistical analysis. To this end we will use a novel shape classification method based on the underlying geometry of the shapes. To capture this underlying geometry we use the model presented in [12] and extend it and this will be the core of this thesis. One of our goals is to model shape variability within and between classes of shapes. Like with any other population, shape populations show variability and we use probability distributions to model it; these models can be later used for the classification of shapes. Having described this variability, we can then obtain samples from shape populations and classify them in one of the pre-determined classes, assigning a confidence level for each class. Our approach is to perform Maximum a Posteriori (MAP) determination of each class given the data. In this thesis we follow the work of Srivastava and Jermyn [12] and we extend the way that the MAP is performed in a more efficient way.

To test our methods and our model, we use examples from the KIMIA database [13] and an alphabet database that we created ourselves. The Kimia database is comprised of binary images of several types such as animals and objects whereas the alphabet database is comprised of binary letters of the latin alphabet in 6 different fonts. Another example where we apply the methods we developed is on geological data, namely sand bodies. The classification of geological sand bodies is an important problem in geology, however current classification methods are characterised as simplistic and ad-hoc. There has been a need for more efficient, scientific and statistical classification methods that capture the geometry of the sand bodies since it is the most important feature that determine their nature, class and oil capacity.

Furthermore, we extend this work to the learning of the parameters that describe the studied data shapes. By using the learned parameters, we can then

classify new and previously unobserved data with confidence. In addition, we try to observe and identify the emergence of new, previously unknown classes of shapes for a given data set based on some measure of similarity. With the help of the Expectation-Maximisation algorithm we identify clusters of shapes whose properties can be inferred and used for future classification of new data.

The structure of this thesis is as follows: the current chapter presents a sample of the previous work done in shape analysis. We discuss the advances of three areas of shape analysis: shape pre-processing, shape transformations and shape classification. We focus on certain aspects of these areas which have been of use in our work, namely shape acquisition, shape representation and classification.

In the second chapter we introduce the work of Srivastava and Jermyn [12] and we discuss the methods which they used for the classification of observed shapes. We refer to the statistical framework they established and how shape spaces are utilised for the results of the classification methods they used. We then start to extend their work, presenting the models used for the description of the parameters capturing the shape variability. One of the important features of the second chapter is the presentation of our work in which we replace some of Srivastava and Jermyn's approximating methods by their analytic equivalents which give rise to our proposed classification algorithm of continuous, planar shapes.

In the third chapter we evaluate the effectiveness and the accuracy of the proposed algorithm. We present the confidence and success results of the algorithm as examined in experiments conducted with the help of the Kimia and the alphabet database. We extend the experiments on evaluating the confidence and classification results on the geological sand body database which we had to simulate in absence of any real geological data. For this part of the third chapter we evaluate methods of supervised learning applied in this case in anticipation to learn the parameters of the models we utilise.

The fourth chapter is an attempt to compare our suggested method to classification methods using clustering. For this comparison, we use the Kimia and alphabet

databases from which we extract properties that describe each class of shapes. We form feature vectors that are constructed from properties which, we assume, describe each of the classes. By using the Expectation-Maximisation (EM) algorithm we try to infer the number of existing clusters in the data and the statistical properties that capture the variability of each class. We assume that each of the classes can be described by a multidimensional Gaussian so that the distribution of the data is thus explained as a mixture of multidimensional Gaussians. Based on these properties we classify new, unobserved data in the classes as they were decided by the EM and compare its accuracy to the results acquired from our proposed algorithm. The purpose of this experiment is to compare the classification results when using a classification method that employs the geometrical information of the shape to the classification method that makes use of less detailed information. The last part of the fourth chapter presents an attempt at an adapted version of the EM algorithm in the case of the sand body database where we replace the Gaussian mixtures by mixtures that use our proposed likelihood and algorithm from chapter [2].

Up to this point we will have focussed on the study of two dimensional planar shapes. The fifth chapter is an extension of the work presented in the second chapter for the case of two dimensional surfaces. Some of the methods presented in chapter [2] are extended for surfaces so that the algorithm can be extended to three dimensions. These methods include the integration of three dimensional translations and rotations. However, the attempted methods didn't produce fruitful results and left open questions for future consideration.

The sixth and final chapter presents a summary and discussion of the results presented in the previous chapters, a review of the overall attempt and method of shape classification with a set of open questions posed.

## 1.1 Previous work on shape analysis

The heuristic geometrical definition of shape proposed by Kendal [11] is "the geometrical information that remains when the differences which can be attributed to translations, rotations and dilatations have been quotiented out of an object." Thus, we expect shapes to be invariant under the transformations of rotations, scale and translations. The definition of shape in terms of invariant quantities can be found extensively in the literature [14, 15]. However there is criticism [16] that these type of transformations do not specify what an object or a data set is. It has also been criticised that shapes that have undergone other types of transformations (e.g. affine transformations), or shapes that could still be recognised as equivalent by humans, are not explained by similarity transformations. An alternative definition of shape is given by Costa [16]: "a shape is a single visual entity or object." The concept of single, whole, united is also used to describe a shape. Adding to the above the notion of connectivity, we have a formal definition: a shape is any connected set of points. This definition includes both continuous and discrete shapes, however it does not reflect the geometric underpinning of shapes nor the invariance under similarity transformations. In this thesis we will follow Kendall's definition since similarity transformations and geometric invariance play a big role in our study of shapes.

Shape analysis aims to explain, describe and predict the shapes of objects and is used as a tool in many sciences. Applications can be found in medical image analysis [1, 2], bioinformatics [3], morphometrics [4, 5], text recognition [6, 7], archaeology [8, 9], anthropology [10] and others. Shape analysis is divided into three main classes: shape preprocessing, shape transformations and shape classification. Figure (1.1) shows these three classes and their sub-levels [16]. Although this thesis focuses on novel classification methods we will briefly refer to each of the above categories to set the ground before we describe how we use them in our work.

Although we already gave an accurate geometrical definition of shapes, for the next two sections we will consider shape to be a one dimensional line denoting the boundary of an object and work with this definition for the sake of illustration.

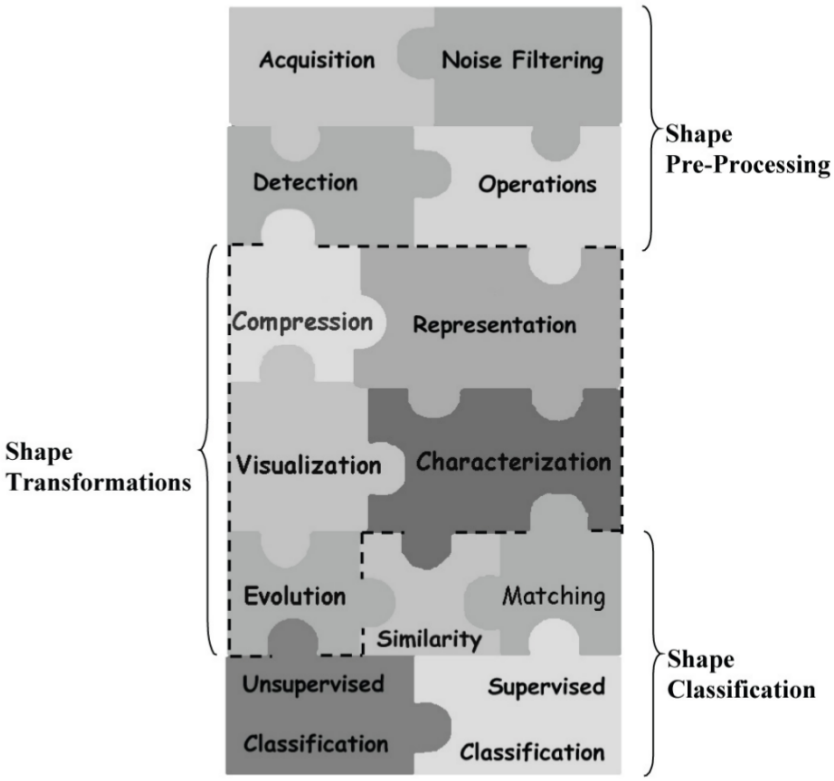



Figure 1.1: Shape analysis and its sub-categories [16]. The image is used after permission was granted by the Copyright Clearance Centre.

## 1.2 Shape preprocessing

The main goal of shape preprocessing is the acquisition and detection of the boundary of an object from a given image in the presence of noise or other objects. Usually this is the first step towards the analysis of a shape. This area of research is mostly addressed by computer science and machine vision. Shape detection and the extraction of shape contours can be done by automated image segmentation and edge detection algorithms [17]. Image segmentation techniques identify and locate the boundary of a shape by partitioning the object into smaller segments. Edge detection identifies the outline and boundary of a shape by comparing the contrast between the image and its background. There are many different approaches and many different algorithms to the image segmentation and edge detection problem which we now briefly recapitulate.

The first attempts of image segmentation were done by Attneave [18] who used spline functions to approximate the boundaries of shapes. Splines functions are piecewise polynomials which interpolate between fixed points, fulfilling certain continuity conditions [19]. A different approach was made by Wallace [20] who used polynomial functions for the approximation of the contour. Marr [21] suggested that shape edges coincide with changes in the boundary intensity and hence constitute a primal sketch of the shape in question. Marr [22] also combined his primal sketch with information such as depth for a more complete representation of the shape edges. Marr's primal sketch inspired Asada and Brady [23] for their curvature primal sketch where they represented curvature changes of the boundary rather than intensity changes. This was extended to three-dimensional shapes by Ponce and Brady [24]. Mathematical morphology came to add to the area of edge detection [25, 26] with the images being analysed based on the description of the boundaries as sets of points and their logical relationships. Other approaches to edge detection include Kirsch masks [27], wavelets based approaches [28] and Markov techniques [29]. Methods that evolved towards the edge detection of shapes also include the detection of particular features such as corners [30], curves [31, 32] and pattern analysis [33, 34].

Although the shape pre-processing category also includes noise filtering and operations, as shown in figure (1.1), we only refer to detection and acquisition since these are the only methods used in our work. The above mentioned methods, and many more, can be used for the acquisition of the boundary of the shapes with the help of edge detection algorithms. We shall use built-in edge detection algorithms in MATLAB for the purpose of extracting boundaries of some test shapes in chapter [3]. This will be sufficient since the focus of this thesis is on the classification of the resulting curves. Having obtained the contour of a shape, we can now choose the appropriate mathematical representation for it. In the next section we discuss some of these choices that have been used broadly in the literature.

## 1.3 Shape transformations

Shape transformations are the second step of shape analysis. Once the boundary of the shape is available through edge detection then valuable information can be extracted from it so that the shape can be analysed. Shape transformations can help towards the extraction of such information which can then be used for classification purposes. The main aim of shape transformations is to decide how to represent a shape appropriately and quantify its properties. It also allows us to quantify the difference between shapes which is vital for the classifications stage of the process. Since shape representation is an important subject of our work it will be the only subcategory we will turn our focus on from the category of shape transformations presented in figure 1.1. Pavlidis [35], suggests that the representation should be chosen according to either information preserving or information non-preserving techniques. Information preserving techniques allow the reconstruction of the initial shape so that different shapes have different representations [16]; non-information preserving techniques do not allow the reconstruction of the initial shape and in many cases different shapes can have the same representation. Such techniques are sometimes used for shape classification as we will see in chapter [3]. This thesis focuses on the study of planar shapes and since we will treat them as parametrised curves, there are many possibilities for the choice of the representation. We will

now present some of the most important representations and we will discuss which of them we will choose for the purposes of this thesis.

### Feature extraction

The representation of a shape can be done by feature extraction (i.e. by extracting properties that are important for the shape) and we then identify the shape by those features. A set of features that is usually used are the following: perimeter [36], area [37, 38], center of mass, major and minor axes, statistical moments [39], number of holes or even the class that the shapes belongs to. This representation falls into the non-preserving information techniques because one can represent more than one shapes with the same area and perimeter etc. Differences between the features provide a description of the differences between the shapes those features represent. We shall see an example of these techniques in chapter [4].

### Landmarks

One of the most commonly used ways for representing shapes is landmarks. Landmarks are placed on parametrised curves. These are points of correspondence for objects that match between and within populations [14, 40]. This selects a finite set of points to act as a discrete representation of the shape boundary. They play the role of the minimum adequate representation of a shape from an infinite set of points that would make the continuous version of the planar shape [41] and they are regarded as shape features. The collection of all landmarks is referred to as a configuration. Equally, a configuration can be represented by the polygons that are created by connecting the landmarks with lines, splines etc. Landmarks are distinguished between three types: anatomical, mathematical and pseudo-landmarks. In this thesis we will only discuss mathematical landmarks which are points that have been placed on the object based on some geometrical property of it. Landmarks also come in two sub-types, ordered (or labelled) and free landmarks. Labelled landmarks come with an associated label so that two shapes can be compared to one another by comparing their corresponding landmarks. Free landmarks are the ones where the order of the points is not taken into account.

Different choices of landmarks offer different representations of the same shape. Choices of landmarks can be made on different bases; for example at points of maximum curvature [42], distance from the centroid [43] or any other criterion set by the experimentalist. One simple approach is the placement of points in equal intervals around the boundary [44, 45]. Other approaches include landmarks placed on lines tangent to the boundary; their position is then chosen iteratively by sliding them forwards or backwards along the tangents and choosing the optimal position when a cost function (bending energy) is minimised [5, 46, 47]. Another approach includes the placement of points by following the centroid radii model [48]; this model places landmarks at points of high curvature. It is clear that using landmarks adds to the loss of important information from the shape. It is also obvious that a greater number of landmarks give a better approximation to the true representation of the shape.

Early attempts of shape analysis have been merely based on the representation of shapes by landmarks [14]. This approach is referred to as **classical shape analysis**. The inception of classical shape analysis was made by Thompson [49] but it was established formally by Kendall [11], Bookstein [50], Dryden and Mardia [14], and Kent and Mardia [51]. This work offered the basis for the modern theory of shape by borrowing ideas from differential geometry and gave the first definition of the idea of the **shape space**. The common feature in their work is that all shapes are represented by a certain number of landmarks in $\mathbb{R}^2$. Following Kendall's definition given at the beginning of this chapter, sets of landmarks that are connected through similarity transformations represent the same shape. Quotienting out such transformations creates the shape space for the representation of each class of shapes. This quotient space is a shape manifold; imposing a metric on the manifold allows the comparison and quantification of differences between shapes (shape spaces will be discussed in section [1.5]).

Landmark based techniques have been extended to many applications and used in several contexts [14, 52] for the accurate representation of shapes. Another big breakthrough in the representation of shapes is the "snakes or active contour mod-

els" representation [53]. Snakes are "energy minimising splines guided by external constraint forces that pull towards the object's contour." However, snakes are not that flexible since some knowledge of the contour is required and they are only useful in situations where the shape of the object is rather amorphous. Snakes were then extended to the "active shape models or smart snakes" [54, 55, 56, 57, 58, 59]; an application of them is presented in [60]. Active shape models (ASM) represent shapes by sets of landmarks that are connected by lines and polygons. ASM are learnt from the training phase of images that have been annotated by an experimentalist. They then use principal component analysis [61, 62] on the selected landmarks to capture the shape variation in the observed samples. The observed samples of shapes are aligned and then correspondences between equivalent landmarks are established. This allows the calculation of the mean position and variation of each of the landmarks. The main breakthrough of this work is that the model reflects the patterns of the shapes and the variations of each class however it ignores the non-linear nature of shape spaces. Similar work was produced by Kervrann and Heinz [57] who used the equivalent unsupervised approach to learn the deformations of two dimensional polygonal objects. The three dimensional equivalent was studied by Pentland [59, 63] who proposed the "finite element model".

### Deformable templates

A different shape representation to the other landmark based approaches is that of deformable templates. Deformable templates represent classes of shapes which have been generated by an idealised shape which takes the place of the representative template of the class. Given a template and an image we can then find the optimum map between them [64]. Deformation templates have been studied extensively [65, 66, 67, 68, 69, 70]. In this approach shapes are elements of infinite dimensional, differentiable manifolds and the differences between the shapes are modelled as action of Lie groups on the manifolds [71]. A drawback of the deformable template theory is the need to consider the action of diffeomorphisms in $\mathbb{R}^2$ and $\mathbb{R}^3$. This is computationally expensive which makes the approach difficult to use when the shape database is large. In addition, this framework doesn't provide an appropriate

distance for comparing shapes in images [72].

### Continuous planar curves

The classical shape framework set out by Kendall and Bookstein for the representation of curves discussed above has contributed largely to the development of the modern theory of shape. Although this approach has proven to be limited, landmark representations have been extensively used in the literature in cases where landmarks are easily and readily available [14, 73, 74]. However, the usage of landmarks as shape descriptors for general purposes is a great limitation to statistical shape analysis since landmarks are a subjective way of defining shapes. Furthermore, automating the choice of landmarks is a hard and unrigorous procedure; there are many works that study the problem of automatic landmark choice [75, 76] however the resulting shapes remain dependent on the landmarks. The same drawback is present in the active shape models too. In many cases, this approach is biased and although the simplicity of the method is luring, many times it leads to undersampled or coarsely sampled contours, unsatisfactory interpolations and inaccurate results since the number of landmarks and their location can totally change the polygonal shape.

The biggest drawback of the above mentioned representation methods is that they do not take into account the continuous nature of the boundary of the shape. They rather represent it by a discretised version of the curve which contributes to the loss of important information that could be effectively used for shape analysis. This created the need for a concrete statistical framework for the theory of shapes that can also be used for efficient computational applications. The demand for more efficient techniques of shape analysis has initiated the research of the modern theory of shapes. Thus, modern shape analysis is focusing on the study of shapes as continuous curves and surfaces by following some of Kendall's and Bookstein's initial framework. There has been a shift in paradigm so there is now extensive literature that treats shapes as continuous curves (rather than discretised versions of them). In order to represent continuous curves we need a way of parameterising their embedding in $\mathbb{R}^2$ or $\mathbb{R}^3$. Towards this end, one of the early attempts at the

representation of shapes by curves was done by Raudseps [77]; Raudseps presented some initial ideas on the representation of shapes by angle functions [78] i.e. that each point on a shape can be parametrised by the angle of the tangent of the contour with reference to the x axis. Zahn and Roskies [79] and Bennet and McDonald [80] compared angle function versus arc length representations. Arkin et al. [81] used this representation for the comparison of polygonal shapes. Zahn and Rooskies provided a formal extension of Raudseps' work by representing shapes by the Fourier coefficients of the angle functions. These works were the first steps towards the representation of shapes as planar closed curves.

A common theme in the representation of shapes as closed curves is that of the **shape space**. Shape spaces will be discussed in detail in section [1.5], however we will give a brief definition to ease the reader's understanding in the context of closed planar curves. Following Kendall's definition, the study of shapes is done by establishing equivalences between them with respect to similarity transformations i.e. shapes coming from the same class are equivalent up to rotations, scalings and translations. In all the representations that will be discussed, the construction of shape spaces is done in two steps. After the representation is chosen we are led to the pre-shape space. Then, elements of the pre-shape space that belong to the same orbits of shape similarity transformations are regarded as equivalent. The resulting quotient space forms the shape space and it is the space of the orbits under the group actions of the similarity transformations. If the pre-shape space is a manifold then the shape space inherits the manifold structure and becomes a manifold of orbits (orbifold). This provides us with a natural way to compare the curves. We do this by imposing a metric of our choice on the manifold i.e. a measure of the distances between shapes or orbits of shapes, which is then interpreted as specifying the similarities and differences between shapes.

Being familiar with the definition of a shape space, we can continue with the descriptions of continuous curves of the available shapes. This work was initiated by Younes [82, 83], who defined shape spaces of continuous planar curves and imposed Riemannian metrics on this spaces measuring the deformations between curves. In

the same spirit, Klassen et al. [84] studied the shapes of continuous, closed curves in $\mathbb{R}^2$ parametrised by their arc-length without the need for landmarks or any of the previous frameworks. They were the first to compute geodesics between closed curves in a diffeomorphism invariant way. This is a very important result since it treats different parametrisations of the boundary of the shape as the same curve. Klasen et al. removed similarity transformations from the space of closed planar curves, imposed a Riemannian metric and took advantage of the geometry to perform inferences and solve optimisation problems. In particular, this work suggested that shape representation can be done via angle functions (as Zahn [79] did) or curvature functions i.e. functions that express the curvature as a function of the contour's arc length. Srivastava et al. [85], advanced Klassen et al.'s ideas and developed the appropriate tools for shape analysis. They applied and tested these ideas on databases such as the Surrey database [86]. However, their work doesn't take into account the elasticity of shapes, resulting in non-optimal shape correspondences. An extension of this work was studied in [87] where the variational methods used were faster and more numerically stable.

There have been several studies on the choices of metrics utilised in the spaces of closed planar curves for the purpose of comparing shapes. Some studies include Minchor and Mumford [88], Menucci and Yezzi [89] studying choices of different Riemannian metrics on the space of regular smooth curves and Sundaramoorthi et al. [90] suggesting a novel metric on the space of closed planar curves. Mumford and Sharon [91] studied metric spaces using conformal mappings of two dimensional space.

Mio et al. [92, 93] represented shapes as elastic strings that can be stretched and bent. They were the first to construct shape spaces with the elastic metric that incorporated the elastic properties of the shapes. Due to these properties, they quantified the amount of stretching shapes need to deform into one another. A drawback of the method is that the algorithms used for the calculation of the geodesics were cumbersome [72]. In the same spirit, Shah [94] derived geodesics by using a collection of different elastic metrics and representations of curves. Joshi et

al. [95, 96] proposed a novel representation of continuous closed curves in $\mathbb{R}^n$ by combining the elastic shape metric and path-straightening methods. In Srivastava et al. [97], this was presented in more detail. They presented a shape representation for analysing shapes motivated by the Fisher-Rao metric which is used in the space of probability densities to impose Riemannian structure. They introduced the square root velocity (SRV) representation which produces a simpler Euclidean structure so that geodesics, statistics and distances are simpler to calculate. This work is similar to the work of Younes [82, 98] but more complete since it applies to curves in arbitrary dimension. They have also applied and demonstrated the advantages of the method on 3D shape databases.

The benefits of representing shapes as continuous planar curves are multiple. Firstly, shapes are analysed as their underlying curves without having a sparse collection of landmarks since there is no need to introduce special points. Shapes are continuous in nature and the placement of points and landmarks is a man-made way of analysing shapes. By establishing the continuous curve framework, it is then easy to develop models for the sampling of these curves and link finite realisations to infinite-dimensional models. Secondly, all representations (either discrete or continuous) share the fact that the resulting shape spaces are nonlinear. This nonlinear geometry allows the calculation of statistics and the performance of inferences. That means though that simple operations like addition, multiplication etc cannot be performed on such spaces. Thus, one has to perform operations between shapes on frameworks that allow them. Representing shapes as continuous curves and establishing nonlinear manifolds allows us to establish full statistical frameworks, define probability densities on shapes, perform operations such as integration and differentiation, create priors for our beliefs and use them for operations between shapes and also Bayesian inferences. Lastly, these methods are not computationally expensive making them easier to use. Kendall's approach is similar in nature with the difference that the study is performed on discretised curves instead of continuous ones. As mentioned, the use of landmarks complicates and biases the setting up of the problem. Active shape models are faster than other landmark-based models but they don't remove similarity transformations and hence don't utilise the non-linear

nature of shape spaces. Grenander's formalisms are similar to Klassen's however are computationally slow for real time applications. Srivastava et al.'s [85] approach offers a complete framework which can be used in various applications.

What we discussed above is only a small flavour of all the available representations that one can choose for the purposes of shape analysis. As discussed above, in our opinion, the representation of shapes as continuous planar curves is the most natural and the most effective one. This is the representation that we choose to use for the purpose of this thesis since it will help us treat the problem of classification in a continuous way. We will discuss the chosen representation and the full statistical framework we establish in chapter [2].

## 1.4 Shape classification

After the preprocessing and choosing the appropriate representation, the next step of shape analysis is classification. Shape classification is a correspondence problem which compares shapes and is the process of assigning a shape to a category or vice-versa. Duda et al. [99], describe it as the task of recovering the model that generated the patterns. There are two types of shape classification: supervised and unsupervised. The former is used for the assignment of shapes into predetermined classes. The latter is a more difficult procedure where the object is assigned into unknown classes which must themselves be inferred from the data. Both classification types require the comparison of shapes to determine how similar they are, which was the theme motivating the use of different representations in the previous section. The similarity comparison is done by comparing corresponding points (these can be labelled landmarks for example) of the shapes.

The problem of classification in a supervised setting is the main subject of this thesis. In this setting, we consider samples or templates of the classes that generate the shapes we have in our data. In particular, in this thesis we will consider the Bayesian classification which is a powerful approach to classification since, having a statistical model, Bayes' laws can supply us with the statistically optimal solution to

this problem. A common route to statistical classification is to construct a classifier for the particular problem. The classifier is constructed according to the chosen shape representation. We will present some examples of such classifiers below.

### Nearest neighbour

In the nearest neighbour approach, the classification is achieved by the extraction of a feature vector. In this case, shapes are represented by a single feature (for example their area) or by a commonly used vector of multiple features (for example area, perimeter, convex hull and others) to describe the shapes that will be classified. If we impose a metric on the feature space we can then calculate distances between two feature vectors. Here, we have two categories of shapes: the training shapes (a database of templates coming from the existing classes) and test shapes (shapes that come from the same shape population as the training ones) which assess the performance of the classifier. There are many algorithms and classifiers which classify based on training shapes.

A commonly used classifier in the case of feature spaces is the nearest neighbour or k-nearest neighbours. The classifier compares the feature of the test shape to the features of all training sets and finds the k-nearest to it based on a measure of similarity which is usually calculated using the metric on the feature space. The test shape is assigned to the most common class amongst the $k$ nearest neighbours and the classifier outputs the class membership of the particular shape. The k-nearest neighbours assign the feature vector of the test shape to the k-nearest feature vectors from the training set and then assigns it to the most common among the k. The role of k is a smoother since the larger it is the more noise is tolerated from the training shapes. The classifier assumes that the similarity between all the shapes can accurately be represented by their feature vectors and hence k-nearest neighbour is a good classifier to be used [100]. An extension and improvement of the results of the k-nearest neighbour algorithm is Bayesian Aggregation [100]. The k-nearest neighbour can also be used in the case that we choose to represent our shape by its contour; the contour is treated as being the feature that describes the shape. The

metric is used to find the k-closest elastic closed curves to the test one.

### Support vector machines

Support vector machines (SVM) is another popular classification method. The popularity of the method lies in the fact that it replaces the distances and the inner products of the shape spaces with those of higher-dimensional spaces. In SVM one creates new feature spaces that are of much higher dimension than the shape space itself. Here, we have all the training data being represented as points in this feature space and then been assigned to one of two categories. Then SVM becomes a non-probabilistic binary classifier and is trained to assign new shapes into one of the two pre-determined classes [101].

Firstly, the training shapes are separated into the two categories. SVM constructs a hyperplane in the high-dimensional feature space and test shapes can be mapped to this space. They are mapped to the training shapes so that the separation between the categories is as clear as possible. In other words, the SVM finds the best hyperplane that gives the clearest separation between points that belong to different classes. The best hyperplane is the one that gives the largest margin (the wider gap) with no interior points, between two classes by classifying new test shapes on either side of the gap.

### Maximum likelihood

Other shape classifiers can be model-based classifiers. In this case, classes of shapes are assigned a probability distribution. Then, the aim is to use the models of shape variations so that each test shape is assigned to the class that maximises the value of the likelihood. A similar in nature classifier is the naive Bayes classifier. The Bayesian classifier is usually combined with Bayesian decision theory (alternatively Bayes classification or Bayesian decision rule) and uses strong independence assumptions for the features that describe the shapes. In this case, class labels are assigned to the training shapes which can be represented by certain features making use of

the fact that certain features of shapes are independent of any other features conditioned on the class. One common approach is when the classifier chooses the class that maximises the posterior probability of that class given the observed data which is known as the Maximum a Posteriori rule. Bayesian classification is a powerful approach; having a good statistical model, Bayesian classification can provide the statistically optimum solution of classification because they minimise the chance of misclassification [99].

In this thesis we will make use of a model-based classification technique; we will assign probabilistic models to the shapes but also to the classes of shapes. We will then construct a Bayesian classifier and use a Maximum a Posteriori (MAP) decision rule for the classification of each of the test shapes so that they are assigned to the class that maximises the posterior probability of the class. We will present our approach in chapter [2]. Thus, this work has set its own statistical framework for the study and classification of shape in a Bayesian way.

Before we present in the next chapter the statistical framework chosen for this thesis and the classification methods used, we will discuss in the next section a very important theme which is shape spaces.

## 1.5 Shape spaces

Having an appropriate formulation for classes of shapes, a shape space needs to be constructed for their study and classification. Whatever the chosen representation is, the study of shape is done by establishing equivalences between them with respect to similarity transformations. This firstly includes the construction of the pre-shape space by imposing appropriate constraints on the chosen representation of curves. This leaves some elements of the pre-shape space belonging to the same orbits of shape similarity transformations. The quotient space that comes from this results in the shape space. If the pre-shape space happens to be a manifold, then the shape space inherits the Riemannian structure and becomes a Riemannian manifold of orbits (orbifold). We will explicitly describe how shape spaces are created in the

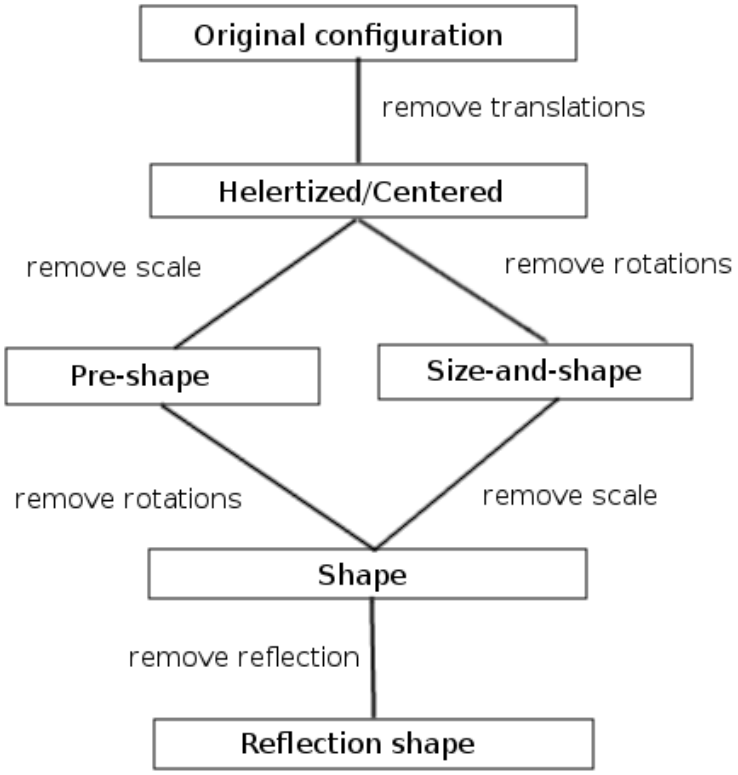



Figure 1.2: The shape spaces [102]

following paragraphs.

Shapes can be described geometrically and it is their geometry that can help us estimate their properties and use them to make inferences about populations of shapes. Inferences require us to use probability distributions and appropriate spaces of shapes to represent them. To obtain the representation of a shape according to the definition we are following [11], similarity transformations need to be filtered out so that all shapes are aligned to a common reference coordinate system. We described how one can acquire such a coordinate system with the Procrustes method which places shapes into shape spaces.

In order to compare and classify shapes into their respective categories we need to establish a common coordinate system as motivated by the definition of shape that we adopted. In this coordinate system, all the shapes will be aligned with similarity transformations removed. The classical alignment procedure was firstly described by Kendall [11] which can also be found in [14] and incorporates the aspects of classical shape analysis which describes shapes by landmark points. The classical alignment procedure is called Procrustes analysis and has been routinely used in the literature [5, 14, 103]. Ordinary Procrustes Analysis (OPA) [104] aligns any two shapes so that they have a common reference coordinate system by minimising the difference of shapes according to a measure of similarity or a chosen metric. OPA uses least squares techniques to match the shape configurations with respect to similarity transformations and is used in the case that only two shapes are to be considered.

Procrustes analysis removes from shapes all similarity transformations and places them into the shape space. The imposition of a shape metric turns the space into a Riemannian manifold. Common shape metrics that have been used are the Hausdorf metric [105] and the Procrustean metric or Procrustean distance [5, 14, 106, 107] which is the most broadly used and it is the one that we will describe now. The Procrustean metric aligns the two shapes so that the best correspondence between the shapes is estimated in the following way: it scales the shapes to have equal size (for example unitary length), aligns the shapes with respect to their centroid (the

centre of mass) at the origin and then aligns them with respect to their orientation by rotating them. A reference rotation is chosen as the rotation of one of the two shapes and then the other is rotated so that the sum of the squared distances of the points is minimised. The optimum rotation can also be found by Singular Value Decomposition (SVD). Now, differences between shapes can be measured by the Procrustean metric that is the square root of the sum of the squared distances of the points of the superimposed shapes.

In the case that the comparison involves more than two shapes then the similarity transformation removal is done by the Generalised Procrustes Analysis (GPA) [108, 109] which in contrary to OPA superimposes optimally rather than to a shape that was chosen at random. Goodall [110] and Goodall and Bose [111] adapted it for the particular case of shape analysis. In GPA, translations and scalings are removed in exactly the same way as in OPA. The only difference between the two methods is how the rotation is chosen. Rather than arbitrarily aligning the shapes to a reference rotation GPA finds the optimal rotation. The way GPA finds the optimal superimposition is: choose an arbitrary shape as a reference (also known as mean/average shape) and translate all remaining shapes so they are superimposed to the reference. Then, calculate the Procrustes mean shape (also known as Frecét or Karcher mean [112, 113, 114]) of the aligned shapes and if its difference from the reference has changed recalculate the mean shape. Convergence is achieved when the difference between the mean shapes is less than a threshold.

Kendall [11] was the first to construct the mathematical framework of shape spaces. Kendall represented shapes as points of a non linear manifold. The use of differential and Riemannian geometry is the tool used for shape analysis and it studies shapes as non-linear objects. Since we assume that shapes are objects invariant to Euclidean similarity transformations we can also use algebra and group theory to represent their action on shape spaces so that shapes can be described as orbits of these groups. We will now give the definitions of concepts that will be useful for the construction of shape spaces.

**Definition 1.5.1** *Equivalence relation*: a relationship denoted by ~ that satisfies

a) reflexivity ($a \sim a$), b) symmetry ($a \sim b \Rightarrow b \sim a$) and c) transitivity ($a \sim b$ and $b \sim c \Rightarrow a \sim c$). Then, an equivalence class of a set $X$ under the equivalence $\sim$ is $[a] = \{x \in X | x \sim a\}$.

**Definition 1.5.2** *Quotient space*: a quotient space is the set of all equivalence classes in a set $X$ under a certain relation and is denoted as $[X] = X/\sim$.

**Definition 1.5.3** *Manifold*: a manifold is a topological space that is locally Euclidean: there are compatible maps, $\phi : M \to \mathbb{R}^n$, where $n$ is known as the dimension of the manifold.

**Definition 1.5.4** *Riemannian metric*: a Riemannian metric is a map that satisfies certain conditions and relates each point on the manifold to points on the tangent space – a flat approximation of the manifold in question. Metrics allow us to measure infinitesimal distances on the tangent space.

**Definition 1.5.5** *Geodesic*: The minimum path between point $a$ and point $b$ on the manifold as measured by the metric is called a geodesic between $a$ and $b$. The distance between two finitely separated points on the manifold is then the length of the geodesic joining them.

**Definition 1.5.6** *Lie group*: a Lie group is a group that has a manifold structure where the operation of multiplication and the operation of inversion are smooth maps. Examples of Lie groups include all the Euclidean similarity transformations. Translations are a manifold $\mathbb{R}^d$ and a group equipped with the operation of vector addition. Similarly, scalings are a manifold $\mathbb{R}^+$ and a group equipped with multiplication and rotations $SO(3)$ are a manifold $S^3$ and a Lie group under composition.

If we represent a shape by a set of ordered $n$-ads in $\mathbb{R}^m$ say $X = (x_1, ..., x_n)$ then each configuration could be represented into a $nm$-vector in $\mathbb{R}^{m \times n}$ (In this thesis we will focus on planar shapes in $m = 2$ dimensions of $n$ points so the configuration space will be of dimension $2n$). The groups that describe the Euclidean similarity transformations of a configuration are: the group of translations under addition $G_t = \mathbb{R}^m$ where the action is: $(t, X) \mapsto X + t\mathbf{1}_n = (x_1 + t, ..., x_n + t)$. The group of scalings

$G_a = \mathbb{R}^+$ under multiplication where the action is: $(a, X) \mapsto a \cdot X = (a \cdot x_1, ..., a \cdot x_n)$ and finally the group of rotations $G_R = SO(m)$ under composition where the action is: $(R, X) \mapsto RX = (Rx_1, ..., Rx_n)$.

The actions of rotations and scalings do not commute with the action of translations. That is because $Rx + t \neq R(x + t)$ for rotations and also $a(x + t) \neq ax + t$ for scalings. That means that there are two separate actions: $\mathbb{R}^+ \times SO(m)$ and the action of $\mathbb{R}^m$ but not the action of the whole group as $\mathbb{R}^m \times (\mathbb{R}^+ \times SO(m))$. However, we can define the action of the semi-direct product as $G = \mathbb{R}^m \ltimes \mathbb{R}^+ \times SO(m)$ which is well defined since $\mathbb{R}^+ \times SO(m)$ acts on translations in the same way as it acts on the vector space $\mathbb{R}^m$, defined as $(t, a, R) * x = aRx + t$. The action of the semi-direct product $\mathbb{R}^m \ltimes \mathbb{R}^+ \times SO(m)$ on an element is called a rigid motion.

Kendall separated the pre-shape space from the shape space. The former is the last step before the true shape space; it is the space where scale and translation are removed from the object with the rotations still present. It is a hypersphere of unit radius in $(n-1) \times m$ dimensions and is constituted by configurations that represent the same shape since rotations have not been filtered out yet. The elements of the pre-shape space are called pre-shapes and are invariant under translations and scalings.

In summary, a configuration that has been rotated, scaled and translated can be represented as: $[X] = \{aRX + t\mathbf{1}_n : a \in \mathbb{R}^+, R \in SO(m), t \in \mathbb{R}^m\}$ with $G = \mathbb{R}^m \ltimes \mathbb{R}^+ \times SO(m)$. These are called *orbits* of the three groups and all elements of $[X]$ describe equivalent shapes. Each equivalence class represents a unique shape and it is an element of the quotient space $\mathbb{R}^{m \times n}/(\mathbb{R}^m \ltimes (\mathbb{R}^+ \times SO(m)))$. This quotient space is the set of all the orbits of $\mathbb{R}^m \ltimes (\mathbb{R}^+ \times SO(m)$ in $\mathbb{R}^{m \times n}$ and constructs the **shape space** of all possible configurations that are the quotient space of the pre-shape space. If a pre-shape is a Riemannian manifold then the shape spaces inherits this Riemannian structure and becomes a Riemannian orbifold. An action that intersects all the orbits of the quotient space is called an orthogonal section.

Kendall's shape space [11] is based on the coordinates of landmarks which describe a shape geometrically. The space of all possible landmarks is the configuration

space [14]. The Shape space [11] is the space of all possible shapes of the object in question. This space is isomorphic to $\mathbb{R}^{mn}$ which is a differential manifold of dimension $mn$ and each shape is a point on it. This manifold comes with a natural Riemannian metric: $d(u,v) = \sqrt{\sum_{i=1}^{n} ||u_i - v_i||^2}, \quad u, v \in \mathbb{R}^m$ which is the sum of Euclidean distances in $\mathbb{R}^m$ and is invariant under translations and rotations. However, the metric is not invariant under scalings since it transforms as $d(au, av) = a^2 d(u, v)$. Removing translations removes $m$ dimensions leaving $nm - m$ degrees of freedom. The isotropic non-rigid scaling removes one dimension and the rotations remove $\frac{1}{2}m(m-1)$. Overall, the dimensionality of the shape space is: $nm-m-1-\frac{1}{2}m(m-1)$. This is also known as the "Kendall shape space" [5]. Removing rotational, translational and scale effects from a shape is known as *pose*. Kendall's shape space is a finite dimensional Riemmanian manifold. Different shapes are different elements of the manifold and the differences between them are calculated and quantified by the imposed Riemannian metric one chooses. In these shape manifolds we can define probability distributions to statistically study shapes' estimation.

By using OPA or GPA shapes are brought to a common coordinate system and now the variation of the shape classes can be studied in this framework. Popular methods for modelling the class variation is Principal Component Analysis (PCA). PCA is a way of removing redundancy from the dataset and was first introduced by Pearson [61] and established by Hotelling [62]. It is a way of identifying existing data structure and explaining their variation. PCA is useful in shape analysis since it allows dimensional reduction in these high dimensional spaces. Principal components can explain in which direction the highest variability of the data lies. Shape variables usually are not statistically independent so most of the times we expect them to be correlated. This is because they describe aspects and features of the shape that are connected in a way; either genetically or mathematically. Here by the term genetically we mean all shapes that share the same properties due to the physiology of their shape. For example skulls share certain genetic landmarks that are identical and act as reference for the experimentalists. The aim of PCA is to explain this variation and reveal the patterns between the features by transforming the variables into a set of new ones that are an independent linear combination of

the old ones. Most of the sample variation can be then explained by only a few principal components and make the task of statistical inference much easier. Its deep goal is to explain the directions of largest proportion of the total variance. Figure (1.2) shows a diagrammatic relationship between the spaces we have discussed by now.

To sum up, we have seen all the stages of shape analysis: shape pre-processing, shape transformations and shape classification. In particular, we have talked about acquisition and detection, shape representation and classification; we focused only on these notions since these will be of use for the purposes of this thesis. Shape acquisition and detection are notions discussed in chapter [3] where we will describe how these methods help us acquire the boundaries and outlines of the shapes in question. In chapter [2] we will start building the framework needed for the classification of the acquired shapes. We will briefly discuss the chosen representations under which we will perform the shape classification and how the shape spaces come of use. Then, the big task is to establish and build probability distributions on these spaces and model the variability of the parameters we chose to describe the shapes. For the classification of the shapes, we will describe how we utilise the Bayesian statistical framework and extend the work presented in [12].

# Chapter 2

# A novel shape classification method

## 2.1 Introduction

In this chapter, we introduce a classification method which is the main contribution of this thesis. We revisit and extend the results presented in Srivastava and Jermyn [12] by studying the problem of Bayesian classification of planar shapes in an unbiased way. We utilise the Bayesian statistical framework presented in [12], however the novelty of our approach is based on the way that the similarity transformations are removed from the shapes in question in such a way that previous numerical methods are replaced by their equivalent closed form solutions. Unless stated otherwise, much of the following sections are based on the work presented by Srivastava and Jermyn [12].

Section [2.2] presents the problem of shape classification and explains the way we have chosen to treat it in the Bayesian paradigm by employing the results of [12]. Sections [2.3] to [2.7] present the models we have utilised for the description of the parameters that take part in the formulation of the problem. In particular, section [2.7] explains one of the novelties of our approach by presenting the evaluated Jeffreys prior for the parameters. However, Jeffreys prior introduces irregularities and divergences of the result. Section [2.8] discusses our method of alleviating the

divergences produced by Jeffreys prior which was done by employing regularisation methods for the prior. Overall, this section presents the solution of the classification problem and presents a result which replaces some of the approximating methods presented in [12]. Section [2.9] presents a comparison between results when using the method evaluated at section [2.8] and results when using Jeffreys prior of section [2.7]. Section [2.10] presents the remaining computational methods we used for the solution of the classification problem . The final result constitutes our proposed classification algorithm in replacement of the algorithm in [12]. Finally, section [2.11] presents the concluding remarks of this chapter.

## 2.2 The problem of classification

As mentioned before, shape is an important feature of objects in question. One common theme and problem in the study of continuous, closed planar curves is that in computer simulations one has to deal with noisy and undersampled data where the use of landmarks and primitives is imperative. Towards this end, we will study how to classify shapes that are generated by such continuous curves and look how we can probabilistically classify them into their respective categories; given a set of pre-determined classes we would like to classify the observed data shapes – we here define a **data shape** to be one of the shapes that we observed i.e. an ordered set of points in $\mathbb{R}^2$.

In the approach we take we will represent the objects of interest and their boundaries as continuous planar curves (i.e. one-dimensional lines which denote the outline of the object) and study their shapes. Our goal is to develop shape models, statistical procedures and classification methods of continuous planar shapes and establish the statistical framework needed for their classification. However, since the testing of our theory involves computer implementation of algorithms, we will eventually discretise these shapes by sampling their boundaries; however, we follow the philosophy of discretising as late as possible [115].

The problem of classification is to state how probable it is that a given data

shape $y$ belongs to a class $C$. It can be described probabilistically and can be mathematically formulated as the posterior probability of the class in question given the observed data, that is by $\mathbb{P}(C|\boldsymbol{y})$ where $C \in \mathcal{C}$ the class of the object, represented by the dataset and $\boldsymbol{y} \in Y$ is the set of all the observed data shapes i.e. a finite set of primitives. More specifically, $\mathbb{P}(C_j|\boldsymbol{y_i})$ is the probability that the observation $\boldsymbol{y_i}$ has been generated by class $C_j$. For the time being we will study planar shapes so we can take $Y$ to be $Y = \mathbb{R}^{2n}$ for $n$ primitives. In a Bayesian framework, classification is performed by maximising the posterior probability of the class so constructing a Bayesian classifier and using the Maximum a Posteriori decision rule, the classification of the data set $\boldsymbol{y}$ can be done with the help of Bayes theorem:

$$\mathbb{P}(C|\boldsymbol{y}) = \frac{\mathbb{P}(\boldsymbol{y}|C)\mathbb{P}(C)}{\mathbb{P}(\boldsymbol{y})}. \tag{2.2.1}$$

The classification is then achieved by deciding on $\tilde{C} = \text{argmax}_C\ \mathbb{P}(C|\boldsymbol{y})$. The prior probability over the classes $\mathbb{P}(C)$ can be freely chosen. Without any evidence on which to base this prior, we choose it to be uniform and hence give equal chance of each class to appear in our data. Then, the greatest task is to calculate the likelihood which describes how likely it is for the data to have been generated by a fixed class. To calculate the likelihood, we will partition it over nuisance parameters that correspond to the data formation process so that the likelihood describes how the data are formed by the object class; this is the novelty of the geometrical approach in [12]. The marginalisation of the likelihood also helps us to break down its approximation into simpler steps which makes the calculation easier. We now introduce the variables needed for the partition of the likelihood that provide an overview of the formation stages and the rise of the algorithm used for the purposes of classification. Each of the variables will be thoroughly explained in the following sections.

Let $g \in G$ be the group action of rotations, translations and scalings with $G = \mathbb{R}^m \ltimes (\mathbb{R}^+ \times SO(m))$, the semi-direct product as described previously in chapter [1], section [1.5]. Let $\beta \in \mathcal{B} \equiv \mathbb{R}^{m\times n}/(\mathbb{R}^m \ltimes (\mathbb{R}^+ \times SO(m)))$ be a shape which is an object's outline modulo Euclidean similarity transformations that from now on we will call

an **example shape**; a specific shape outline is then given by $g\beta$. Let $s \in \mathcal{S}$, be a sampling function that places $n$ points around the boundary of the shape. We will see in section [2.3] that the model of sampling functions is chosen so that it reflects our belief that samplings should favour the even sampling and vary around that. Placing these points on the boundary, the continuous curve $\beta s$ becomes a set of $n$ discrete landmarks modulo similarity transformations. A particular shape on which a similarity transformation has acted upon is given by $g\beta s$. Lastly, let $b : [0, ...n] \to [0, ...m] \in \mathbb{B}$ be a bijection relating points from $\beta$ uniquely to points of the data shape $y$. There will also be some inherent noise as part of the data collection process so we include a parameter $\sigma$ which represents the variance of this noise. As this is an unknown parameter, we will integrate over it by imposing a prior which we introduce in later sections. These parameters can be used for the marginalisation of the likelihood so that:

$$\mathbb{P}(y|C) = \sum_{b \in \mathcal{B}} \int \mathscr{D}\beta \; \mathscr{D}s \; \mathscr{D}g \; d\sigma \; \mathbb{P}(y|b, \beta, s, g, \sigma)\mathbb{P}(b)\mathbb{P}(s)\mathbb{P}(g)\mathbb{P}(\sigma)\mathbb{P}(\beta|C) \quad (2.2.2)$$

The integration measures use calligraphic $\mathscr{D}$ to denote their spaces are infinite dimensional. In the above expression, in the formulation of the likelihood we have made the necessary independence assumptions. In particular, $b \perp\!\!\!\perp \{\beta, g, C\}$ as bijections are conditionally independent of a particular shape, transformation or class. One can say here that bijections implicitly depend on the samplings $s$ only with respect to the number of the sample points. Samplings around the boundary of the shape can be done in many ways, for example with respect to the curvature of the curve however in our work we take $s \perp\!\!\!\perp \{g, \beta, C\}$ so that samplings have no dependence on any particular transformation, shape curve or class. For similarity transformations we take $g \perp\!\!\!\perp \beta$ because no curve depends on a particular similarity transformation and they are thus independent of how the curves were formed. We will discuss the individual marginal distributions of the nuisance parameters in the sections to follow.

The difficulty of the classification problem lies in the fact that in order to compute the posterior probability via MAP one must evaluate the integrals and thus

somehow sum over all possible bijections and integrate over all similarity transformations, samplings and shape curves. In past work [12, 14, 116] the integration and summation over all nuisance parameters were approximated by Monte Carlo integration and Laplace's approximation (saddle point approximation). In particular, the integration over the curves $\beta$ and samplings $s$ were calculated by Monte Carlo integration. To do so, realisations from the distributions were generated and the values of the integrand evaluated at these realisations were summed over. The integration over the similarity group $g$ and the sum over bijections $b$ were carried out by a Laplace's approximation which finds the maximal bijection $b$ for the best transformations $g$. This used a combination of the Procrustes and the Hungarian algorithm [117, 118] to find the optimum solution to the combined registration-transformation problem with the likelihood being the cost function. The solution to this combined registration-transformation optimisation problem maximises the integrand and the result of these two integration procedures is an approximation of the value of the likelihood $\mathbb{P}(\boldsymbol{y}|C)$ which, when normalised, gives the MAP estimation of the posterior probability over the classes given the observed data.

In this thesis, we present in section [2.8] how, under the right choice of priors, the integration over the nuisance parameters is feasible using analytical methods which result in closed form solutions. To construct a fully statistical framework, we develop probability models and computational methods for our choice of probabilistic models $\mathbb{P}(\beta|C)$, $\mathbb{P}(s|\beta, C)$, $\mathbb{P}(y|b, g\beta s)$ which respectively describe the variability in shape, samplings and the observation noise.

## 2.3 Sampling models

In this work, by sampling a continuous, planar curve we mean the placement of a certain number of ordered points (landmarks) on the curve by a sampling function. As discussed in chapter [1], the placement of points around a planar curve most often depends on external factors such as the particular experiment or the experimentalist. This action of discretisation of a curve contributes to the loss of information about the curve that represents the original shape. Samplings generate primitives

or landmarks by certain procedures such as edge detection or the experimentalist himself. This makes the position of the landmarks dependent on the chosen sampling method and for this reason we treat samplings in a probabilistic way. In the next section we describe the sampling model $\mathbb{P}(s)$ and the space that such sampling functions are being generated from.

### Representation of samplings

The problem of matching sampled shapes has been discussed in the past [119] however the mathematical representation of samplings has not been studied extensively in the literature. Now, a sampling involves the placement of $n$ points on a curve. To describe this placement mathematically we need a good representation and parametrisation of the curve. A natural way a sampling can be parametrised is with respect to its arc length; then the $n$ points can be placed along the length of the curve i.e. in the interval $[0, L]$, where $L$ is the curve's Euclidean length. It is usual practice to standardize lengths so that the points are placed between 0 and 1, which makes the actual representation of the samplings easier. As stated in the previous section, the probability of a sampling is independent of the position, orientation or scale of the curve which is implicit in equation (2.2.2).

The placement of $n$ points in the interval $[0, 1]$ is equivalent to partitioning it into $n$ sub–intervals. The first point to be placed will be assumed to be the origin $\tau$ of the samplings and will become an element of the representation. The partition of the unit interval by $n$ points can be thought of as a probability mass function with $n$ elements. This allows consistency between probabilities of samplings with different number of points which implies that the number of the points and their placement should be considered as separate actions. We now see how samplings can be represented.

Let $\Gamma$ be the set of all increasing differentiable functions from $[0, 1]$ to itself with the constraint for all $\gamma \in \Gamma$ be: $\gamma(0) = 0$ and $\gamma(1) = 1$. This is a positive diffeomorphism of the unit interval. Partitioning uniformly the unit interval in $n$ sub–intervals we get $U = [0, 1]/n$. A sampling $s$ is then represented by an equivalence

class $[n, \tau, \gamma] \in \mathbb{N} \times \mathbb{S}^1 \times \Gamma$ where parameters that lead to the same sampling are identified. Thus, an equivalence class $[n, \tau, \gamma]$ forms a sampling by the action of $\gamma$ on $U$ starting at $\tau$ – the diffeomorphisms push forward the points in $U$ to new positions where we sample. Since diffeomorphisms are increasing functions in the unit interval they can be thought of as cumulative distribution functions on $[0, 1]$. There is a number of possibilities for the representation of such functions of which we must choose the most efficient for our applications:

1. **Diffeomorphisms**: an element of $\Gamma$ can be represented by itself; as an increasing function in the interval $[0, 1]$ such that $\gamma(0) = 0$ and $\gamma(1) = 1$. The action of the group of diffeomorphisms is composition which is relatively simple.

2. **Probability density**: an element of $\Gamma$ can be represented as a positive probability density so that $p = \dot{\gamma} = \frac{d\gamma(\tau)}{d\tau}$ which is a positive function that integrates to 1.

3. **Square-root form**: an element of $\Gamma$ can be represented by the square root of a probability density so that $\psi = \sqrt{p}$ with $\psi \in \Psi$ and $p \in \mathcal{P}$ a probability density. This representation coincides with the positive functions whose square integrates to 1; this is the positive orthant of the unit sphere in the space $\mathbb{L}^2([0, 1])$. This representation simplifies the form of the functions and induces a simple natural Riemannian metric on $\Gamma$. A big advantage of this representation and the nature of the underlying space is that geodesics and exponential maps can be calculated in closed form. In the past the usage of a different metric was used for the approximation of geodesics by numerical methods. Srivastava et al. [120] demonstrate the computational superiority of the square-root form representation.

We must now choose the appropriate representation of sampling functions to have as much efficiency as possible for our work and applications. As we mentioned above, one can show that the square-root form of a probability density results in a simpler manifold, the unit sphere, under the much simpler $\mathbb{L}^2$ metric; for the proof

of this result refer to [120]. For these reasons, the square root form is our chosen representation and we now discuss how the space $\Gamma$ of the increasing differential functions is structured under our chosen representation.

### 2.3.1 The structure of $\Gamma$

Our ultimate goal is to construct probability distributions that represent our trust in the samplings i.e. the placement of $n$ points on the boundary of an object. The positive diffeomorphisms on the unit interval form a non-linear manifold on which we can impose Riemannian structure by choosing an appropriate metric. This allows us to perform statistics and calculate geodesics between different diffeomorphisms on the manifold. Thus, we must decide on the choice of the representation of the increasing differentiable functions and also choose the metric to impose on the manifold.

There are unlimited choices of Riemannian metrics one could impose on $\Gamma$. However, there is a natural choice of metric for the space of probability distributions, the so called Fisher-Rao metric, which defines an inner product on variations of two probability distributions and has been extensively used in the literature [121, 122, 123]. The choice of this metric has a geometrical meaning since it is invariant to reparametrisations of the unit interval and the action of the diffeomorphism group as proved by Čencov [124]. Since, as stated in point 2 of the previous section, the space of probability distributions $\mathcal{P}$ is isomorphic to $\Gamma$, $\Gamma$ inherits this natural metric which is the one we choose to use. For the proof of the Fisher-Rao metric's invariance on $\Gamma$ refer to [12] or [120].

Under the choice of the probability density representation $p$ for the increasing differentiable functions, the Fisher-Rao metric takes the following form: the inner product on variations in probability densities at any point $p \in \Gamma$ is:

$$\langle \delta p, \delta p' \rangle = \int_0^1 \delta p(s) \delta p'(s) \frac{1}{p(s)} ds, \qquad \delta p, \delta p' \in T_p(\Gamma). \tag{2.3.3}$$

However, under the square root representation $\psi = \sqrt{p} \in \Psi$, the metric becomes significantly simpler and transforms into the $\mathbb{L}^2$ metric that appeared as part of the

constraints on $\psi \in \Psi$, which is:

$$\langle \delta\psi, \delta\psi' \rangle_\psi = \int_0^1 \delta\psi(s)\delta\psi'(s)ds, \qquad \delta\psi, \delta\psi' \in T_\psi(\Psi). \tag{2.3.4}$$

Overall, the space $\Psi$ is the positive orthant of the unit sphere and the Fisher-Rao metric transforms into the $\mathbb{L}^2$ Riemannian metric on $\mathbb{L}^2([0,1])$ restricted to $\Psi$. With this simpler form of the metric, geodesics are great circles on the sphere. The solution to the geodesic equation for $\Gamma$ under the Fisher-Rao metric and the $\mathbb{L}^2$ inner product is:

$$d(\gamma_1, \gamma_2) = \cos^{-1}(\langle \dot{\gamma}_1, \dot{\gamma}_2 \rangle_\Psi). \tag{2.3.5}$$

The geodesics on $\Psi$ between two points $\psi_1$ and $\psi_2$ are given by:

$$\psi(t) = \frac{1}{\sin(\theta)} \left[ \sin((1-t)\theta)\psi_1 + \sin(t\theta)\psi_2 \right] \tag{2.3.6}$$

with $\langle \psi_1, \psi_2 \rangle_\Psi = \cos(\theta)$. Finally, for the desired geodesics in $\Gamma$, one can derive that $\gamma(t)(s) = \int_0^s \psi^2(t)(\tau)d\tau$, using that $\psi_i = \dot{\gamma}^{1/2}$. Since $\Psi$ in general is an easier space to calculate mathematical and statistical quantities, it is easier to compute the wanted quantities on $\Psi$ and then project the results back to $\Gamma$. This is of particular use for the sampling functions and for constructing the desired probabilities on $\Gamma$. For the projection of the results between $\Psi$ and $\Gamma$ we need to describe and use the exponential map.

The exponential map is a map between the tangent space $T(\Psi)$ of the manifold and the manifold itself which in our case is the $\mathbb{L}^2$ unit sphere. We have been referring to the tangent space in the previous two equations and it is defined in section [1.5]. The only complication here is that the elements of the manifold are sampling functions (one possible basis could be Fourier components) so elements of the tangent space are infinitesimal differences between these functions. The geodesic on $\Psi$ starting from the point $\psi$ in the direction of $v \in T_\psi(\Psi)$ is: $\cos(t)\psi + \sin(t)\frac{v}{\|v\|}$ so that the exponential map from the tangent space $T_\psi(\Psi)$ of $\Psi$ to $\Psi$ is defined by: $\exp_\psi(v) = \cos(\|v\|)\psi + \sin(\|v\|)\frac{v}{\|v\|}$. However $\|v\|$ must be restricted in $[0, \pi)$ to avoid

negative values of $\psi$ and hence avoid $\exp_\psi(v)$ lying outside the manifold $\Psi$. For any two $\psi_1$, $\psi_2 \in \Psi$ the definition of the inverse exponential of $\psi_2$ is $v = \exp_{\psi_1}^{-1}(\psi_2)$ which can be calculated by: $u = \psi_2 - \langle \psi_2, \psi_1 \rangle \psi_1$ and $v = u \cos^{-1}\left(\frac{\langle \psi_1, \psi_2 \rangle}{\sqrt{\langle u, u \rangle}}\right)$.

In summary, the space $\Gamma$ of increasing differentiable functions is described by a manifold equipped with a Riemannian metric and we can now construct probability distributions on it. Srivastava and Jermyn [12] use the fact that a sampling $s$ can be represented as $\langle n, \tau, \gamma \rangle \in \mathbb{N} \times \mathbb{S}^1 \times \Gamma$. The probability for a sampling $s$ was then calculated as: $\mathbb{P}(s|C) = \mathbb{P}(n)\mathbb{P}(\tau|C)\mathbb{P}(\gamma|\tau, C)$. However, in our work we assume that $\mathbb{P}(s) = \mathbb{P}(\gamma)$, which implies that $s$ is conditionally independent of the class $C$ and we use a uniform distribution for $\mathbb{P}(n)$. For the distribution over the diffeomorphisms $\gamma$ the choices are enormous however not arbitrary. For example, the points in $s$ represent the sample points selected by the person extracting the shape from the laser cloud data and they tend to be evenly spread. We follow [12] and we use the generalised Gaussian probability distribution which is:

$$\mathbb{P}(\gamma) = \frac{1}{Z} \exp\left(-\frac{1}{2\sigma_s^2} d^2(\dot{\gamma}^{1/2}, \psi_0)\right) \tag{2.3.7}$$

where $d$ is the geodesic distance calculated under the chosen metric and $\psi_0 = \dot{\gamma_0}^{1/2}$ the mode of the Gaussian distribution. We now must make a choice for $\gamma_0$. The most natural choice one could make is $\gamma_0(s) = s$ and $\psi_0 = 1$ because it favours uniform samplings of the curve with respect to its arc-length parametrisation[1]; we choose a Gaussian distribution that has this uniform sampling as its mean and varies along that. Of course, other choices are possible which may depend on geometrical properties such as curvature but we leave this for future consideration.

To simulate from such a probability density we need to generate functions that satisfy the desired restrictions. To do so, we use the exponential map. We need to randomly generate a function $f \in T_{\psi_0}(\Psi)$ such that $\|f\| = 1$. We assume that a function $f \in T_{\psi_0}(\Psi)$ can be written as an infinite sum of its Fourier components

[1]The points in $U$ could remain fixed under $\gamma_0$, so the curve would be split into $n$ points equally spaced along its length.

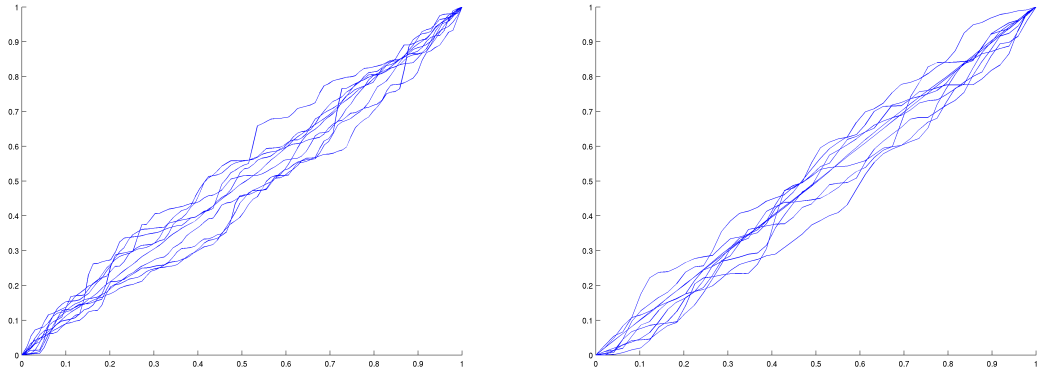


Figure 2.1: Examples of 10 different diffeomorphisms.

i.e. $f(t) = \sum_{-\infty}^{\infty} C_m \exp(2\pi imt)$. We chose this representation because although we could generate a random function in the time domain, its Fourier series' transform in the frequency domain offers a greater smoothness. For computational purposes we take $f(t) = \sum_{-N/2}^{N/2-1} C_m \exp(2\pi imt)$ therefore providing an approximation of $f$. The complex realisations $C_m$ are generated from a complex Gaussian distribution and by imposing the constraint $C_{-m} = C_m^*$ we ensure the reality condition of the Fourier components. To also ensure that $f$ is an element of the tangent space $T_{\psi_0}(\Psi)$ of $\Psi$ and that it integrates to zero, we set $C_0 = 0$. Finally, the function $f$, is calculated via the inverse Fourier transform and its normalization by Parseval's theorem. Then, we generate a distance $x$ of a normal distribution so that $x \sim N(0, \sigma^2)$ and compute a random point $\psi(x) = \cos(x)\psi_0 + \sin(x)f$ via the pushforward. Here, $\psi_0$ represents the mode of the distribution and the starting point on the manifold. In other words, a random element of $\Psi$, which effectively represents a sampling function, can be calculated as going along the geodesic of $\Psi$ starting from $\psi_0$ in the direction of $f$. Finally, the random sampling function we are interested in is calculated by: $\gamma(s) = \int_0^s \psi^2(s')ds'$ which takes us from the square root on $\Psi$ to the functional representation $\gamma \in \Gamma$. Figure (2.2) gives a pictorial explanation of the above and figure (2.1) shows examples of such generated diffeomorphisms. Figure (2.3) shows how such a diffeomorphism pushes forward (in red) the sampled points (in blue).

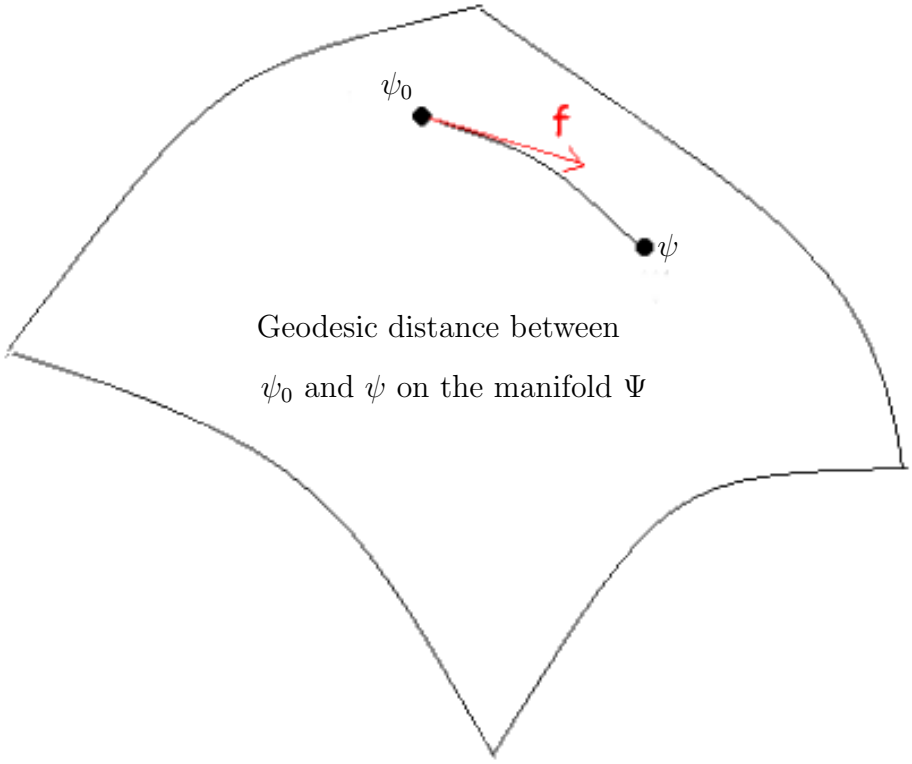



Figure 2.2: A pictorial representation of the observation model

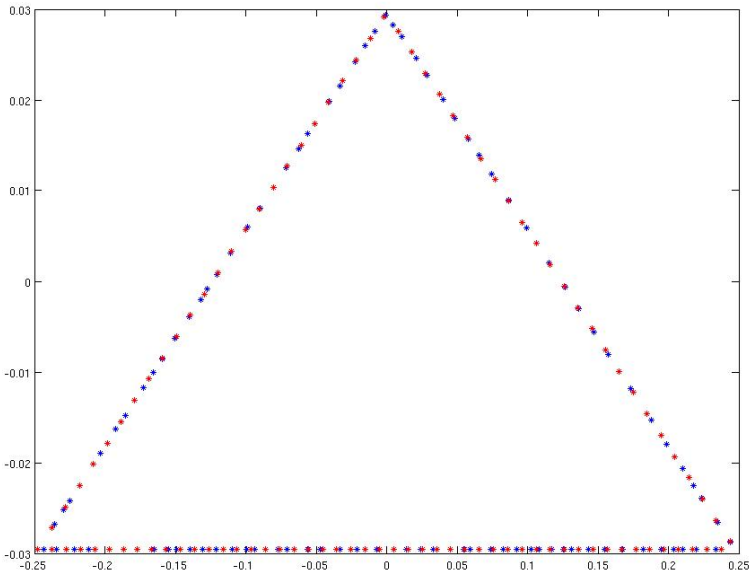



Figure 2.3: A diffeomorphism pushes forward (in red) the sampled points (in blue) of the triangle's boundary.

## 2.4 Shape models

We now discuss the representation of shapes and how we construct shape models $\mathbb{P}(\beta|C)$. These models reflect our belief that objects coming from the same shape class present natural variability within it. In this thesis, we choose different models to describe particular shape applications; for example the KIMIA database model is uniform whereas the geological sand bodies' model is a $\Gamma$ distribution on the aspect ratio (see chapter [3], sections [3.2] and [3.3] respectively for more details). In both cases we represent shapes as closed planar curves that are parametrised by their arc-length.

## 2.5 Bijection models

One of the challenges of the calculation of the marginalised likelihood (2.2.2), is the summation of all possible bijections. The bijection model $\mathbb{P}(b)$ is assumed to

be discrete uniform and in particular we assume that there are $n$ rather than $n!$ bijections. This is a point that we will extensively discuss in section [2.10] due to the irregularities introduced by the closed form results.

## 2.6 The observation model

In many cases, the data we observe may be found in a noisy environment so that the observed data points might be different than the corresponding curve points. Since any data shape point collection introduces uncertainty and errors in the experiment, we need to find a way to include this in the mathematical model that describes the observed shapes. One way to treat this variability is to introduce observational noise that perturbs the points from the original boundary of the shape according to a probability distribution. The choices here are once again enormous. We choose the probability distribution imposed on the noise to be white and additive Gaussian for simplicity. In [12] Srivastava and Jermyn also include clutter that is introduced from the background however this type of noise will not be considered in our work[2]. Modelling the noise and dissimilarity between any two shapes via a Gaussian likelihood the form of the model is:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,g,\sigma) &= \frac{1}{(2\pi)^n\sigma^{2n}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{y_{b_i}} - g\circ\boldsymbol{\beta}(s(b_i^{-1}))|^2\right)\\ &= \frac{1}{(2\pi)^n\sigma^{2n}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{y_{b_i}} - a\mathrm{R}\boldsymbol{\beta}(s(b_i^{-1})) - \boldsymbol{t}|^2\right)\end{aligned} \tag{2.6.8}$$

which models errors in shape point collection as Gaussian white noise where $\sigma^2$ is the noise variance which can be regarded as a free parameter. Varying $\sigma^2$ changes the shape of the posterior but not its mode [12]. Depending on the method of the extraction of the points, our observational points might be different from the corresponding points on the curves. The noise is taken to be responsible for the perturbation of the points from their original place. This shows that a given set of data is supposed to have arisen as a result of a rigid transformation of an ideal

[2]This reflects the mechanism by which data are provided from geological contexts

example shape $\beta$ from a particular class with Gaussian noise added to each sampled point. To express our ignorance of the transformation between the shapes, the likelihood will be broken into small parts and in particular it will be partitioned over the nuisance parameters that will be eventually integrated out.

The likelihood function for the complete data is then:

$$\begin{aligned}\mathbb{P}(y|C) &= \sum_{b\in\mathcal{B}} \int \mathscr{D}\beta\ \mathscr{D}s\ \mathscr{D}g\ d\sigma\ \mathbb{P}(y|b,\beta,s,g,\sigma)\mathbb{P}(b)\mathbb{P}(s)\mathbb{P}(g)\mathbb{P}(\sigma)\mathbb{P}(\beta|C) \\ &= \sum_{b\in\mathcal{B}} \int \mathscr{D}\beta\ \mathscr{D}s\ \mathscr{D}g\ d\sigma\ \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{y_{b_i}} - a\mathrm{R}\boldsymbol{\beta}(s(b_i^{-1})) - \boldsymbol{t}|^2\right)\times \\ &\qquad \mathbb{P}(b)\mathbb{P}(s)\mathbb{P}(g)\mathbb{P}(\sigma)\mathbb{P}(\beta|C) \quad (2.6.9)\end{aligned}$$

The detailed explanation of the above expression is as follows: a particular data shape $\boldsymbol{y_{b_i}}$ can be regarded as coming from one of the representative shapes $\boldsymbol{\beta_i}$ which comes from one of the classes $C$. The model assumes that the data shape $\boldsymbol{y_i}$ has arisen by such a representative shape that has been rotated by R, scaled by $a$ and translated by $\boldsymbol{t}$; this group action is represented by $g$. A sampling function $\boldsymbol{s}$ places $\boldsymbol{N}$ points around the boundary of the shape. Due to data collection errors, Gaussian noise $\sigma$ is added which perturbs the points from their original places. To compare the data to the representative shapes of each class the model assumes a bijection $b : [1,...n] \to [1,...n]$ relating each point of the data shape to a unique point of the idealised example shape. A pictorial explanation of expression (2.6.8) is given in figure (2.4). Figure (2.5a) shows the comparison between an "observed" rectangle (in red) and an "idealised" example shape $\beta$ from the same class. Figure (2.5b) shows the comparison between an "observed" circle (in red) and an "idealised" rectangle.

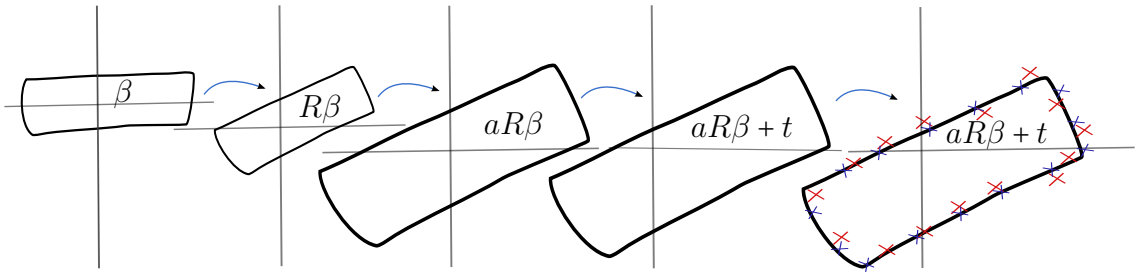
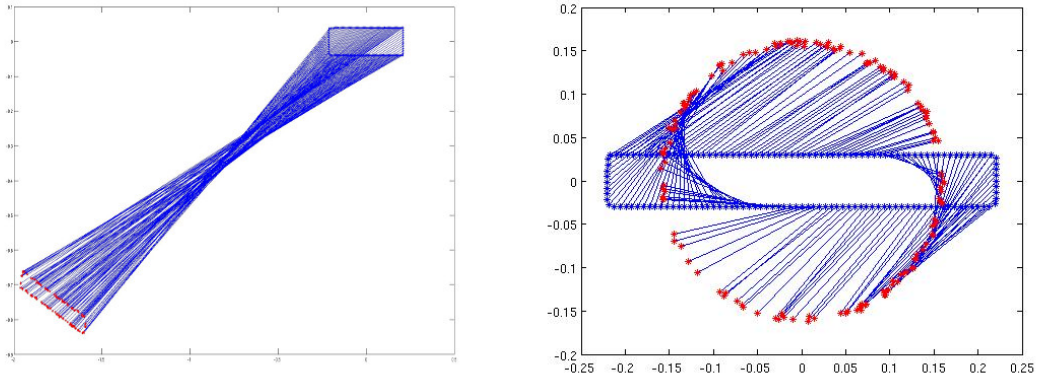



Figure 2.4: A pictorial representation of the observation model




(a) Comparison between shapes of the same class

(b) Comparison between shapes of different classes

Figure 2.5: A pictorial explanation of the observation model $\mathbb{P}(y|b, \beta, s, g, \sigma)$

By making appropriate changes of variables, it is easy to verify that the complete likelihood enjoys the following behaviours. It has translational symmetry under $\boldsymbol{y} \to \boldsymbol{y} + \boldsymbol{u}$ for any vector $\boldsymbol{u}$. It is invariant to rotations over the data shape $\boldsymbol{y} \to S\boldsymbol{y}$ where $S \in SO(2)$ and under a scaling $\boldsymbol{y} \to \lambda \boldsymbol{y}$ for $\lambda \in \mathbb{R}^+$, the likelihood scales as $\lambda^{-2n}$. We can illustrate these claims by the latter example: to compensate for the scaling of $y$ one may let $\sigma' = \frac{\sigma}{\lambda}$, $a' = \frac{a}{\lambda}$ and $t' = \frac{t}{\lambda}$. Then it is easy to check that the likelihood scales as claimed provided the priors scale in the appropriately. In particular, with our use of Jeffreys prior, which corresponds to the Haar measure on the variables, then this behaviour holds. Similar changes of variable (with the same behaviour of the priors) suffice to show invariance under translations and rotations. We will soon see the need to regulate divergences by modifying Jeffreys prior which will spoil the scaling behaviour unless the regulators are removed but translational and rotational invariance will be preserved throughout. However, when this is used to calculate the posterior, such scale factors cancel out so as to leave the posterior to be scale invariant.

There is one important matter to be discussed at this point. Although the likelihood appears to scale as $\lambda^{-2n}$, the posterior is invariant under such scalings i.e. $\mathbb{P}(C|y) = \mathbb{P}(C|\lambda y)$ which follows from the scaling of the likelihood $\mathbb{L}(y) = \lambda^{-2n}\mathbb{L}(\lambda y)$.

In particular, one can see by a change of variables that for a constant $b$, the likelihood satisfies the relationship: $\mathbb{L}(y|\sigma/b, a) = b^p\mathbb{L}(by|\sigma, ab) = b^{p-2n}\mathbb{L}(y|\sigma, ab)$ where $p$ is some integer. This shows that changes in the scale of $\sigma$ can be reinterpreted (up to a constant scaling of the likelihood) as changes in the scaling parameter, a. This generates a question as to whether the inclusion of both the scaling and noise parameters are needed in the model. Perhaps we could remove the scalings and then take only $\sigma$ into account or vise versa; in other words we will investigate whether both parameters are crucial to our analysis. We will ultimately answer this later on in this chapter when we calculate the posterior.

To investigate the impact of restricting our model to include only one of these variables, we have to evaluate Jeffreys prior when each of these parameters are removed from the calculation. We will investigate this in section [2.7.1].

## 2.7 Similarity transformations and noise model

If we want to describe a configuration that has a particular position, orientation and scale in space, we need to act on the configuration by a similarity transformation $g \in G$. Srivastava and Jermyn use a uniform model $\mathbb{P}(g)$ on the space $G$. In our work, we use a different model to describe similarity transformations and the noise taking into account our ignorance over them. To reflect our ignorance over the similarity transformations and the noise variance we chose to model them as the joint Jeffreys' prior over this space. Our ignorance in the distribution of these nuisance parameters stems from the fact that we will not have sufficient information of the properties of the database that we will discuss in chapter [3]. This is because we won't have access to sufficient geological data to form subjective priors. For this reason, Jeffreys prior is the most appropriate choice as it is unbiased with respect to the parameters. We shall see later in this chapter, that it is necessary to modify this slightly by introducing what will appear to be subjective priors to regulate divergences. However, we will only be interested in a certain limit where these priors reproduce the result arrived at with the Jeffreys prior we shall derive in section [2.9].

It is necessary to explain the choice of the prior model we made for similarity transformations and the noise variance $\sigma$. The prior distribution is the joint Jeffreys prior and was calculated over this complex joint five dimensional space. Jeffreys' prior [125] is a locally uniform [126] and non-informative prior whose density can be calculated as the square root of the determinant of the Fisher information matrix $\mathcal{I}(\phi)$ i.e.

$$\mathbb{J}(\phi) \propto \sqrt{\det(\mathcal{I}(\phi))} \propto \sqrt{\det \mathbb{E}\left[\frac{\partial \ln L}{\partial \phi_i}\frac{\partial \ln L}{\partial \phi_j}\right]} \tag{2.7.10}$$

where $L$ is the likelihood function which is differentiated over the parameters $\phi$.

This prior distribution has the special property that it is invariant under re-parametrizations of $\phi$ something that is called "Jeffreys' invariance" and it is required for the construction of non-informative priors. Jeffreys' invariance ensures that a change in the parametrisation of $\phi$ doesn't change the answer of an integration and should yield at the same result; this implies invariance under the diffeomorphism group because any changes of the parameters is just a diffeomorphism of that space of variables. Another special property of Jeffreys' prior is that it always corresponds to the left Haar measure; it is conventional to use the left Haar measure because this corresponds to an "active" transformation rather than a "passive" transformation. The Haar measure is invariant under the action of the group in the sense that $\int \mathscr{D}U = \int \mathscr{D}(Uv)$ for any group element $v$. For Jeffreys prior we have that $\int |J|dx = \int |J'|dx'$ where $|J'|$ and $x'$ are the variables in some other parameterisation.

Although Jeffreys prior violates the likelihood principle and is flat relative to the likelihood function, its choice is useful when we have no *a priori* knowledge for our problem and hence used as the most unbiased representative measure whose geometrical interpretation is in terms of our ignorance of the variables in question. In addition, Jeffreys prior can be improper for many models and although improper priors are allowed, they may produce improper posteriors. In the next section we calculate Jeffreys prior for similarity transformations $g \in G$ and the noise parameter $\sigma$.

### 2.7.1 Jeffreys prior

In this section we calculate the five dimensional joint Jeffreys prior for the similarity transformations $g \in G$ and the noise parameter $\sigma$. To do so, we will form the Fisher information matrix by differentiating the likelihood function with respect to the parameters over which the prior is built. These parameters are: the noise $\sigma$, the scalings $a$, the translations $\boldsymbol{t}$ and the rotations R. Since we are only encountering two dimensional shapes, we will regard translations to be two dimensional so we take into account both the $x$ and the $y$ component. This results in a $5 \times 5$ Fisher information matrix. The marginalised likelihood is:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,g,\sigma) &= \frac{1}{(2\pi)^n\sigma^{2n}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left|\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{\beta}(s(b_i^{-1})) + \boldsymbol{t}\right|^2\right) \\ &= \frac{1}{(2\pi)^n\sigma^{2n}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left|\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t}\right|^2\right) \end{aligned} \tag{2.7.11}$$

where we have substituted $\boldsymbol{v_i} = \boldsymbol{\beta}(s(b_i^{-1}))$ and $\boldsymbol{y_{b_i}} = \boldsymbol{y_i}$ for simplicity. The log marginalised likelihood is then:

$$\begin{aligned}\mathbb{L} = \log(\mathbb{P}(y|b,\beta,s,g,\sigma)) &= -n\log(2\pi) - 2n\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t}|^2 = \\ &= -n\log(2\pi) - 2n\log(\sigma) \\ &\quad - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t})\cdot(\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t})\end{aligned} \tag{2.7.12}$$

For the calculation of the scores of the Fisher matrix, we will calculate all the second derivatives of the log likelihood with respect to the parameters $\phi = \{\sigma, a, \boldsymbol{t}, \mathrm{R}\}$ and then compute the expectation of the derivatives with respect to $y$. The calculation of the derivatives will be described in the following sections. We firstly calculate all the diagonal terms of the Fisher matrix and then proceed to the computation of the off diagonal terms.

### Derivatives with respect to noise $\sigma$

The first term we calculate is the first diagonal entry of Fisher's matrix. We find the derivative of the log likelihood (2.7.12) with respect to the noise variance $\sigma$. The derivatives of the marginalised log-likelihood with respect to $\sigma$ were found to be:

$$\frac{\partial \mathbb{L}}{\partial \sigma} = -\frac{2n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n} |\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t}|^2 \tag{2.7.13}$$

$$\frac{\partial^2 \mathbb{L}}{\partial \sigma^2} = \frac{2n}{\sigma^2} - \frac{3}{\sigma^4}\sum_{i=1}^{n} |\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t}|^2. \tag{2.7.14}$$

To complete the calculation of Fisher's entry we must take the expectation of the score element (2.7.14) with respect to $y$. That is:

$$\begin{aligned}
\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial^2 \sigma}\right) &= \int \prod_i d^2 y_i \; \mathbb{P}(y|b,\beta,s,g,\sigma)) \frac{\partial^2 \mathbb{L}}{\partial^2 \sigma} \\
&= \frac{1}{(2\pi)^n \sigma^{2n}} \int \prod_i d^2 y_i \; \frac{2n}{\sigma^2} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n} |\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} - \boldsymbol{t}|^2\right) \\
&- \frac{1}{(2\pi)^n \sigma^{2n}} \int \prod_i d^2 y_i \; \frac{3}{\sigma^4}\sum_{i=1}^{n} |\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t}|^2 \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n} |\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} - \boldsymbol{t}|^2\right) \\
&= \frac{2n}{\sigma^2}\frac{(2\pi\sigma^2)^n}{(2\pi\sigma^2)^n} - \frac{1}{(2\pi)^n \sigma^{2n}}\frac{3}{\sigma^4} \int \prod_{j\neq i} d^2 y_j \exp\left(-\frac{1}{2\sigma^2}\sum_{j\neq i} |\boldsymbol{y_j} - a\mathrm{R}\boldsymbol{v_j} - \boldsymbol{t}|^2\right) \times \\
&\int \sum_{i=1}^{n} |\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} - \boldsymbol{t}|^2 \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n} |\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} - \boldsymbol{t}|^2\right) \\
&= \frac{2n}{\sigma^2} - \frac{1}{(2\pi)^n \sigma^{2n}}\frac{3}{\sigma^4}(2\pi\sigma^2)^{n-1}\sum_{i=1}^{n} 2\;\sigma^2(2\pi\sigma^2) = \frac{2n}{\sigma^2} - \frac{6n}{\sigma^2} = -\frac{4n}{\sigma^2}.
\end{aligned} \tag{2.7.15}$$

Thus, the diagonal entry of the Fisher matrix with respect to $\sigma$ is:

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial^2 \sigma}\right) = -\frac{4n}{\sigma^2}}$$

### Derivatives with respect to scalings $a$

The second term to be calculated is the diagonal term of the Fisher matrix with respect to $a$. The derivatives of the marginalised log likelihood with respect to scalings $a$ were found to be:

$$\begin{aligned}\frac{\partial \mathbb{L}}{\partial a} &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left[(-\mathrm{R}\boldsymbol{v_i})\cdot(\boldsymbol{y_i}-a\mathrm{R}\boldsymbol{v_i}+\boldsymbol{t})+(-\mathrm{R}\boldsymbol{v_i})\cdot(\boldsymbol{y_i}-a\mathrm{R}\boldsymbol{v_i}+\boldsymbol{t})\right] = \\ &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left[(-2\mathrm{R}\boldsymbol{v_i})\cdot(\boldsymbol{y_i}-a\mathrm{R}\boldsymbol{v_i}+\boldsymbol{t})\right] = \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{n}\left[(\mathrm{R}\boldsymbol{v_i})\cdot(\boldsymbol{y_i}-a\mathrm{R}\boldsymbol{v_i}+\boldsymbol{t})\right] \qquad (2.7.16)\\ \frac{\partial^2 \mathbb{L}}{\partial a^2} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}\left[(\mathrm{R}\boldsymbol{v_i})\cdot(-\mathrm{R}\boldsymbol{v_i})\right] = -\frac{1}{\sigma^2}\sum_{i=1}^{n}(\mathrm{R}\boldsymbol{v_i})\cdot(\mathrm{R}\boldsymbol{v_i}) \\ &= -\frac{1}{\sigma^2}\sum_{i=1}^{n}\left|\mathrm{R}\boldsymbol{v_i}\right|^2. \qquad (2.7.17)\end{aligned}$$

However, because rotations R act as isometries, expression (2.7.17) becomes $-\frac{1}{\sigma^2}\sum_{i=1}^{n}|\boldsymbol{v_i}|^2$. To complete the calculation of the Fisher entry one must calculate the expectation of equation (2.7.17) with respect to $y$. This is:

$$\begin{aligned}\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial a^2}\right) &= -\frac{1}{(2\pi)^n\sigma^{2n}}\int\prod_i d^2y_i\,\frac{1}{\sigma^2}\sum_{i=1}^{n}|\boldsymbol{v_i}|^2\,\mathbb{P}(y|b,\beta,s,g,\sigma) \\ &= -\frac{1}{(2\pi)^n\sigma^{2n}}\int\prod_i d^2y_i\,\frac{1}{\sigma^2}\sum_{i=1}^{n}|\boldsymbol{v_i}|^2\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{y_i}-a\mathrm{R}\boldsymbol{v_i}+\boldsymbol{t}|^2\right) \\ &= -\frac{1}{\sigma^2}\sum_{i=1}^{n}|\boldsymbol{v_i}|^2. \qquad (2.7.18)\end{aligned}$$

Thus, the diagonal entry of the Fisher matrix with respect to scalings $a$ is:

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial a^2}\right) = -\frac{1}{\sigma^2}\sum_{i=1}^{n}|\boldsymbol{v_i}|^2}$$

### Derivatives with respect to translations $\boldsymbol{t}$

We now calculate the diagonal term of the Fisher matrix with respect to translations. We will calculate the translations' derivatives with respect to its constituent components $t_x$ and $t_y$. Writing the equation of the marginalised likelihood (2.7.12) in component form, we have:

$$\begin{aligned}\mathbb{L} &= -2n\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{y_i}-a\mathrm{R}\boldsymbol{v_i}-\boldsymbol{t}|^2 = \\ &= -2n\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left[(y_i^x - a\,(\mathrm{R}v_i)^x + t_x)^2 + (y_i^y - a(\mathrm{R}v_i)^y + t_y)^2\right] \qquad (2.7.19)\end{aligned}$$

Taking the derivatives of equation (2.7.19) with respect to the $x$ component of translations first:

$$\begin{aligned}\frac{\partial \mathbb{L}}{\partial t_x} &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n} 2(y_i^x - a\mathrm{R}v_i^x + t_x) = \\ &= -\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i^x - a\mathrm{R}v_i^x + t_x) \qquad (2.7.20)\\ \frac{\partial^2 \mathbb{L}}{\partial {t_x}^2} &= -\frac{1}{\sigma^2}\sum_{i=1}^{n} 1 = -\frac{n}{\sigma^2}. \qquad (2.7.21)\end{aligned}$$

Similarly and by symmetry, the derivative of equation (2.7.19) with respect to the $y$ component of translations is:

$$\begin{aligned}\frac{\partial \mathbb{L}}{\partial t_y} &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n} 2(y_i^y - a\mathrm{R}v_i^y + t_y) = \\ &= -\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i^y - a\mathrm{R}v_i^y + t_y) \qquad (2.7.22)\\ \frac{\partial^2 \mathbb{L}}{\partial {t_y}^2} &= -\frac{1}{\sigma^2}\sum_{i=1}^{n} 1 = -\frac{n}{\sigma^2} \qquad (2.7.23)\end{aligned}$$

The expectation of equation (2.7.21) and (2.7.23) enter into the Fisher information matrix; they are equal and since they are constants, their expectation with respect to $y$ is also a constant and equal to:

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial t_x^2}\right) = \mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial t_y^2}\right) = -\frac{n}{\sigma^2}}$$

**Derivatives with respect to rotations** R

Rotations can be represented in various ways. However, the calculation of the derivatives of Fisher information matrix with respect to rotations becomes significantly simpler if we represent rotations R in standard form parametrised by $\theta$:

$$\mathrm{R} = \begin{pmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{pmatrix}$$

Writing equation (2.7.19) in component form including rotations, it becomes:

$$\mathbb{L} = -2n\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\Big[(y_i{}^x - a(\cos\theta v_i{}^x - \sin\theta v_i{}^y) + t^x)^2 + (y_i{}^y - a(\sin\theta v_i{}^x + \cos\theta v_i{}^y) + t^y)^2\Big] \tag{2.7.24}$$

We now find the derivative of equation (2.7.24) with respect to $\theta$ which is:

$$\begin{aligned}\frac{\partial\mathbb{L}}{\partial\theta} &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\Big[2\left[y_i{}^x - a(\cos\theta v_i{}^x - \sin\theta v_i{}^y) + t^x\right]\cdot\left[-a(-\sin\theta v_i^x - \cos\theta v_i^y)\right] \\ &\qquad + 2\left[y_i{}^y - a(\sin\theta v_i{}^x + \cos\theta v_i{}^y) + t^y\right]\cdot\left[-a(\cos\theta v_i^x - \sin\theta v_i^y)\right]\Big] = \\ &= -\frac{1}{\sigma^2}\sum_{i=1}^{n}\Big[a\left[(y_i{}^x - a\left(\cos\theta v_i{}^x - \sin\theta v_i{}^y\right) + t^x)\left(\sin\theta v_i{}^x + \cos\theta v_i{}^y\right)\right] \\ &\qquad - a\left[(y_i{}^y - a(\sin\theta v_i{}^x + \cos\theta v_i{}^y) + t^y)(\cos\theta v_i{}^x - \sin\theta v_i{}^y)\right]\Big].\end{aligned} \tag{2.7.25}$$

For simplicity we will differentiate the above expression's brackets separately and combine the result. For the first term in square brackets we find that the second derivative is:

$$\begin{aligned}&-\frac{1}{\sigma^2}\sum_{i=1}^{n}\Big[a(a(\sin\theta v_i{}^x - \cos\theta v_i{}^y))(\sin\theta v_i{}^x + \cos\theta v_i{}^y) \\ &\qquad + a(y_{b_i}{}^x - a(\cos\theta v_i{}^x - \sin\theta v_i{}^y) + t^x)(\cos\theta v_i{}^x - \sin\theta v_i{}^y))\Big] \\ &= -\frac{1}{\sigma^2}\sum_{i=1}^{n}\Big[a^2(\sin\theta v_i{}^x + \cos\theta v_i{}^y)^2 \\ &\qquad + a(y_i{}^x - a(\cos\theta v_i{}^x - \sin\theta v_i{}^y) + t^x)(\cos\theta v_i{}^x - \sin\theta v_i{}^y))\Big].\end{aligned} \tag{2.7.26}$$

For the second term in the square brackets of equation (2.7.25) the second derivative with respect to $\theta$ is:

$$
\begin{aligned}
&-\frac{1}{\sigma^2}\sum_{i=1}^{n}\Big[-a\big(-a(\cos\theta {v_i}^x-\sin\theta {v_i}^y)(\cos\theta {v_i}^x-\sin\theta {v_i}^y)\\
&\qquad -a\big({y_i}^y-a(\sin\theta {v_i}^x+\cos\theta {v_i}^y)+t^y\big)(-\sin\theta {v_i}^x-\cos\theta {v_i}^y)\big)\Big]=\\
&=-\frac{1}{\sigma^2}\sum_{i=1}^{n}\Big[a^2(\cos {v_i}^x-\sin\theta {v_i}^y)^2\\
&\qquad +a\big({y_i}^y-a(\sin\theta {v_i}^x+\cos\theta {v_i}^y)+t^y\big)(\sin\theta {v_i}^x+\cos\theta {v_i}^y)\big)\Big] \qquad (2.7.27)
\end{aligned}
$$

Again, for simplicity we will take the expectations of these two expressions separately and combine them in the end. The expectation of equation (2.7.26) with respect to $y$ is:

$$
\begin{aligned}
&\mathbb{E}\Bigg(-\frac{1}{\sigma^2}\sum_{i=1}^{n}\Big[a^2(\sin\theta {v_i}^x+\cos\theta {v_i}^y)^2\\
&\qquad +a\big({y_i}^x-a(\cos\theta {v_i}^x-\sin\theta {v_i}^y)+t^x\big)(\cos\theta {v_i}^x-\sin\theta {v_i}^y)\big)\Big]\Bigg)=\\
&=-\frac{1}{(2\pi)^n\sigma^{2n}\sigma^2}\int\prod_{i=1}^{n}d^2y_i\sum_{i=1}^{n}\Big[a^2(\sin\theta {v_i}^x+\cos\theta {v_i}^y)^2+\\
&\qquad a\big({y_i}^x-a(\cos\theta {v_i}^x-\sin\theta {v_i}^y)+t^x\big)(\cos\theta {v_i}^x-\sin\theta {v_i}^y)\big)\Big]\times\\
&\qquad \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{y_i}-a\mathrm{R}\boldsymbol{v_i}+\boldsymbol{t}|^2\right) \qquad (2.7.28)
\end{aligned}
$$

However we notice that the second term of the integrand is odd in $({y_i}^x-a(\cos\theta {v_i}^x-\sin\theta {v_i}^y)+t^x)$ so when integrated against the marginalised likelihood which is an even function, the result will be zero. Thus, the expectation of the above expression is: $-\frac{1}{\sigma^2}\sum_{i=1}^{n}a^2(\sin\theta {v_i}^x+\cos\theta {v_i}^y)^2$. Similarly, the expectation of the second term which we calculated in equation (2.7.27) is:

$$\mathbb{E}\Bigg( -\frac{1}{\sigma^2}\sum_{i=1}^{n} a^2(\cos {v_i}^x - \sin\theta {v_i}^y)^2 + \\ a({y_i}^y - a(\sin\theta {v_i}^x + \cos\theta {v_i}^y) + t^y)(\sin\theta {v_i}^x + \cos\theta {v_i}^y))\Bigg) = \\ = -\frac{1}{(2\pi)^n\sigma^{2n}\sigma^2}\int \prod_{i=1}^{n} d^2 y_i \sum_{i=1}^{n} a^2(\cos {v_i}^x - \sin\theta {v_i}^y)^2 + \\ a\left({y_i}^y - a(\sin\theta {v_i}^x + \cos\theta {v_i}^y) + t^y)(\sin\theta {v_i}^x + \cos\theta {v_i}^y)\right) \times \\ \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n} |\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t}|^2\right). \tag{2.7.29}$$

The expectation of the second term of the above expression with respect to $y$ is odd in $({y_i}^y - a(\sin\theta {v_i}^x + \cos\theta {v_i}^y) + t^y))$ so when integrated against the marginalised likelihood (2.6.8), which is an even function, the result will be zero. The result of the above expression is: $-\frac{1}{\sigma^2}\sum_{i=1}^{n} a^2(\cos {v_i}^x - \sin\theta {v_i}^y)^2$. Combining the results from both expressions we have:

$$\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial\theta^2}\right) = -\frac{1}{\sigma^2}\sum_{i=1}^{n} a^2\left[(\cos\theta {v_i}^x - \sin\theta {v_i}^y)^2 + (\sin\theta {v_i}^x + \cos\theta {v_i}^y)^2\right] = \\ = -\frac{1}{\sigma^2}\sum_{i=1}^{n} a^2(({v_i}^x)^2 + ({v_i}^y)^2) = \\ = -\frac{1}{\sigma^2}\sum_{i=1}^{n} a^2|\boldsymbol{v_i}|^2 \tag{2.7.30}$$

Thus the diagonal element of the Fisher information matrix with respect to rotations $\theta$, which is the last diagonal term to be calculated, is:

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial\theta^2}\right) = -\frac{1}{\sigma^2}\sum_{i=1}^{n} a^2|\boldsymbol{v_i}|^2}$$

### Derivatives of cross terms

In this section we carry out the differentiation of the log-likelihood for all the terms that fill the off diagonals of the Fisher information matrix. To start with, we calculate the score with respect to $\sigma$ and scalings $a$. This is:

$$\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial a} = \frac{1}{\sigma^3}\sum_{i=1}^{n}\left[(\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t})\cdot(-\mathrm{R}\boldsymbol{v_i}) + (\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i})\cdot(-\mathrm{R}\boldsymbol{v_i})\right] = \\ = \frac{-2}{\sigma^3}\sum_{i=1}^{n}(\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t})\cdot(\mathrm{R}\boldsymbol{v_i}). \tag{2.7.31}$$

The expectation of equation (2.7.31) with respect to $y$ is:

$$\mathbb{E}\left(\frac{-2}{\sigma^3}\sum_{i=1}^{n}(\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t}) \cdot (\mathrm{R}\boldsymbol{v_i})\right) = 0, \tag{2.7.32}$$

because the expression is odd in $(\boldsymbol{y_i} - a\mathrm{R}\boldsymbol{v_i} + \boldsymbol{t})$ so when integrated against the likelihood which is an even function, the result will be zero. Thus:

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial a}\right) = 0}$$

The next calculation is for $\sigma$ and the $x$ component of translations $\boldsymbol{t}$:

$$\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial t_x} = \frac{1}{\sigma^3}\left[2\sum_{i=1}^{n}({y_i}^x - a\mathrm{R}{v_i}^x + t^x)\right] \tag{2.7.33}$$

The expectation of equation (2.7.33) is:

$$\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial t_x}\right) = \mathbb{E}\left(\frac{1}{\sigma^3}\left[2\sum_{i=1}^{n}({y_i}^x - a\mathrm{R}{v_i}^x + t^x)\right]\right) = 0. \tag{2.7.34}$$

The expression is odd in $({y_i}^x - a\mathrm{R}{v_i}^x + t^x)$ so when integrated against the likelihood which is an even function, the result will be zero. Similarly and by symmetry for the $y$ component of translations we have:

$$\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial t_y} = \frac{1}{\sigma^3}\left[2\sum_{i=1}^{n}({y_i}^y - a\mathrm{R}{v_i}^y + t^y)\right]. \tag{2.7.35}$$

The expectation of equation (2.7.35) is:

$$\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial t_y}\right) = \mathbb{E}\left(\frac{1}{\sigma^3}\left[2\sum_{i=1}^{n}({y_i}^y - a\mathrm{R}{v_i}^y + t^y)\right]\right) = 0. \tag{2.7.36}$$

Thus, the expectation of (2.7.33) and (2.7.35) is:

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial t_x}\right) = \left(\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial t_y}\right) = 0}$$

Turning now to the $\sigma\theta$ component of the Fisher information matrix, we have:

$$\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial\theta} = \frac{2}{\sigma^3}\sum_{i=1}^{n}\big[a(y_i^x - a(\cos\theta {v_i}^x - \sin\theta {v_i}^y) + t^x)(\sin\theta {v_i}^x + \cos\theta {v_i}^y) \\ -a(y_i^y - a(\sin\theta {v_i}^x + \cos\theta {v_i}^y) + t^y)(\cos\theta {v_i}^x - \sin\theta {v_i}^y)\big] \tag{2.7.37}$$

The expectation of equation (2.7.37) is:

$$\mathbb{E}\left(\frac{2}{\sigma^3}\sum_{i=1}^{n}\Big[a(y_i^x - a(\cos\theta {v_i}^x - \sin\theta {v_i}^y) + t^x)(\sin\theta {v_i}^x + \cos\theta {v_i}^y) \\ -a(y_i^y - a(\sin\theta {v_i}^x + \cos\theta {v_i}^y) + t^y)(\cos\theta {v_i}^x - \sin\theta {v_i}^y)\Big]\right) = 0. \tag{2.7.38}$$

The above expectation is zero because it is odd in $(y_i{}^x - a(\sin\theta {v_i}^x + \cos\theta {v_i}^y) + t^x)$ and in $(y_i^y - a(\sin\theta {v_i}^x + \cos\theta {v_i}^y) + t^y)$ so that when integrated against the even likelihood the result will be zero. Thus:

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial\sigma\partial\theta}\right) = 0}$$

The derivative of the log-likelihood with respect to the scalings $a$ and the $x$ component of translations $t_x$ is:

$$\frac{\partial^2 \mathbb{L}}{\partial t_x \partial a} = -\frac{1}{\sigma^2}\sum_{i=1}^{n}(-(\mathrm{R}v_i)^x) = \frac{\sum_{i=1}^{n}\mathrm{R}v_i^x}{\sigma^2} \tag{2.7.39}$$

The expectation of equation (2.7.39) with respect to $y$ is a constant since it is independent of $y$ and thus:

$$\mathbb{E}\left(\frac{\sum_{i=1}^{n}(\mathrm{R}v_i)^x}{\sigma^2}\right) = \frac{\sum_{i=1}^{n}\mathrm{R}v_i^x}{\sigma^2} \tag{2.7.40}$$

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial t_x \partial a}\right) = \frac{\sum_{i=1}^{n}\mathrm{R}v_i^x}{\sigma^2}}$$

By symmetry, the expression for the $y$ component of the translations $t_y$ and the scalings $a$ is:

$$\frac{\partial^2 \mathbb{L}}{\partial t_y \partial a} = -\frac{1}{\sigma^2}\sum_{i=1}^{n}(-(\mathrm{R}v_i)^y) = \frac{\sum_{i=1}^{n}\mathrm{R}v_i^y}{\sigma^2} \tag{2.7.41}$$

The expectation of equation (2.7.41) is a constant since it is independent of $y$:

$$\mathbb{E}\left(\frac{\sum_{i=1}^{n}(\mathrm{R}v_i)^y}{\sigma^2}\right) = \frac{\sum_{i=1}^{n}\mathrm{R}v_i^y}{\sigma^2} \tag{2.7.42}$$

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial t_y \partial a}\right) = \frac{\sum_{i=1}^{n}\mathrm{R}v_i^y}{\sigma^2}}$$

We also note that the second derivative of the likelihood with respect to both the $x$ and the $y$ component of translations is zero:

$$\frac{\partial^2 \mathbb{L}}{\partial t_y \partial t_x} = 0 \tag{2.7.43}$$

so that the expectation of this term is also equal to zero.

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial t_y \partial t_x}\right) = 0}$$

The derivative of the log-likelihood with respect to the $x$ component of translations and the rotations parametrised by $\theta$ is:

$$\begin{aligned}\frac{\partial^2 \mathbb{L}}{\partial t_x \partial \theta} &= \frac{\partial}{\partial t_x}\left[-\frac{1}{\sigma^2}\sum_{i=1}^{n} a\left(y_i{}^x - a(\cos\theta v_i{}^x - \sin\theta v_i{}^y) + t^x\right)(\sin\theta v_i{}^x + \cos\theta v_i{}^y)\right] = \\ &= -\frac{1}{\sigma^2}\sum_{i=1}^{n} a(\sin\theta v_i{}^x + \cos\theta v_i{}^y)\end{aligned} \tag{2.7.44}$$

The expectation of equation (2.7.44) with respect to $y$ is:

$$\mathbb{E}\left(-\frac{1}{\sigma^2}\sum_{i=1}^{n} a(\sin\theta v_i{}^x + \cos\theta v_i{}^y)\right) = -\frac{a}{\sigma^2}\sum_{i=1}^{n}(\sin\theta v_i{}^x + \cos\theta v_i{}^y). \tag{2.7.45}$$

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial t_x \partial \theta}\right) = -\frac{a}{\sigma^2}\sum_{i=1}^{n}(\sin\theta v_i{}^x + \cos\theta v_i{}^y)}$$

Similarly, the derivative of the $y$ component of the translations and the rotations $\theta$ is:

$$\frac{\partial^2 \mathbb{L}}{\partial t_y \partial \theta} = \frac{\partial}{\partial t_y}\left[-\frac{1}{\sigma^2}\sum_{i=1}^{n}(-a)(y_i{}^y - a(\cos\theta v_i{}^x + \sin\theta v_i{}^x) + t^y)(\cos\theta v_i{}^x - \sin\theta v_i{}^y)\right] =$$
$$= \frac{1}{\sigma^2}\sum_{i=1}^{n} a(\cos\theta v_i{}^x - \sin\theta v_i{}^y) \tag{2.7.46}$$

The expectation of equation (2.7.46) is:

$$\mathbb{E}\left(\frac{1}{\sigma^2}\sum_{i=1}^{n} a(\cos\theta v_i{}^x - \sin\theta v_i{}^y)\right) = \frac{a}{\sigma^2}\sum_{i=1}^{n}(\cos\theta v_i{}^x - \sin\theta v_i{}^y) \tag{2.7.47}$$

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial t_y \partial \theta}\right) = \frac{a}{\sigma^2}\sum_{i=1}^{n}(\cos\theta v_i{}^x - \sin\theta v_i{}^y)}$$

Finally, the derivative of the likelihood with respect to the rotations $\theta$ and the scalings $a$ is:

$$\begin{aligned}\frac{\partial^2 \mathbb{L}}{\partial \theta \partial a} = -\frac{1}{\sigma^2}\sum_{i=1}^{n}\big[&y_{b_i}{}^x - a(\cos\theta v_i{}^x - \sin\theta v_i{}^y) + t^x)(\cos\theta v_i{}^y + \sin\theta v_i{}^x)\\ &+a(-(\cos\theta v_i{}^x - \sin\theta v_i{}^y))(\sin\theta v_i{}^x + \cos\theta v_i{}^y)\\ &-(y_{b_i}{}^y - a(\sin\theta v_i{}^x + \cos\theta v_i{}^y) + t^y)(\cos\theta v_i{}^x - \sin\theta v_i{}^y)\\ &+a(\sin\theta v_i{}^x + \cos\theta v_i{}^y)(\cos\theta v_i{}^x - \sin\theta v_i{}^y)\big]\end{aligned} \tag{2.7.48}$$

However, we note here that the two expressions $(y_i{}^x - a(\cos\theta v_i{}^x - \sin\theta v_i{}^y) + t^x)$ and $(y_i{}^y - a(\sin\theta v_i{}^x + \cos\theta v_i{}^y) + t^y)$ are odd and when integrated against the even likelihood they will give zero as a result. The expectation of equation (2.7.48) is zero overall:

$$\boxed{\mathbb{E}\left(\frac{\partial^2 \mathbb{L}}{\partial \theta \partial a}\right) = 0}$$

We have calculated and now have all the terms to form Fisher's information matrix. For convenience in the equations we denote the variables associated to each row and column. The matrix $\mathcal{I}(\phi)$ is:

$$
\begin{array}{c}
\\
\sigma \\
a \\
t_x \\
t_y \\
\theta
\end{array}
\begin{array}{c}
\begin{array}{ccccc}
\sigma & a & t_x & t_y & \theta
\end{array} \\
\left(\begin{array}{ccccc}
-\frac{4n}{\sigma^2} & 0 & 0 & 0 & 0 \\
0 & -\frac{\sum_{i=1}^n |\boldsymbol{v_i}|^2}{\sigma^2} & \frac{\sum_{i=1}^n \cos\theta v_i^x - \sin\theta v_i^y}{\sigma^2} & \frac{\sum_{i=1}^n \sin\theta v_i^x + \cos\theta v_i^y}{\sigma^2} & 0 \\
0 & \frac{\sum_{i=1}^n \cos\theta v_i^x - \sin\theta v_i^y}{\sigma^2} & -\frac{n}{\sigma^2} & 0 & -a\frac{\sum_{i=1}^n \sin\theta v_i^x + \cos\theta v_i^y}{\sigma^2} \\
0 & \frac{\sum_{i=1}^n \sin\theta v_i^x + \cos\theta v_i^y}{\sigma^2} & 0 & -\frac{n}{\sigma^2} & a\frac{\sum_{i=1}^n \cos\theta v_i^x - \sin\theta v_i^y}{\sigma^2} \\
0 & 0 & -a\frac{\sum_{i=1}^n \sin\theta v_i^x + \cos\theta v_i^y}{\sigma^2} & a\frac{\sum_{i=1}^n \cos\theta v_i^x - \sin\theta v_i^y}{\sigma^2} & -a^2\frac{\sum_{i=1}^n |\boldsymbol{v_i}|^2}{\sigma^2}
\end{array}\right)
\end{array}
$$

In order to calculate Jeffreys' prior, we have to compute the determinant of $\mathcal{I}$. This is a long-running calculation and the details can be found in the Appendix. Here, we present the final result of this calculation. The determinant of Fisher's information matrix was found to be:

$$|\mathcal{I}(\phi)| = -\frac{a^2 n^3}{\sigma^{10}} \mathrm{Var}^2(\boldsymbol{v}) \tag{2.7.49}$$

where $\mathrm{Var}(\boldsymbol{v}) = \mathrm{Var}(\boldsymbol{\beta}(s(b_i^{-1}))$. The overall Jeffrey's prior is then the square root of the above expression:

$$\mathbb{J} \propto \frac{\sqrt{n^3} a \mathrm{Var}(\boldsymbol{v})}{\sigma^5}$$

It is standard practice to only take into account the parameters we partitioned over and absorb any other factors and constants into normalisation, so Jeffreys prior is proportional to:

$$\mathbb{J} \propto \frac{a\ \mathrm{Var}(\boldsymbol{v})}{\sigma^5} \tag{2.7.50}$$

To conclude, the model we use for the solution of the classification problem for similarity transformations and the noise variance is: $\mathbb{P}(g, \sigma) \propto \frac{a\ \mathrm{Var}(\boldsymbol{v})}{\sigma^5}$. In the next section, we discuss how Jeffreys prior can be used for the integration over similarity transformations and the noise variance in order to achieve the desired approximation of the marginalised likelihood and hence the desired shape classification.

Before continuing we pause to point out to the reader the different Jeffreys priors which arise depending upon our choice of variables in the model. We now investigate what Jeffreys prior is in case we remove either the scalings $a$ or the noise variance $\sigma$ as we mentioned in section [2.6]. Keeping $a$ and removing $\sigma$ Jeffreys prior is calculated to be $n\ a\ \mathrm{Var}(v)$. We will keep this result to investigate the behaviour of the posterior when we perform the integration over the scalings in section [2.8.4]. Similarly, the resulting Jeffreys prior when we keep $\sigma$ and remove $a$ the calculated Jeffreys prior is found to be $\frac{\sqrt{\mathrm{Var}(v)}}{\sigma^3}$. We will see the behaviour of the posterior when using this result in section [2.8.5].

Although the remainder of this thesis is based upon keeping all 5 variables we will point out how the result of the posterior changes with the choices we mention above. The effects of removing one of the variables do not seem to be advantageous which we believe gives weight to the decision to include all five parameters. We will present these modifications in section [2.8].

## 2.8 Solving the classification problem

In the previous sections we discussed the models needed to construct the marginalised likelihood in order to perform the desired classification. The classification of a given shape in a predetermined class is done by maximising the posterior probability $\mathbb{P}(C|\boldsymbol{y})$ of the class given the data. To perform MAP, one has to integrate over all nuisance parameters and build the likelihood. To do this we will have to map the data shapes to the sample shapes of the existing classes and then compare them. As mentioned before, in previous work, *e.g.* Dryden and Mardia [14] or Srivastava and Jermyn [12], an algorithmic approach was taken to the integrals over the group $G$ thus providing an approximation of the posterior probability. In particular, Monte Carlo integration was used for the integrals over the shape variable $\beta$ and the samplings $s$. In [12], the integral over transformations $g$ and the summation over bijections $b$, which is the joint registration and alignment problem, finds the optimum rotation, scaling and translation which minimises the Euclidean distance between two configurations. The integration over this group was carried out by maximizing

the integrand over the integration variables by using both the Procrustes and the Hungarian algorithm to compute a zeroth order Laplace approximation.

The main contribution of this thesis is the analytic calculation of the Bayesian integrals in expression (2.6.9). In our work, we carry out the geometrical integration over the similarity transformations' group, and the integration over the noise parameter $\sigma$ analytically, resulting in a closed form expression. After the transformational integrations are carried out and the result is in a closed form, the remaining integrals of the likelihood (2.6.9) are approximated by simple Monte Carlo techniques. In the next section we will describe how the integration of the rigid transformations was performed and discuss the results.

### 2.8.1 Calculation of the posterior

For the calculation of the Maximum a Posteriori approximation we need to calculate the integrated likelihood which is represented by the model in equation (2.6.9). The likelihood will be integrated over the nuisance parameters that were marginalised over and that give rise to the formation of the data. Assuming that we have at our disposal a set of planar shapes such that each is represented by $n$ two dimensional points around its boundary the marginalised likelihood is:

$$\mathbb{P}(y|b,\beta,s,g,\sigma) = \frac{1}{(2\pi)^n\sigma^{2n}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left|\boldsymbol{y}_{\boldsymbol{b_i}} - a\mathrm{R}\boldsymbol{\beta}(s(b_i^{-1})) + \boldsymbol{t}\right|^2\right) \tag{2.8.51}$$

where $\left(\frac{1}{2\pi\sigma^2}\right)^n$ is the normalization constant for the collection of $n$ points for each shape. The likelihood for the complete data set is then:

$$\mathbb{P}(y|C) = \sum_{b\in\mathcal{B}}\int \mathscr{D}\beta\ \mathscr{D}s\ \mathscr{D}g\ d\sigma\ \frac{1}{(2\pi)^n\sigma^{2n}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left|\boldsymbol{y}_{\boldsymbol{b(i)}} - a\mathrm{R}\boldsymbol{\beta}(s(b_i^{-1})) + \boldsymbol{t}\right|^2\right)\times$$
$$\mathbb{P}(b)\mathbb{P}(s)\mathbb{P}(g,\sigma)\mathbb{P}(\beta|C) \tag{2.8.52}$$

Of particular interest is the Jeffreys prior we imposed over $g \in G$ and the noise $\sigma$. In contrast to expression (2.6.9) the calculation of Jeffreys prior has taught us

how to combine the priors on $\sigma$ and $g$ so that the expression is now constructed to be invariant under reparametrisation of the nuisance parameters. Unfortunately, the result of integrating the complete likelihood against Jeffreys prior was found to lead to a divergence which in turn leads to a divergent posterior distribution. It is worth mentioning that Jeffreys prior is improper with respect to its parameters. To alleviate the divergence that appeared with the usage of Jeffreys prior, we introduced regulators that regularise the prior and explicitly calculate the resulting integral. The regularization of divergent integrals is a common technique that is used to remove divergences, calculate the integrals and then restore the divergence, hence the original result, by removing the regulators. The result of the integration should not depend on the chosen regularization and the way to do so is to parametrize the integral in terms of the regulators; taking the limit of the regulators must give back the initial integral. There are many types of regularizations, for example the momentum cutoff or the dimensional regularization are famous in physics; the choice really lies within the problem. Nevertheless, they all have the same use: to alleviate all divergences and give a finite result. We chose our regulators so that the end result of the integration with respect to the group actions of $G$ and $\sigma$ in the limit is the same as the initial result when using Jeffreys prior.

To overcome the divergence we used a regularised version of Jeffreys prior as the measure for the integration over the group $G$. We employed prior distributions on the parameters $g \in G$ that would act as regulators of the divergences that Jeffreys prior introduced and make the integrals converge and also smooth out the domain of each variable. The particular priors were chosen as reasonable choices that also reflect our beliefs in these parameters and our knowledge about the objects. For this reason, a Gaussian prior was introduced for translations which had the effect of artificially removing the previous translational invariance of the posterior with a physical interpretation of limiting the size of the 2 dimensional domain $\mathbb{R}^2$ in which the shape points lie. For the rest of the variables, a flat prior was used for rotations, a $\Gamma$ prior for the noise parameter $\sigma$. A Rayleigh prior was used for scalings $a$, which had the effect of breaking the scaling behaviour of the likelihood and thus braking the scale invariance of the posterior by effectively limiting the range of scales

considered.

In this thesis we will only work with the regularised version of Jeffreys prior. In the next section this new route employed will be discussed. We will show how we can alleviate the divergence with the help of the regulators and how we can return to the old result by removing them. The geometrical meaning and the origin of the divergence will also be discussed as a conclusion of the usage of the particular priors.

### 2.8.2 Integration of translations $t$ with a Gaussian prior

The expression that needs to be integrated with respect to translations is:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,g,\sigma) &= \mathbb{P}(y|b,\beta,s,t,\mathrm{R},a,\sigma)\\ &= \frac{1}{(2\pi)^n\sigma^{2n}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{y_i}-a\mathrm{R}\boldsymbol{\beta}(s(b_i^{-1}))-\boldsymbol{t}|^2\right)\end{aligned} \tag{2.8.53}$$

Extracting the translational dependence from expression (2.8.53) one has:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,t,\mathrm{R},a,\sigma) &= \frac{1}{Z}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{y_i}-a\mathrm{R}\boldsymbol{v_i}-\boldsymbol{t}|^2\right) =\\ &= \frac{1}{Z}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{Y_i}-\boldsymbol{t}|^2\right)\end{aligned} \tag{2.8.54}$$

where we have defined: $\boldsymbol{Y_i} = \boldsymbol{y_i} - a\mathrm{R}\boldsymbol{\beta}(s(b_i^{-1}))$. We had found that integrating translations against Jeffreys prior produces the first divergence of the result. To regulate the divergence that the integration of translations introduces we impose a Gaussian prior which has an effect on the domain of translations. Such a prior has as an effect to smooth out the divergence and effectively limit the two-dimensional domain. It provides a cut-off by "concentrating" the favoured translations around the mean of the prior Gaussian, which is of the form:

$$\mathbb{P}(t) = \left(\frac{1}{\sqrt{2\pi}D\sigma}\right)^2\exp\left(-\frac{|\boldsymbol{t}|^2}{2\sigma^2D^2}\right) \tag{2.8.55}$$

where $D$ is the regulator we introduced. At the end of the calculation we take the limit $D \to \infty$. Combining equations (2.8.54) and (2.8.55), the expression that needs to be integrated is:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,\mathrm{R},a,\sigma) &= \iint d^2\boldsymbol{t}\ \ \mathbb{P}(y|b,\beta,s,t,\mathrm{R},a,\sigma)\mathbb{P}(t) = \\ &= \frac{1}{(2\pi)^n\sigma^{2n}}\frac{1}{2\pi D^2\sigma^2}\iint d^2\boldsymbol{t}\ \ \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{Y_i}-\boldsymbol{t}|^2\right)\exp\left(-\frac{|\boldsymbol{t}|^2}{2\sigma^2D^2}\right)\end{aligned} \tag{2.8.56}$$

We extract the normalization constant of the distributions which is now:

$$\frac{1}{Z} = \frac{1}{(2\pi)^n\sigma^{2n}2\pi D^2\sigma^2} = \frac{1}{(2\pi)^{n+1}\sigma^{2n+2}D^2} \tag{2.8.57}$$

Expression (2.8.56) becomes:

$$\mathbb{P}(y|b,\beta,s,\mathrm{R},a,\sigma) = \frac{1}{Z}\iint d^2\boldsymbol{t}\ \ \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}|\boldsymbol{Y_i}-\boldsymbol{t}|^2\right)\exp\left(-\frac{|\boldsymbol{t}|^2}{2\sigma^2D^2}\right) \tag{2.8.58}$$

We notice that the expression (2.8.58) is Gaussian on translations $\boldsymbol{t}$. Taking into account the exponent only and completing the square on $t$:

$$\begin{aligned}&-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(|\boldsymbol{Y_i}|^2+|\boldsymbol{t}|^2-2\boldsymbol{Y_i}\cdot\boldsymbol{t}\right)-\frac{|\boldsymbol{t}|^2}{2\sigma^2D^2} = \\ &-\frac{1}{2\sigma^2}\frac{nD^2+1}{D^2}\left|\boldsymbol{t}-\frac{D^2\sum_{i=1}^{n}\boldsymbol{Y_i}}{nD^2+1}\right|^2-\frac{\sum_{i=1}^{n}|\boldsymbol{Y_i}|^2}{2\sigma^2}+\frac{D^2|\sum_{i=1}^{n}\boldsymbol{Y_i}|^2}{2\sigma^2(nD^2+1)}\end{aligned} \tag{2.8.59}$$

Setting $\tilde{n} = \frac{nD^2+1}{D^2}$, the exponent of the integrand in (2.8.59) becomes:

$$-\frac{\tilde{n}}{2\sigma^2}\left|\boldsymbol{t}-\frac{1}{\tilde{n}}\sum_{i=1}^{n}\boldsymbol{Y_i}\right|^2-\frac{\sum_{i=1}^{n}|\boldsymbol{Y_i}|^2}{2\sigma^2}+\frac{|\sum_{i=1}^{n}\boldsymbol{Y_i}|^2}{2\tilde{n}\sigma^2} \tag{2.8.60}$$

We substitute expression (2.8.60) in the exponent of the integral (2.8.58) and we continue with its calculation:

$$\begin{aligned}
&\mathbb{P}(y|b,\beta,s,\mathrm{R},a,\sigma) = \\
&= \frac{1}{Z}\iint d^2\boldsymbol{t}\ \ \exp\left[-\frac{\tilde{n}}{2\sigma^2}\left|\boldsymbol{t}-\frac{1}{\tilde{n}}\sum_{i=1}^{n}\boldsymbol{Y_i}\right|^2\right]\exp\left[\frac{|\sum_{i=1}^{n}\boldsymbol{Y_i}|^2}{2\tilde{n}\sigma^2}-\frac{\sum_{i=1}^{n}|\boldsymbol{Y_i}|^2}{2\sigma^2}\right] = \\
&= \frac{1}{Z}\exp\left[\frac{|\sum_{i=1}^{n}\boldsymbol{Y_i}|^2}{2\tilde{n}\sigma^2}-\frac{\sum_{i=1}^{n}|\boldsymbol{Y_i}|^2}{2\sigma^2}\right]\left(\sqrt{\frac{2\pi\sigma^2}{\tilde{n}}}\right)^2
\end{aligned} \tag{2.8.61}$$

One notices that the above expression $\left[\frac{(\sum_{i=1}^{n}\boldsymbol{Y_i})^2}{2\tilde{n}\sigma^2}-\frac{\sum_{i=1}^{n}|\boldsymbol{Y_i}|^2}{2\sigma^2}\right]$ is similar to the variance of $\boldsymbol{Y}$ although at this stage this does not have a significant interpretation in terms of our data. Later, we shall see statistical properties of the data to come naturally out of our calculations. Now, absorbing all the new constants into our normalisation we have:

$$\begin{aligned}
\frac{1}{Z} &= \frac{1}{(2\pi)^{n+1}\sigma^{2n+2}D^2}\frac{2\pi\sigma^2}{\tilde{n}} \\
&= \frac{1}{(2\pi)^{n}\sigma^{2n}D^2}\frac{D^2}{nD^2+1} \\
&= \frac{1}{(nD^2+1)(2\pi)^{n}\sigma^{2n}}
\end{aligned} \tag{2.8.62}$$

so that the result after integrating translations is:

$$\boxed{\mathbb{P}(y|b,\beta,s,\mathrm{R},a,\sigma) = \frac{1}{Z}\exp\left[\frac{|\sum_{i=1}^{n}\boldsymbol{Y_i}|^2}{2\tilde{n}\sigma^2}-\frac{\sum_{i=1}^{n}|\boldsymbol{Y_i}|^2}{2\sigma^2}\right]}$$

### 2.8.3 Integration of rotations R with a flat prior

The variable to be integrated now is rotations R which we parametrised by $\theta$. For the integration of rotations we use the flat prior since the space of rotations is compact so this choice of prior does not introduce any divergences. Furthermore as we have seen Jeffreys prior does not depend on the rotations $R$ - the group element in question. This is because the Haar measure for this group is $\frac{d\theta}{2\pi}$ so the expression that needs to be integrated is:

$$\mathbb{P}(y|b,\beta,s,\mathrm{R},a,\sigma) = \frac{1}{Z}\exp\left[\frac{|\sum_{i=1}^{n}\boldsymbol{Y_i}|^2}{2\tilde{n}\sigma^2}-\frac{\sum_{i=1}^{n}|\boldsymbol{Y_i}|^2}{2\sigma^2}\right] \tag{2.8.63}$$

with $\boldsymbol{Y_i} = \boldsymbol{y_{(i)}} - a\mathrm{R}\boldsymbol{v_i}$. A convenient representation we chose for rotations is $\mathrm{R} = e^{i\theta}$ so we will transform everything to the complex plane. The inner product between two vectors of the complex plane takes the form:

$$\begin{aligned}\sum_i \sum_j \boldsymbol{Y_i} \cdot \boldsymbol{Y_j} &= \frac{1}{2}\sum_i \sum_j \left(Y_i \overline{Y}_j\right) + \frac{1}{2}\sum_i \sum_j \left(\overline{Y}_i Y_j\right) \\ &= \frac{1}{2}\sum_i \sum_j \left(Y_i \overline{Y}_j\right) + \frac{1}{2}\sum_i \sum_j \left(Y_i \overline{Y}_j\right) \\ &= \sum_i \sum_j Y_i \overline{Y}_j \end{aligned} \tag{2.8.64}$$

where we relabelled and reordered the second term. For simplicity we study separately how each of the terms of the exponent in expression (2.8.63) transforms into the complex plane. The first term of the exponent becomes:

$$\begin{aligned}|\sum_i \boldsymbol{Y_i}|^2 &= \sum_i \sum_j Y_i \overline{Y}_j = \\ &= \sum_i \sum_j y_i \bar{y}_j - \bar{a}\overline{\mathrm{R}} \sum_i \sum_j \bar{v}_j y_i - a\mathrm{R} \sum_i \sum_j v_i \bar{y}_j + \sum_i \sum_j |a|^2 |\mathrm{R}|^2 v_i \overline{v_j}\end{aligned} \tag{2.8.65}$$

Notice that $\bar{a} = a$ since $a \in \mathbb{R}$ and that $|\mathrm{R}|^2 = \mathrm{R}\overline{\mathrm{R}} = e^{i\theta}e^{-i\theta} = 1$. We now expand the second term of the exponent of expression (2.8.63):

$$\begin{aligned}\sum_{i=1}^{n} |\boldsymbol{Y_i}|^2 = \sum_{i=1}^{n} Y_i \overline{Y_i} &= \sum_{i=1}^{n} \left[y_i \bar{y}_i - \bar{a}\overline{\mathrm{R}} \bar{v}_i y_i - a\mathrm{R} v_i \bar{y}_i + |a|^2 |\mathrm{R}|^2 |v_i|^2\right] = \\ &= \sum_{i=1}^{n} |\boldsymbol{y_i}|^2 - \bar{a}\overline{R} \sum_{i=1}^{n} \overline{v}_i y_i - aR \sum_{i=1}^{n} v_i \overline{y_i} + |a|^2 \sum_{i=1}^{n} |\boldsymbol{v_i}|^2\end{aligned} \tag{2.8.66}$$

Combining the two parts of the exponent in (2.8.65) and (2.8.66) the desired quantity (2.8.63) becomes:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,\mathrm{R},a,\sigma) &= \frac{1}{2\sigma^2}\left[\frac{1}{\tilde{n}}\sum_i \sum_j Y_i \overline{Y}_j - \sum_i |Y_i|^2\right] = \\ &= \frac{1}{2\sigma^2}\left[\frac{1}{\tilde{n}}\sum_i \sum_j y_i \bar{y}_j - \sum_i |y_i|^2 + |a|^2 \frac{1}{\tilde{n}}\sum_i \sum_j v_i \bar{v}_j - |a|^2 \sum_i |v_i|^2 \right. \\ &\quad \left. -\mathrm{R}\left(\frac{1}{\tilde{n}}\sum_i \sum_j a v_i \bar{y}_j - \sum_i a v_i \bar{y}_i\right) - \overline{\mathrm{R}}\left(\frac{1}{\tilde{n}}\sum_i \sum_j \bar{a}\bar{v}_j y_i - \sum_i \bar{a}\bar{v}_i y_i\right)\right]\end{aligned} \tag{2.8.67}$$

Here, we introduce three new quantities because they have an interpretation in terms of regulated versions of statistical properties of the data and example shapes:

$$\widetilde{\mathrm{Cov}(v,y)} = \frac{1}{\tilde{n}}\left[\sum_i v_i\bar{y}_i - \frac{1}{\tilde{n}}\sum_i\sum_j v_i\bar{y}_j\right] \tag{2.8.68}$$

$$\widetilde{\mathrm{Var}(v)} = \frac{1}{\tilde{n}}\left[\sum_i |v_i|^2 - \frac{1}{\tilde{n}}\sum_i\sum_j v_i\bar{v}_j\right] \tag{2.8.69}$$

$$\widetilde{\mathrm{Var}(y)} = \frac{1}{\tilde{n}}\left[\sum_i |y_i|^2 - \frac{1}{\tilde{n}}\sum_i\sum_j y_i\bar{y}_j\right] \tag{2.8.70}$$

These properties have arisen by introduction of the priors required to regulate divergences. Thus, expression (2.8.67) then becomes:

$$\mathbb{P}(y|b,\beta,s,\mathrm{R},a,\sigma) = \frac{1}{2\sigma^2}\left[-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)} - \mathrm{R}\left(\frac{1}{\tilde{n}}\sum_i\sum_j av_i\bar{y}_j - \sum_i av_i\bar{y}_i\right)\right.$$
$$\left. -\bar{\mathrm{R}}\left(\frac{1}{\tilde{n}}\sum_i\sum_j \bar{a}\bar{v}_j y_i - \sum_i \bar{a}\bar{v}_i y_i\right)\right] \tag{2.8.71}$$

Re-labelling and re-ordering the summation of the last term in square brackets we find that:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,\theta,a,\sigma) &= \frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right)\\ &+ \frac{1}{2\sigma^2}\left[-R\left(\frac{1}{\tilde{n}}\sum_i\sum_j av_i\bar{y}_j - \sum_i av_i\bar{y}_i\right) - \bar{\mathrm{R}}\left(\frac{1}{\tilde{n}}\sum_j\sum_i \bar{a}\bar{v}_i y_j - \sum_i \bar{a}\bar{v}_i y_i\right)\right] =\\ &= \frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right) +\\ &+ \frac{1}{2\sigma^2}\left[-(\cos\theta + i\sin\theta)k - (\cos\theta - i\sin\theta)\bar{k}\right]\end{aligned} \tag{2.8.72}$$

where we have substituted:

$$k = \frac{1}{\tilde{n}}\sum_i\sum_j av_i\bar{y}_j - \sum_i av_i\bar{y}_i \tag{2.8.73}$$

It is convenient to collect all the terms involving sines and cosines and express equation (2.8.72) as:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,\theta,a,\sigma) =& \frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right) + \frac{1}{2\sigma^2}\left[-\cos\theta(k+\bar{k}) - i\sin\theta(k-\bar{k})\right] \\ =& \frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right) + \frac{1}{2\sigma^2}\left[-2Re(k)\cos\theta + 2Im(k)\sin\theta\right] \\ =& \frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right) + \frac{A}{2\sigma^2}\cos(\theta-\alpha) \qquad (2.8.74)\end{aligned}$$

where we made use of the following properties:

$$\alpha = \arctan\frac{2Im(k)}{-2Re(k)} \qquad (2.8.75)$$

$$A = \sqrt{(-2Re(k))^2 + (2Im(k))^2} = 2|k| \qquad (2.8.76)$$

We now return to equation (2.8.74) in order to carry out the integral over the rotation group with respect to $\theta$:

$$\mathbb{P}(y|b,\beta,s,\theta,a,\sigma) = \frac{1}{Z}\ \exp\left[\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right)\right]\ \exp\left[\frac{2|k|}{2\sigma^2}\cos(\theta-\alpha)\right] \qquad (2.8.77)$$

Using a flat prior for $\theta$ since the space of rotations is compact we have:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,a,\sigma) =& \int_0^{2\pi} d\theta\ \mathbb{P}(y|b,\beta,s,\theta,a,\sigma)\mathbb{P}(\theta) = \\ =& \frac{1}{Z}\ \exp\left[\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right)\right]\times \\ & \int_0^{2\pi}\ \frac{d\theta}{2\pi}\ \exp\left[\frac{2|k|}{2\sigma^2}\cos(\theta-\alpha)\right] \qquad (2.8.78)\end{aligned}$$

Setting $\theta - \alpha = z$ and making use of the periodicity of the cosine function, we now have:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,a,\sigma) =& \frac{1}{Z}\ \exp\left[\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right)\right]\int_{-\alpha}^{2\pi-\alpha}\frac{dz}{2\pi}\ \exp\left[\frac{|k|}{\sigma^2}\cos(z)\right] \\ =& \frac{1}{Z}\ \exp\left[\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right)\right]\ I_o\left(\frac{|k|}{\sigma^2}\right). \qquad (2.8.79)\end{aligned}$$

Recall that:

$$k = a\left(\frac{1}{\tilde{n}}\sum_i\sum_j v_i\bar{y}_j - \sum_i v_i\bar{y}_i\right) = -a\tilde{n}\widetilde{\mathrm{Cov}(v,y)} \tag{2.8.80}$$

Thus, expression (2.8.79) becomes:

$$\mathbb{P}(y|b,\beta,s,a,\sigma) = \frac{1}{Z}\ \exp\left[\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right)\right]\ I_o\left(\frac{a\ \tilde{n}\left|\widetilde{\mathrm{Cov}(v,y)}\right|}{\sigma^2}\right) \tag{2.8.81}$$

where $I_o$ is the modified Bessel function of the first kind and of zero-th order. The result after integrating rotations is:

$$\boxed{\mathbb{P}(y|b,\beta,s,a,\sigma) = \frac{1}{Z}\ \exp\left[\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(y)} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(v)}\right)\right]\ I_o\left(\frac{a\ \tilde{n}\left|\widetilde{\mathrm{Cov}(v,y)}\right|}{\sigma^2}\right)}$$

with the overall normalization coefficient $\frac{1}{Z} = \frac{1}{(nD^2+1)(2\pi)^n\sigma^{2n}}$.

Before continuing we return to our choice of prior for rotations. To lend further weight that $d\theta$ is an invariant measure. We can adapt the work of Wood [127] and Prentice [128] who showed how to integrate over the generators of rotations in a compact way. Introduce:

$$x = \begin{pmatrix}\cos(\frac{z}{2})\\ \sin(\frac{z}{2})\end{pmatrix}$$

which satisfies $x^Tx = 1$. Then $X(x) = e^{2iz}$ represents a rotation in $SO(2)$. Wood noted that $X$ is uniform in $SO(2)$ if and only if $x$ is uniform on $S^1 \subset \mathbb{R}^2$ . We can rewrite the exponential in the integral of expression (2.8.79) as $e^{\mathrm{Tr}(A\ xx^T)-1}$ where $A = \begin{pmatrix}1+\frac{|k|}{\sigma^2}\\ 1-\frac{|k|}{\sigma^2}\end{pmatrix}$. We then need $\int_{S^1} d[x]\ e^{\mathrm{Tr}(A\ xx^T)-1}$ which with the uniform measure can be written in polar coordinates ($r = 1$):

$$\int_0^{2\pi}\frac{dz}{2\pi}\exp\left(\frac{|k|}{\sigma^2}\cos z\right) \tag{2.8.82}$$

as above in equation (2.8.79). So the uniform measure in $S^1$ induces our chosen prior which is the Haar measure on this space. We will revisit this idea for the more complicated calculation of the three-dimensional rotations in chapter [5].

Wood has shown how to evaluate the integral in (2.8.82) by diagonalising $xx^T$. In such a basis the exponent simplifies and we have checked that this method does indeed reproduce the Bessel function in (2.8.79). This was a useful check for the calculation and is a powerful approach which generalises to higher dimensional spaces.

### 2.8.4 Integration of scalings $a$ with a Rayleigh prior

Before we proceed to the integration of expression (2.8.3) with respect to scalings $a$ we will extract part of the integrand that depends on this variable and re-write everything back in Cartesian coordinates:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,a,\sigma) &= \frac{1}{Z}\ \exp\left[\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - |a|^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}\right)\right]\ I_o\left(\frac{a\ \tilde{n}\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|}{\sigma^2}\right) = \\ &= \frac{1}{Z}\exp\left(\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}\right)\right)\ \exp\left(-|a|^2\frac{\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}}{2\sigma^2}\right) \\ &\qquad \times I_o\left(\frac{a\ \tilde{n}\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|}{\sigma^2}\right) \qquad (2.8.83)\end{aligned}$$

For scalings we introduce a Rayleigh prior which will effectively remove the scaling invariance of the posterior by limiting the effective domain of the scaling parameters. The Rayleigh prior also has the advantage of being naturally defined on positive parameters which is what we want for the scale factor $a$ and decays exponentially to cut off large values of its argument. Furthermore, it will give the required linear dependence on $a$ when we remove our regulator $B$ so as to reproduce Jeffreys prior in that limit. This is true because such a prior is of the form:

$$\mathbb{P}(a|s) = \frac{a}{s^2}\ \exp\left[\frac{-a^2}{2s^2}\right]. \qquad (2.8.84)$$

We take $s \propto \sigma$ so that $s = B\sigma$, where $B$ is the scalings' regulator that we will to take to infinity to return to the initial result:

$$\mathbb{P}(a|B\sigma) = \frac{a}{B^2\sigma^2}\ \exp\left[\frac{-a^2}{2B^2\sigma^2}\right]. \tag{2.8.85}$$

We acknowledge this is not the only choice of regulator, nor was our choice of regulator for translations nor will be our choice for the noise parameter $\sigma$. However, these choices suffice in regularising our divergences and returning the Jeffreys prior for the appropriate limiting values of the regulators. It is important to note that our result should be independent of our choice of regulators so that the priors we have introduced in this section, which appear to be subjective, do not represent physical information or knowledge of our data but (as we take these limiting values) are rather mathematical tools enabling us to understand and describe the form of the divergence that arises in the likelihood.

Integrating expression (2.8.83) with respect to scalings $a$:

$$\begin{aligned}
\mathbb{P}(y|b,\beta,s,\sigma) &= \int_0^\infty \mathbb{P}(y|b,\beta,s,a,\sigma)\mathbb{P}(a) \\
&= \frac{1}{Z}\frac{1}{B^2\sigma^2}\exp\left(\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}\right)\right) \\
&\quad \int_0^\infty da\ a\ \exp\left[-a^2\left(\frac{\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}}{2\sigma^2}+\frac{1}{2B^2\sigma^2}\right)\right]\ I_o\left(a\ \frac{\tilde{n}\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|}{\sigma^2}\right) = \\
&= \frac{1}{Z}\frac{1}{B^2\sigma^2}\ \exp\left(\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}\right)\right)\times \\
&\qquad \frac{1}{2(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)}\exp\left(\frac{1}{2\sigma^2}\frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)}\right)
\end{aligned} \tag{2.8.86}$$

where we have used the following property [129]:

$$\int_0^\infty da\ a\ \exp(-a^2E)\ I_o(aF) = \frac{1}{2E}\exp\left(\frac{F^2}{4E}\right)$$

After integrating the marginalised likelihood with respect to scalings and rearranging the terms, expression (2.8.86) becomes:

$$\mathbb{P}(y|b,\beta,s,\sigma) = \frac{1}{Z}\frac{1}{2(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)}\exp\left(\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}+\frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)}\right)\right) \tag{2.8.87}$$

We again absorb the leading constants in order to make a redefinition of $Z$:

$$\frac{1}{Z} = \frac{1}{2(nD^2+1)(2\pi)^n\sigma^{2n}}\frac{1}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} \tag{2.8.88}$$

The final result of integrating scalings is:

$$\boxed{\mathbb{P}(y|b,\beta,s,\sigma) = \frac{1}{Z}\exp\left(\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}+\frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)}\right)\right)}$$

As we suggested before, we will investigate the scaling behaviour of the likelihood when either of scalings $a$ or $\sigma$ is removed from our calculations. As we mentioned in the previous section when we remove $\sigma$ then Jeffreys prior becomes $n\,a\,\mathrm{Var}(v)$. This has the same dependence on $a$, so little would have changed up to this point. The marginalised likelihood when using this prior for the group of the transformations is easily determined by inspection to be:

$$\mathbb{P}(y|b,\beta,s) = \frac{1}{Z}\frac{1}{2(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)}\exp\left(\frac{1}{2}\left(-\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}+\frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)}\right)\right) \tag{2.8.89}$$

with $Z$:

$$\frac{1}{Z} = \frac{1}{2(nD^2+1)(2\pi)^n}\frac{1}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} \tag{2.8.90}$$

Despite our discussion that it may not be necessary to include both $\sigma$ and $a$ in our model, in the case that $\sigma$ is removed, the integrated likelihood (2.8.89) suffers

from bad scaling properties under $y \to \lambda y$ because the exponent picks up a factor of $\lambda^2$; since this scaling component is in the exponent that would imply that even the posterior would not be scale invariant with this choice. This is not what we wanted to achieve with our model of the formation of data shapes, as one of the key desires was that scale should not matter. In the next section we will see what happens with the choice of removing $a$ and keeping $\sigma$.

### 2.8.5 Integration of $\sigma$ using a $\Gamma$ prior

We write the result after integrating scalings as:

$$\mathbb{P}(y|b,\beta,s,\sigma) = \frac{1}{Z}\exp\left(\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} + G\right)\right) \tag{2.8.91}$$

with $G$ being:

$$G = \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} \tag{2.8.92}$$

Expression (2.8.91) needs to be integrated with respect to $\sigma$. For this integration we will be using a $\Gamma(\alpha,\zeta)$ prior on $\frac{1}{\sigma^2}$ since $\sigma$ enters the likelihood with this functional form so it is the natural variable to use. This is a good distribution to use on parameters that are positive such as $\frac{1}{\sigma^2}$ and has the right asymptotic properties to limit the effect of large values of the noise parameter. As with the Rayleigh prior it also gives the correct limiting behaviour which yields Jeffreys prior when we remove the regulator. This will follow from the additional inverse powers of $\sigma$ that will be introduced by this choice of prior, which we can tune to provide the required result. To make these statement precise, the prior on $\sigma$ is of the form:

$$\mathbb{P}(\sigma) = \frac{\zeta^\alpha}{\Gamma(\alpha)}\left(\frac{1}{\sigma^2}\right)^{\alpha-1}\exp\left(-\frac{\zeta}{\sigma^2}\right)\, d\left(\frac{1}{\sigma^2}\right) \tag{2.8.93}$$

whose exponent provides a damping to large values of $\frac{1}{\sigma^2}$. The expression that will be integrated is:

$$\mathbb{P}(y|b,\beta,s) = \int_0^\infty \mathbb{P}(y|b,\beta,s,\sigma)\mathbb{P}(\sigma) = \\ = \int_0^\infty d\left(\frac{1}{\sigma^2}\right)\ \frac{1}{Z}\ \exp\left(\frac{1}{2\sigma^2}\left(-\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}+G\right)\right)\frac{\zeta^\alpha}{\Gamma(\alpha)}\left(\frac{1}{\sigma^2}\right)^{\alpha-1}\exp\left(-\frac{\zeta}{\sigma^2}\right) \tag{2.8.94}$$

Extracting the $\sigma$ dependence from the normalization constant and re-ordering the expression:

$$\mathbb{P}(y|b,\beta,s) = \frac{1}{Z}\int_0^\infty d\left(\frac{1}{\sigma^2}\right)\frac{1}{\sigma^{2n}}\exp\left[\left(-\frac{1}{\sigma^2}\right)\left(\frac{\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}-G}{2}\right)\right]\left(\frac{1}{\sigma^2}\right)^{\alpha-1}\exp\left(-\frac{\zeta}{\sigma^2}\right) = \\ = \frac{1}{Z}\int_0^\infty d\left(\frac{1}{\sigma^2}\right)\frac{1}{\sigma^{2n+2\alpha-2}}\exp\left[\left(-\frac{1}{\sigma^2}\right)\left(\frac{2\zeta+\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}-G}{2}\right)\right] \tag{2.8.95}$$

where we have stripped the $\sigma$ dependence out of the normalization so that the remaining constants are now:

$$\frac{1}{Z} = \frac{1}{(nD^2+1)(2\pi)^n}\frac{1}{(B^2\tilde{n}\widetilde{\mathrm{Var}}(\boldsymbol{v})+1)}\frac{\zeta^\alpha}{\Gamma(\alpha)} \tag{2.8.96}$$

Consider the integral of (2.8.95) with respect to $\sigma$:

$$\mathbb{P}(y|b,\beta,s) = \frac{1}{Z}\int_0^\infty d\left(\frac{1}{\sigma^2}\right)\ \frac{1}{\sigma^{2(n+\alpha)-2}}\exp\left[\left(-\frac{1}{\sigma^2}\right)\left(\frac{2\zeta+\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}-G}{2}\right)\right] \tag{2.8.97}$$

This can be simplified by changing variables to $x = \frac{1}{\sigma^2}$ so that expression (2.8.97) becomes:

$$\mathbb{P}(y|b,\beta,s) = \frac{1}{Z}\int_0^\infty dx\ x^{n+\alpha-1}\exp\left[\left(\frac{2\zeta+\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}-G}{2}\right)x\right] = \\ = \frac{1}{Z}\Gamma(n+\alpha)\left[\frac{2\zeta+\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})}-G}{2}\right]^{-n-\alpha} \tag{2.8.98}$$

The final expression of the likelihood with the noise $\sigma$ integrated out is:

$$\mathbb{P}(y|b,\beta,s) = \frac{1}{Z}\left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \tag{2.8.99}$$

where we have absorbed the constants into $Z$ for the final time so that:

$$\frac{1}{Z} = \frac{1}{(nD^2+1)(2\pi)^n} \frac{\Gamma(n+\alpha)}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} \frac{\zeta^\alpha}{\Gamma(\alpha)} \frac{1}{2^{-n-\alpha}} \tag{2.8.100}$$

We here have to note that the above result does not scale in the same fashion as the likelihood in (2.6.8) since the prior we have imposed on scalings breaks the scale behaviour of the model. Only when $\zeta$ is taken to zero do we find that the posterior is scale invariant.

$$\boxed{\mathbb{P}(y|b,\beta,s) = \frac{1}{Z}\left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha}}$$

This is the final result of the integration over the similarity transformations $g \in G$ and the noise parameter $\sigma$ which we calculated with the help of the regularised version of Jeffreys prior. Employing this result, the complete likelihood is:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s) &= \sum_b \int \mathscr{D}\beta\ \mathscr{D}s\ \mathbb{P}(y|b,\beta,s)\mathbb{P}(b)\mathbb{P}(\beta)\mathbb{P}(s) \\ &= \sum_{b\in\mathcal{B}} \int \mathscr{D}\beta\ \mathscr{D}s\ \frac{1}{Z}\left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \mathbb{P}(b)\mathbb{P}(\beta)\mathbb{P}(s)\end{aligned} \tag{2.8.101}$$

As we have investigated the scaling properties of the likelihood when removing $\sigma$ we will also investigate it when we remove $a$. So if instead, we keep $\sigma$ and remove $a$, then Jeffreys prior is $\frac{\sqrt{\mathrm{Var}(v)}}{\sigma^3}$. To find the marginalised likelihood when removing $a$ we must integrate (2.8.81) with respect to $\sigma$ against the Haar measure (which is Jeffreys prior) on $a = 1$. The integration leads to a hypergeometric function and such result exhibits scale invariance; in contrary to when we remove $\sigma$ and keep $a$, the marginalised likelihood scales as $\lambda^r$ (where $r$ is some power) which leaves the posterior invariant. The future simulations in this thesis will be based on including all five original variables. Given the differences in behaviour that arise when either $\sigma$ or $a$ is removed there is an open question as to how to decide upon which parameter is superfluous. We intend to investigate both of these possibilities in future work in

order to better understand the consequences of such a choice. Perhaps then it will be possible to justify which of the variables is needed; for now we proceed with both and present the results of doing so as a first exploration of our approach to shape classification.

In the next section we compare the result of expression (2.8.101) to the result obtained when integrating against Jeffreys prior. In the sections to follow, we discuss how the remaining integrals of expression (2.8.101) have been calculated and how they give rise to a classification algorithm.

## 2.9 Comparison of the two results

To compare the results from the integration of the marginalised likelihood when using Jeffreys prior and when using its regularised version it suffices to take the limits of the regulators. Before we do that, we discuss the choice of the regularised priors. One could argue that many choices of priors could generate the same results as when integrating against the original Jeffreys prior. However, we chose the ones presented above not only because they give us the expected result but also because they reflect our belief in the parameters. In particular, a Gaussian prior on translations shows our belief that some translations are favoured and are concentrated around the mean. Such a prior also cuts off very large translations that have small probability of occurring. The Rayleigh prior on scalings was chosen because it introduces the extra multiplicative factor of $a$ as in the case of Jeffreys prior but also because only the positive scalings are included. Lastly, the choice over the Gamma prior on $\frac{1}{\sigma^2}$ was chose to reflect our belief that the noise can only be positive. We now state the result of the calculation of the integral of similarity transformations against Jeffreys prior:

$$\mathbb{P}(y|b,\beta,s) \propto \left[ n\mathrm{Var}(\boldsymbol{y}) - \frac{n\,|\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})|^2}{\mathrm{Var}(\boldsymbol{v})} \right]^{-n-\frac{1}{2}} \tag{2.9.102}$$

The result of (2.9.102) can be reproduced by the right choice of the values of the regulators in expression (2.8.99) and we are in the position to do so. It is straight

forward to decide on the choice of $D$ and $B$ which are taken to be $D \to \infty$, $B \to \infty$. In the case of $\zeta$ is also easy to choose $\zeta \to 0$ however this is not the case with $\alpha$. We take $\alpha$ to be $\alpha \to \frac{3}{2}$ and the reason behind that is the fact that we do not compare the chosen prior against Jeffreys prior but we compare the individual priors to Jeffreys prior when it is thought of as a measure on $\frac{1}{\sigma^2}$. It is easy to see that expression (2.9.102) corresponds to (2.8.99) when we take the limits of the regulators.

We now discuss the divergence produced by the result in (2.9.102). It is obvious that the divergence of this distribution is caused when the quantity in square brackets tends to zero. This is what we investigate now. The vanishing of the denominator of (2.9.102) occurs when:

$$\begin{aligned} \mathrm{Var}(\boldsymbol{y}) - \frac{|\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})|^2}{\mathrm{Var}(\boldsymbol{v})} &= 0 \\ \Leftrightarrow \frac{|\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})|^2}{\mathrm{Var}(\boldsymbol{v})\mathrm{Var}(\boldsymbol{y})} &= 1 \\ \Leftrightarrow \mathrm{Corr}(\boldsymbol{v},\boldsymbol{y}) &= \pm 1 \end{aligned} \tag{2.9.103}$$

That shows us that the divergence appears when $\mathrm{Corr}(\boldsymbol{v},\boldsymbol{y}) = \pm 1$ , i.e. when the data shape $\boldsymbol{y}$ is either positively or negatively correlated with the sample shape $\boldsymbol{v}$. It is fairly easy to understand the meaning of the correlation of two random variables, however in the case of two vector valued random variables it needs a little thought. When can two vectors be correlated? Since correlation is linked with the idea of linearity one could say that two vectors are correlated when they are linearly dependent i.e. when there is a linear transformation that could generate one from the other. We notice that the correlation is a quantity invariant to translations and rotations of both $\boldsymbol{y}$ and $\boldsymbol{v}$. The posterior is also invariant under scalings since the extra $a^2$ introduced by the covariance in the numerator can be cancelled by the $a^2$ introduced by the variances multiplied in the denominator. Since correlation is invariant under similarity transformations, we understand that $\boldsymbol{y}$ and $\boldsymbol{v}$ are correlated either positively or negatively when one of $\boldsymbol{y}$ or $\boldsymbol{v}$ is generated by the other by some similarity transformations. This means that the origin of the divergence is apparent

if and only if $\boldsymbol{y}$ and $\boldsymbol{v}$ are shapes that belong to the same orbit under the action of the similarity group $G$. To understand more the geometrical interpretation and meaning of the particular property we discuss this in the following paragraph.

At this point, it would be useful to introduce a new notation for equation (2.9.102). The notation is called "bra-ket" and it is a standard notation in quantum mechanics but is also used to denote abstract vectors in linear algebra. The inner product of two $n$-vectors in two dimensions, say $\boldsymbol{x}$ and $\boldsymbol{y}$, is denoted by $\boldsymbol{x}^T\boldsymbol{y} = \langle \boldsymbol{x}|\boldsymbol{y}\rangle$ and the outer product is denoted as $\boldsymbol{x}\boldsymbol{y}^T = |\boldsymbol{x}\rangle\langle\boldsymbol{y}|$. Using this notation, and making the substitution $\boldsymbol{x}' = \boldsymbol{x} - \bar{\boldsymbol{x}}$ and $\boldsymbol{y}' = \boldsymbol{y} - \bar{\boldsymbol{y}}$ we can write:

$$\mathrm{Cov}(\boldsymbol{x}', \boldsymbol{y}') = \sum_{i=1}^{n} (\boldsymbol{x_i} - \bar{\boldsymbol{x}}) \cdot (\boldsymbol{y_i} - \bar{\boldsymbol{y}}) = \boldsymbol{x}'^T\boldsymbol{y}' = \langle \boldsymbol{x}|\boldsymbol{y}\rangle \tag{2.9.104}$$

$$\mathrm{Var}(\boldsymbol{x}', \boldsymbol{x}') = \sum_{i=1}^{n} (\boldsymbol{x_i} - \bar{\boldsymbol{x}}) \cdot (\boldsymbol{x_i} - \bar{\boldsymbol{x}}) = \boldsymbol{x}'^T\boldsymbol{x}' = \langle \boldsymbol{x}'|\boldsymbol{x}'\rangle \tag{2.9.105}$$

This notation can be very useful for re-writing the result of equation (2.9.102) as:

$$\begin{aligned}
\mathbb{P}(y|b,\beta,s) &\propto \left[ n\mathrm{Var}(\boldsymbol{y}') - \frac{n\,|\mathrm{Cov}(\boldsymbol{v}', \boldsymbol{y}')|^2}{\mathrm{Var}(\boldsymbol{v}')} \right]^{-n-\frac{1}{2}} \\
&\propto \left[ \langle \boldsymbol{y}'|\boldsymbol{y}'\rangle - \frac{\langle \boldsymbol{y}'|\boldsymbol{v}'\rangle \langle \boldsymbol{v}'|\boldsymbol{y}'\rangle}{\langle \boldsymbol{v}'|\boldsymbol{v}'\rangle} \right]^{-n-\frac{1}{2}} \\
&\propto \left[ \left\langle \boldsymbol{y}' \middle| I - \frac{|\boldsymbol{v}'\rangle\langle\boldsymbol{v}'|}{\langle \boldsymbol{v}'|\boldsymbol{v}'\rangle} \middle| \boldsymbol{y}' \right\rangle \right]^{-n-\frac{1}{2}}
\end{aligned} \tag{2.9.106}$$

The quantity $\mathbf{P} = I - \frac{|\boldsymbol{v}'\rangle\langle\boldsymbol{v}'|}{\langle \boldsymbol{v}'|\boldsymbol{v}'\rangle}$ in equation (2.9.106) is also recognised as the projection operator that projects orthogonally to $\boldsymbol{v}'$. Writing the result as such it is easier to interpret it geometrically and we note that the projection operator acts on the data $\boldsymbol{y}$. The quantity in square brackets gives rise to the divergence when it is equal to zero, i.e. when the projection operator acting on $\boldsymbol{y}'$ equals zero i.e. $\mathcal{P}|\boldsymbol{y}'\rangle = 0$. That happens when $\boldsymbol{y}'$ is on the orbit of $\boldsymbol{v}'$ under the action of $G$ i.e. when $\boldsymbol{y}'$ and $\boldsymbol{v}'$ are linked via some similarity transformations so that $|\boldsymbol{y}'\rangle = g\,|\boldsymbol{v}'\rangle = a\mathrm{R}\,|\boldsymbol{v}'\rangle + \boldsymbol{t}$.

One could argue here that it would be useful to remove the divergence by removing and extracting all such $\boldsymbol{y}$ that produce the divergence i.e. all the data shapes that are linked to specific example shapes by specific similarity transformations. That could happen by excluding this hypersurface from the integration region. However, even doing so doesn't alleviate the problem. Although we are dealing with continuous shapes, when it comes to simulate such data shapes for computational purposes the problem will be apparent; there will always be a neighborhood around the region that we extracted in which the quantity in (2.9.106) will be close to singular which may lead to numerical errors that are hard to predict. However, such a situation is difficult to appear even if data and example shapes are linked via such similarity transformations since the intrinsic Gaussian noise would always make sure to alleviate such a problem. For the purposes of simulation we retain the regulators $\zeta, B$ and $D$, choosing sufficiently large or small values to alleviate the divergence without changing the classification accuracy.

Here, we should also mention that the primary underlying reason of the cause of the divergence is the likelihood itself (2.6.8). The reason is that in the case that $\boldsymbol{y}$ and $\boldsymbol{v}$ are linked via such similarity transformations so that $\boldsymbol{y}^* = a\mathrm{R}\boldsymbol{v} + \boldsymbol{t}$, then the likelihood becomes:

$$\mathbb{P}(y|b,\beta,s,a,\mathrm{R},t,\sigma) = \frac{1}{Z}\exp\left(-\frac{1}{2\sigma^2}|\boldsymbol{y}^* - a\mathrm{R}\boldsymbol{\beta}(s(i)) + \boldsymbol{t}|^2\right) = \frac{1}{Z} \tag{2.9.107}$$

which must be integrated with respect to $\boldsymbol{t}, \mathrm{R}, a, \sigma$ and $\beta$. Equation (2.9.107) is constant and so independent of the data and any of the parameters. Integrating it with respect to $\boldsymbol{t}$ for example, over $\mathbb{R}^2$ gives rise to the divergence. By default that shows that choosing the appropriate priors for the integration which smoothed the integration domain was the correct thing to do.

## 2.10 The remaining integrals

To obtain the full likelihood we have:

$$\mathbb{P}(\boldsymbol{y}|C) = \sum_{b} \int \mathscr{D}\beta \; \mathscr{D}s \; \mathbb{P}(y|b,\beta,s)\mathbb{P}(b)\mathbb{P}(\beta|C)\mathbb{P}(s). \tag{2.10.108}$$

For the remaining integrals over sampings $s$, shape curves $\beta$ and the summation over the bijections $b$ we employed numerical techniques and the summation was carried out exhaustively. These choices are explained below.

For the integration over samplings and shape curves we employed simple Monte Carlo techniques [130, 131] of the generalised Gaussian prior that we described in section [2.3.1]. The form of $\mathbb{P}(y|b,\beta,s)$ is complicated and the integration over samplings cannot be evaluated explicitly. We thus use Monte Carlo techniques by generating realisations from the probability distributions of the samplings and the shape curves and then sum the values of the integrand evaluated at these realisations.

The summation over bijections was carried out exhaustively. In previous work, where the integration was done numerically, the sum over bijections could be approximated, again in a zeroth order Laplace estimation, using the Hungarian algorithm, where the summation was treated as a simple linear assignment problem. Our analytic calculation has introduced problems with such a summation. The presence of the term involving $\mathrm{Cov}^2(\boldsymbol{\beta}(s(b_i^{-1})), \boldsymbol{y_i})$ complicates the situation and turns the simple assignment into a quadratic assignment problem. The quadratic assignment (QAP) [132] is one of the major problems in the branch of optimization and is the same in nature as the assignment problem; the crucial difference is that the cost function is quadratic and hence the assignments are not independent. Quadratic assignment problems are NP-hard [133], which are a class of decision problems with no known algorithm that solves them to optimality in polynomial time [134]. What is even more interesting, as proven by Sahni [133], is that any routine that finds even an $\epsilon$–approximate solution is also NP-hard, thus making QAP problems "the hardest of the hard" of all combinatoric problems. One solution could be to actually calculate all possible bijections but this will take a very big amount of time ( e.g. for 30 points around the boundary there are $30! \simeq 2^{32}$ bijections) and still be prohibitive for any brute force simulation.

Instead of using a Laplace approximation we could approximate the full sum-

mation using Markov Chain Monte Carlo integration. However, we came up with a more realistic solution for this. Assuming that the points around the boundary are ordered and (this is the case of labeled landmarks we discussed in chapter [1]) the summation can be carried out exhaustively. The reason for this is that the number of ordered bijections is only $n$, since each map is uniquely fixed by the starting point out of $n$ possibilities. We based our decision to reflect the fact that in any realistic situation an experimentalist would choose to place the landmarks around the outline of the shape in an ordered rather than in a random way. For example, in the case of geological sand bodies, although the three-dimensional point cloud is obtained by a laser scanner, the extraction of the three-dimensional curves is done by an experimentalist geologist. Since this task involves human interaction, we would expect a deterministic approach to the placement of the points around the boundary of the three-dimensional shapes. It is part of the deterministic nature of human beings that they would not treat a problem in a random way but rather in a concise and structured way. Thus, one would expect that an experimentalist geologist or anyone involved in an experiment of placing points around the boundary of a shape would do it in order that the end result would be a collection of ordered points.

This sum over cyclic bijections also implies that our choice of starting position –that is which of the data shape points and which of the example shape point– is labeled as the first of the set. This is true because every one of the data shape points will at some point in the sum be associated to every one of the example shape points after which the remaining points are compared to one another in sequence. Furthermore, this decision to treat the boundary points in an ordered way allowed us to exhaustively calculate the summation over bijections and approximate the remaining integrals by Monte Carlo techniques.

Summing exhaustively over the bijections and integrating over samplings and shape curves provides an approximation of the complete likelihood $\mathbb{P}(\boldsymbol{y}|C)$. Then the Monte Carlo estimate for a given class is given by:

$$\mathbb{P}(C_i|\boldsymbol{y}) \approx \frac{\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L}\mathbb{P}(\boldsymbol{y}|b_j,\beta_k,s_l)\mathbb{P}(\beta_k|C_i)\mathbb{P}(C_i)}{\sum_i\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L}\mathbb{P}(\boldsymbol{y}|b_j,\beta_k,s_l)\mathbb{P}(\beta_k|C_i)\mathbb{P}(C_i)} \tag{2.10.109}$$

where $k, l$ are the Monte Carlo iterations of the curves and samplings respectively and $j$ is the number of bijections. The above forms the Maximum a Posteriori estimate of a given class given the observed data shapes. Once the posterior is approximated, its values can be used for classifying $\boldsymbol{y}$ into a shape class by picking the class that gives the highest posterior. Using this approximation in combination with the result of (2.8.101), we summarise the steps needed to approximate the posterior $\mathbb{P}(C_i|\boldsymbol{y})$ for a given $\boldsymbol{y}$.

**Algorithm**:

*For j = 1,2,...,J*
*For k = 1,2,...,K*:
*For l = 1,2,...,L*:

1. Randomly generate a shape class $C_i$ and simulate a shape $\beta_k \sim \mathbb{P}(\beta|C_i)$.
2. Generate a sampling function $\gamma_l \sim \mathbb{P}(\gamma)$ and use this to place points on $\beta_k$.
3. Associate the points of the data to those of the shape $\beta_k$ through the $j^{\text{th}}$ bijection.
4. Approximate the likelihood function $\mathbb{P}(\boldsymbol{y}|C_i)$ by using the result in (2.10.108).
5. Approximate posterior $\mathbb{P}(C_i|\boldsymbol{y})$ by using (2.10.109).

The above steps constitute our proposed algorithm for the classification of planar shapes. For testing this result and evaluating the effectiveness and the stability of the proposed algorithm, we utilize shape databases and present experimental results in the next chapter.

## 2.11 Concluding remarks

We have investigated the problem of classification and how it can be approached and resolved in the Bayesian paradigm. We have presented a Bayesian approach which finds shape classes for a given configuration of points in the presence of undersampled shape curves and observational noise. We have presented how this problem can be constructed and the models we employ for the description of the parameters. We have seen that the difficulty of the problem lies in the calculation of the likelihood which had to be marginalised over nuisance parameters that take part in the formation of the data. This marginalisation introduced the integration and summation over the nuisance parameters for which we had to employ probability models.

The models we have used for the description of the parameters followed the work presented in [12]. However, the novelty of our approach was that for some of the nuisance parameters, namely the similarity transformations and the noise variance, we have used a different model which in contrary to the methods of [12], enabled us to evaluate some of the integrals in a closed form. In particular, the model we employed was the joint Jeffreys prior which has the special property of invariance under reparametrisations. Although the results were found in a closed form, Jeffreys prior introduced irregularities and divergences after integrating with respect to the above parameters. To alleviate the problem, we used a regularised version of the prior. The effect of the regularisation was to smooth and restrict the domains of the parameters and at the same time remove the invariances of the likelihood with respect to these parameters. However, in some sense, the two results can be regarded as the same since the removal of the regulators by taking their limit to appropriately big or small values restores the invariance of the likelihood.

After integrating with respect to the similarity transformations and the noise variance, the remaining integrals were evaluated by simple Monte Carlo techniques. The final result is the Maximum a Posteriori approximation of a given class and this gives rise to a new classification algorithm. The above proposed algorithm can be implemented computationally for the evaluation of its accuracy and its classification efficacy and efficiency when given a shape data set. In the next chapter we

describe experimental results on estimating the posterior probability $\mathbb{P}(C_i|\boldsymbol{y})$ where we simulate data $\boldsymbol{y}$ according to the data model and then apply the algorithm. In the next chapter we also investigate the confidence levels of our algorithm and the success results it returns for a known shape data set.

# Chapter 3

# Experimental results

## 3.1 Introduction

In this chapter we discuss the experimental results that we acquired for the estimation of $\mathbb{P}(C_i|\boldsymbol{y})$ using the classification algorithm proposed in chapter [2]. For each experiment we use data that we have simulated according to the data model we chose for the particular application. We make use of the result of the analytic integration over similarity transformations and the noise variance that we have presented in chapter [2] to classify shapes into their respective categories. In this way, we can evaluate the effectiveness and the confidence we can have for the algorithm. We also discuss the confidence levels and the success rates of the algorithm when performing classification. In section [3.2] we present the experimental results we acquired for the Kimia database. We explain how the data shapes are simulated and discuss the success rates and the confidence levels of the algorithm for different values of some of the parameters. We also examine the classification results of the algorithm for different simulated data sets. In section [3.3] we present the experimental results acquired for the alphabet database. We perform the same experiments as with the Kimia database and evaluate the confidence and success results. Section [3.4] poses the problem of sand body classification. In this section we discuss about the definition of a geological sand body and their current classification methods. We discuss how the sand bodies can be extracted and discuss how statistical classification can be used and our algorithm is utilised so that the underlying geometry of the sand

bodies is included in contrary to existing classification methods. We also discuss the confidence results and the success rates of the algorithm. In section [3.5], we suggest a way of learning the parameters of the shape models for a given sand body dataset. However, the optimisation algorithm employed for the learning of the parameters doesn't return particularly confident results and we debate the reasons behind this occurrence. Although the learned parameters were different to the ones expected we substitute them with the ones we assume to be correct. Using these parameters we evaluate the classification results of the proposed algorithm for the sand body datasets. Finally, section [3.6] presents the concluding remarks of this chapter.

## 3.2 Experiments on Kimia database

For the experimental results in this section we utilise the Kimia database [12, 13, 135] and in particular the combination of Kimia216 and Kimia99. The database is comprised of binary images and consists of 22 classes of shapes: birds, bones, bricks, camels, cars, children, man, elephants, spectacles, faces, forks, fountains, glasses, hammers, hands, hearts, keys, misks, rabbits, rays, tools and turtles. Each of the classes contains roughly 12 shapes so in total we had approximately 265 shapes at our disposal. Figure (3.1) shows examples of shapes coming from some of the classes. The Kimia database was used to evaluate the accuracy of the proposed algorithm that classifies shapes into a given class. We now describe the process followed for the experiments.

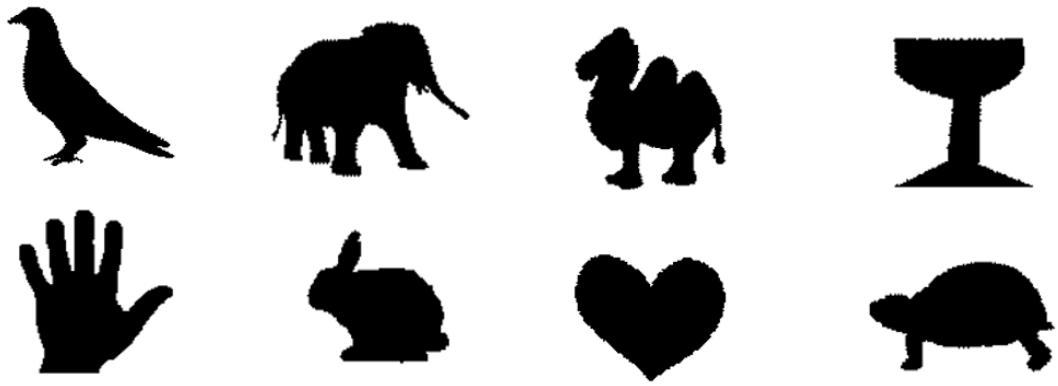


Figure 3.1: Examples of binary Kimia images

### 3.2.1 Acquisition of the shape boundaries

The Kimia database is comprised of binary images of a ".pgm" form. To work with the shapes i.e. the one dimensional line of each image, we had to extract their outlines and acquire their boundaries. Although in our work we assume that the one dimensional boundaries are available to us, following section [1.2] we describe the method by which the outlines were obtained. For the extraction of the outlines, we used the "bwboundaries" [136] MATLAB function which can trace the exterior boundary of a binary image. It implements the Moore–Neighbor tracing algorithm [137] which finds the contour of a given graph and terminates when it visits the first visited pixel for the second time. MATLAB uses an improved stopping condition "Jacob's stopping criterion" [138] which stops the algorithm when the start pixel has been visited for the second time in the same direction it was originally entered. The algorithm returns a matrix which contains the coordinates of the boundary pixels that constitute the outline of the binary image; this is the one dimensional outline that represents a particular shape. Figure (3.2) shows examples of obtained boundaries of shapes coming from the Kimia database which were extracted with the above method.

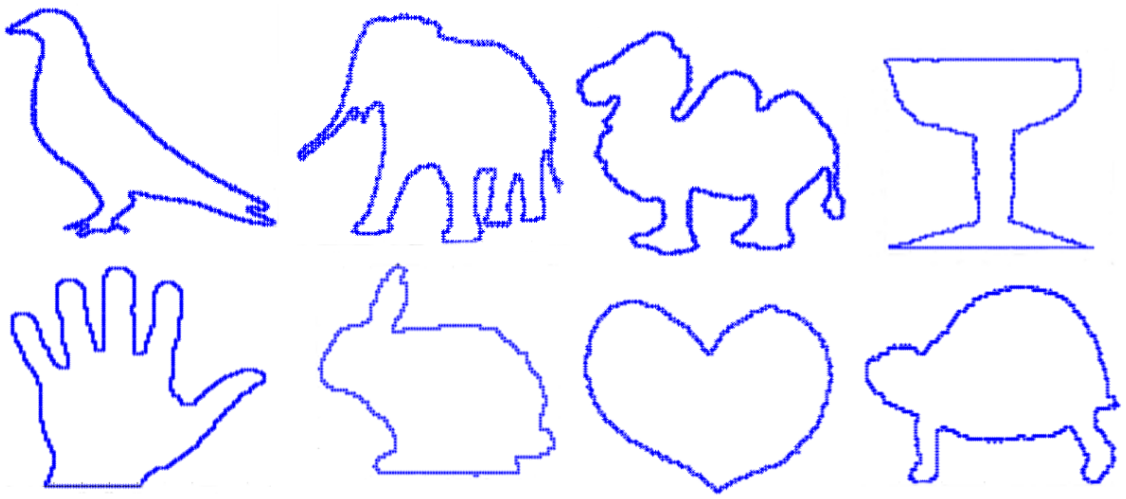


Figure 3.2: Extracted outlines with the Moore-Neighbor algorithm

### 3.2.2 Generation of data shapes

For the classification of new observed data shapes we simulate a data set with realisations from random classes of the Kimia database. Since we assume that each of the acquired observations has been generated by an idealised closed, planar curve we use these for the generation of the observed data shapes. The process is as follows: we choose a random binary image from a random class of the KIMIA database. The boundaries of the chosen images are extracted by the described Moore-Neighbor algorithm. The extraction method returns the coordinate values of the boundary pixels which is the discretisation of the underlying closed planar curve that we chose to represent all our shapes. Since the discretisation of the underlying curve returns more landmarks than is necessary we randomly select a subset of these to represent the data shape. We then assign a random rotation, scale and translation to the chosen landmarks and add isotropic Gaussian noise to each of them. This is the final result which represents the observed data shape. Figure (3.3) illustrates examples of generated data shapes from the Kimia database as described above.

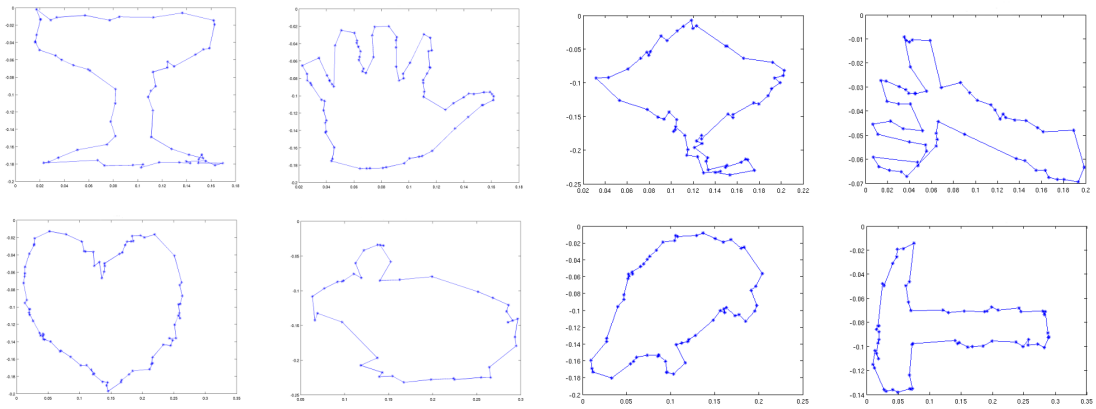


Figure 3.3: Examples of data shapes

### 3.2.3 Generation of example shapes

As we mentioned in the previous chapter, to perform classification of new observed data shapes the probability of each class has to be evaluated through the approximation algorithm we proposed. This involves the marginalisation of the likelihood with respect to nuisance parameters that are involved in the data formation process.

Although some of the results were found in closed form solutions, we were not as lucky with the integration of others for which we employed Monte Carlo integration. One of the nuisance parameters that has to be integrated is the shape curve $\mathscr{D}\beta$ which effectively compares the data shapes to all possible idealised curves of all available classes under the chosen prior shape model $\mathbb{P}(\beta|C)$. The shape model is then used for the integration of the marginalised likelihood over all possible curves $\mathscr{D}\beta$ $\mathbb{P}(\beta|C)$. Since in the Kimia database we have no prior information about the curves or any special characteristics that they bear we choose to represent the shape models by a uniform distribution.

The shapes in the Kimia database are not described by parameters following a probability distribution; instead each class has a few ideal shapes. To sum over all curves, the observed data shapes must be compared to the available idealised example shapes of the given class. For this comparison we need to form all idealised example shapes of the available classes. Since the Kimia database has 12 images for every class, we suggest that these can be assumed to be all the representative idealised shapes of a given class. Then the integration over the curves $\beta$ is done by comparing the data shapes to all 12 idealised example shapes of all classes (in total 256 shapes) and in the end by summing over all classes against $\mathbb{P}(\beta|C)$. For the generation of the example shape we use the following technique. We firstly extract the outline of shapes by using the described Moore-Neighbor algorithm. As before, we randomly choose a subset of the generated landmarks. We linearly interpolate between each pair of points and then apply a diffeomorphism, as described in section [2.3.1], so that the diffeormorphism pushes forward the chosen points from their original place to a new position. The diffeomorphism is applied for practical reasons so that we can evaluate the Monte Carlo integration of the likelihood for both the curves $\beta$ and the samplings $s$ simultaneously. In other words, we generate random realisations from $\mathbb{P}(\beta|C)$ and each one of them is evaluated at a different realisation from $\mathbb{P}(s)$. Examples of generated sample shapes are shown in figure (3.4).

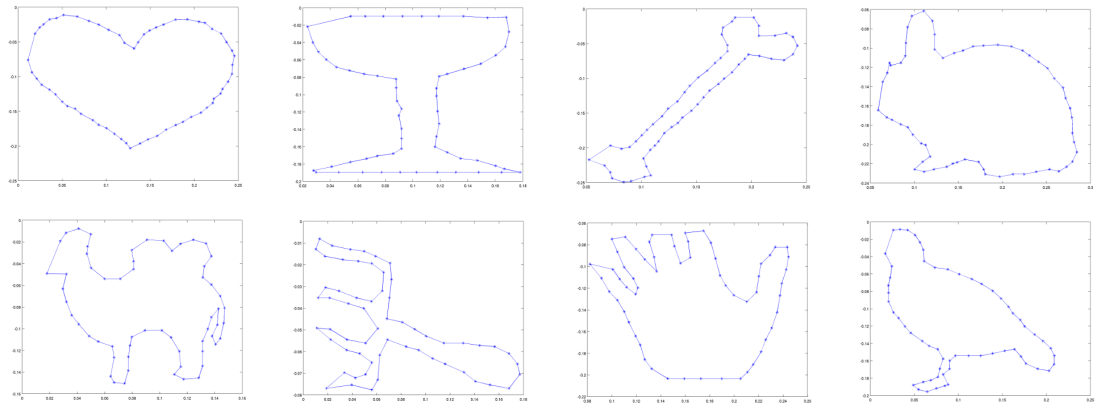


Figure 3.4: Examples of sampled idealised example shapes

### 3.2.4 Confidence and success results

In this section we investigate the properties of the classification algorithm we proposed in chapter [2]. The posterior probability of a class $\mathbb{P}(C|\boldsymbol{y})$ can be approximated by:

$$\mathbb{P}(C|y) \propto \sum_{b\in\mathcal{B}} \int \mathscr{D}\beta \; \mathscr{D}s \; \mathbb{P}(y|b,\beta,s)\mathbb{P}(b)\mathbb{P}(s)\mathbb{P}(\beta|C)\mathbb{P}(C)$$

where we have analytically integrated over all similarity transformations. To fully approximate the posterior distribution of a class $C$ one needs to integrate over the remaining nuisance parameters namely the samplings $s$, the shape curves $\beta$ and sum over all bijections. The summation is calculated exhaustively and we would like to remind the reader that because we are summing over cyclic bijections the result is invariant to the choice of the first data point; this is the case for both the Kimia and the alphabet database. However, the integrals over $s$ and $\beta$ must be done by simple Monte Carlo techniques. This increases our uncertainty over the results due to the randomness of the procedure. For this reason, in this section we examine the confidence levels and the success rates of the algorithm as we vary these parameters that can affect the accuracy of the classification results of the algorithm. In the next subsections we present the results we found for the different values of the samplings $s$, the number of landmarks and, in the case of the sand body database,

the iterations over the curves $\beta$; this is not included in the Kimia and the alphabet databases since they both have a discrete number of idealised curves in each class.

### Confidence level for Monte Carlo iterations of samplings

In this section we present the confidence levels of the algorithm and explore how much they vary as the iterations over the sampling integration increase. To produce the graphs of the Monte Carlo iterations we used the following process. We simulated a data shape whose underlying shape class was picked randomly with equal probability. It was created by the method explained in section [3.2.2] with 40 landmarks around its boundary. The added isotropic Gaussian noise was kept relatively low at $\frac{\sigma}{L} = 0.3 \times 10^{-2}$. For comparison, the noise level is given in terms of the arc length of the curve, $L$ which is usually taken to be equal to 1. That means that if $\sigma = 0.3 \times 10^{-2}$ then the observed data shape $y$ is simulated and the noise perturbs each of the boundary points at $0.3 \times 10^{-2}$ times the length of the true curve. We then performed classification by using our proposed algorithm whilst varying the number of the sampling iterations and keeping other parameters constant. In the next subsection, we investigate the sensitivity of our results to larger and smaller values of $\sigma$ in order to better understand how resilient this approach is to noisy data.

Since we work with the regularised version of the likelihood (3.2.1),

$$\begin{aligned}\mathbb{P}(y|b,\beta,s) &= \sum_{b} \int \mathscr{D}\beta \; \mathscr{D}s \; \mathbb{P}(y|b,\beta,s)\mathbb{P}(b)\mathbb{P}(\beta|C)\mathbb{P}(s) \\ &= \sum_{b\in\mathcal{B}} \int \mathscr{D}\beta \; \mathscr{D}s \; \frac{1}{Z}\left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \mathbb{P}(b)\mathbb{P}(\beta|C)\mathbb{P}(s)\end{aligned} \tag{3.2.1}$$

we need to choose the values of the regulators. For the following runs the values of the regulators were chosen to be: $B = 10^5$, $D = 10^5$, $\alpha = 1.5$ and $\zeta = 0.1$. The variance of the generalised Gaussian of the diffeomorphisms (2.3.7) was chosen to be $\sigma_{\mathrm{s}} = 1.5$. Once we have approximated the posterior probability of the classes $\mathbb{P}(C_i|y)$ the observed data shape $y$ can be classified into the shape class by ranking the classes

according to the posteriors and picking the highest. Since the data shape $y$ was simulated from a known class we can compare the estimated class to the true class. For the following results the Monte Carlo iterations of the sampling were increased to 500 for 18 different runs (simulations of the data shape $y$). A deeper analysis would ideally involve the study of the dependence of the classification results on the limiting values chosen for the regulators. It is important that these values are suitably chosen such that our results are not unduly sensitive to variations in these parameters which therefore defines how large or small these variables ought to be taken in any future application of the work in this thesis. We leave this study for future work in the interests of demonstrating here the validity of our model.

The following graphs provide the confidence levels for 6 out of the 18 different runs of the algorithm. One notices that the confidence level is stabilised for a threshold $\epsilon = 0.01$ after 20 iterations.

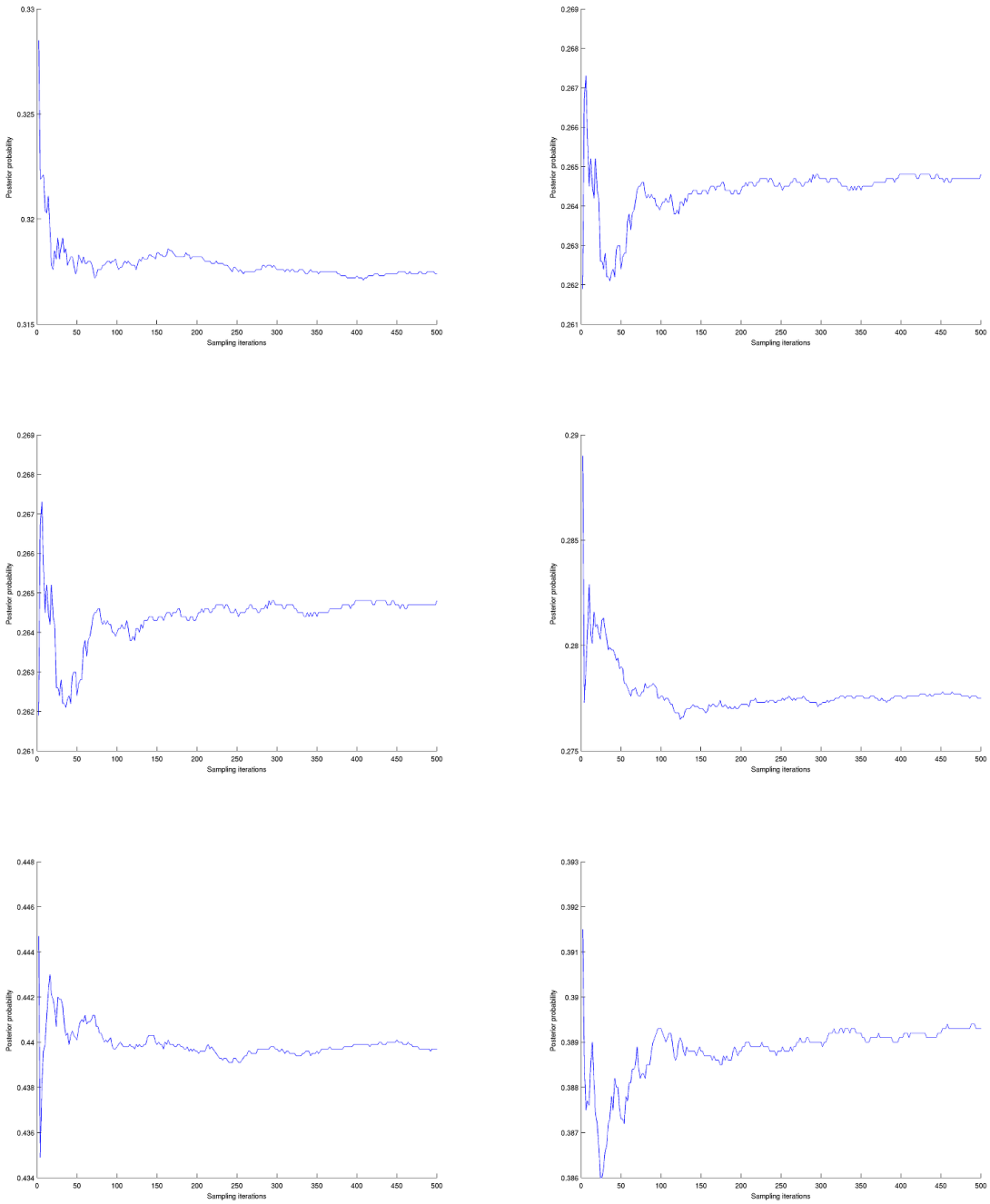



Figure 3.5: Confidence levels against 500 sampling iterations $s$

In the following 4 graphs the scale of the graphs helps to confirm that the algorithm stabilises after 20 iterations. The confidence levels for the above mentioned parameters vary from 26 to 46 percent which are relatively small but one has to bear in mind that the confidence levels are also affected by other parameters such

as the number of points.

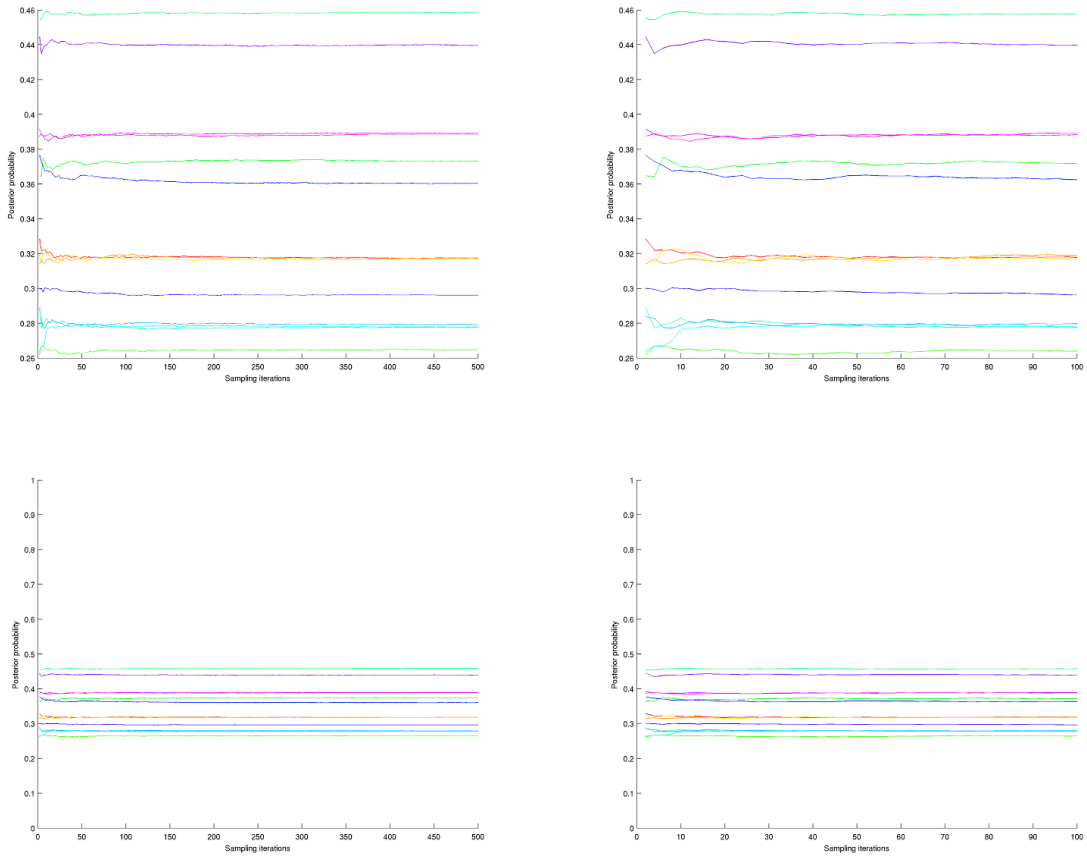



Figure 3.6: Confidence levels against the sampling iterations $s$

### Confidence levels for $\sigma$

In this section we present how the confidence levels of the algorithm vary as the Gaussian noise $\sigma$ increases. This experiment is important to determine at what point the noise is too great for the algorithm to perform as intended. To produce the graphs of the Gaussian noise we used the following process. We simulated 40 data shapes whose underlying shape class was picked randomly with equal probability from the Kimia database. They were created by the method explained in section [3.2.2] with 40 landmarks around their boundary. Starting from the base data shape with zero noise we have added Gaussian noise in increments of $0.15 \times 10^{-2}$ to each of the remaining 39 shapes. We then performed classification using the minimum number of Monte Carlo iterations needed for the classification. This experiment

was conducted in order to identify the impact of the observational noise on the classification results of the generated data shapes.

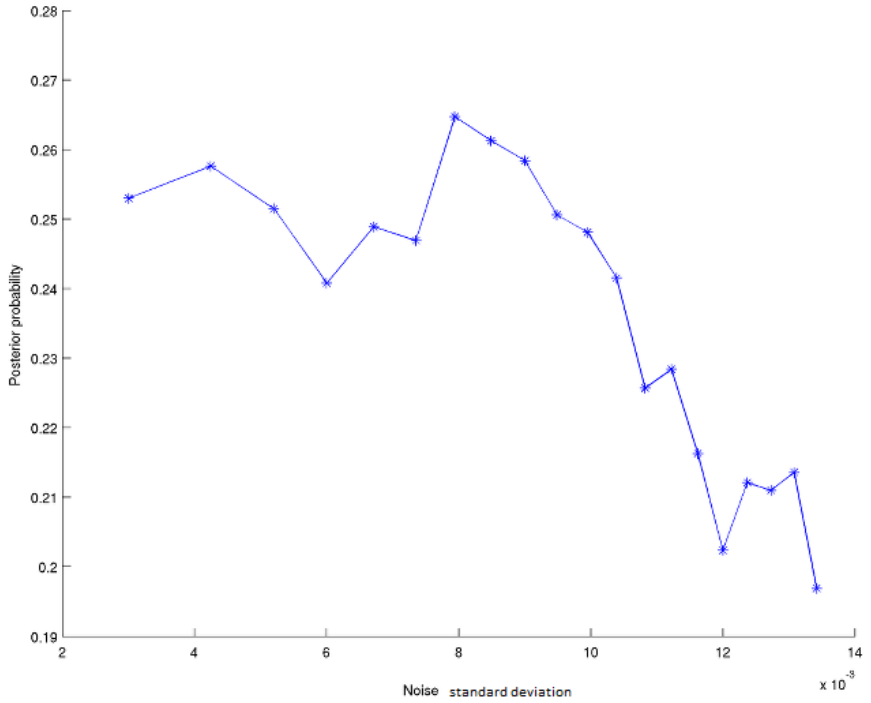



Figure 3.7: Confidence results against Gaussian noise $\sigma$

The graph above represents the average classification rates for each level of noise added in the 40 data shapes. One notices that the classification levels drop from 25% to 20% as the standard deviation of the added noise increases from $0.2\times10^{-2}$ to $1.4\times10^{-2}$. There are two points to make here. Firstly we notice that the classification levels are relatively low and secondly the increase of the Gaussian noise has the effect one would expect. As the noise $\sigma$ increases the classification levels drop by 5%. We would expect the same behaviour if the noise would increase even more, but have focused on small increases of $\sigma$ in the interest of brevity and simulation efficiency.

### Success rates for Monte Carlo iterations of samplings

The following graphs present the success rates of the algorithm against the sampling iterations. The $y$-axis represents the number of correct classifications for the 20 shapes of each run. For each individual run the data shapes were generated with 25

points and the noise was kept relatively low at $\sigma = 0.3 \times 10^{-2}$. One notices that the success rate stabilises after 20 iterations (in some cases after 10 iterations) and the success rate ranges from 85 to 90 percent. For simplicity, we present 6 out of the 20 runs.

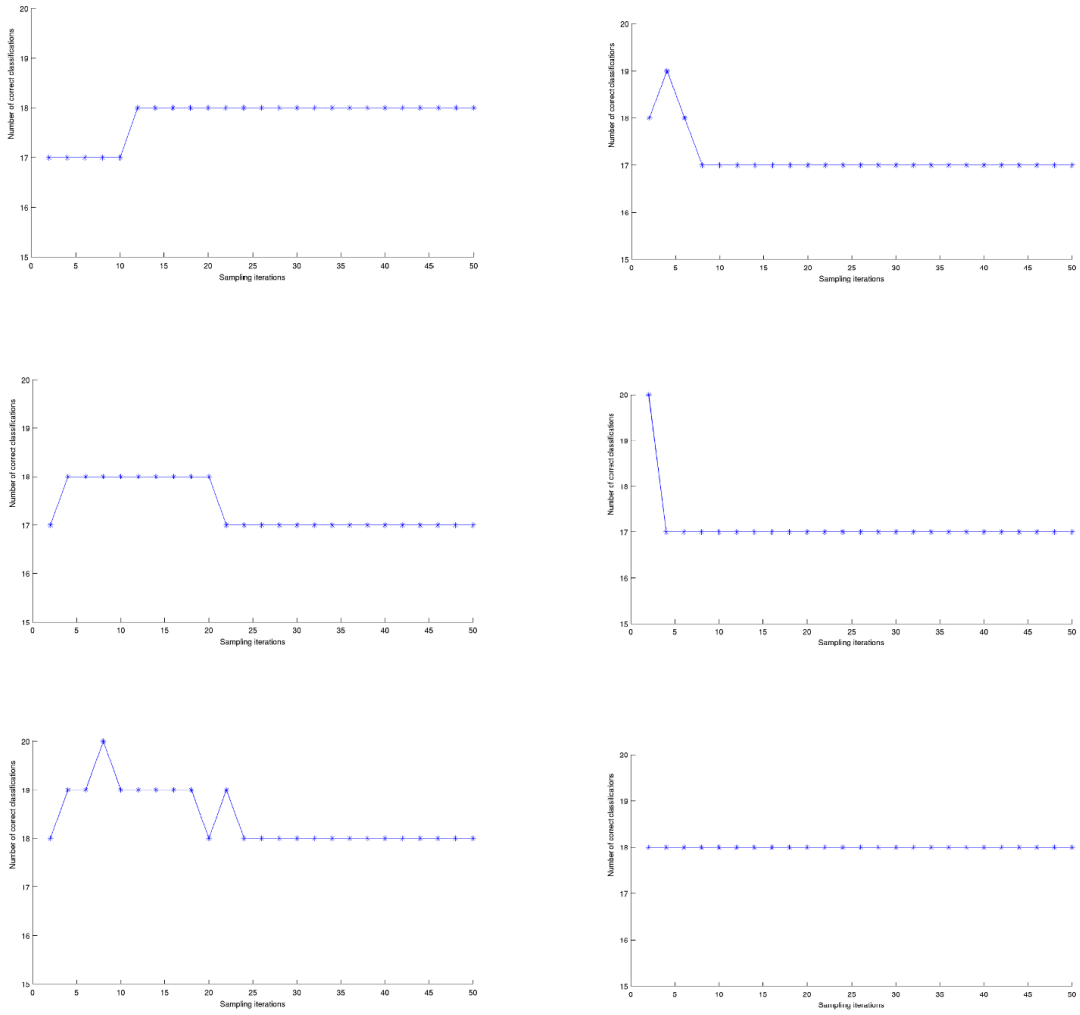



Figure 3.8: Success rate against the sampling iterations for 20 different shapes

In summary, although the confidence levels of the algorithm against the sampling iterations remain relatively low, 26 to 46 percent, the algorithm converges to the level value after 20 iterations. The same behaviour is noted when we examine the success rates whilst the sampling iterations vary. The success rates range from 85 to 90 percent and the results imply that 20 sampling iterations are enough for such

a high success rate although the confidence levels are low.

### 3.2.5 Classification results

To classify the observed data shapes, we evaluate and approximate the posterior probability for each of the classes via the proposed algorithm in chapter [2]. The observed data shapes are then classified to the class that assigns the highest posterior probability to it. Since the data shapes $\boldsymbol{y}$ have been generated by known classes it is easy to evaluate how the algorithm performs by comparing the estimated classes to the true classes.

Obtaining the values of the minimum Monte Carlo iterations needed for stable confidence levels we can now proceed to the classification of observed data shapes. For the following experiment we simulated 10 different shapes from the same class for 10 different runs. We then approximated the posterior $\mathbb{P}(C_i|y)$ and picked the highest posterior which effectively gave the class in which the shapes were classified which evaluates the performance of the algorithm. The following 10 shapes were generated from the class of "bones". The shapes were generated with 30 points and the noise's standard deviation was $\sigma = 0.4 \times 10^{-2}$. The sampling iterations were fixed to be 20, the minimum number that is needed for the confidence results to stabilise. The first graph presents the 10 simulated shapes to be superimposed whereas the second one shows one example of such a shape. For simplicity, we present 4 out of 10 classification results. The success rate for these 10 runs was 90 percent with 9 out of 10 shapes being classified into their respective category correctly. For the correctly classified shapes the confidence levels ranged between 26 to 35 percent with the average confidence level at 33 percent. In the misclassification case, presented in the last graph, the observed shape is classified as a "tool." This is an expected behaviour since the classes of bones and tools are very similar. However one notices that although the confidence level for the class tool is 26 percent, the second higher posterior is for the class of bones with confidence level of 25 percent.

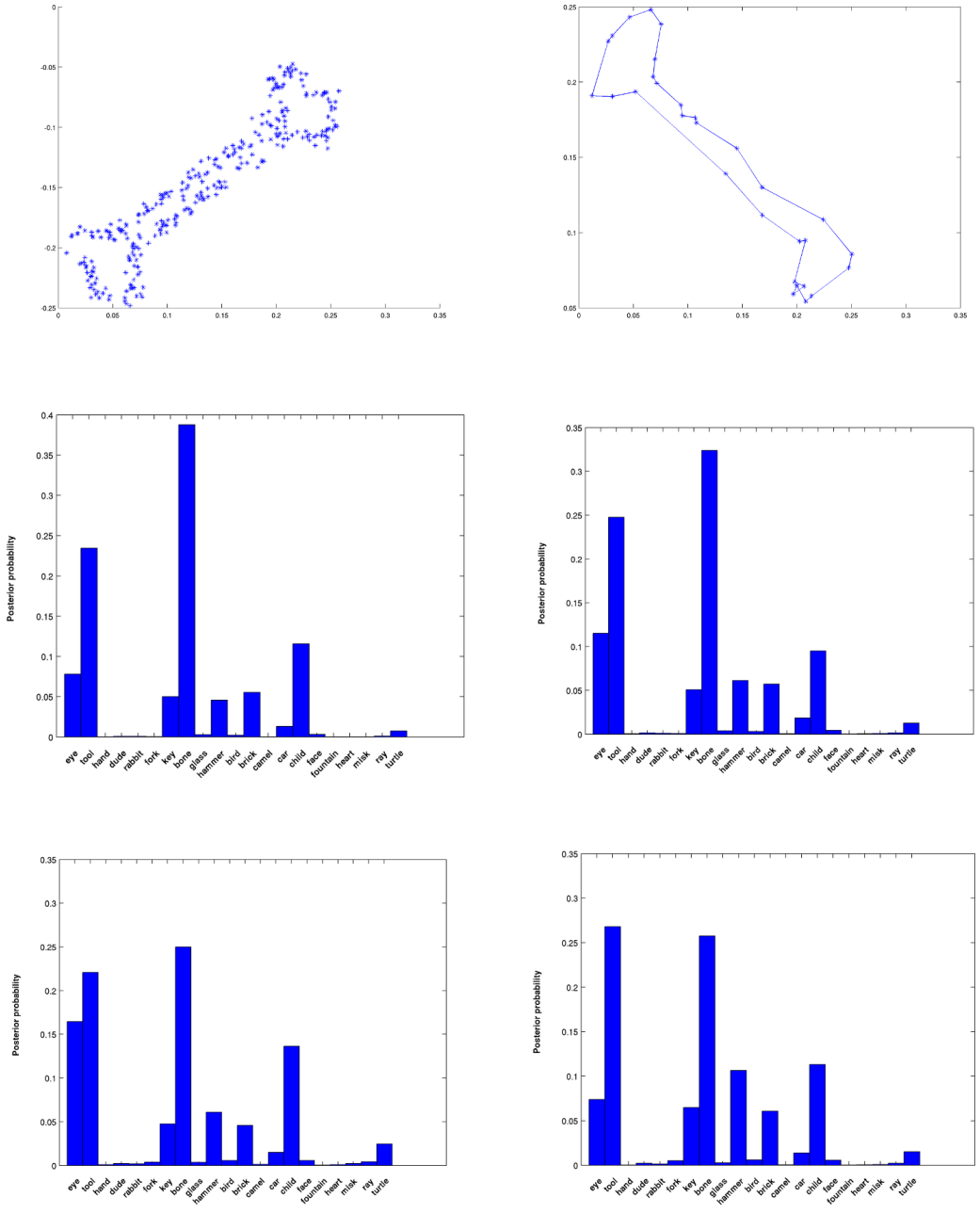



Figure 3.9: Classification results for the class of bones

The next 10 shapes were simulated from the class of "camel." For the simulation of the observed data shapes we used 40 landmarks and the standard deviation of the added noise was $\sigma = 0.5 \times 10^{-2}$ which is regarded to be high. This is also reflected in the results where the success rate is 40 percent with only 4 out of 10 shapes classified in the correct category. The last three graphs present three misclassification rates

where the data shapes are either classified as face, rabbit or misk. This is due to the similarity of the classes but also due to the presence of the high noise which causes the classification levels, even when correct, to be kept quite low between 13 and 23 percent with an average confidence level at 19 percent.

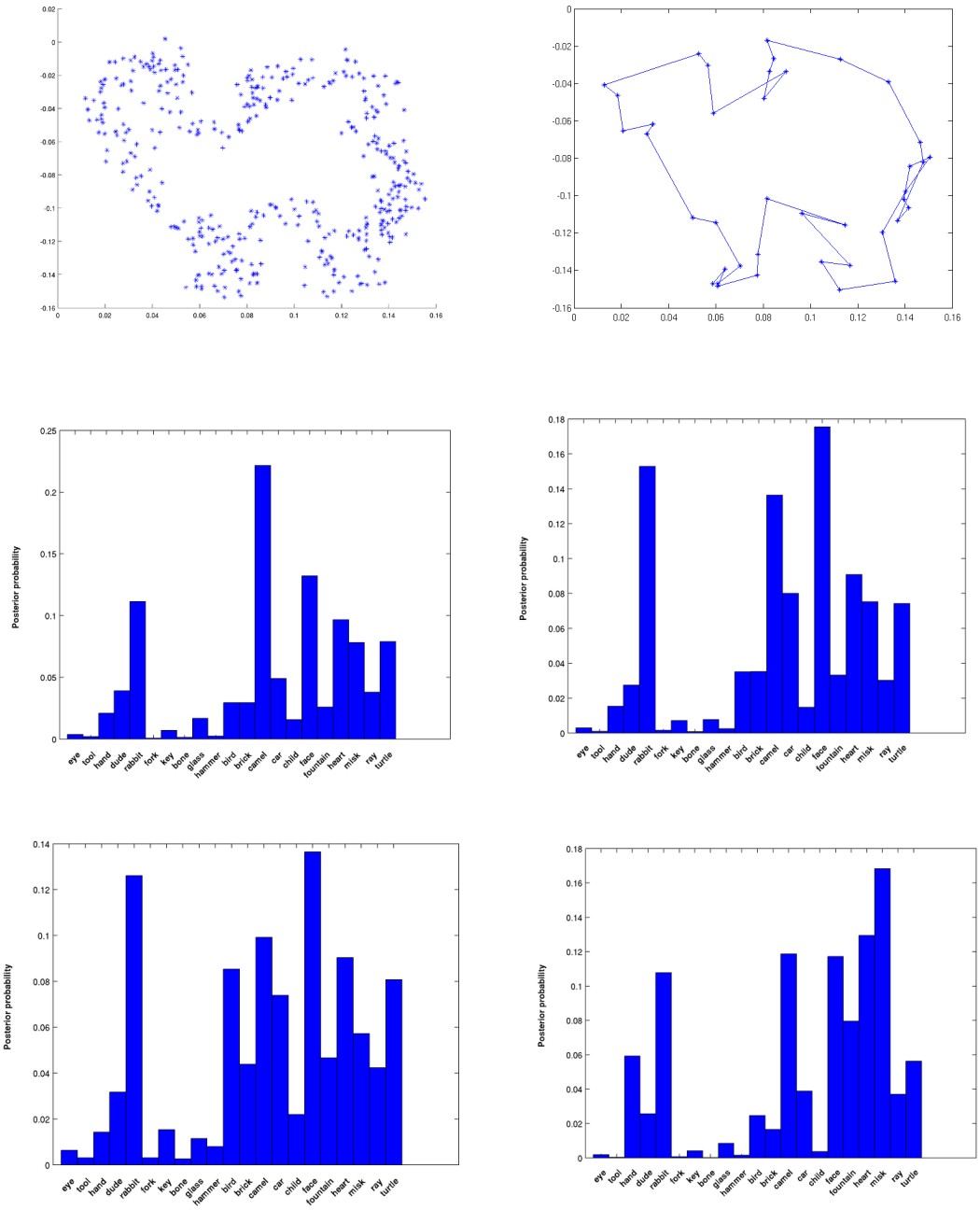



Figure 3.10: Classification results for the class of camels

The next 10 shapes were simulated from the class of "forks." For their simulation

we used 40 points and the standard deviation of the observed noise was $\sigma = 0.4 \times 10^{-2}$. Although this run was done with the same number of points as in the previous run, the presence of the lower noise pushed the success rate to 90 percent; 9 out of 10 shapes were classified correctly. The second highest posterior is usually attributed in the class of camels as can be seen in the second graph. In addition, in the one case of misclassification the shape was classified in the class of camel. The lowest classification level was 23 percent whilst the highest was 90 percent with the average confidence level to be 58 percent. This shows that the classification levels are sensitive to the presence of noise and an increase of $10^{-3}$ can cause a fall of 70 percent in the confidence.

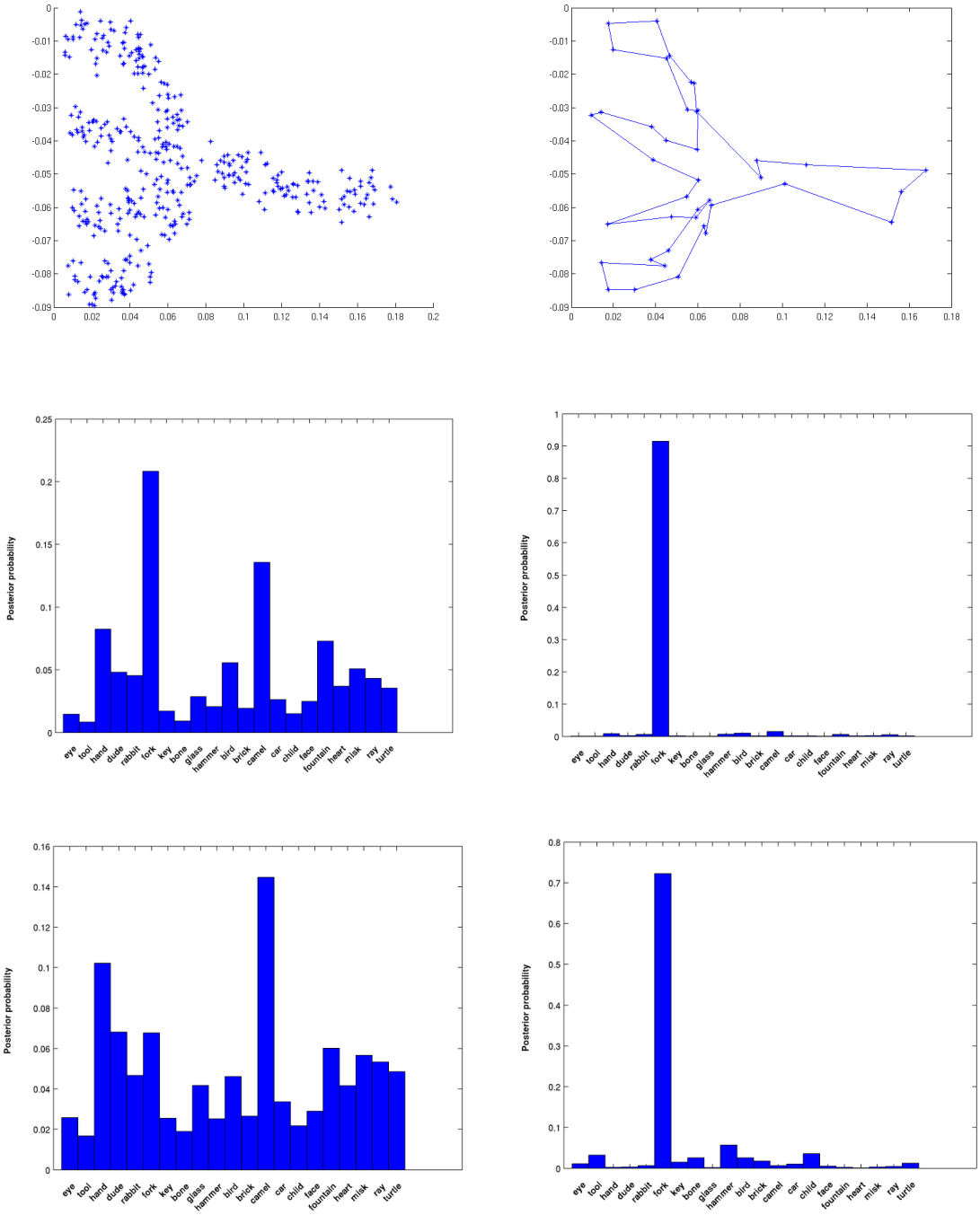



Figure 3.11: Classification results for the class of forks

The next 10 shapes were simulated from the class of "hammers." For the generation of the data shapes we used 50 points and the standard deviation of the noise was $\sigma = 0.4 \times 10^{-2}$. In this case, the success rate was 80 percent and the classification levels ranged from 45 to 92 percent with 7 out of 8 correct classifications to have

confidence more than 60 percent. The average confidence level was 78 percent.

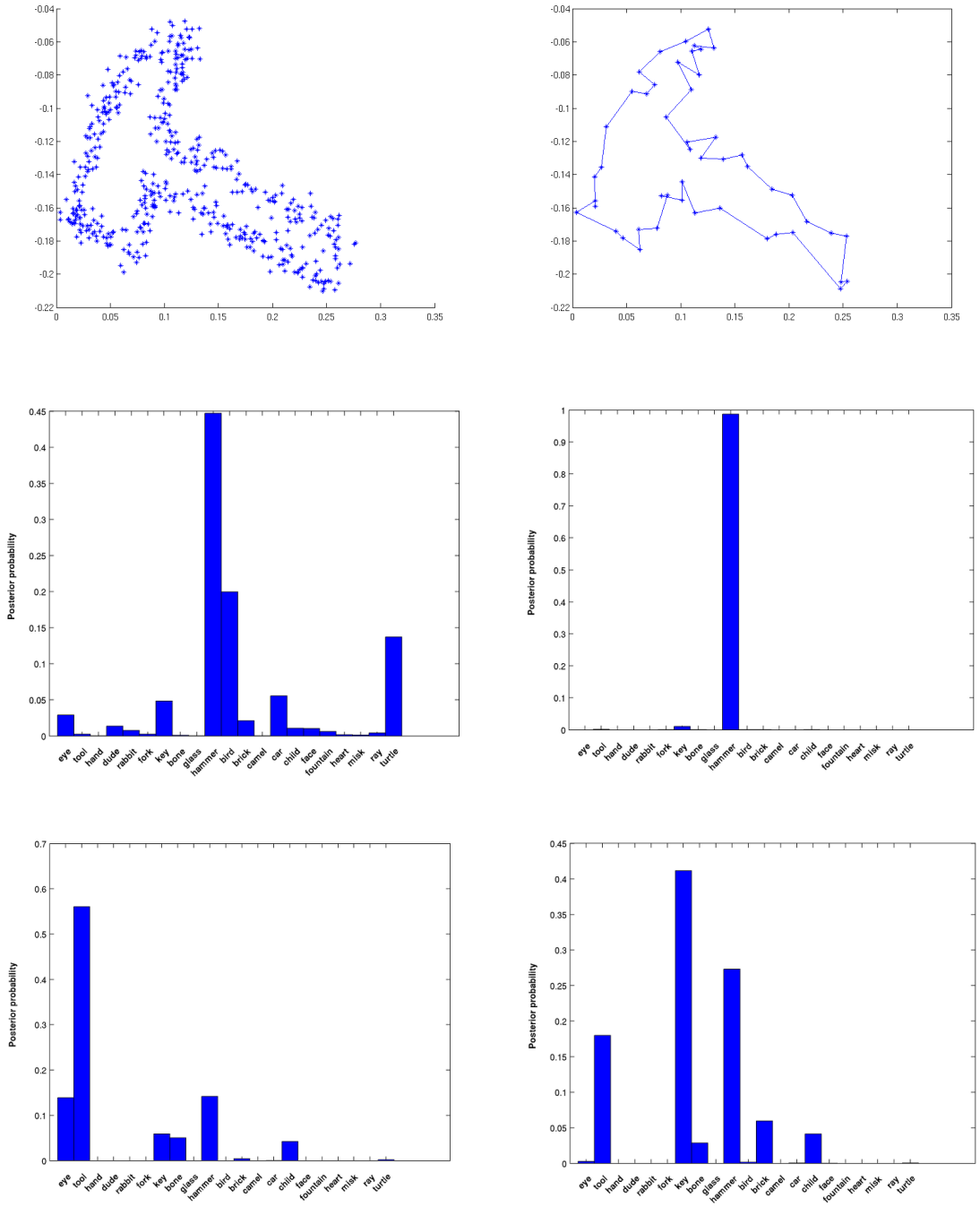



Figure 3.12: Classification results for the class of hammers

The next 10 shapes were simulated from the class of "hands." For the generation of the data shapes we used 50 points and the standard deviation of the noise was $\sigma = 0.8 \times 10^{-2}$ which is regarded to be extremely high. The 10 simulated shapes were

all correctly classified into their respective category with confidence levels ranging from 26 to 75 percent with the average confidence at 40 percent. Although in 8 out of 10 cases the confidence levels were close to 30 percent, the highest posterior was very distinguishable in comparison to the second highest posterior which was close to 10 percent. Although the noise in this case is extremely high and the number of points medium, the high success rate of this experiment is due to the fact that the particular class is quite distinct and recognisable in comparison to other classes (for example the class of bones is easily mistaken and misclassified as a tool or a hammer).

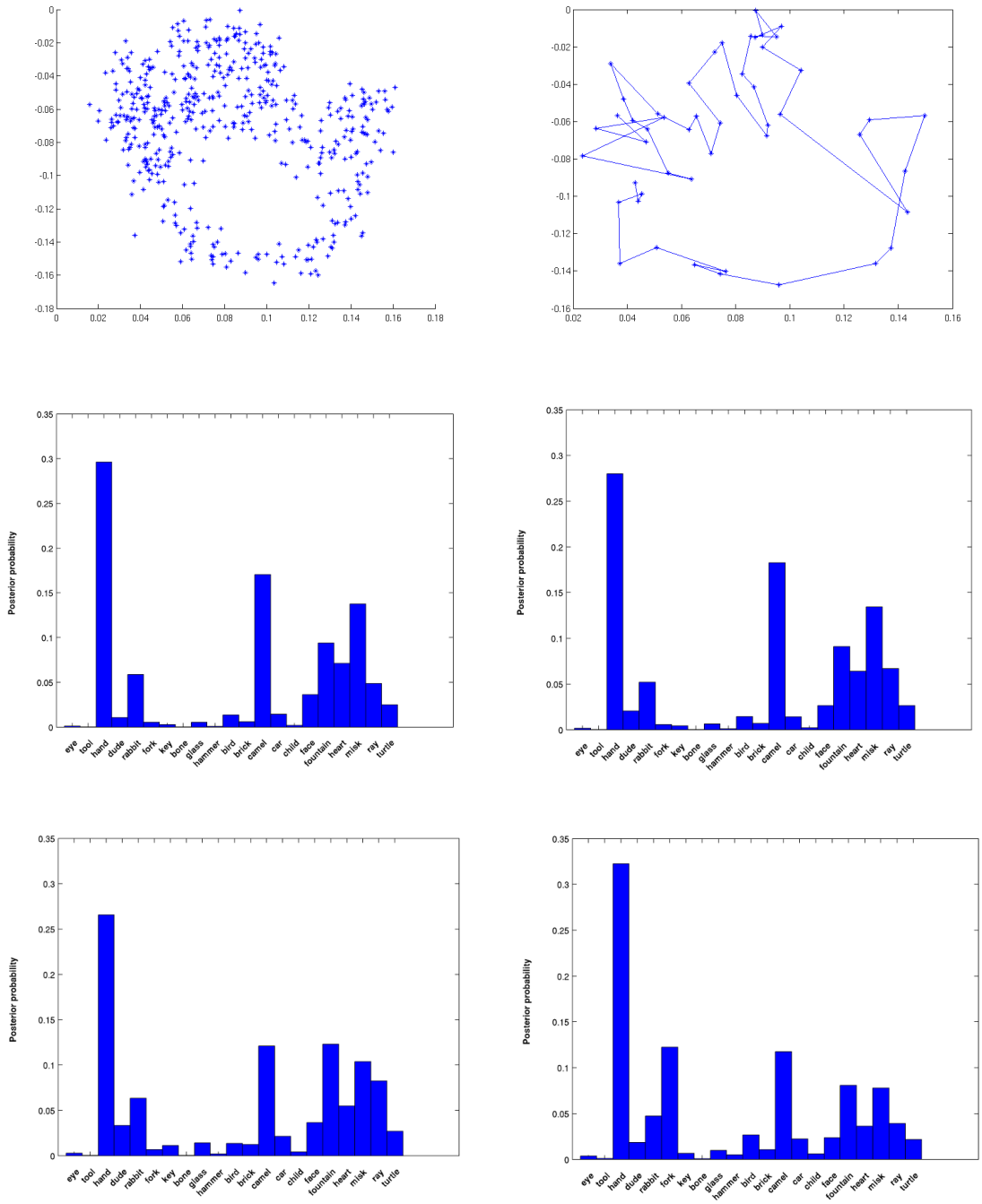



Figure 3.13: Classification results for the class of hands

The last 10 shapes were generated from the category of "tools." For the simulation of the data shapes we used 80 points and the standard deviation of the noise was $\sigma = 0.5 \times 10^{-2}$. There were 9 out of the 10 shapes classified correctly and the confidence results were in all cases more than 90 percent with the average confidence

reaching 93 percent. In the misclassification case, the data shape was classified as a bone with probability almost 80 percent. Although the noise was kept at a high level we see that the confidence levels are much higher than when the noise was kept low but the number of point was 40 or 50. It appears that the higher number of points compensates for the high level of noise.

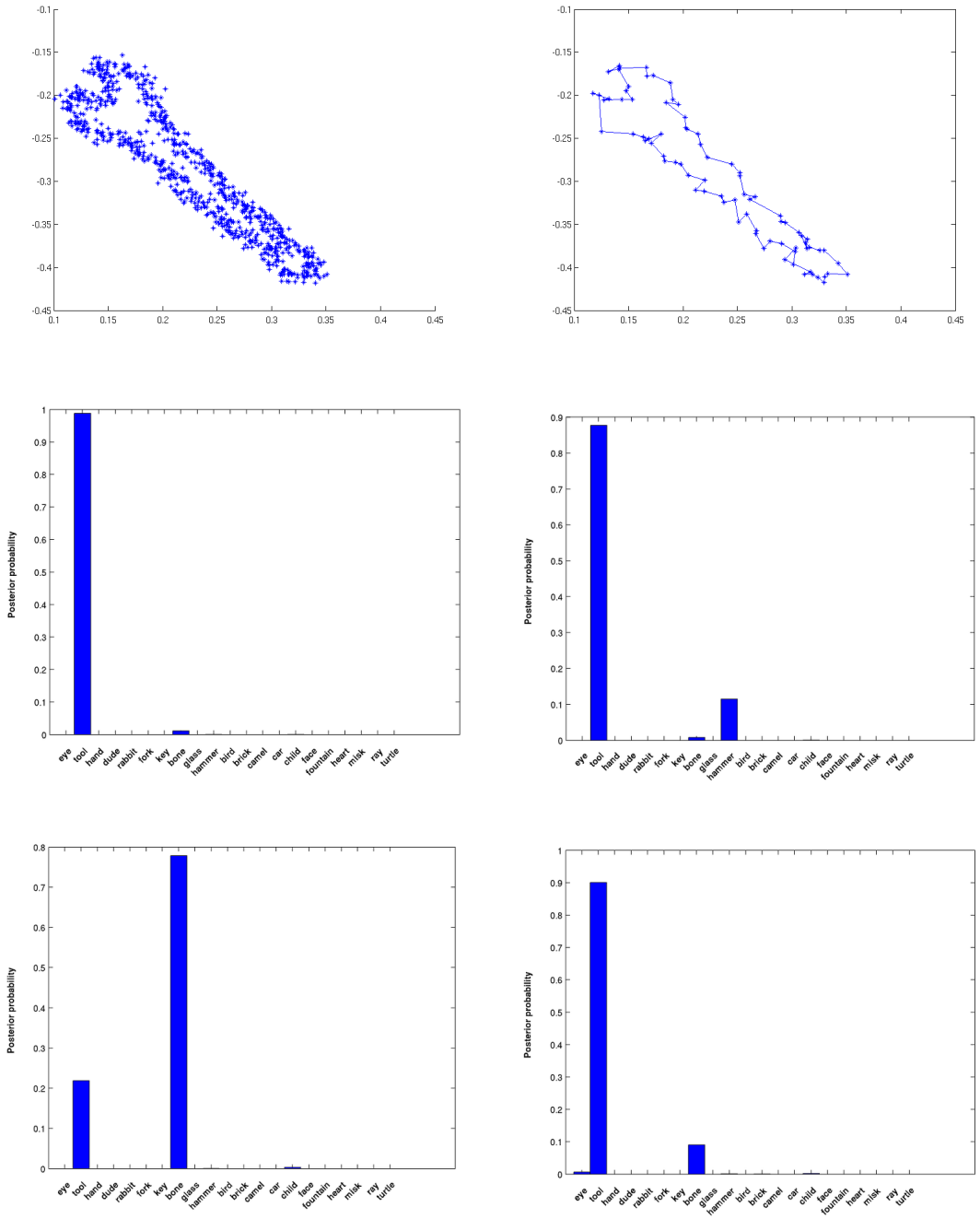
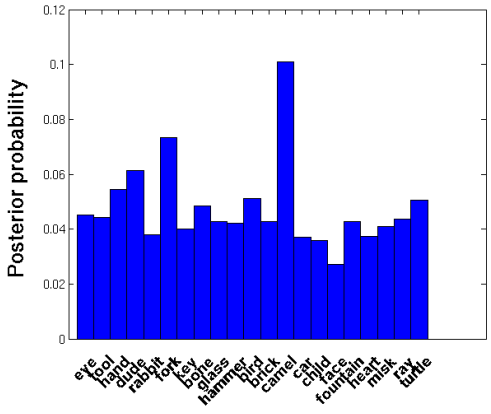



Figure 3.14: Classification results for the class of tools

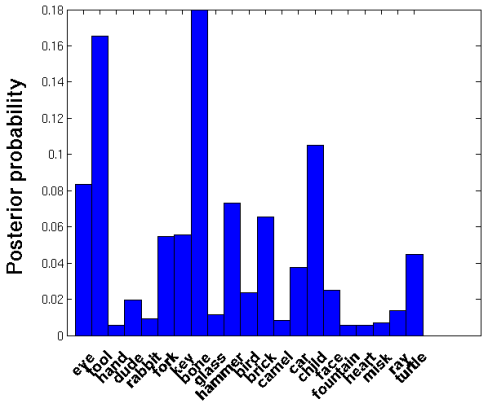
To investigate the classification levels for different numbers of points we classified data shapes from the same class and with stable noise as the number of points increased. The experiment was run 10 times with the number of points ranging from 10 to 100 and the noise being $\sigma = 0.2 \times 10^{-2}$. The 5 graphs in figure (3.15) present the results of 5 runs for the class of "tools." One notices that in the case of 10 and 20 points the data shapes are misclassified whereas for 30 points the classification level is 25 percent. For 50 points the classification levels approach 70 percent and for more than 60 points the confidence is more than 90 percent. The last graph presents the time in hours for a single run as a function of points which shows that for a large number of points the algorithm is computationally expensive.

The following experiment was run 10 times with the number of points ranging from 10 to 100 and the noise being $\sigma = 0.2 \times 10^{-2}$. The 5 graphs in figure (3.16) present the results of 5 runs for the class of "hands." One notices that in the case of 10, 20 and 30 points the data shapes are misclassified whereas for 40 points the classification level is 45 percent. For 50 points the classification levels approach 68 percent and for more than 60 points the confidence is at 100 percent. The last graph presents the time in hours for a single run as a function of points which shows a similar behaviour as in the case of class of tools.

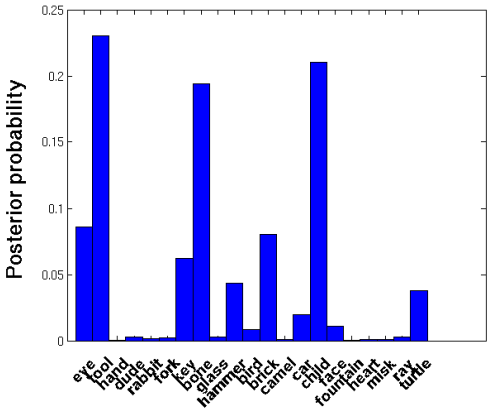
Overall, we performed 10 runs for 10 shapes each. These 100 shapes came from the classes of bones, camels, forks, glass, hammer, hands, hearts, key, rabbit and tools. For the 10 runs, the average classification level was $\hat{\mu} = 59\% \pm 7\%$ and the average success rate was $80\% \pm 5\%$. That means that in average 8 out of 10 shapes were classified correctly with an average classification confidence of 59%. The algorithm is sensitive in the presence of too few points or too high noise however it seems that a high number of points can compensate the high noise. It is also noticeable that the algorithm does not perform well when the number of points is quite small. However, when the number of points increases to 40 points the classification levels become high and in most cases more than 60 percent. In this case, the success rates are also quite high and in most cases more than 80 percent. As soon as the number of points increases to more than 50 the confidence levels



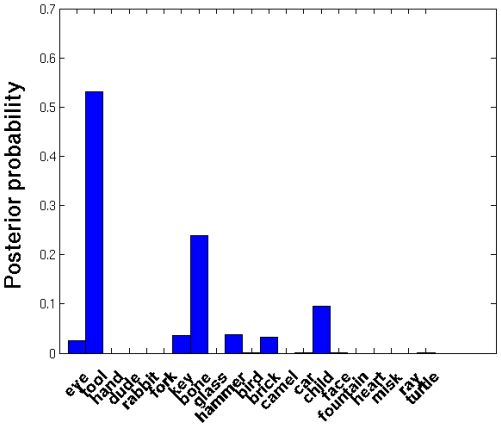

(a) Classification results for 10 points



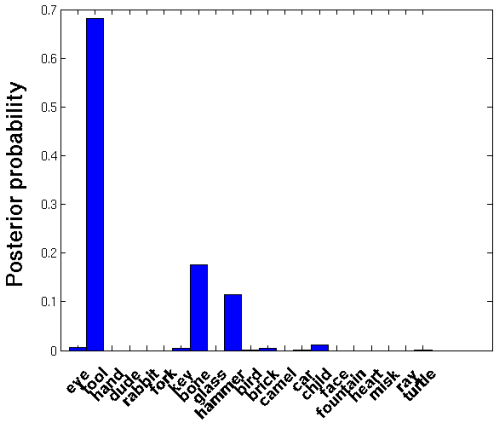

(b) Classification results for 20 points



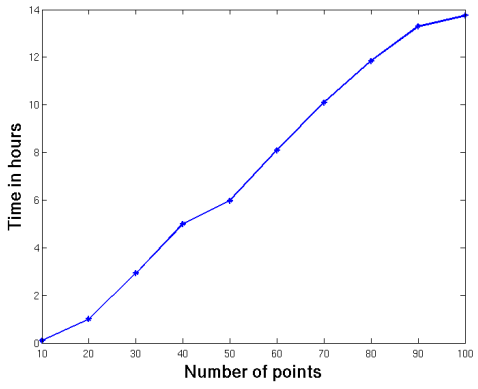

(c) Classification results for 30 points




(d) Classification results for 40 points

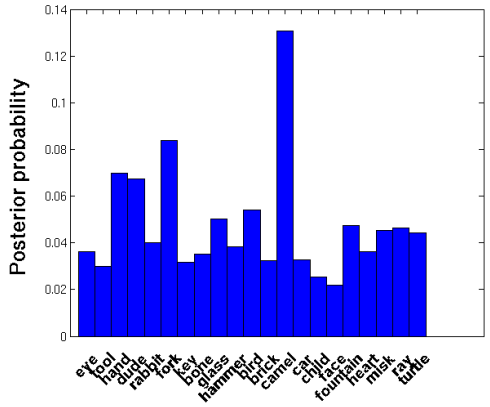



(e) Classification results for 50 points

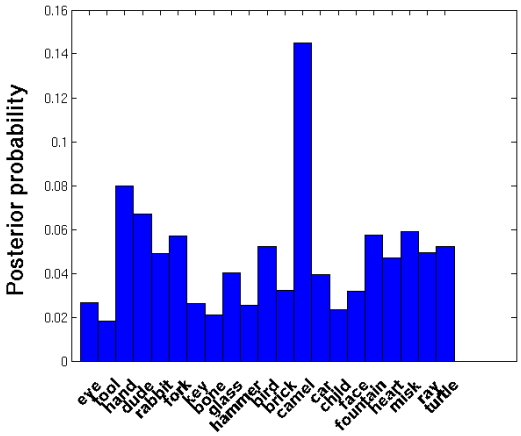



(f) Computational time as a function of the number of points

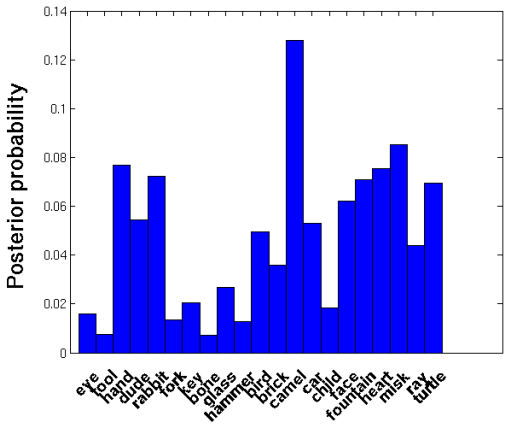
Figure 3.15: Classification results for the class of tools



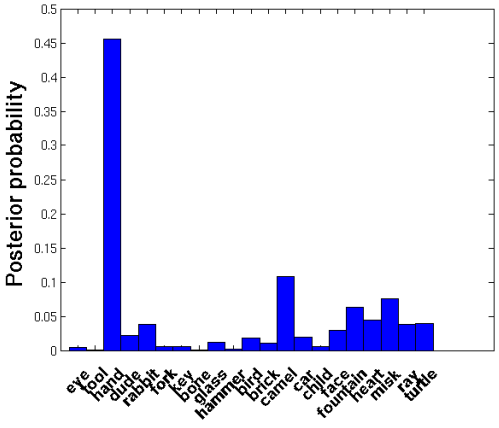

(a) Classification results for 10 points



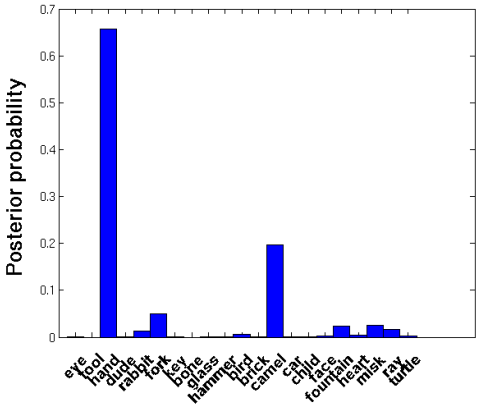

(b) Classification results for 20 points



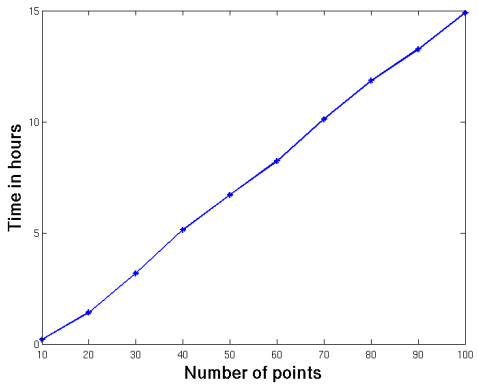

(c) Classification results for 30 points




(d) Classification results for 40 points




(e) Classification results for 50 points




(f) Computational time as a function of the number of points

Figure 3.16: Classification results for the class of hands

become almost more than 90 percent depending on the noise. However, when the number of points becomes more than 50 the algorithm becomes computationally expensive with the computational time being more than 5.5 hours.

## 3.3 Experimental results on letters

The second database we used for the experimental results was the alphabet database which we created ourselves. We made the database as a collection of binary images of the alphabet. Each class of letters (i.e. each letter of the alphabet) was comprised of six different types of fonts: tahoma, times new roman, arial, calibri, sans sherif and courier. In total, the database was comprised of 156 binary images. As with the Kimia database, the shape models $\mathbb{P}(\beta|C)$ we employed here are uniform since we have no prior information about the curves or any special characteristics that they have. For the generation of the example and the data shapes we used the same techniques as with the Kimia database which were described in section [3.2.2] and [3.2.3]. Figure (3.17) shows examples of letters coming from all six fonts of letters T and W and figure (3.18) shows the extracted boundaries of the letter T with the Moore-Neighbor algorithm. Figure (3.19) shows examples of sampled example shapes and figure (3.20) shows examples of generated data shapes. In this instance of course the application of our algorithm comes with a warning; ordinarily the orientation of letters is crucial (for example W versus M and C versus U) whereas our likelihood has been constructed to be invariant under rotations of the data. This section should be understood as a general test of our algorithm which is used for demonstrational purposes and not as a serious proposal for recognition of written letters.

In the next sections we describe the success rates and confidence levels of the algorithm with respect to different varying parameters and we discuss its classification accuracy.

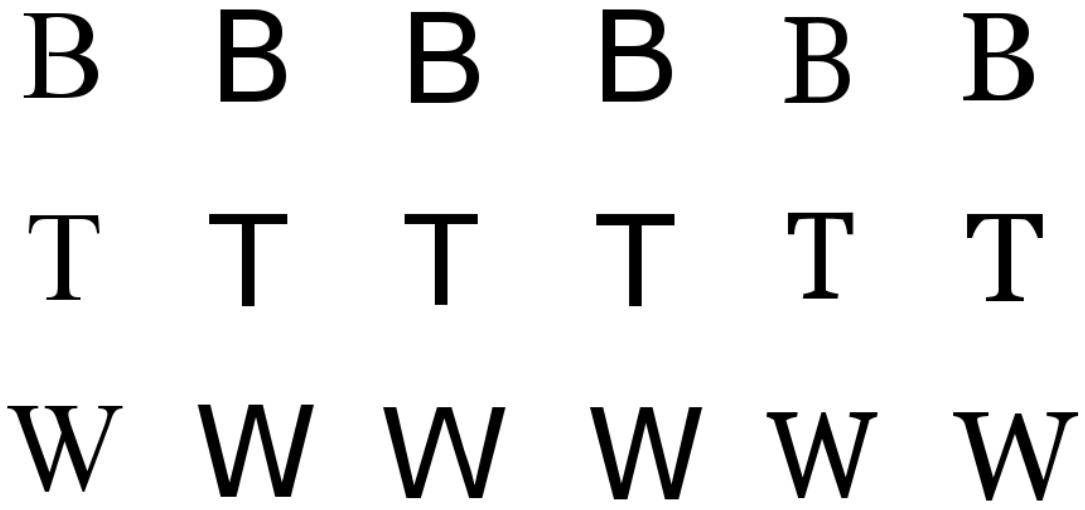



Figure 3.17: Examples of binary letters

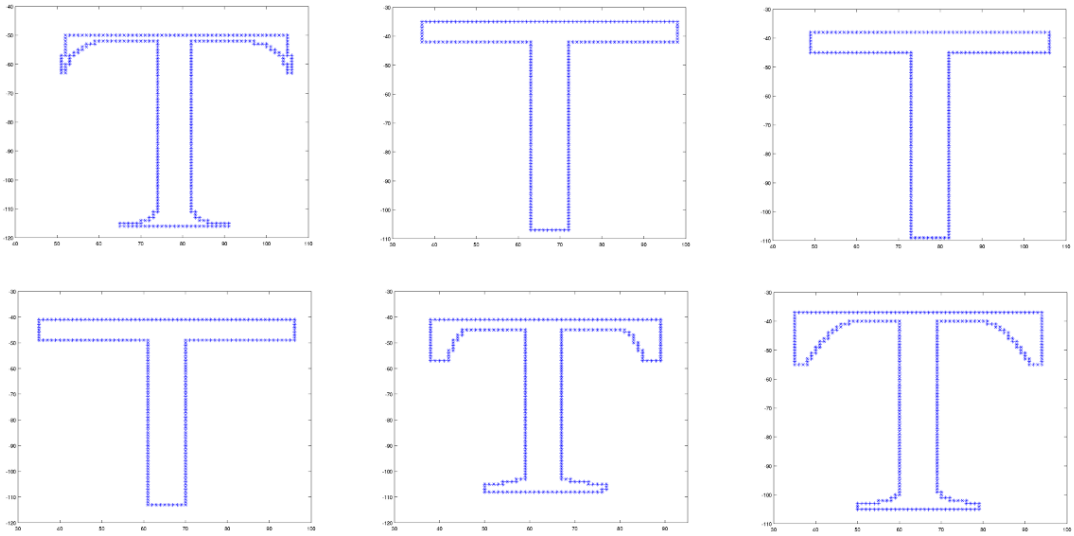


Figure 3.18: Extracted outlines with the Moore-Neighbor algorithm

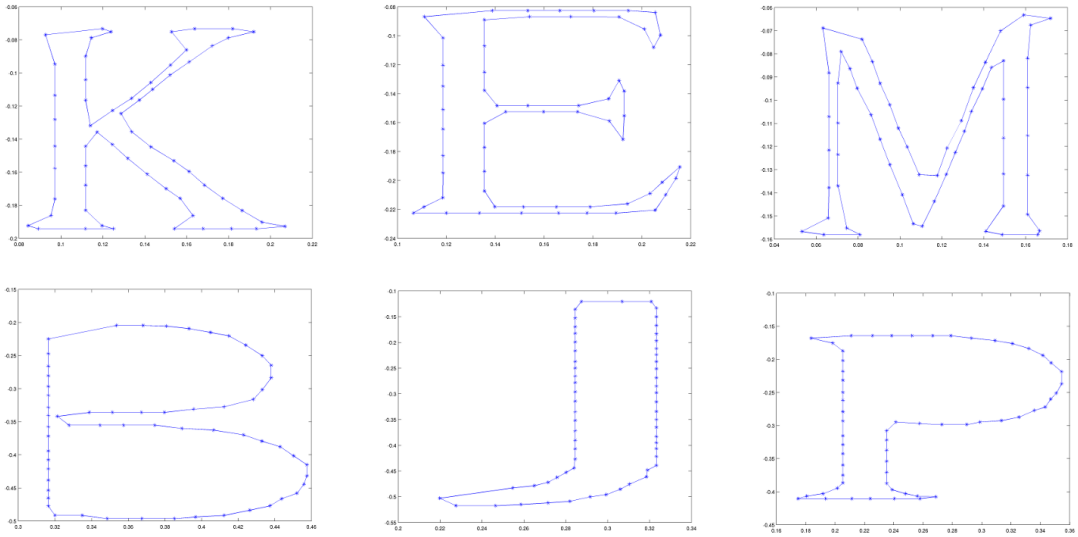


Figure 3.19: Examples of sampled idealised example shapes

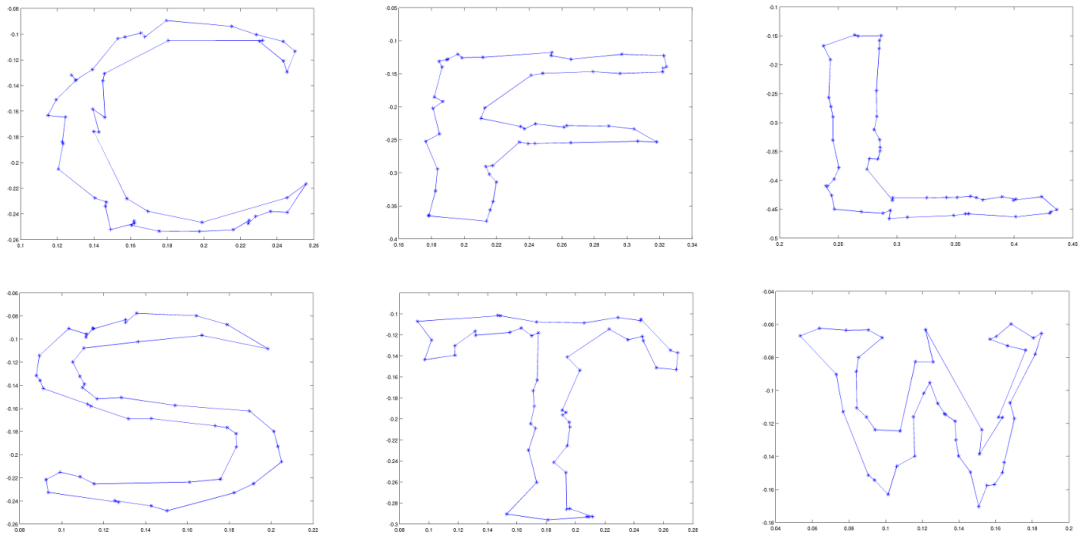


Figure 3.20: Examples of data shapes

### 3.3.1 Confidence and success results

In this section we present the confidence levels and the success results of the algorithm as we vary different parameters. The procedure we followed is the same as for Kimia database as described in section [3.2.4].

### Confidence levels for Monte Carlo iterations of samplings

In this section we present the confidence levels of the algorithm and how much they vary as the iterations over the sampling integration increase for the letter database. To produce the graphs of the Monte Carlo iterations we used the following process. We simulated a data shape whose underlying shape class was picked randomly with equal probability from one of the letter classes. It was created by the method explained in section [3.2.2] with 40 landmarks around its boundary. The added isotropic Gaussian noise was kept relatively low at $\sigma = 0.2 \times 10^{-2}$. As explained previously, the noise level is in terms of the arc length of the curve which means that if $\sigma = 0.2 \times 10^{-2}$ then the observed data shape $y$ is simulated and the noise perturbs each of the boundary points at $0.2 \times 10^{-2}$ times the length of the true curve. We then performed classification by using our proposed algorithm whilst varying the number of the sampling iterations and keeping other parameters constant. Since we work with the regularised version of the likelihood (3.2.1), we need to choose the values of the regulators. For the following runs the values of the regulators were: $B = 10^5$, $D = 10^5$, $\alpha = 1.5$ and $\zeta = 0.1$. The variance of the generalised Gaussian of the diffeomorphisms (2.3.7) was chosen to be $\sigma_s = 1.5$. For the following results the Monte Carlo iterations of the sampling were increased to 500 for 20 different runs (simulations of the data shapes $y$).

The following graphs provide the confidence levels for 6 out of the 20 different runs of the algorithm. One notices that the confidence level is stabilised for a threshold $\epsilon = 0.02$ after 20 iterations as with the case of the Kimia database.

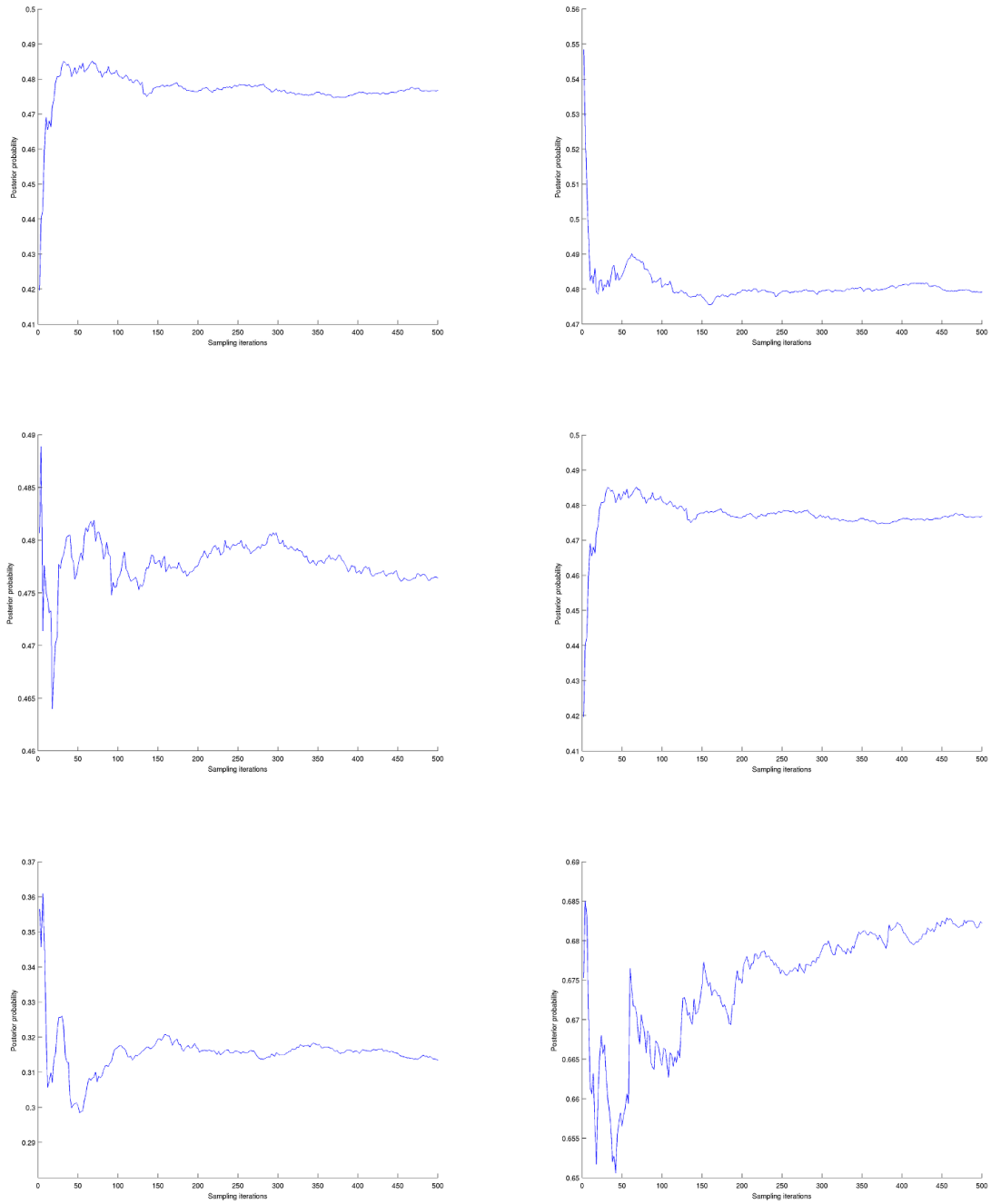



Figure 3.21: Confidence level against the sampling iterations

In the following 4 graphs the scale helps to confirm that the algorithms stabilises after 20 iterations. The confidence levels for the above mentioned parameters, vary from 30 to 90 percent.

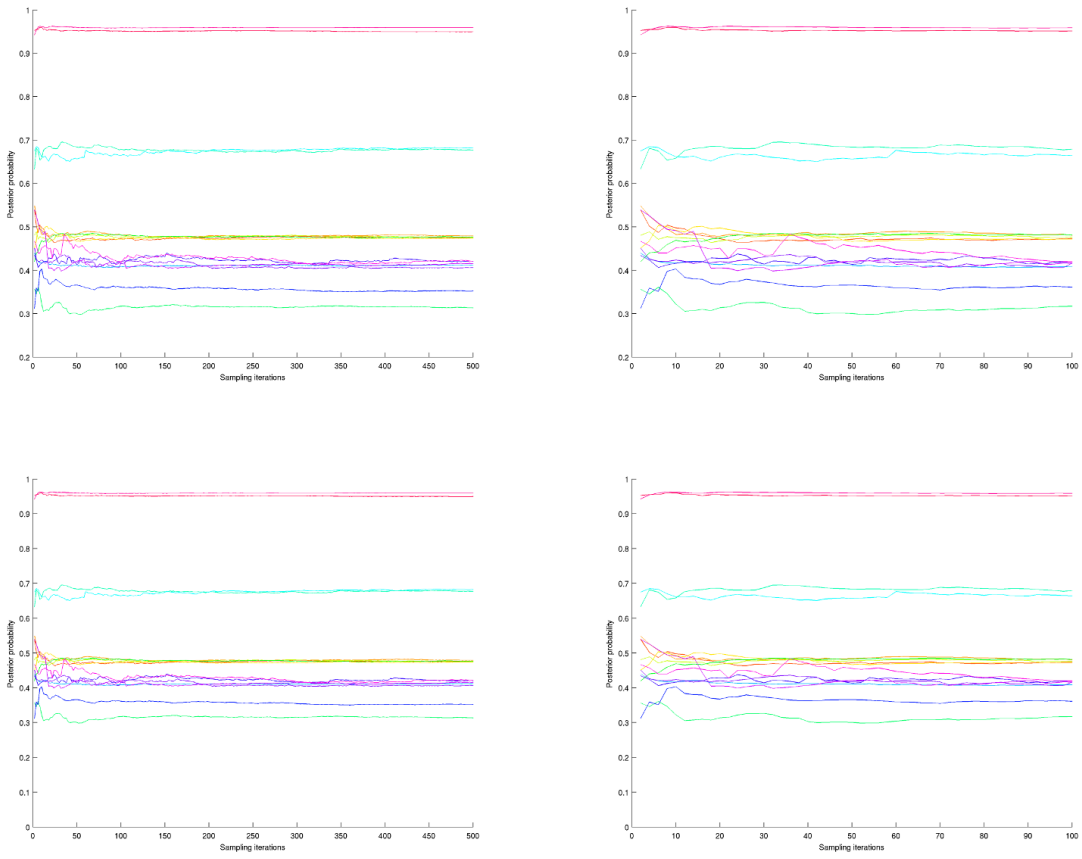
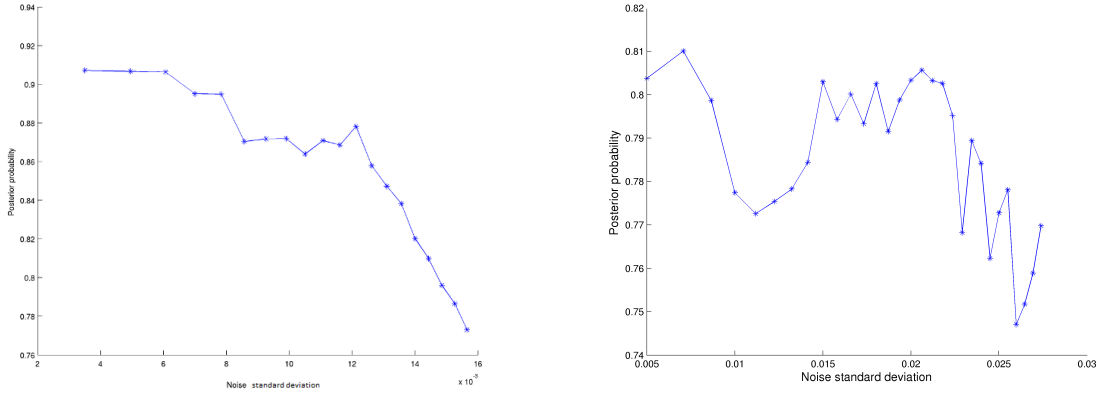



Figure 3.22: Confidence level against the sampling iterations

### Confidence levels for $\sigma$

In this section we present how the confidence levels of the algorithm vary as the Gaussian noise $\sigma$ increases. This experiment is important to determine at what point the noise is too great for the algorithm to perform as intended. To produce the graphs of the Gaussian noise we used the following process. We simulated 40 data shapes, twice, whose underlying shape class was picked randomly with equal probability from the letter database. They were created by the method explained in section [3.2.2] with 40 landmarks around their boundary. Starting from the base data shape with zero noise we have added Gaussian noise in increments of $0.2 \times 10^{-2}$ for figure (3.23a) and increments of $0.5 \times 10^{-2}$ for figure (3.23b) in each of the remaining 39 shapes. We then performed classification using the minimum number of Monte Carlo iterations needed for the classification. This experiment was conducted in

order to identify the impact of the observational noise on the classification results of the generated data shapes.




(a) (b)

Figure 3.23: Confidence results against Gaussian noise $\sigma$

The graphs above represent the average classification rates for each level of noise added in each of the two runs of the 40 data shapes. One notices that, as in the case of the Kimia database, the classification levels presented in figure (3.23a) show that classification results drop from 90% to 77% as the standard deviation of the added noise increases from $0.2 \times 10^{-2}$ to $1.6 \times 10^{-2}$. The classification levels presented in figure (3.23b) drop from 80% to 74% as the standard deviation of the added noise increases from $0.5 \times 10^{-2}$ to $2.5 \times 10^{-2}$. As in the case of the Kimia database, the results are as expected; the increase of the Gaussian noise impacts the classification results which drop almost 20%.

### Success rates for Monte Carlo iterations of samplings

The following graphs present the success rates of the algorithm against the sampling iterations. The $y$-axis represents the number of correct classifications for the 20 shapes of each run. For each individual run the data shapes were generated with 30 points and the noise was kept relatively low at $\sigma = 0.3 \times 10^{-2}$. One notices that the success rate stabilises after 20 iterations and the success rate ranges from 45 to 75 percent. For simplicity, we present 6 out of the 20 runs.

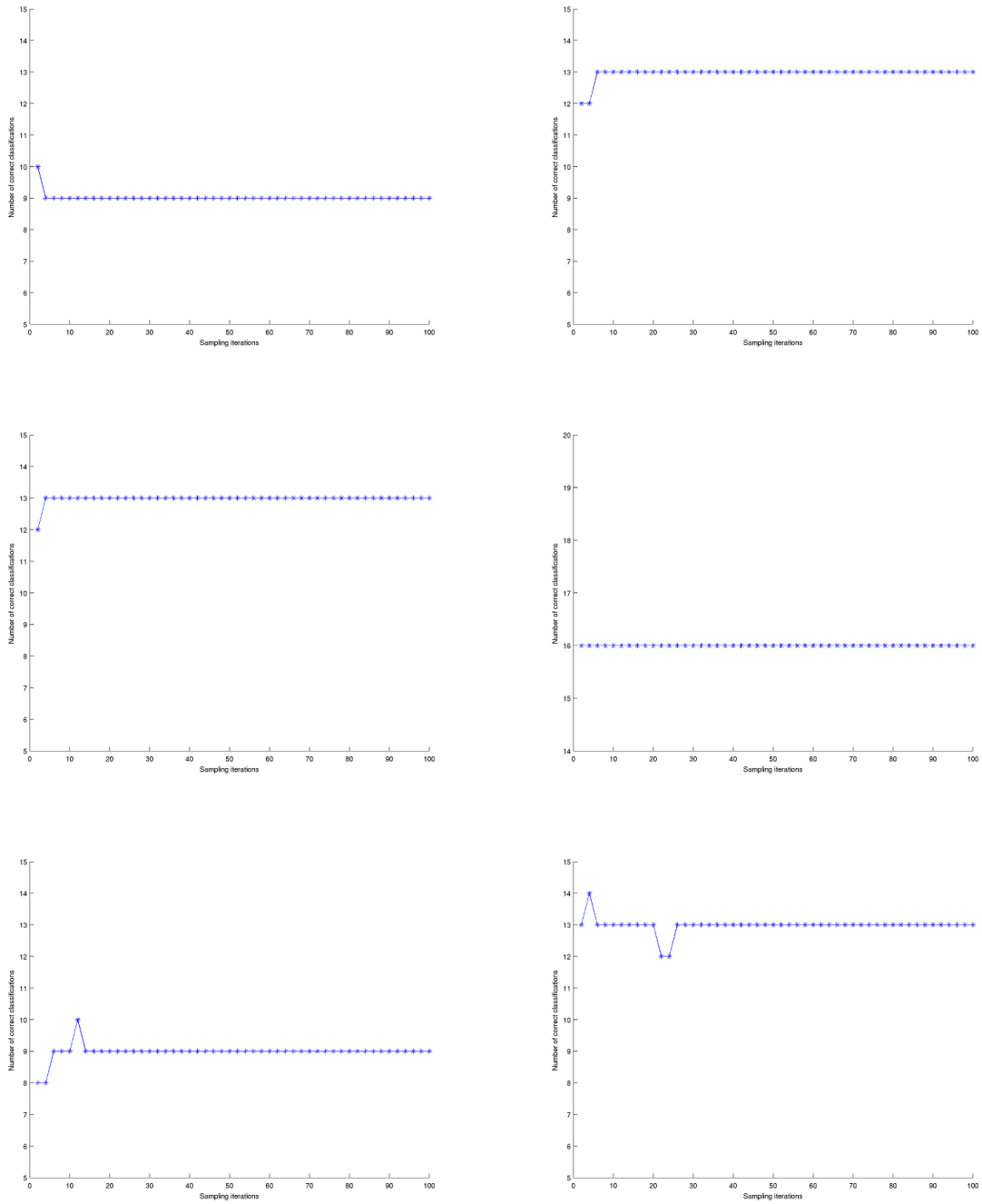



Figure 3.24: Success rates against the sampling iterations

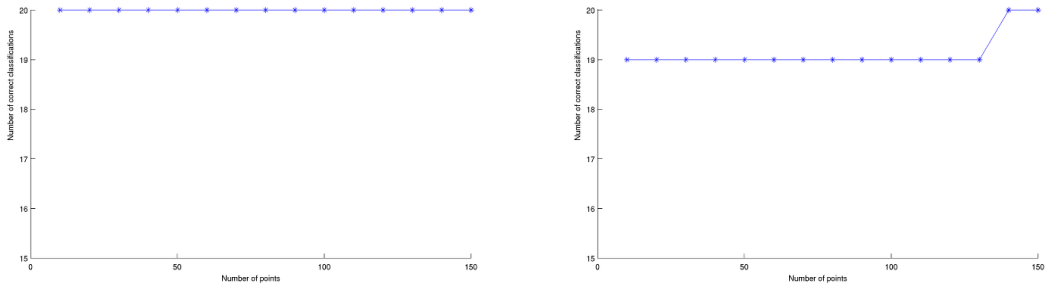



Figure 3.25: Success rate against the number of points

### 3.3.2 Classification results

To classify the observed data shapes, we evaluate and approximate the posterior probability for each of the classes via the proposed algorithm in chapter [2]. The observed data shapes are then classified to the class that assigns the highest posterior probability to it. Since the data shapes $\boldsymbol{y}$ have been generated by known classes it is easy to evaluate how the algorithm performs by comparing the estimated classes to the true classes.

As in the case of the Kimia database, for the following experiment we simulated 10 different shapes coming from the same class for each of the 10 runs. We then approximated the posterior $\mathbb{P}(C_i|y)$ and picked the highest posterior which effectively gave the class in which the shapes were classified. We can then evaluate the performance of the algorithm since the shapes are generated from a known class. The following 10 shapes were generated from the letter "B". The shapes were generated with 30 points and the noise's standard deviation was $\sigma = 0.4 \times 10^{-2}$. The sampling iterations were fixed to be 20, the minimum number that is needed for the confidence results to stabilise. The first graph presents the 10 simulated shapes to be superimposed whereas the second one shows one example of such a shape. For simplicity, we present 4 out of 10 classification results. The success rate for these 10 runs was 70 percent with 7 out of 10 shapes being classified into their respective category correctly. For the correctly classified shapes the confidence levels ranged

between 38 to 68 percent with the average confidence level at 52 percent. In the misclassification cases, the letter B is classified either as a D or an O. The data shapes are created with 30 points in the presence of noise so this is an expected behaviour since these classes are very similar. One notices that even in the cases that the data shapes were classified correctly, the two higher posteriors after class B belong to class D and class O.

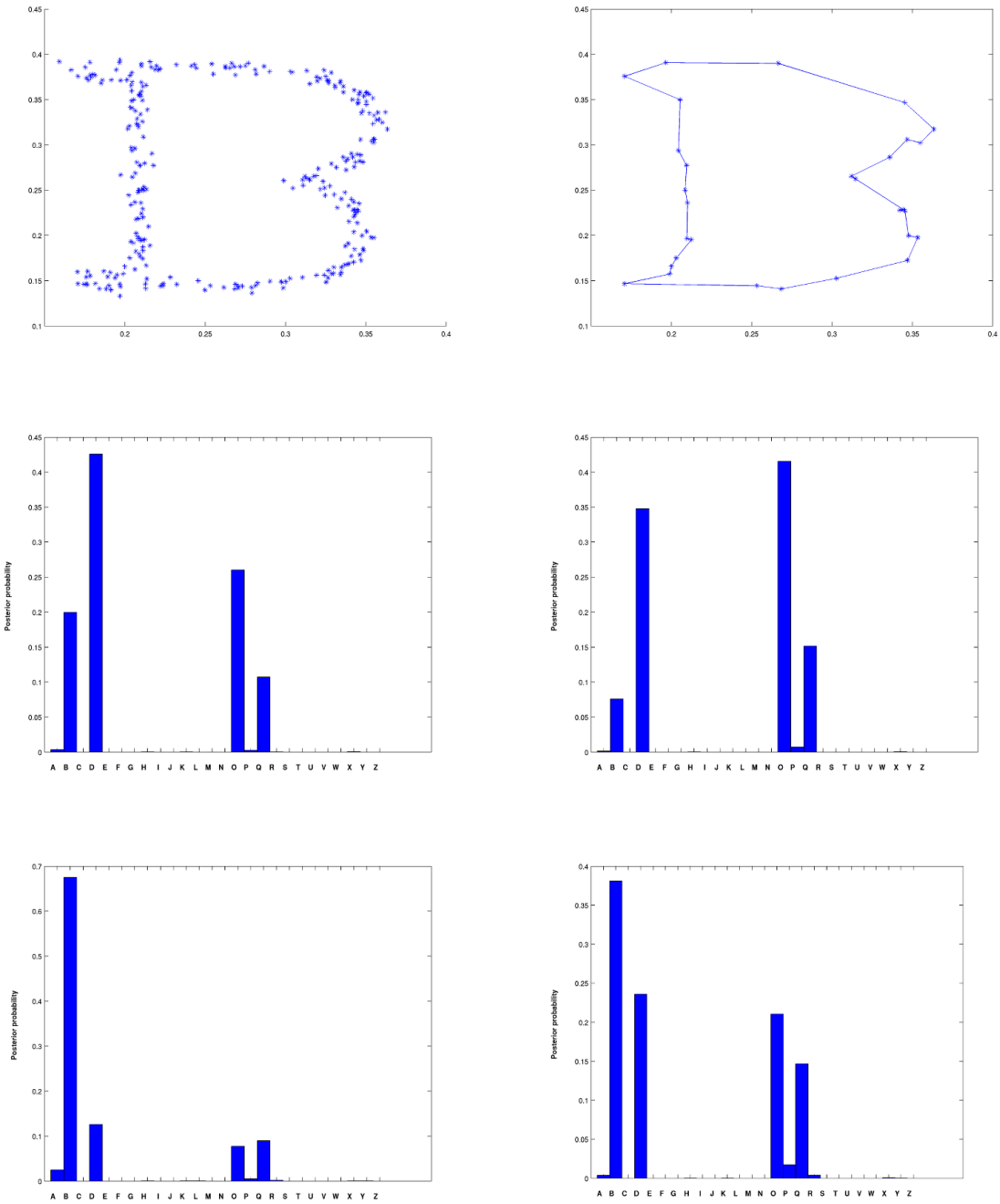



Figure 3.26: Classification results for the letter B

The next 10 shapes were simulated for letter E. For their simulation we used 50 points and the standard deviation of the observed noise was $\sigma = 0.7 \times 10^{-2}$ which is considered to be relatively high. The success rate for this run was 90 percent with 9 out of 10 shapes classified correctly.The classification levels for the letter E ranged

from 60 to 100 percent with an average classification level of 87 percent. One should mention that the high confidence levels are mainly due to the fact that letter E is quite distinguishable in comparison to other classes of letters.

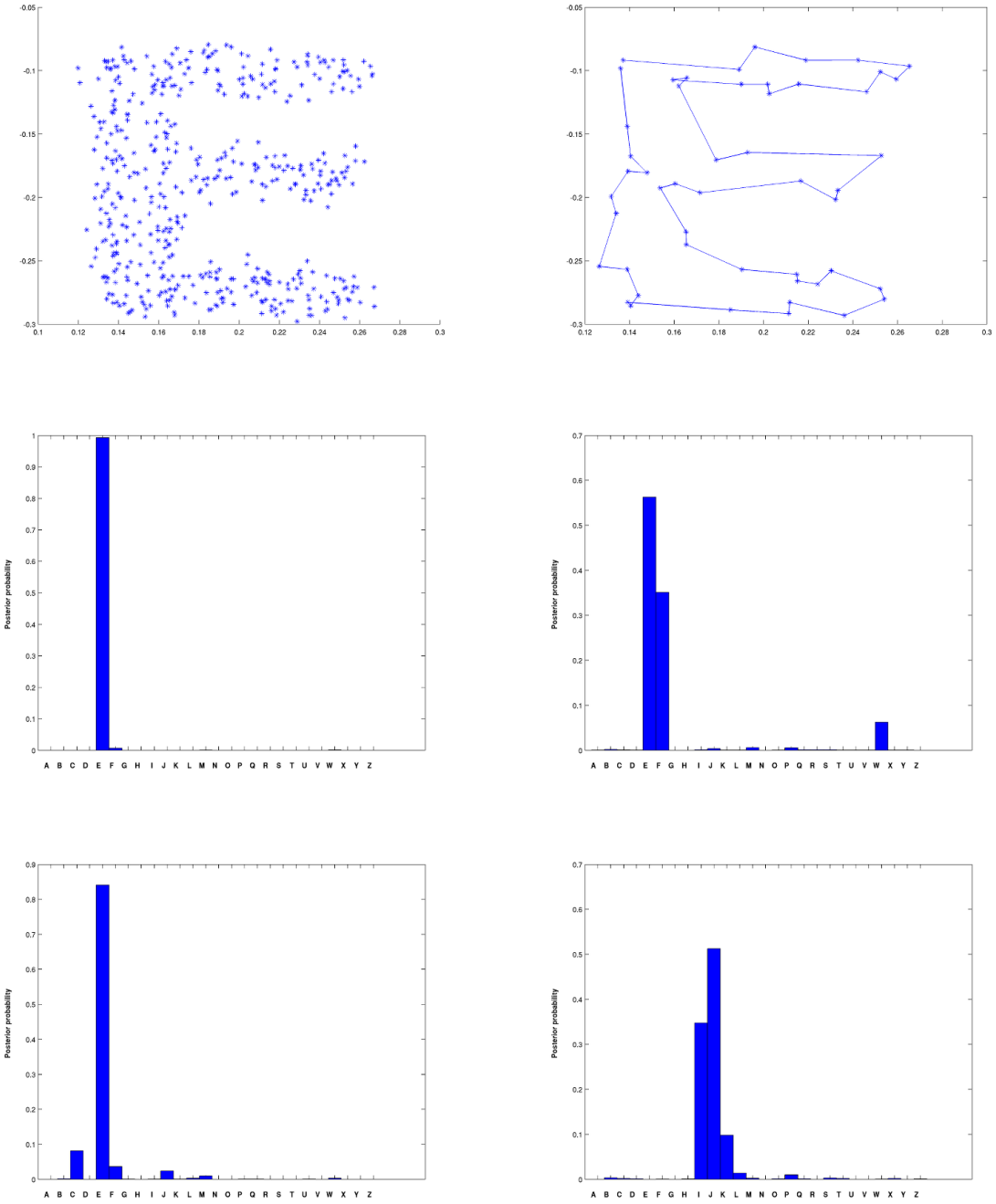



Figure 3.27: Classification results for the letter E

The next 10 shapes were simulated for the letter G. For their simulation we used

50 points and the standard deviation of the observed noise was $\sigma = 0.6 \times 10^{-2}$ which is lower than the noise used in the case of letter E of the previous run. Although this run was done with the same number of points as in the previous run and in the presence of lower noise the success rate was 70 percent. In the misclassification cases, 2 shapes were classified as a C and one as an F. In the case where letter G was correctly classified, the next higher posterior was for letter C which shows that the two classes of letters are quite similar and not easily distinguishable. The lowest classification level for letter G was 55 percent whilst the highest was 98 percent with the average confidence level to be 74 percent.

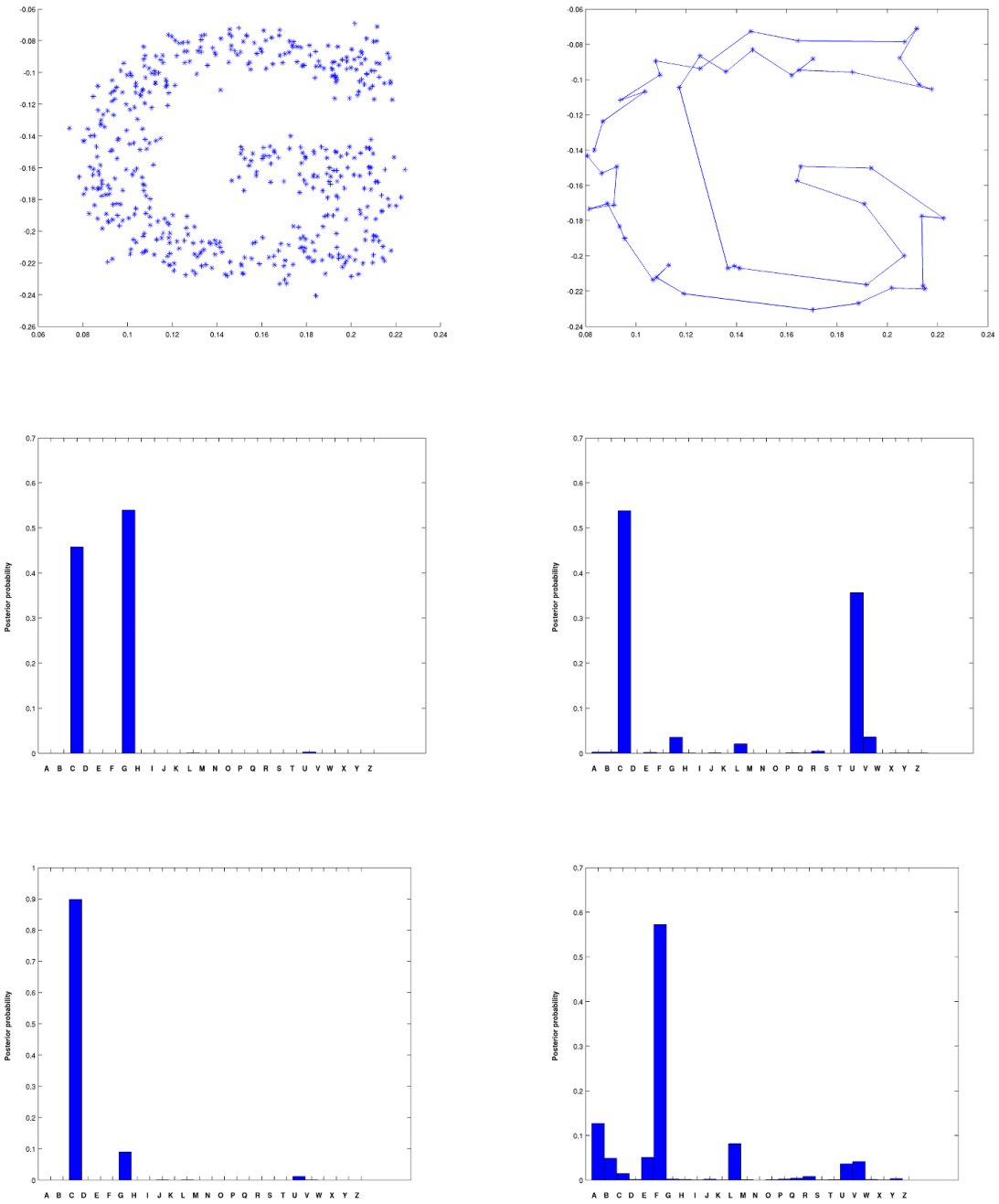



Figure 3.28: Classification results for the letter G

The next 10 shapes were simulated from the letter Q. For their simulation we used 40 points and the standard deviation of the observed noise was $\sigma = 0.6 \times 10^{-2}$. The success rate was 40 percent since letter Q is very similar to letter O or letter D. In all misclassification cases letter Q was classified either as an O or a D. In the case

of correct classifications, the confidence level ranges from 30 to 88 percent with the average confidence being 55 percent which makes letter Q in the threshold of being distinguished from other letters.

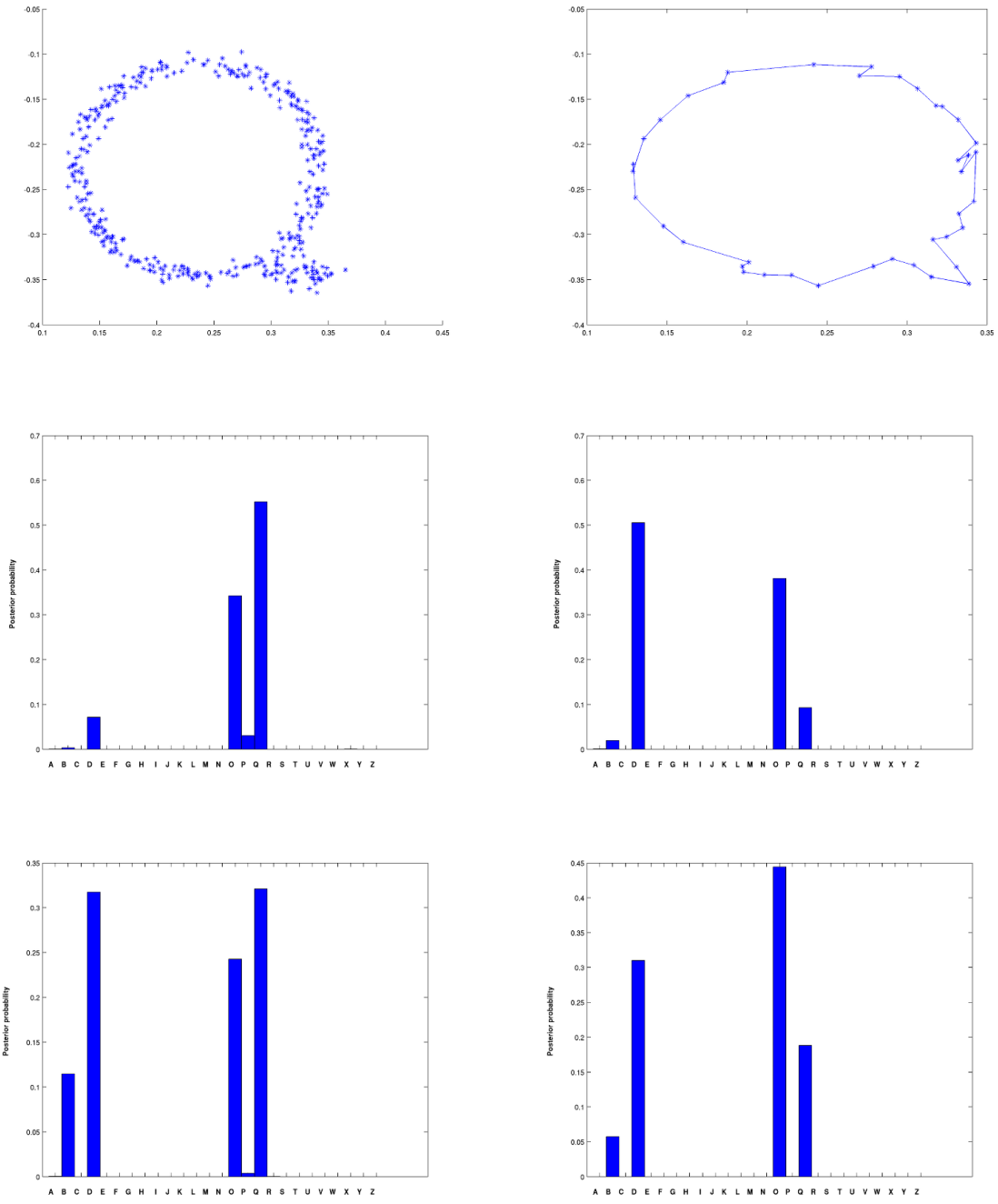



Figure 3.29: Classification results for the letter Q

The next 10 shapes were simulated from the letter T. For their simulation we

used 40 points and the standard deviation of the observed noise was $\sigma = 0.5 \times 10^{-2}$. Although in this run we used the same number of points and lower noise than for letter Q the success rate was 60 percent. The confidence level ranged from 43 to 98 percent with the average being 81 percent. The choice of a more distinguishable class of letter increases the confidence for about 30 percent.

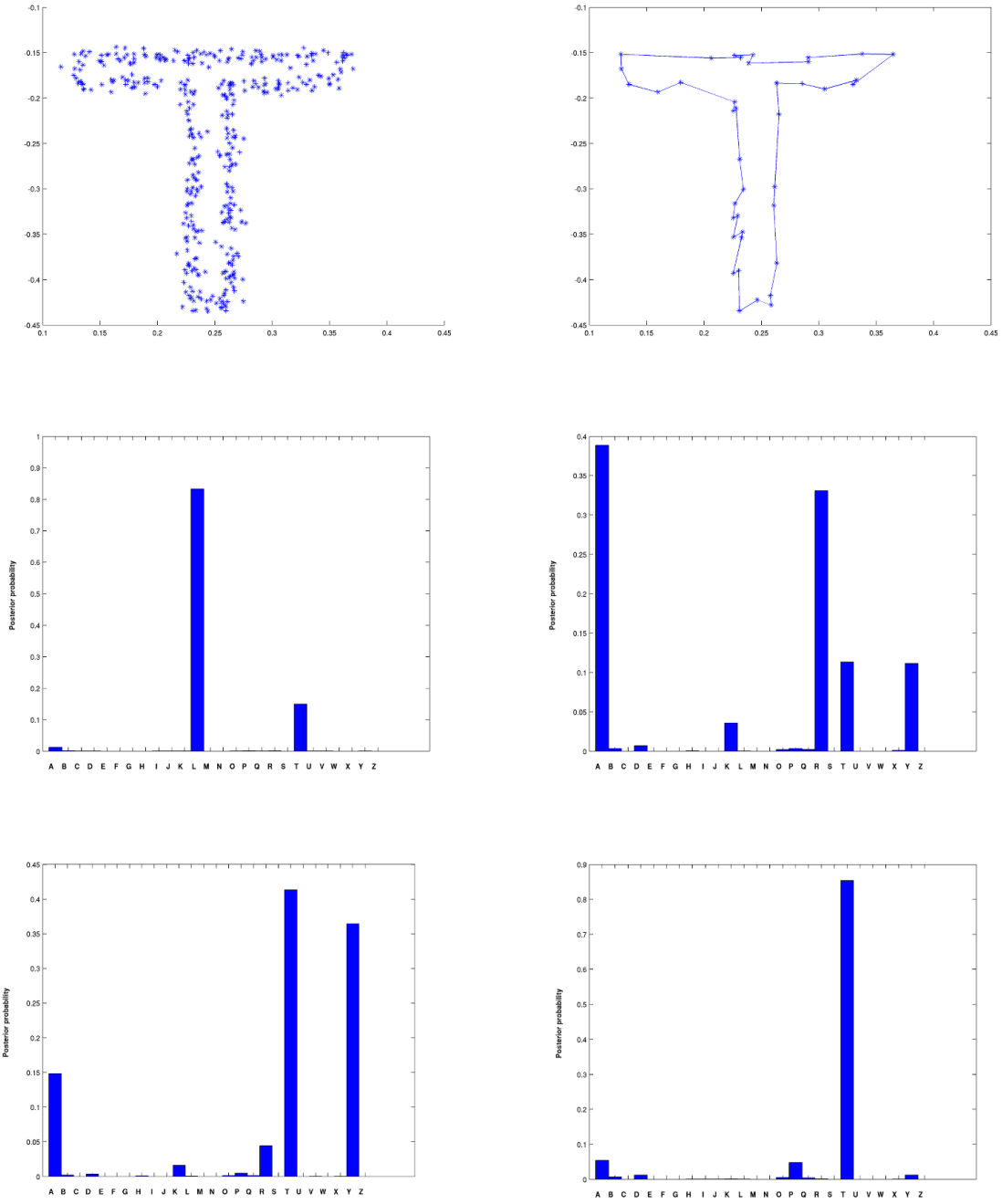



Figure 3.30: Classification results for the letter T

The last 10 shapes were generated for letter Y. For the simulation of the data shapes we used 80 points and the standard deviation of the noise was $\sigma = 0.8 \times 10^{-2}$ which is considered to be extremely high. There were 5 out of the 10 shapes classified correctly and the confidence results were in all cases more than 90 percent with the average confidence reaching 99 percent. In the misclassification case, the data shape was classified as an A or a V. Although the success rate is only 50 percent, we see that in the case of correct classifications the confidence levels is extremely high something that happens due to the fact that the number of points is relatively high.

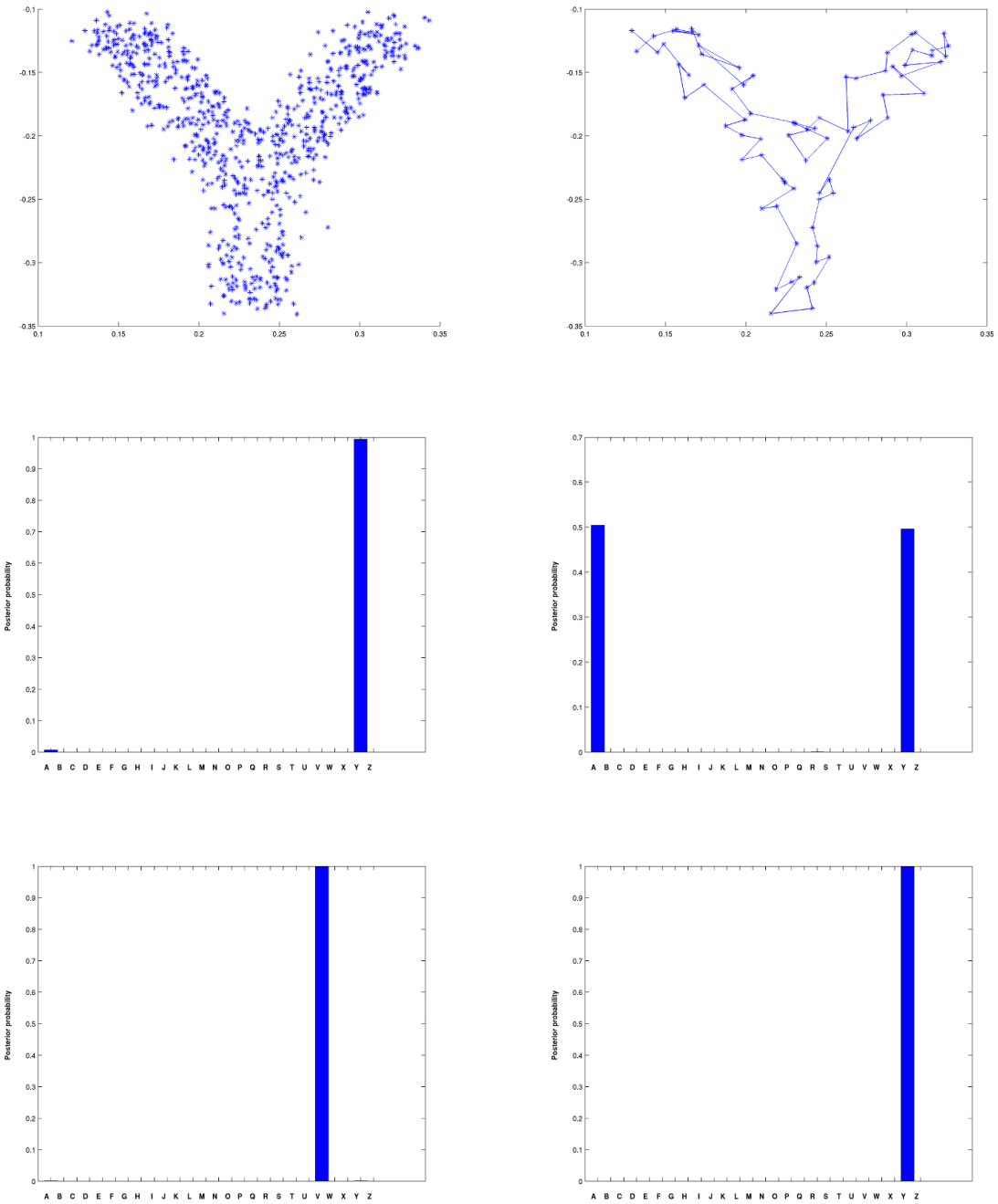
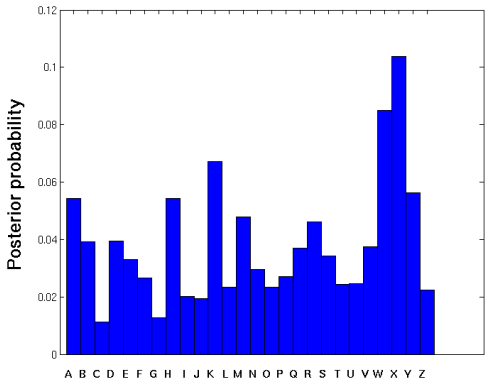



Figure 3.31: Classification results for the letter Y

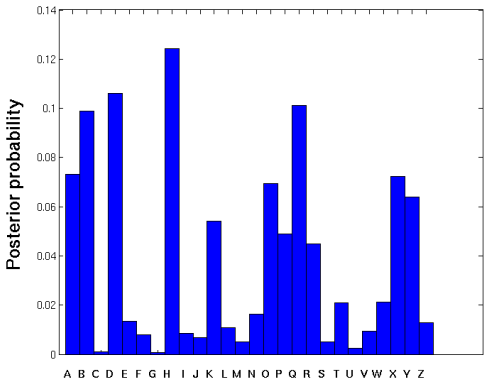
To investigate the classification levels for different numbers of points we classified data shapes from the same class and with stable noise as the number of points increased. The experiment was run 10 times with the number of points ranging from 10 to 100 and the noise being $\sigma = 0.2 \times 10^{-2}$. The following 5 graphs present

the results of 5 runs for letter B. One notices that in the case of 10, 20 and 30 points the data shapes are misclassified whereas for 40 points the classification level is 45 percent. For more than 50 points the confidence reaches more than 90 percent. The last graph presents the time in hours for a single run as a function of points which shows that for a large number of points the algorithm is computationally expensive.

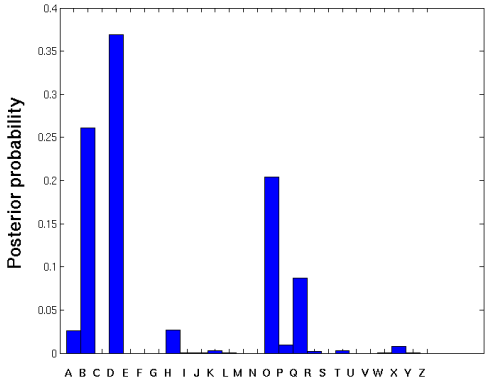
The next 5 graphs present the results of 5 runs for letter E. This experiment was run 10 times with the number of points ranging from 10 to 100 and the noise being $\sigma = 0.2 \times 10^{-2}$. One notices that in the case of 10 to 40 points the data shapes are misclassified whereas for 50 points the classification level is more than 90 percent. The last graph presents the time in hours for a single run as a function of points which shows a similar behaviour as in the case of letter B.



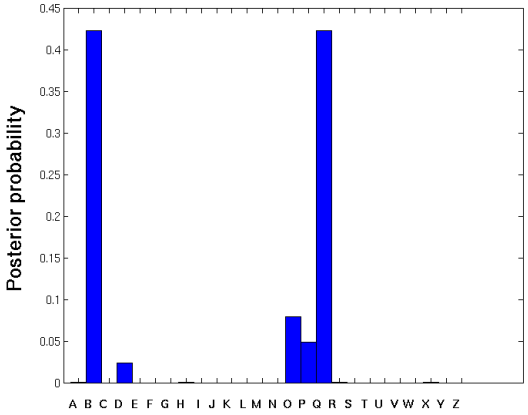

(a) Classification results for 10 points



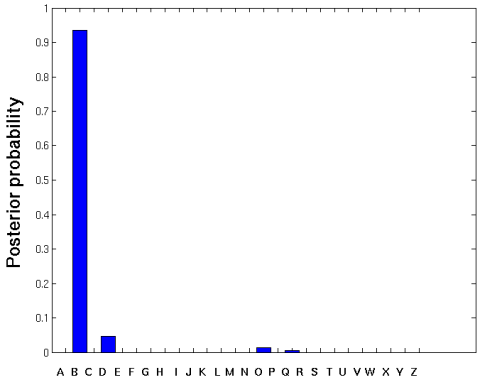

(b) Classification results for 20 points



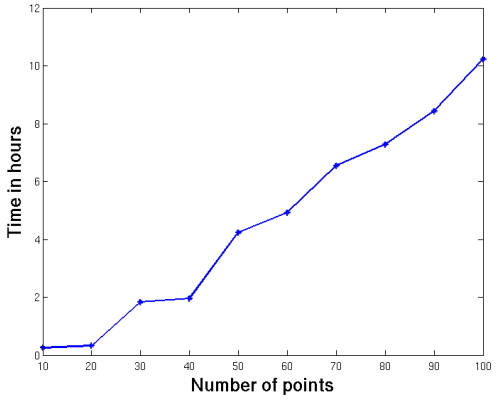

(c) Classification results for 30 points




(d) Classification results for 40 points

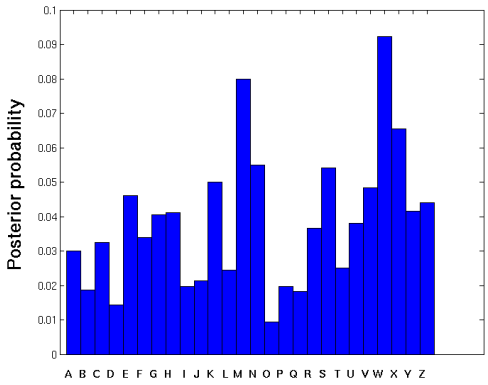



(e) Classification results for 50 points

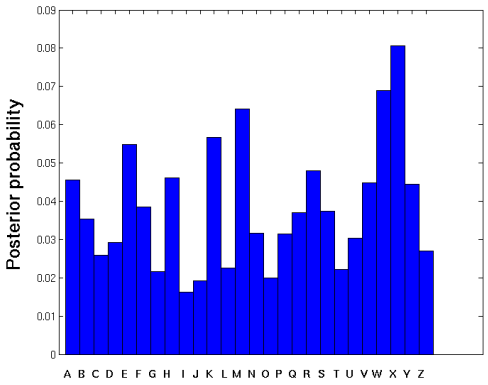



(f) Computational time as a function of the number of points

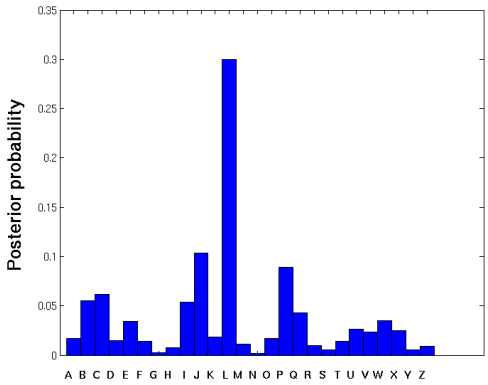
Figure 3.32: Classification results for letter B



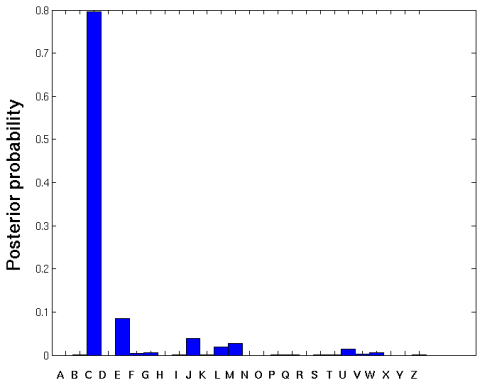

(a) Classification results for 10 points



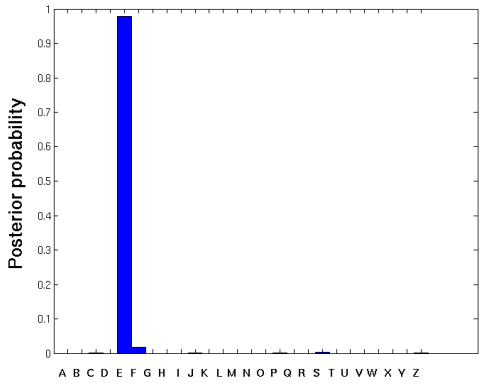

(b) Classification results for 20 points



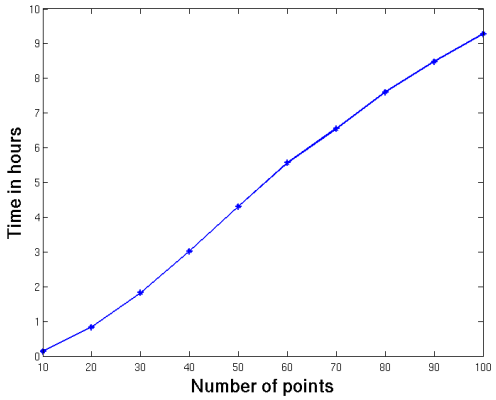

(c) Classification results for 30 points




(d) Classification results for 40 points




(e) Classification results for 50 points




(f) Computational time as a function of the number of points

Figure 3.33: Classification results for letter E

Overall, we performed 10 runs for 10 shapes each. These 100 shapes came from

the classes of letters B, T, G, P, Q, J, I, E, D, Y. For the 10 runs, the average success rate was $\hat{\mu}$ = 73% ± 6% and the average classification level was 77% ± 5%. That means that on average 7 out of shapes were classified correctly with an average classification confidence of 77%. We can see that in the alphabet database, the algorithm returned slightly better classification confidence levels than the Kimia database but the success rates were comparable. The algorithm is sensitive in the presence of too few points or too high noise however it seems that a high number of points can't compensate the high noise as in the case of the Kimia database. It is worth mentioning that the number of classes is higher as well as the number of classes that are similar to each other. It is also noticeable that when the success rate is not that high the classification levels for the correct classifications are usually more than 80 percent especially when the number of points increases to more than 50. However, when the number of points becomes more than 40 the algorithm becomes computationally expensive with the computational time being more than 5 hours.

## 3.4 Geological sand bodies

In this section we describe the motivation behind the choice of the sand body database which we created ourselves. We describe the geological definitions needed for the study of the database and we discuss current classification schemes that are inadequate to capture the variation of sand body classes. We then propose the statistical classification of sand bodies and examine the experimental results for this database.

### 3.4.1 Definition of a sand body and geological classification

According to the Geology dictionary [139] a palaeocurrent is a current which existed during the deposition of a sediment at some period of geological history. A paleochannel is a subterranean remnant of an inactive river or paleocurrent or stream channel that has been filled or buried by younger sediment. Other terms used to describe such paleocurrents or paleochannels is sand bodies or sandstone bodies. The sandy nature of sand bodies makes them very porous which in turn makes them

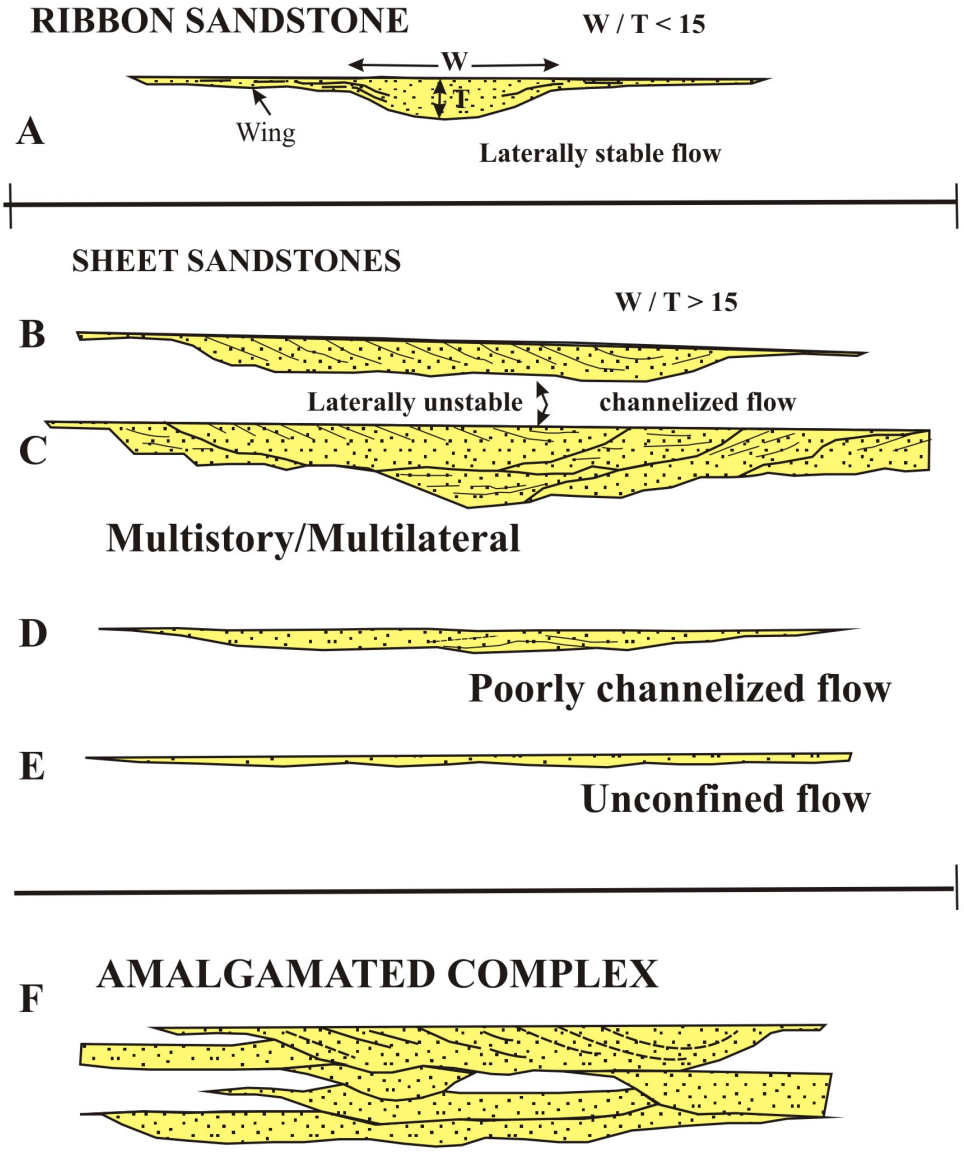
the best oil and hydrocarbon reservoirs. For this reason, sand bodies have become extremely important for both geology and the petroleum industry; in particular, their cross-sectional shapes and their morphology help determine their oil-bearing capacity and porosity.

A different definition of sand bodies comes from Potter [140]. Potter in his review states that "it is not possible to define rigorously a sand body but one might define a sand body as a single, interconnected mappable body of sand." The term interconnected is used to take into account of the branching patterns of many sand bodies and the superposition of sand bodies of different cycles. The term mappable is used to distinguish them from most single beds[1] [141]. Classification of sand bodies' cross-sectional shapes is an important problem to study, however current classification schemes for sand body shapes are qualitative, simplistic, and ad hoc. Thus, there is a need for a quantitative analysis with the help of statistical models. Roughly speaking there are two classes of sand bodies, ribbons and sheets. Figure (3.35) shows the two categories of shapes as the geologists distinguish them. In the section to follow we discuss how geologists established this classification scheme based on experimental observations.

For the classification of the sand bodies we need to define the paleoflow. Paleoflow, paleocurrent direction or paleodirection is the direction of flow of the water or wind at the time the rocks were deposited as sediments [142]. Sedimentologists can deduce this flow direction from sedimentary structures on the rocks such as their ripple marks. Geologists need to know the paleoflow direction because it is one of the parameters they use to define the shape of a sand body. It also enables them to classify them and helps them to understand the environment of the sand bodies' deposition. Another parameter that helps geologists with the classification of sand bodies is their dip angle. The dip angle is the angle a sand body makes with the plane of the horizon. One can think of it as the inclination of a sand body towards the centre of the earth. These two parameters define the plane of projection for a

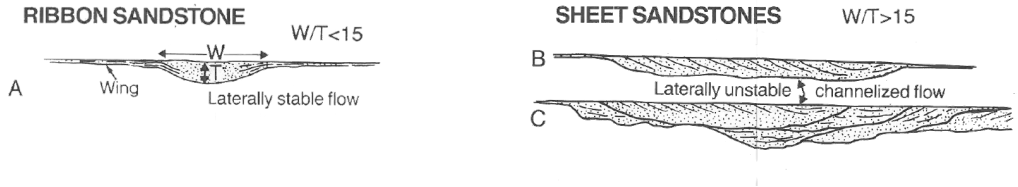
[1] A bed is a layer that is distinctly separated from other layers. It is the smallest division of a geological formation.

three-dimensional sand body to its equivalent planar. This cross section of the sand body can then be used for classification purposes. We will describe these in detail in the next section.




Range of distinctive sandstone-body geometries recognized in the ASF (terminology as summarized in Hirst, 1983 and Friend and others, 1986). A. Ribbon sandstone resulting from the plugging of a channel scour prior to any significant channel migration. Width / thickness ratios (W/T) measured perpendicular to paleoflow are commonly between 5 and 10. The wings attached to each side of the central body are coarse overbank deposits (levees). B to E Sheet sandstone bodies deposited by i. laterally unstable channels (B and C), ii. poorly channelized flow (D), and iii. Unconfined overbank flow (E). Single sweeps of laterally unstable channels produce simple units(B) whereas migration of the channel back and forth results in more complex multistory/multilateral units (C). Unconfined overbank flow deposited units with very high W/T ratios and range from simple parallel-bedded tabular bodies to amalgamations of numerous small scour forms. F. An amalgamated complex of closely spaced sheet-and ribbon-sandstone bodies.

Figure 3.34: Types of sandbodies




(a) A ribbon (b) A sheet

Figure 3.35: The two classes of sandbodies [143]

Early classification attempts were done by Rich [144] who focused on the length and the width of the sand bodies as a combination of longitudinal and cross sectional measures [141]. Potter [145] states that the terms used by Rich [144] are a mixture of the descriptive and the genetic; the descriptive terms are usually geometric and the genetic ones are land form names. Following that, experimentalists have introduced the terms blanket and sheet for equidimensional sand bodies and have used descriptive names such as shoestring, pod and belt [144] for elongated sand bodies whose dimension is 2 to 100 times greater than the width.

Kryinine [146] recognized the utility of a new classification scheme, different to the one proposed by Rich and Potter, the width–to–thickness ratio (W/T) because the ratio was a good means of estimating the sand bodies' area–volume ratio. Using the W/T ratio Kryinine classified sand bodies into four categories. Although Kryinine's classification depended on the size experimentalists preferred to simply use the dimensions and thickness of the sand bodies. McGugan [147] modified this approach by introducing the persistence factor but according to Potter [140] the factor is useful in some studies but does not differentiate elongate and equant sand bodies of equal area and thickness so it hasn't been as widely used as the W/T. Collinson [148] classifies sand bodies according to their channel types: meandering, sinuous and others. Moody-Stuart [149] recognised only these two types of sand body shapes - sinuous and meandering. Friend [150] states that little is known about the process forming the two dimensional or three-dimensional geometry of a sand body and that channels that have lateral migration should be distinguished from those that are characterized by lateral stability. Friend refers to Potter [140]

using his definition of sand bodies and he finally classifies them into sheets and in the second type which is "of elongate form" [145]. Friend characterises the elongate ones ribbons, using a distinguishing value of the W/T of 15:1. As he states: When applying this distinction care must be taken that the width is estimated perpendicular to the local elongation of the sediment body. His work in the Ebro basin has demonstrated a major distinction between ribbons and sheets, formed by laterally stable channels and sheets formed by channels that migrate laterally (also referred in Allen [151]).

The division made by Friend has been generally accepted and accords with the aspect ratio of modern channels [141]. Friend's original choice of 15 as a W/T discriminator was based on channel–body dimensions in the Ebro basin coupled with information from Schumm [152] (P.Friend's written communication with Gibling in 2000 [141]). Friend also investigated the shape properties of sand bodies in the Ebro basin. In the Ebro basin, because of the lack of thick vegetation and soil cover, the bodies weather out and erode so that unusually complete geometrical information is available at outcrop. Atkinson [153] revises Friend's ratio to 25:1 and Nadon [154] revises the ribbon/sheet ratio to 30:1.

Hirst [143] classifies sand bodies after a study in the Huesca system observing that ribbon sand bodies formed when paleochannels became plugged with sediment prior to any lateral migration and have been defined as having W/T<15. His observations in Huesca indicate typical W/T values between 5 and 10. Sheet sand stones have W/T>15 and often W/T>100. In this paper Hirst states that W/T are measured perpendicular to the paleoflow. Ribbons become more prevalent distantly. Here, for the first time we have the clear definitions of the sand body classification and a clear distinction where he suggests the following categorisation.

The first established category is **ribbon**s: these are elongated in plan form and defined by the relatively small W/T of less than 15. When the width is measured we have to make sure not include the thin wings (levees) and to estimate the width perpendicular to the long axis of the ribbon often measured by the paleoflow indicators in the sand body.

The second established category is **sheets**: these sand bodies are defined as having greater W/T ratio than 15 where width is measured perpendicular to the paleoflow. The major difference between the two types: Ribbons are a result of a major episode of channel incision.

This classification scheme has been used in a qualitative way to classify the sand bodies as having risen by meandering (ribbons) or braided (sheets) river systems as observed by Bridge and Tye [155]. However, the most commonly used classification scheme even nowadays is the one one proposed by Hirst, where the discrimination between the two classes occurs at the W/T threshold of 15. Hirst established the existing classification scheme which classifies sand bodies into ribbons and sheets. Modern sedimentologists and geologists still use Friend's classification system. One can argue that although this method is based on observations of experimental geologists, is still quite simplistic and ad–hoc since it does not lend itself to a quantitative analysis. Our goal is to provide a more advanced, scientific and statistical classification scheme which can be based purely on the geometrical deformations and nature of the particular sand body shapes. We describe our methodology in the next sections.

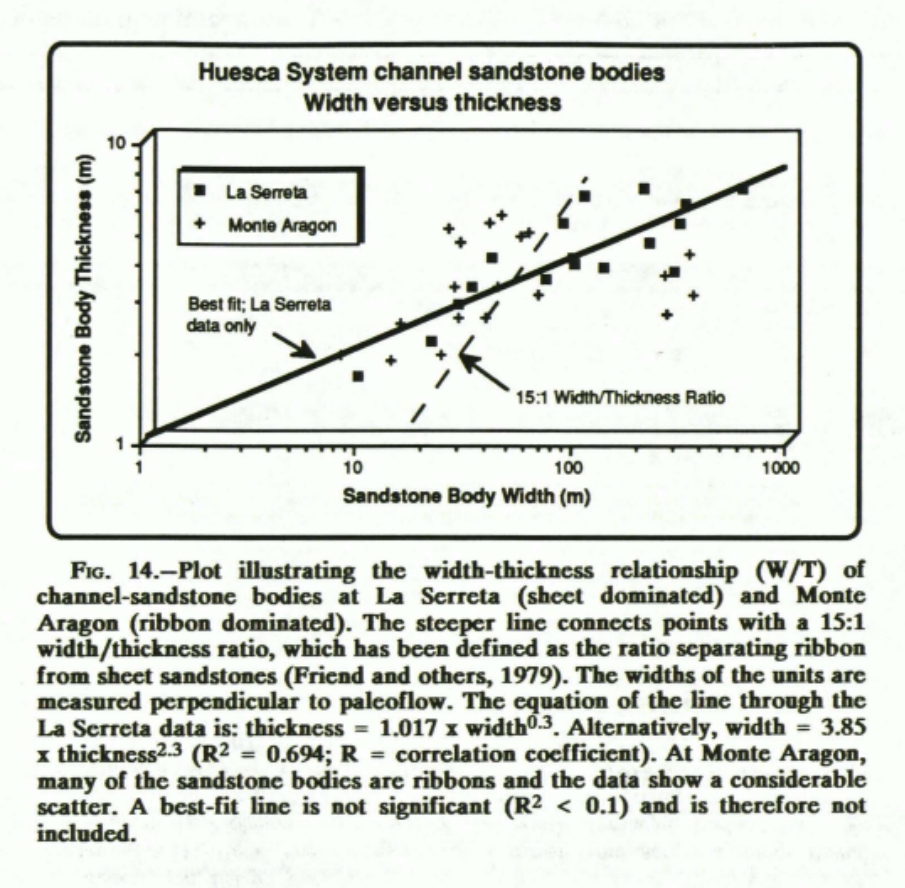



FIG. 14.—Plot illustrating the width-thickness relationship (W/T) of channel-sandstone bodies at La Serreta (sheet dominated) and Monte Aragon (ribbon dominated). The steeper line connects points with a 15:1 width/thickness ratio, which has been defined as the ratio separating ribbon from sheet sandstones (Friend and others, 1979). The widths of the units are measured perpendicular to paleoflow. The equation of the line through the La Serreta data is: thickness = 1.017 x width$^{0.3}$. Alternatively, width = 3.85 x thickness$^{2.3}$ ($R^2$ = 0.694; R = correlation coefficient). At Monte Aragon, many of the sandstone bodies are ribbons and the data show a considerable scatter. A best-fit line is not significant ($R^2$ < 0.1) and is therefore not included.

Figure 3.36: W/T experimental observations by J.P.P Hirst [143]

### 3.4.2 Geological extraction of sand body shapes

Terrestrial laser scanning (also known as ground based lidar) is a remote sensing technique that can be used to acquire a point cloud in three-dimensional space. The scanner emits a laser pulse, which is reflected back from a surface, in this case the geological outcrop. The length of time taken for the reflected pulse to reach the scanner is then used to calculate the distance between the scanner and the surface and to produce a point cloud. A camera mounted on top of the scanner is used to take colour images of the scanned location, allowing the point cloud to be coloured realistically. In addition, GPS data is acquired to enable geo-referencing of the scanned point cloud. These scans can be combined into a single coordinate system using common reflector points seen in several scans. The point cloud is then coloured using the photographs corresponding to each scan, making it easier to pick

out sedimentological characteristics and sand body geometries from the dataset.

To extract the sand body geometries, three-dimensional polylines (i.e. the three-dimensional boundary line) are drawn manually around the sand body areas, which can be seen in the laser scan point cloud in figure (3.37). The sand bodies are identified using a combination of photographs and graphic logs of the section, along with the colour and 3D shape of the point cloud. In some cases it is difficult to identify the edge of the sand body due to the presence of vegetation or shadow. In the case of vegetation, the laser cannot hit the sand body because it is occluded by bushes or trees. In the case of shadow, a section of the point cloud contains no points because the laser is not able to reach around the back of large sand bodies, some of which protrude several metres into the air from the hillside. Where the edge of the sand body is uncertain, several different sand body shapes can be picked in order to capture the range of possible channel geometries. The shape extraction stage thus produces a chain of 3D points for each sand body, representing a curve around the sand body boundary.

In order to characterize the variability in sand body shape in more detail than the W/T ratio, it is necessary to obtain two-dimensional shape boundaries corresponding to these cross-sections from the three-dimensional data. To do this, each three-dimensional chain of points is projected onto the plane perpendicular to the measured paleocurrent direction and defined by the dip angle. Where measured paleocurrent data is not available, paleocurrent directions can be estimated. The projection procedure produces meaningful boundaries, together with uncertainty measures, from which shape properties can be computed.

In this work, we will classify sand bodies based on an analysis of how "similar" two shapes are. This formalism can be used to carry out a full statistical analysis, avoiding the loss of information inherent in choosing a few special shape properties such as the W/T ratio. By building statistical models of sand body shape, we can study the differences of clusters ribbons and sheets, refine these descriptions, and study the links between sand body shape and geological properties in a rigorous way. Due to the fact that the expected geological data were never available to us,

we simulated a sand body database based on the information we had.

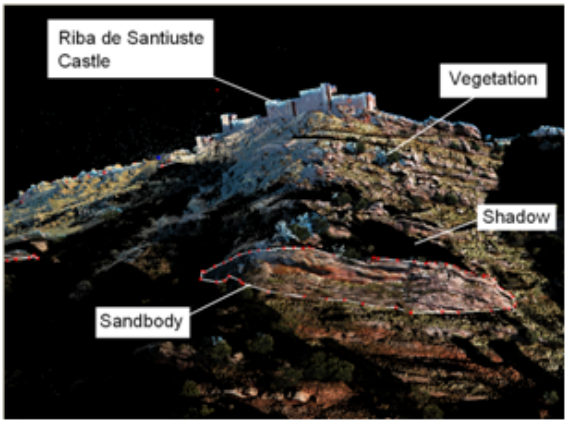



Figure 3.37: Extraction of sand body shapes

### 3.4.3 Statistical classification of sand bodies

The three-dimensional geometry of fluvial channel sand bodies has received considerably less attention than their internal sedimentological structure, despite the inherent importance of sandstone body geometry for subsurface reservoir modelling. The aspect ratio (width/thickness) of fluvial channels is widely used to characterise the geometry of channel sand bodies, with end members of "ribbon" and "sheet" sands. However, these approaches do not typically provide a full characterization of fluvial sand body shape, as a single W/T allows many different channel geometries. Furthermore, using the W/T ratio still requires choosing a classification boundary between "ribbon-like" and "sheet-like", and there can be significant overlap between these values [141]. Over- or under-estimating the cross-sectional area of a sand body can have significant implications for reservoir models and hydrocarbon volume predictions. There is thus a clear need for versatile, quantitative, statistics-based models of sand body shape. The aim of this study is to demonstrate how a new, statistics-based approach provides quantitative data for constraining stochastic fluvial reservoir models.

In order to describe the statistical classification of sand bodies, we need to have a mathematical model for each of the classes. Figure (3.35) shows the two existing classes of sand bodies as reported by [143]. In absence of any geological data however,

we will have to assume that the three-dimensional point cloud, its equivalent three-dimensional polyline (as in figure 3.37) and the corresponding palleocurrent direction and dip angle for each sand body are available to us. In this way, we assume that we know the true projection plane of each sand body and we can thus extract the cross-section of all sand body shapes. In the next section we describe the shape models for ribbons and sheets.

### Shape models for sheets

As described by Hirst [143], sheets are usually elongated objects with W/T ratio of more than 15. For this reason, we model the idealised sample sheet shapes as rectangles. Since sand bodies are characterised by their W/T ratio, we propose to model sheet curves by their aspect ratio $\gamma$ which can be drawn from a $\Gamma(\kappa, \theta)$ distribution since such values will always be positive. From experimental results we know that sheets have W/T ratio bigger than 15. To reflect these values in our analysis, for the moment we assume that aspect ratios are generated by $\gamma \sim \Gamma(50, 0.4)$ so that we can capture the variability of the aspect ratios with a mean W/T to be $\mu = \kappa \cdot \theta = 20$. To avoid generating zeroes, the generated aspect ratios are shifted by 1. We choose the generated shapes to be centred at zero and have unitary length. Figure (3.39b) shows such an idealised sample sheet and figure (3.38) shows the probability density function of the Gamma distribution that the sheets' aspect ratios are generated from.

### Shape models for ribbons

Ribbons are very similar to sheet shapes and we choose to model them as elongated objects with a triangular bump in the middle. To generate such a shape, we assume that the rectangular part's W/T ratio is generated in a similar way as the sheets' W/T. We assume that the height of the triangular bump is placed uniformly between $U(0.25, 0.75)$ of the total height. From experimental results we know that ribbons have W/T ratio smaller than 15. To capture this in the analysis, for the moment we generate ribbons' aspect ratios from a $\gamma \sim \Gamma(7.5, 1)$ with mean $\mu = \kappa \cdot \theta = 7.5$. To avoid generating zeroes, the generated aspect ratios are shifted by 1 and we choose

the generated shapes to be centred at zero and have unitary length. Figure (3.39a) shows an idealised sample ribbon and figure (3.38) shows the Gamma distribution that generates the ribbons' aspect ratios.

To summarise, we model idealised sand body shapes with their aspect ratio coming from two different $\Gamma$ distributions; in this way, the sand body curve $\beta$ is specified uniquely by its aspect ratio $\gamma$. This is reflected in the computational calculation of the likelihood as described by equation (3.2.1) where we implicitly take $\mathscr{D}\beta\ \mathbb{P}(\beta) \to d\gamma\ \Gamma(\gamma;\kappa,\theta)$. The explicit calculation of the integral over the curves $\beta$ is replaced by an implicit Monte Carlo integral over all aspect ratios that specify sand body curves and are being drawn by a $\Gamma$ distribution with hyperparameters $\kappa$ and $\theta$.

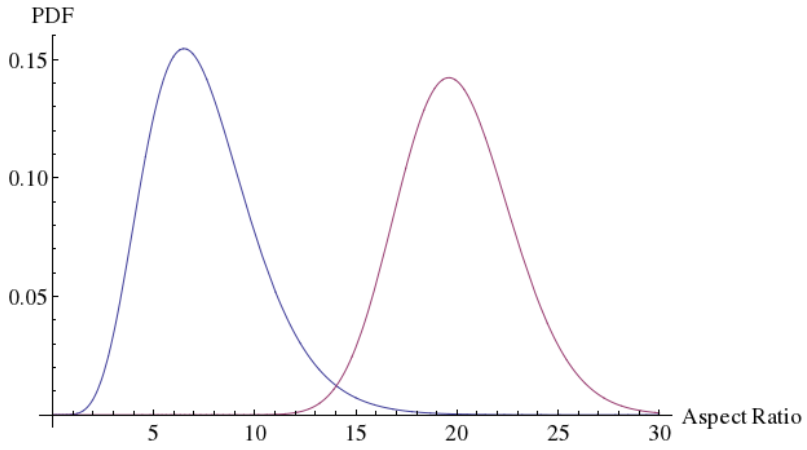



Figure 3.38: $\Gamma(7.5, 1)$ and $\Gamma(50, 0.4)$ the distributions ribbons' and sheets' aspect ratios

(a) An idealised ribbon (b) An idealised sheet

Figure 3.39: Simulated sand bodies

### Generation of sand body data set

In absence of real geological data the sand body data set was simulated. For the generation of simulated sand bodies we use the idealised sheets and ribbons (generated in the way described in the previous section) as the underlying curves which we randomly and uniformly choose for the construction of the data set. We choose a random number of points to assign around the boundary of the generated shape and assign a random rotation, translation and scaling to transform the shape. Finally, we add isotropic Gaussian noise to each of the points that perturbs them from their original place. Figure (3.40) shows examples of such simulated sand bodies with the idealised equivalent ones superimposed. With a complete sand body data set, we can now investigate the confidence levels and the success results of the proposed algorithm. We can then proceed to the statistical classification results of the algorithm for given sand body data shapes and evaluate its effectiveness.

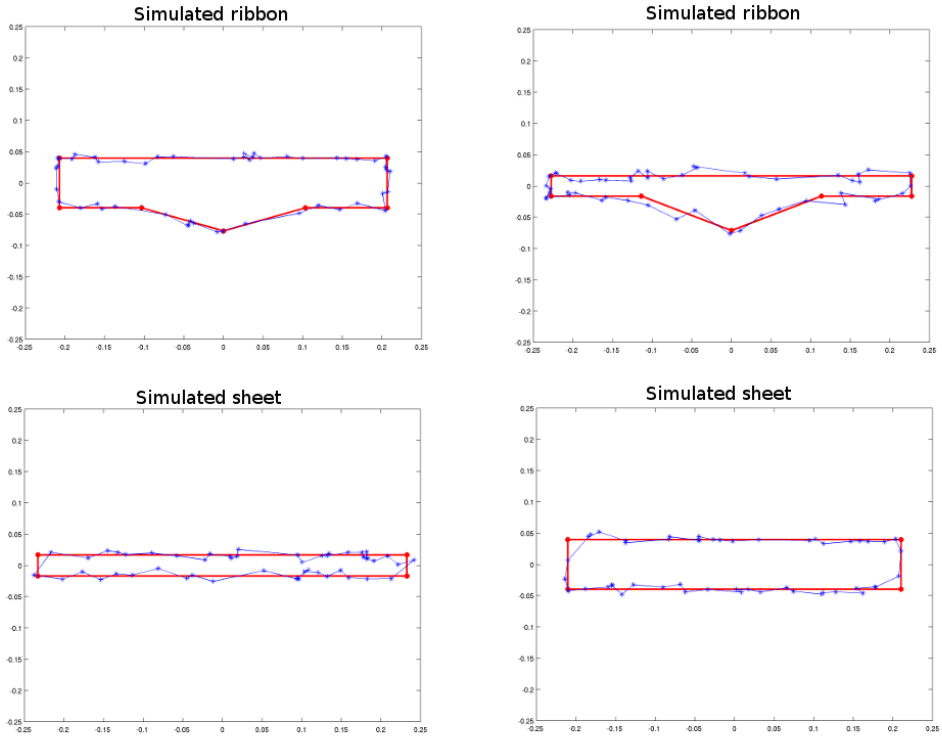



Figure 3.40: Examples of sampled sandbodies

### 3.4.4 Confidence and success results

In this section we investigate the properties of the classification algorithm as in the case of the Kimia and the alphabet database. We examine the confidence levels and the success rates of the algorithm as we vary the parameters that can affect the accuracy of the classification results of the algorithm for the different values of the samplings $s$ and the iterations over the curves $\beta$.

#### Confidence levels for Monte Carlo iterations of samplings

In this section we present the confidence levels of the algorithm and how much they vary as the iterations over the sampling integration increase. To produce the graphs of the Monte Carlo iterations we used the following process. We simulated a data shape whose underlying shape class was picked randomly with equal probability. It was created by the method explained in section [3.4.3] with 30 landmarks around its boundary. The added isotropic Gaussian noise was kept relatively low at $\sigma = 0.2 \times 10^{-2}$. We then performed classification by using our proposed algorithm whilst varying the number of the sampling iterations and keeping other parameters constant. The values of the regulators and the noise of the generalised Gaussian used for the diffeomorphisms were kept the same as for the previous databases. The idealised example shapes for the integration over the curves were generated with aspect ratios generated by the Gamma distributions discussed in section [3.4.3] and [3.4.3]. However, unlike the two previous studied databases, the number of all possible sand body curves is not discrete and hence we cannot sum over all class curves since all curves are generated implicitly by a Gamma distribution. Sand body curves are described by their aspect ratio so in this case we implicitly take $\mathscr{D}\beta\ \mathbb{P}(\beta) \rightarrow d\gamma\ \Gamma(\gamma;\kappa,\theta)$ replacing the explicit integration by an implicit Monte Carlo integration. For the confidence levels against the Monte Carlo iterations of the saplings, the Monte Carlo iterations of the curves were chosen to be 250. For the following results the Monte Carlo iterations of the sampling were increased to 1000 for 20 different runs (simulations of the data shape $y$). The data shapes $y$ coming from the class of sheets were generated in the way described in section [3.4.3] with their aspect ratio drawn from a Gamma distribution $\Gamma(49, 0.3)$ i.e. close to the

anticipated values.

The following graphs provide the confidence levels for 6 out of the 20 different runs of the algorithm when the data shapes come from the class of sheets. One notices that the confidence level is stabilised for a threshold $\epsilon = 0.02$ after 20 iterations.

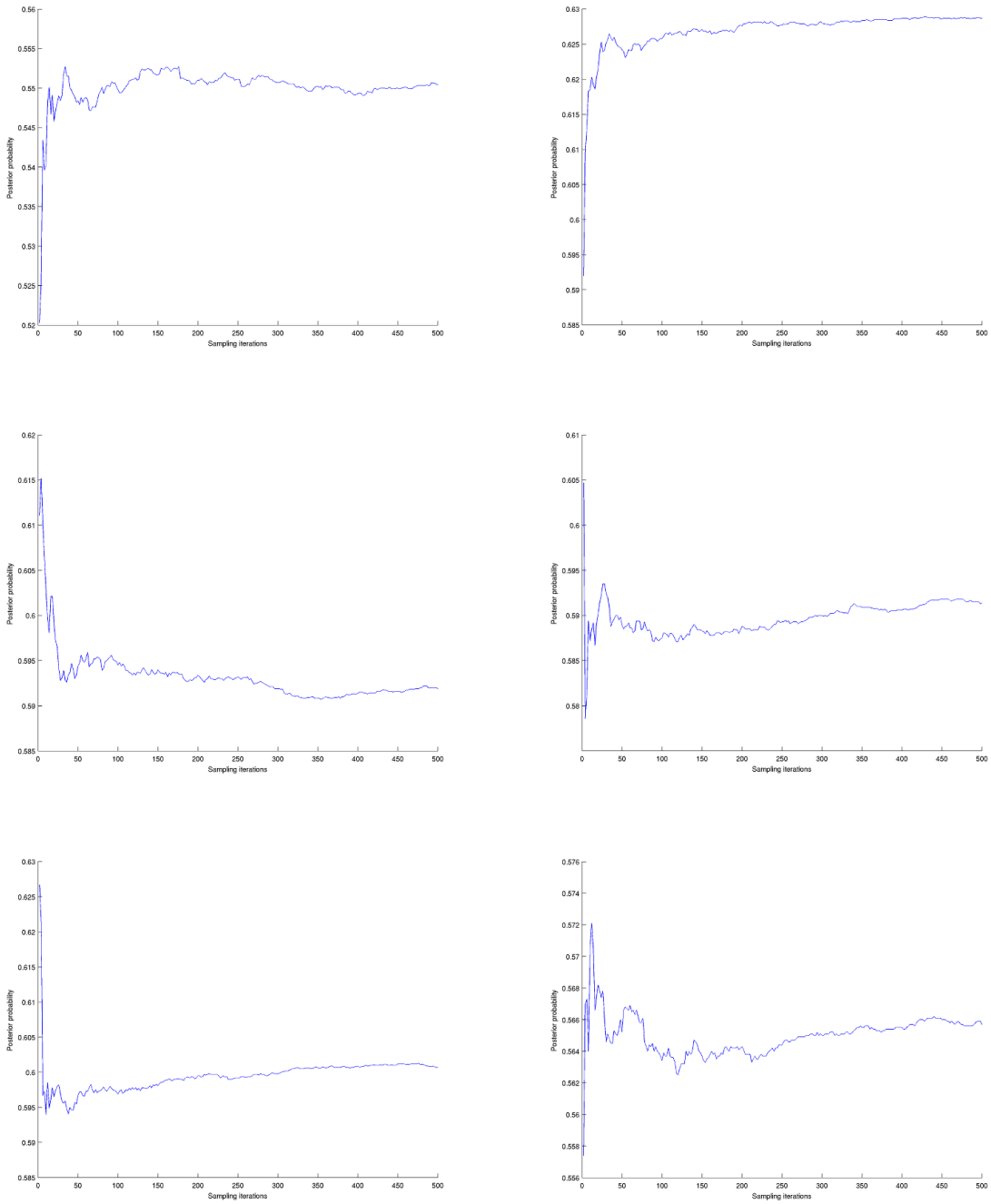



Figure 3.41: Confidence levels against the sampling iterations for sheets

In the following 2 graphs the scale helps to confirm that the algorithm stabilises after 20 iterations for the 20 different runs. The confidence levels for the above mentioned parameters, vary from 55 to 62 percent which implies that the data shapes are distinguishable from the class of ribbons for the chosen values of the varying parameters.

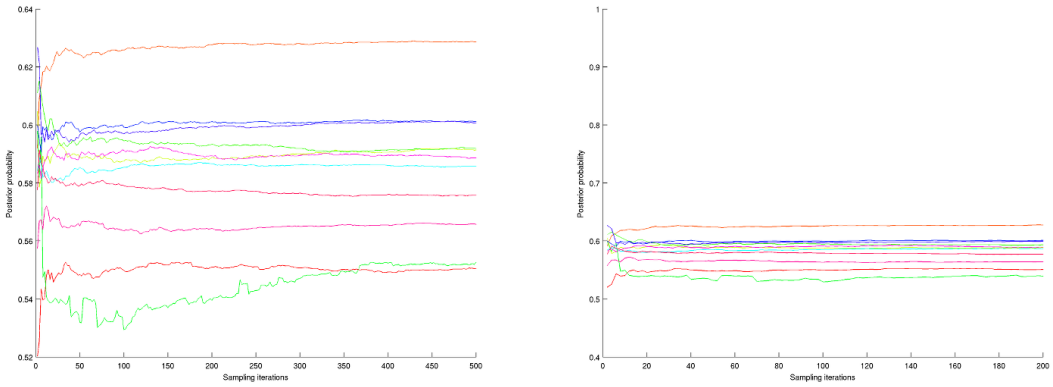



Figure 3.42: Confidence levels against the sampling iterations for sheets

The following graphs present 6 out of the 20 different runs' results of the confidence levels against the sampling iterations for the class of ribbons. The data shapes for this class were generated with 30 points and noise equal to $\sigma = 0.2 \times 10^{-2}$ an the aspect ratio of the data shapes was drawn from a Gamma distribution $\Gamma(7, 0.9)$ close to the "real" values. One notices that the confidence level is stabilised for a threshold $\epsilon = 0.02$ after 20 iterations as in the case of sheets.

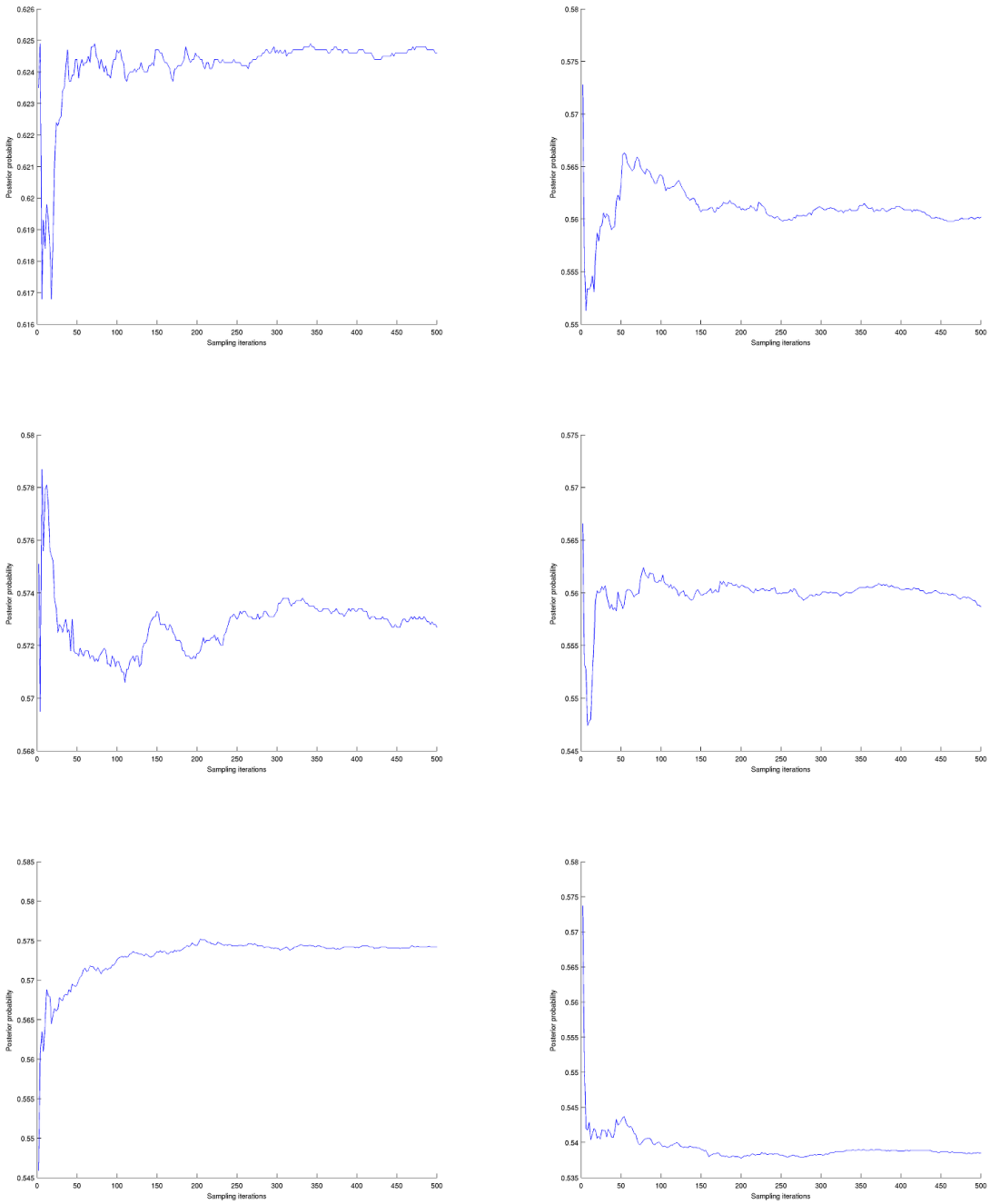



Figure 3.43: Confidence level against the sampling iterations for ribbons

The scale of the following graphs for the 20 different runs for ribbons helps to confirm that the algorithm stabilises after 20 iterations. The confidence levels for the above mentioned parameters, vary from 53 to 74 percent which implies as previously that the data shapes are distinguishable from the class of sheets for the chosen values

of the varying parameters.

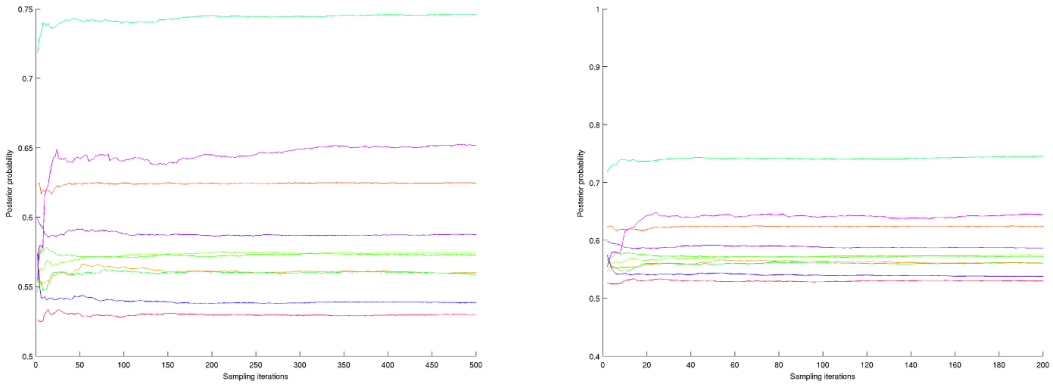



Figure 3.44: Confidence level against the sampling iterations for ribbons

### Confidence levels for Monte Carlo iterations of curves

In this section we perform the same experiment whilst we keep all the parameters constant and vary the Monte Carlo iterations of the curves. For the integration of expression (3.2.1) over the curves $\beta$ we implicitly take $\mathscr{D}\beta\ \mathbb{P}(\beta) \to d\gamma\ \Gamma(\gamma;\kappa,\theta)$. The explicit calculation of the integral over the curves $\beta$ is replaced by an implicit Monte Carlo integral over all aspect ratios that specify sand body curves and are being drawn by a $\Gamma$ distribution with hyperparameters $\kappa$ and $\theta$. Here, we examine the confidence levels that the algorithm returns as we vary the Monte Carlo iterations of the curves. The Monte Carlo iterations of the samplings were chosen to be 250. The following graphs present the confidence levels when the data shapes are sheets whose aspect ratio was drawn from a Gamma distribution $\Gamma(35, 0.2)$ and the noise $\sigma = 0.2 \times 10^{-3}$. The confidence level is stabilised for a threshold $\epsilon = 0.02$ after 20 iterations.

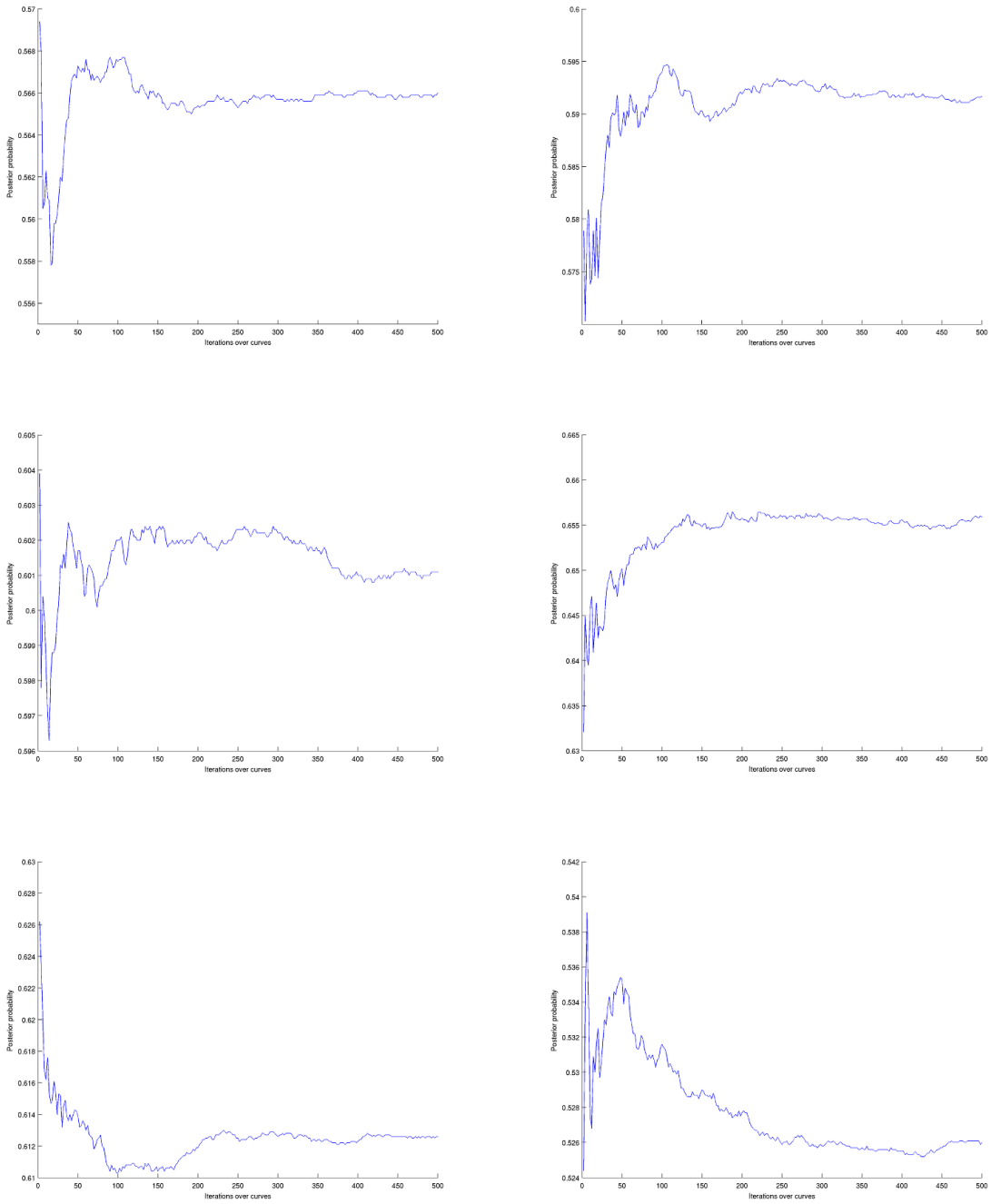



Figure 3.45: Confidence level against the iterations over the curves for sheets

The scale of the following graphs for the 20 different runs for sheets helps to confirm that the algorithm stabilises after 20 iterations. The confidence levels for the above mentioned parameters, vary from 51 to 66 percent which implies as previously that the data shapes are distinguishable from the class of ribbons.

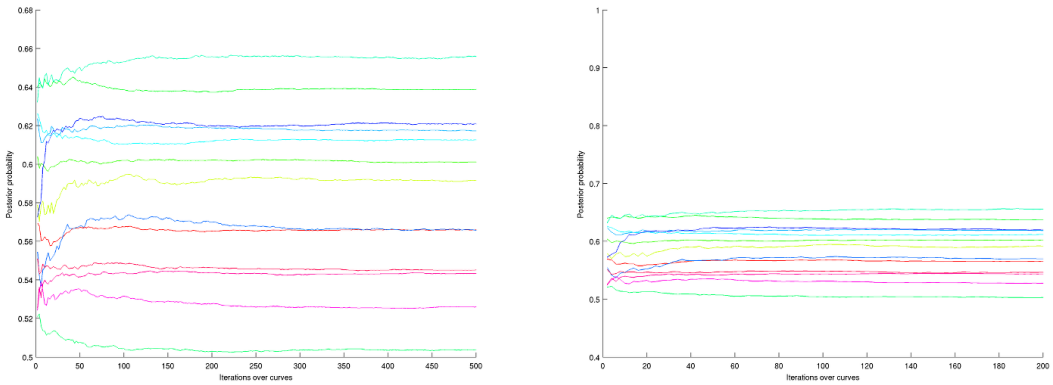



Figure 3.46: Confidence level against the sampling iterations for sheets

The following graphs present 6 out of the 20 different runs' results of the confidence levels against the sampling iterations for the class of ribbons. The data shapes for this class were generated with 30 points and noise equal to $\sigma = 0.2 \times 10^{-2}$ and the aspect ratio of the ribbon data shapes was drawn from a Gamma distribution $\Gamma(6, 0.5)$. One notices that the confidence level is stabilised for a threshold $\epsilon = 0.02$ after 20 iterations as in the case of sheets.

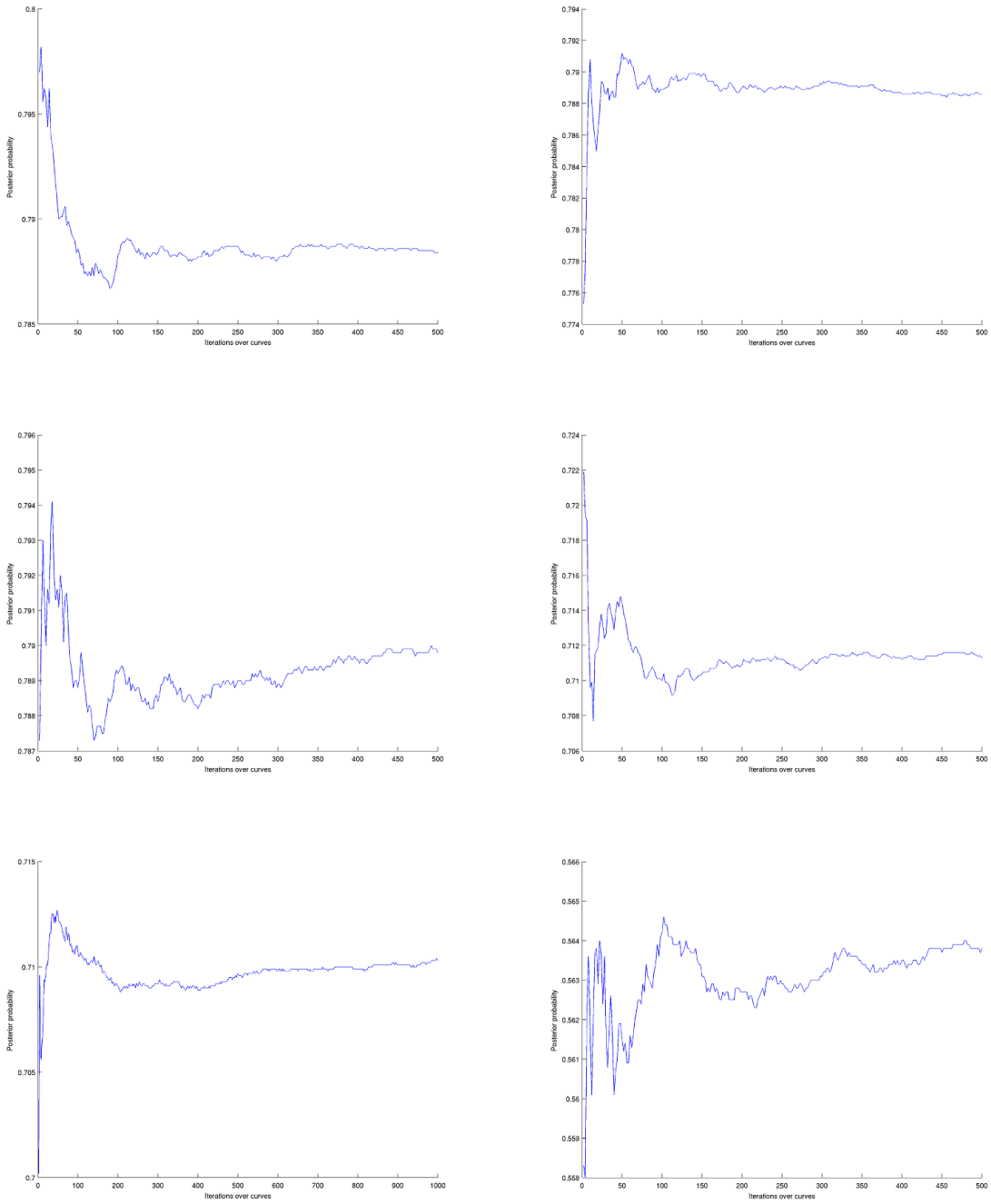



Figure 3.47: Confidence level against the iterations over the curves for ribbons

The scale of the following graphs for the 20 different runs for sheets helps to confirm that the algorithm stabilises after 20 iterations. The confidence levels for the above mentioned parameters, vary from 56 to 83 percent. Once again, the confidence levels show that the class of ribbons is distinguishable from the class of

sheets.

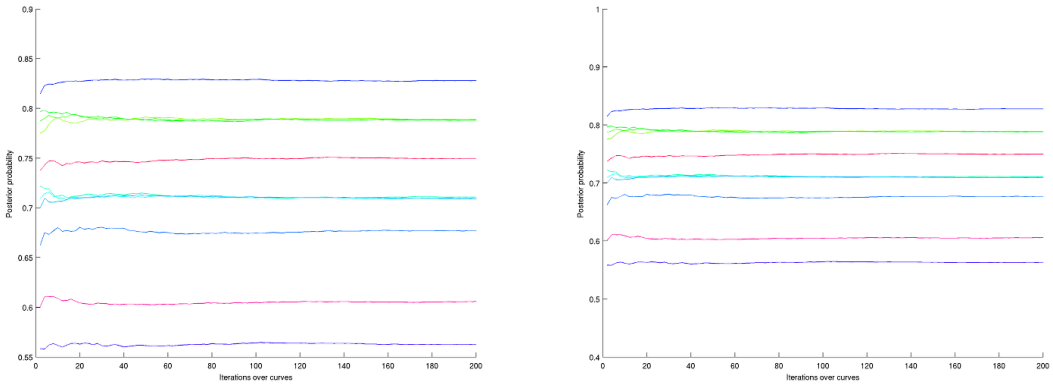



Figure 3.48: Confidence level against the sampling iterations for ribbons

### Success results for Monte Carlo iterations of samplings

In this section we evaluate the success results of the algorithm as we vary the Monte Carlo iterations of the sampling and the curves. The following graphs present the success rates against the sampling iterations. The $y$-axis represents the number of correct classifications for the 20 shapes of each run. For each individual run the sheet data shapes were generated with 30 points and the noise was kept relatively low at $\sigma = 0.3 \times 10^{-2}$. The Monte Carlo iterations of the curves were chosen to be 250 and the aspect ratios of the sheet data shapes $y$ were drawn from a Gamma distribution $\Gamma(49, 0.3)$. One notices that the success rate stabilises after 20 iterations (in some cases after 10 iterations) and the success rate ranges from 85 to 100 percent. For simplicity, we present 6 out of the 20 runs.

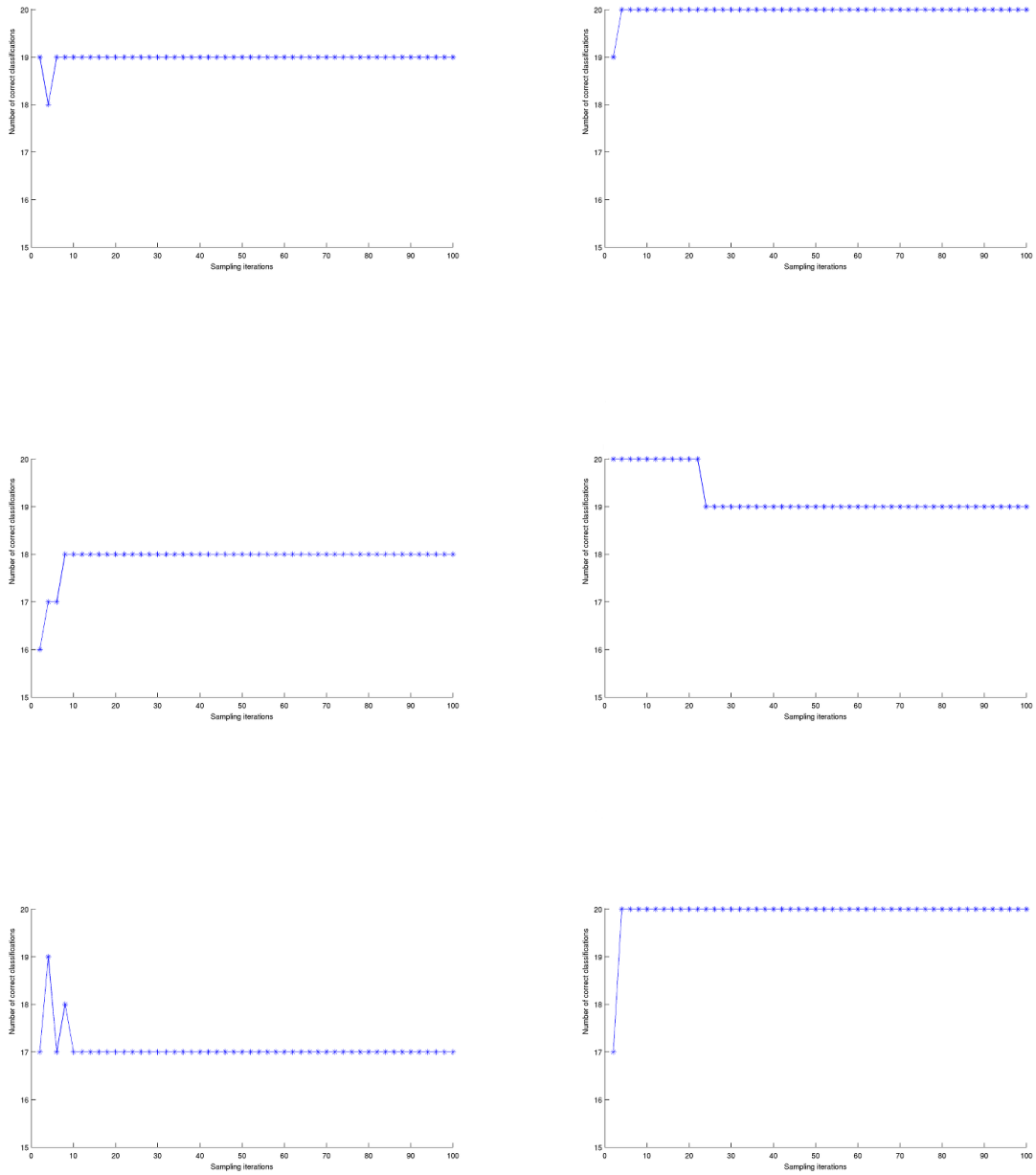



Figure 3.49: Success rates against the sampling iterations for sheets

The following graphs present the results for the success rates for 20 runs of the class of ribbons. For each individual run the sheet data shapes were generated with 30 points and the noise was kept relatively low at $\sigma = 0.3 \times 10^{-2}$. The Monte Carlo iterations of the curves were chosen to be 250 and the aspect ratios of the ribbon

data shapes $y$ were drawn from a Gamma distribution $\Gamma(7, 0.9)$. One notices that the success rate stabilises after 20 iterations and the success rate ranges from 55 to 100 percent. For simplicity, we present 6 out of the 20 runs.

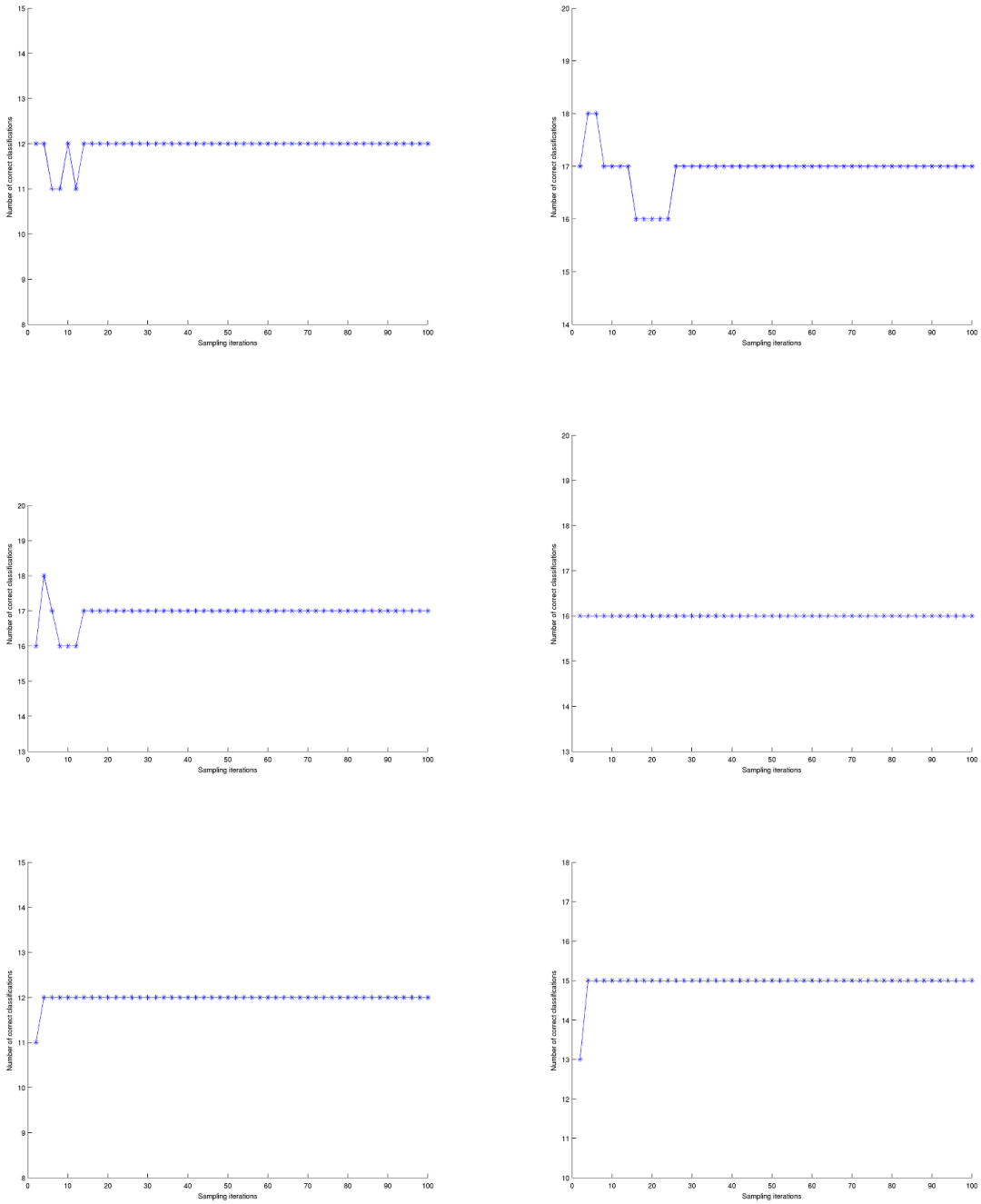



Figure 3.50: Success rates against the sampling iterations for ribbons

### Success rates for Monte Carlo iterations of the curves

In this section we evaluate the success results of the algorithm as we vary the Monte Carlo iterations of the curves $\beta$. For each of the 20 runs the sheet data shapes were generated with 30 points and the noise was kept relatively low at $\sigma = 0.3 \times 10^{-2}$. The Monte Carlo iterations of the saplings were chosen to be 250 and the aspect ratios of the sheet data shapes $y$ were drawn from a Gamma distribution $\Gamma(49, 0.3)$. One notices that the success rate stabilises after 20 iterations and the success rate ranges from 85 to 100 percent. For simplicity, we present 6 out of the 20 runs.

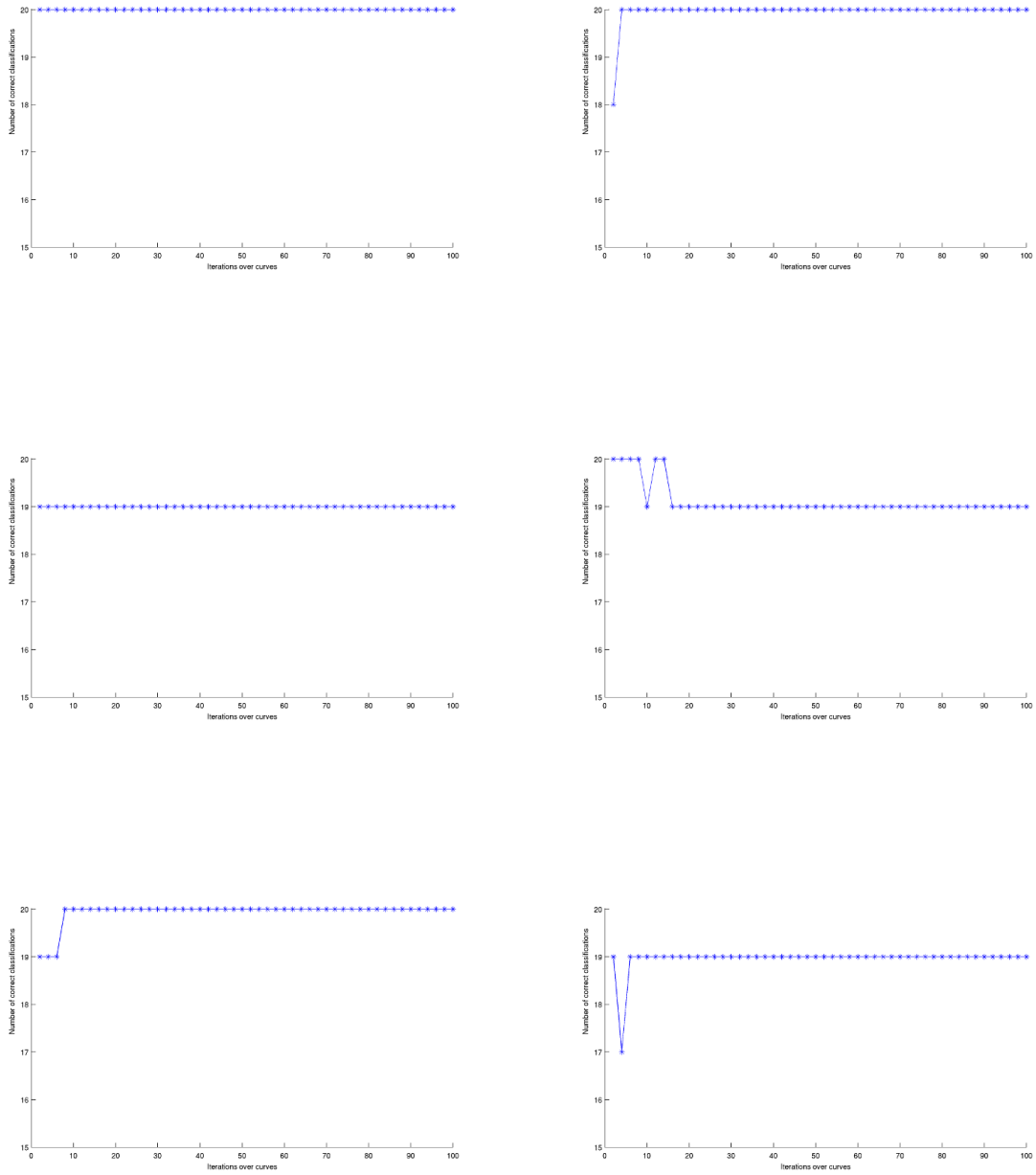



Figure 3.51: Success rates against the sampling iterations for sheets

The following graphs present the results for the success rates for 20 runs of the class of ribbons. For each individual run the sheet data shapes were generated with 30 points and the noise was kept relatively low at $\sigma = 0.3 \times 10^{-2}$. The Monte Carlo iterations of the curves were chosen to be 250 and the aspect ratios of the data

shapes $y$ were drawn from a Gamma distribution $\Gamma(7, 0.9)$. One notices that the success rate stabilises after 20 iterations and the success rate ranges from 40 to 85 percent. For simplicity, we present 6 out of the 20 runs.

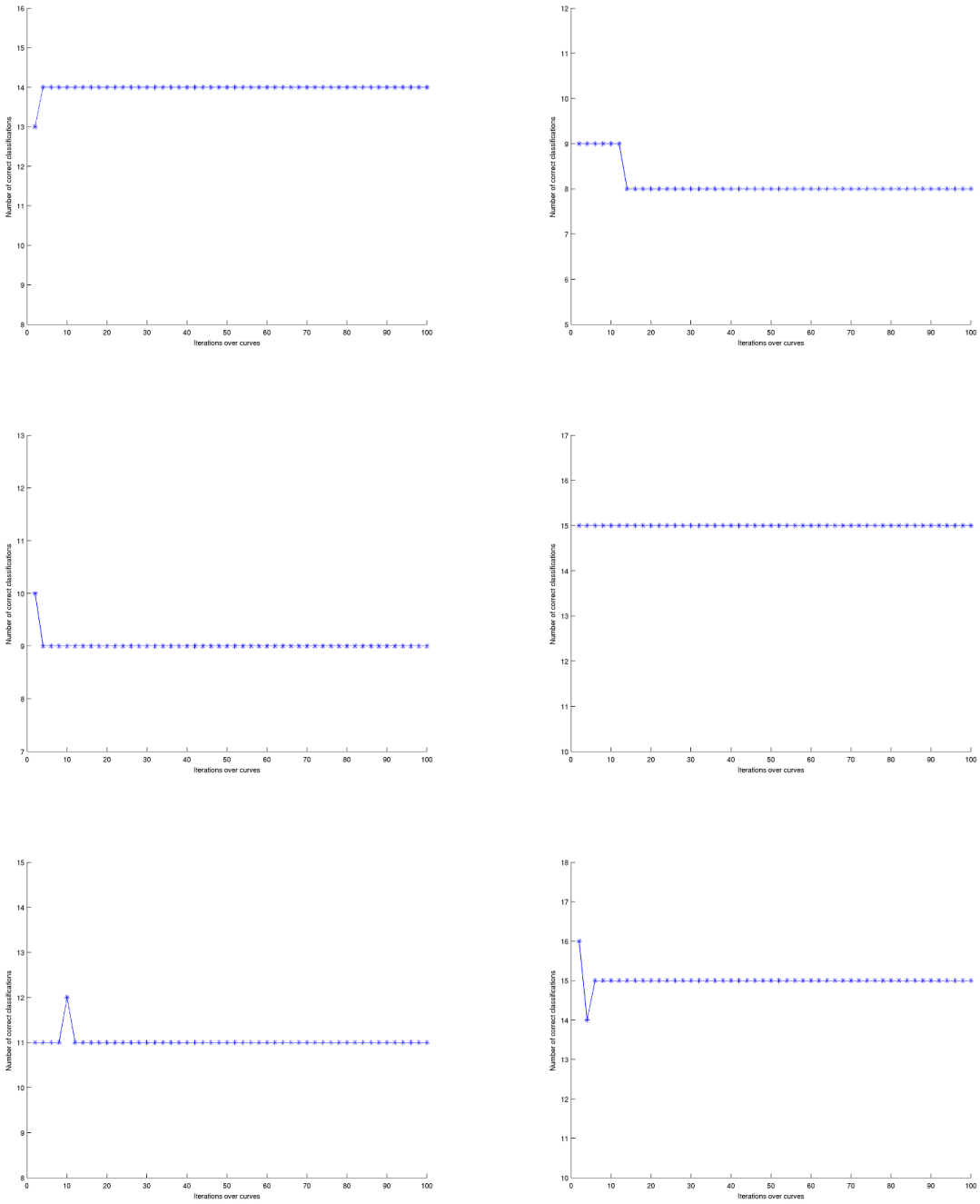



Figure 3.52: Success rates against the iterations over the curves for ribbons

## 3.5 Learning the hyperparameters

The classification of the sand body data set would require us to maximise the posterior distribution of a class $\mathbb{P}(C|\boldsymbol{y})$. To approximate the posterior one has to decide how "similar" a given data shape is to one of the two classes of sheets and ribbons. The comparison is done via our proposed algorithm which utilises a Monte Carlo integration over the shape curves. However, this procedure requires us to know the hyperparameters that generate example shapes from these classes since the shape models are Gamma distributions from which we randomly draw the aspect ratio of a sand body. In the results of the previous section, where we examined the confidence levels and success results, we assumed that the hyperparameters were already known to us. Based on the field results of the sedimentologists and geologists we assumed that such idealised sand bodies' aspect ratios are generated by two $\Gamma(\kappa, \theta)$ distributions with the ribbons' distribution being $\Gamma(7.5, 1)$ and the sheets' distribution being $\Gamma(50, 0.4)$.

In a realistic situation however and had one had real geological data, the parameters of the shape models would be learnt from the available data sets and then used for the classification of new data shapes. The goal would be to learn the parameters so that they can be used to predict the values and the characteristics of a class attribute. Usually this task is called supervised learning. Using supervised learning we can learn a classification model from the existing data (also called training data) which we know they are labelled with pre-defined classes. We can then use the learned model to predict the classes of new, unseen data (also called test data) into these pre-determined classes. The accuracy of the learned model can be tested as the ratio of the number of the correct classifications over the total number of test cases. One of the fundamental assumptions of learning however is that the distribution of the training data is the same as the distribution of the test data. In practice, this assumption is violated quite often leading to poor classification accuracy. Such an assumption and good classification results could be satisfied in the case that the training data sufficiently represent the test data. In the absence of true geological data, we will generate data to play the role of the training data set and assume that

they sufficiently represent the test data on which we will test the accuracy of the model.

To learn the parameters of the model we will perform MAP and maximise the posterior probability distribution with respect to the parameters that we want to learn. This is:

$$\{\kappa_{\max}, \theta_{\max}\} = \text{argmax}_{\kappa,\theta}\mathbb{P}(C|\boldsymbol{y}) \tag{3.5.2}$$

Maximising this posterior probability would require us to evaluate the scores of the posterior with respect to the parameters we are maximising over. However, the derivatives were complicated and there was no closed formed expression for the evaluation of the maximum. To perform MAP then, we need to utilise an optimisation algorithm that maximises the target cost function; we used the method of gradient ascent. The gradient ascent (descent) is a generalised first-order optimisation algorithm that finds the maximum (minimum) of a given function. It starts searching for the optimal solution by some initial guessed values and calculates the gradient of the function at that point (this is the reason it is a first order optimiser; it uses only the first derivative). Then the algorithm takes proportional small steps in the positive (negative) direction of the gradient in order to maximise the function and the process is repeated until convergence. Convergence is achieved either when the derivative of the function is zero or after a certain number of iterations. For example, our cost function is $f(x)$ and we want to find its maximum. Given initial estimate $x_0$ for $x$ we can find the direction in which the function is maximised. This is done by taking small steps proportional to $\nabla f$ in all dimensions of $x$. We take steps proportional to the gradient because it gives the slope of the curve at the point $x$ and its direction shows where the function increases. We change $x$:

$$x_{k+1} = x_k + \epsilon\ \nabla f(x_k) \tag{3.5.3}$$

The parameter $\epsilon > 0$ is a small number that forces the algorithm to take small steps towards the direction of the derivative and also keeps the algorithm stable. In our case the gradient ascent algorithm is performed as follows:

$$\{\kappa_{n+1}, \theta_{n+1}\} = \{\kappa_n, \theta_n\} + \epsilon \, \nabla_{\kappa,\theta} \, \mathbb{P}(C|\boldsymbol{y}) \tag{3.5.4}$$

However, since we perform MAP we know that $\mathbb{P}(C|\boldsymbol{y}) \propto \mathbb{P}(\boldsymbol{y}|C)\mathbb{P}(C)$ so:

$$\{\kappa_{n+1}, \theta_{n+1}\} = \{\kappa_n, \theta_n\} + \epsilon \, \nabla_{\kappa,\theta} \, \mathbb{P}(\boldsymbol{y}|C)\mathbb{P}(C) \tag{3.5.5}$$

The above simplifies even more in our case since the prior distribution over the available classes is uniform so that the gradient ascent simplifies to the following:

$$\{\kappa_{n+1}, \theta_{n+1}\} = \{\kappa_n, \theta_n\} + \epsilon \, \nabla_{\kappa,\theta} \, \mathbb{P}(\boldsymbol{y}|C) \tag{3.5.6}$$

In other words, since the prior distribution over the classes is uniform the model will be maximised when the likelihood is maximal with respect to the parameters we want to learn. From chapter [2], we discussed about the partitioning of the likelihood over the nuisance parameters that give rise to the formation of a planar shape. The likelihood function of the complete data is thus given by:

$$\mathbb{P}(\boldsymbol{y}|C) = \sum_b \mathscr{D}\beta \; \mathscr{D}s \; \frac{1}{Z}\left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \mathbb{P}(\beta)\mathbb{P}(s) \tag{3.5.7}$$

$$\frac{1}{Z} = \frac{1}{(nD^2+1)(2\pi)^n}\frac{\Gamma(n+\alpha)}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)}\frac{\zeta^\alpha}{\Gamma(\alpha)}\frac{1}{2^{-n-\alpha}} \tag{3.5.8}$$

As mentioned, in our case performing Maximum a Posteriori is equivalent to performing Maximum Likelihood (MLE). To perform MLE we assume that the observations that comprise the training data set are all independent and will work under this assumption. This is:

$$\{\kappa_{\max}, \theta_{\max}\} = \max_{\kappa,\theta}\mathbb{P}(\boldsymbol{y}|C) = \max_{\kappa,\theta}\prod_i \mathbb{P}(y_i|C) \tag{3.5.9}$$

It is common practice to maximise the log-likelihood function instead of maximising the likelihood function because the logarithm is often easier to work with. The logarithm, as an increasing function, is maximised at the same points as the function so it makes the calculations of maximisation easier and straight forward. This transforms equation (3.5.9) to:

$$\begin{aligned}\{\kappa_{\max}, \theta_{\max}\} &= \max_{\kappa,\theta} \log\left(\prod_i \mathbb{P}(y_i|C)\right) \\ &= \max_{\kappa,\theta} \sum_i \log\left(\mathbb{P}(y_i|C)\right)\end{aligned} \tag{3.5.10}$$

To perform gradient ascent we need to differentiate the above expression with respect to the hyperparameters that we want to maximise and learn [156]. The derivatives of expression the log-likelihood (3.5.10) are:

$$\begin{aligned}\nabla\left(\log(\mathbb{P}(\boldsymbol{y}|C))\right) &= \left(\frac{\partial \ \sum_i \log\left(\mathbb{P}(y_i|C)\right)}{\partial\kappa}, \frac{\partial \ \sum_i \log\left(\mathbb{P}(y_i|C)\right)}{\partial\theta}\right) \\ &= \left(\sum_i \frac{\partial \ \left(\mathbb{P}(y_i|C)\right)/\partial\kappa}{\left(\mathbb{P}(y_i|C)\right)}, \sum_i \frac{\partial \ \left(\mathbb{P}(y_i|C)\right)/\partial\theta}{\left(\mathbb{P}(y_i|C)\right)}\right)\end{aligned} \tag{3.5.11}$$

To make the calculation easier we calculate the numerators of expression (3.5.11) i.e. the partial derivatives of the likelihood with respect to the hyperparameters separately. The derivative of the likelihood with respect to $\kappa$ is:

$$\frac{\partial \ \left(\mathbb{P}(y_i|C)\right)}{\partial \ \kappa} = \frac{\partial}{\partial \ \kappa}\left(\sum_b \int \mathscr{D}\beta \ \mathscr{D}s \ \mathbb{P}(y_i|\beta, s, b)\mathbb{P}(\beta)\mathbb{P}(s)\right) \tag{3.5.12}$$

We should note here, that the likelihood implicitly depends on the $\kappa$ and $\theta$ parameters since these generate the aspect ratio $\gamma$ on which the likelihood depends on. The planar curves of the sand bodies are specified by their aspect ratios so for computational reasons we substitute: $\mathscr{D}\beta \ \mathbb{P}(\beta) \to d\gamma \ \Gamma(\gamma;\kappa,\theta)$ with $\Gamma(\gamma;\kappa,\theta) = \frac{\gamma^{\kappa-1} \ e^{-\frac{\gamma}{\theta}}}{\theta^{\kappa}\Gamma(\kappa)}$. The partial derivative of the likelihood with respect to $\kappa$ is thus:

$$\frac{\partial\ (\mathbb{P}(y_i|C))}{\partial\ \kappa} = \frac{\partial}{\partial\ \kappa}\left(\sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \mathbb{P}(y_i|\beta,s,b)\Gamma(\gamma;\kappa,\theta)\mathbb{P}(s)\right)$$
$$= \frac{\partial}{\partial\ \kappa}\Bigg(\sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \times$$
$$\Gamma(\gamma;\kappa,\theta)\mathbb{P}(s)\Bigg) \tag{3.5.13}$$

However, the above expression has an implicit $\kappa$ dependence in the likelihood but has an explicit $\kappa$ dependence only through the $\Gamma$ prior on the aspect ratio. The derivative of a $\Gamma$ distribution with respect to $\kappa$ is:

$$\begin{aligned}\frac{\partial}{\partial\ \kappa}\left(\Gamma(\gamma;\kappa,\theta)\right) &= \frac{\partial}{\partial\ \kappa}\frac{\gamma^{\kappa-1}\ e^{-\frac{\gamma}{\theta}}}{\theta^\kappa\Gamma(\kappa)} \\ &= \frac{e^{-\frac{\gamma}{\theta}}}{\Gamma(\kappa)}\ \ \frac{\partial}{\partial\ \kappa}\left(\theta^{-\kappa}\gamma^{\kappa-1}\right) \\ &= \frac{e^{-\frac{\gamma}{\theta}}\theta^{-\kappa}\gamma^{\kappa-1}}{\Gamma(\kappa)}\left(\log(\gamma)-\log(\theta)-\log(\psi)\right) \\ &= \Gamma(\gamma;\kappa,\theta)\left(\log(\gamma)-\log(\theta)-\log(\psi)\right)\end{aligned} \tag{3.5.14}$$

where $\psi$ is the digamma function which is defined as $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$. Expression (3.5.14) enters the derivative of the likelihood with respect to $\kappa$ (3.5.13) which becomes:

$$\frac{\partial\ (\mathbb{P}(y_i|C))}{\partial\ \kappa} = \Bigg(\sum_b \int d\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \mathbb{P}(s)\times$$
$$\Gamma(\gamma;\kappa,\theta)(\log(\gamma)-\log(\theta))\Bigg) \tag{3.5.15}$$

The derivative of the likelihood with respect to $\theta$ is:

$$\frac{\partial\ (\mathbb{P}(y_i|C))}{\partial\ \theta} = \frac{\partial}{\partial\ \theta}\left(\sum_b \int \mathscr{D}\beta\ \mathscr{D}s\ \mathbb{P}(y_i|\beta,s,b)\mathbb{P}(\beta)\mathbb{P}(s)\right) \tag{3.5.16}$$

The partial derivative of the likelihood with respect to $\theta$ is thus:

$$\begin{aligned}\frac{\partial\ (\mathbb{P}(y_i|C))}{\partial\ \theta} &= \frac{\partial}{\partial\ \theta}\left(\sum_b \int d\gamma\ \mathscr{D}s\ \mathbb{P}(y_i|\beta,s,b)\Gamma(\gamma;\kappa,\theta)\mathbb{P}(s)\right) \\ &= \frac{\partial}{\partial\ \theta}\Bigg(\sum_b \int d\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \times \\ &\qquad \Gamma(\gamma;\kappa,\theta)\mathbb{P}(s)\Bigg)\end{aligned} \tag{3.5.17}$$

Similarly, the above expression has an implicit $\theta$ dependence in the likelihood but has an explicit $\theta$ dependence only through the $\Gamma$ prior on the aspect ratio. The derivative of a $\Gamma$ distribution with respect to $\theta$ is:

$$\begin{aligned}\frac{\partial}{\partial\ \theta}\left(\Gamma(\gamma;\kappa,\theta)\right) &= \frac{\partial}{\partial\ \theta}\frac{\gamma^{\kappa-1}\ e^{-\frac{\gamma}{\theta}}}{\theta^\kappa\Gamma(\kappa)} \\ &= \frac{\gamma^{\kappa-1}}{\Gamma(\kappa)}\ \frac{\partial}{\partial\ \theta}\left(\theta^{-\kappa}e^{-\frac{\gamma}{\theta}}\right) \\ &= \frac{\gamma^{\kappa-1}e^{-\frac{\gamma}{\theta}}\theta^{-\kappa}}{\Gamma(\kappa)}\ \left(\frac{\gamma}{\theta^2} - \kappa\theta^{-1}\right) \\ &= \Gamma(\gamma;\kappa,\theta)\left(\frac{\gamma}{\theta^2} - \kappa\theta^{-1}\right)\end{aligned} \tag{3.5.18}$$

Expression (3.5.18) enters the derivative of the likelihood with respect to $\theta$ (3.5.17) which becomes:

$$\begin{aligned}\frac{\partial\ (\mathbb{P}(y_i|C))}{\partial\ \theta} = \Bigg(\sum_b \int d\gamma\ \mathscr{D}s\ &\left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \mathbb{P}(s)\times \\ &\Gamma(\gamma;\kappa,\theta)\left(\frac{\gamma}{\theta^2} - \kappa\theta^{-1}\right)\Bigg)\end{aligned} \tag{3.5.19}$$

Combining expressions (3.5.19) and (3.5.15), the overall gradient of the log-likelihood function with respect to the hyperparameters is:

$$\begin{aligned}\nabla\left(\log(\mathbb{P}(\boldsymbol{y}|C))\right)\Big|_{\kappa} &= \left(\sum_i \frac{\partial\ \left(\mathbb{P}(y_i|C)\right)/\partial\kappa}{\left(\mathbb{P}(y_i|C)\right)}\right) \\ &= \sum_i \left(\frac{\sum_b \int d\gamma\ \mathscr{D}s\ \mathbb{P}(y_i|\beta,s,b)\mathbb{P}(s)\Gamma(\gamma;\kappa,\theta)\left(\log(\gamma)-\log(\theta)\right)}{\sum_b \int d\gamma\ \mathscr{D}s\ \mathbb{P}(\boldsymbol{y}|\beta,s,b)\mathbb{P}(s)\Gamma(\gamma;\kappa,\theta)}\right)\end{aligned} \tag{3.5.20}$$

$$\begin{aligned}\nabla\left(\log(\mathbb{P}(y_i|C))\right)\Big|_{\theta} &= \left(\sum_i \frac{\partial\ \left(\mathbb{P}(y_i|C)\right)/\partial\theta}{\left(\mathbb{P}(y_i|C)\right)}\right) \\ &= \sum_i \left(\frac{\sum_b \int d\gamma\ \mathscr{D}s\ \mathbb{P}(y_i|\beta,s,b)\mathbb{P}(s)\Gamma(\gamma;\kappa,\theta)\left(\frac{\gamma}{\theta^2}-\kappa\theta^{-1}\right)}{\sum_b \int d\gamma\ \mathscr{D}s\ \mathbb{P}(y_i|\beta,s,b)\mathbb{P}(s)\Gamma(\gamma;\kappa,\theta)}\right)\end{aligned} \tag{3.5.21}$$

In the above derivatives, one notices that the expressions in the numerators are similar to the ones in the denominators and involve the same type of integration like with expression (3.2.1). For the calculation of the integrals and hence the calculation of the derivatives, we use the same techniques as with the calculation of expression (3.2.1). The integrals with respect to the aspect ratio $\gamma$ are evaluated by simple Monte Carlo integration by drawing discrete values of $\gamma$ from a $\Gamma$ distribution.

In the next section, we present some of the results we acquired from the gradient ascent optimisation. The learning of the parameters was done for both ribbons and sheets.

### 3.5.1 Parameters for sheets

For the learning of the parameters for the class of sheets, we used ten data shapes which were generated as described in section [3.4.3]. We then used the gradient ascent algorithm for four different sets of starting values of $\kappa$ and $\theta$. Each set of initial starting values was run 10 separate times and the chosen value of $\epsilon$ was chosen to be $\epsilon = 0.0025$. For all runs of the gradient ascent, the algorithm converged either after the maximum number of iterations was achieved or after the value the Euclidean distance $d = \sqrt{d\kappa^2 + d\theta^2}$ was smaller than a threshold which was chosen to be equal to 0.0002. The following table presents the data acquired:

| **Starting values** | **Mean convergent values** |
|---|---|
| $\kappa_0 = 45,\ \theta_0 = 0.3$ | $\kappa = 45.013 \pm 8 \times 10^{-4}$ , $\theta = 1.133 \pm 3.5 \times 10^{-3}$ |
| $\kappa_0 = 53,\ \theta_0 = 0.5$ | $\kappa = 53.010 \pm 7 \times 10^{-4}$ , $\theta = 1.133 \pm 3.3 \times 10^{-1}$ |
| $\kappa_0 = 20,\ \theta_0 = 1$ | $\kappa = 20.040 \pm 1.8 \times 10^{-3}$ , $\theta = 1.653 \pm 19 \times 10^{-2}$ |
| $\kappa_0 = 60,\ \theta_0 = 0.1$ | $\kappa = 60.002 \pm 1.5 \times 10^{-4}$ , $\theta = 1.4816 \pm 16 \times 10^{-2}$ |

The following graphs present the results for the four different sets of starting values, each evaluated for one run.

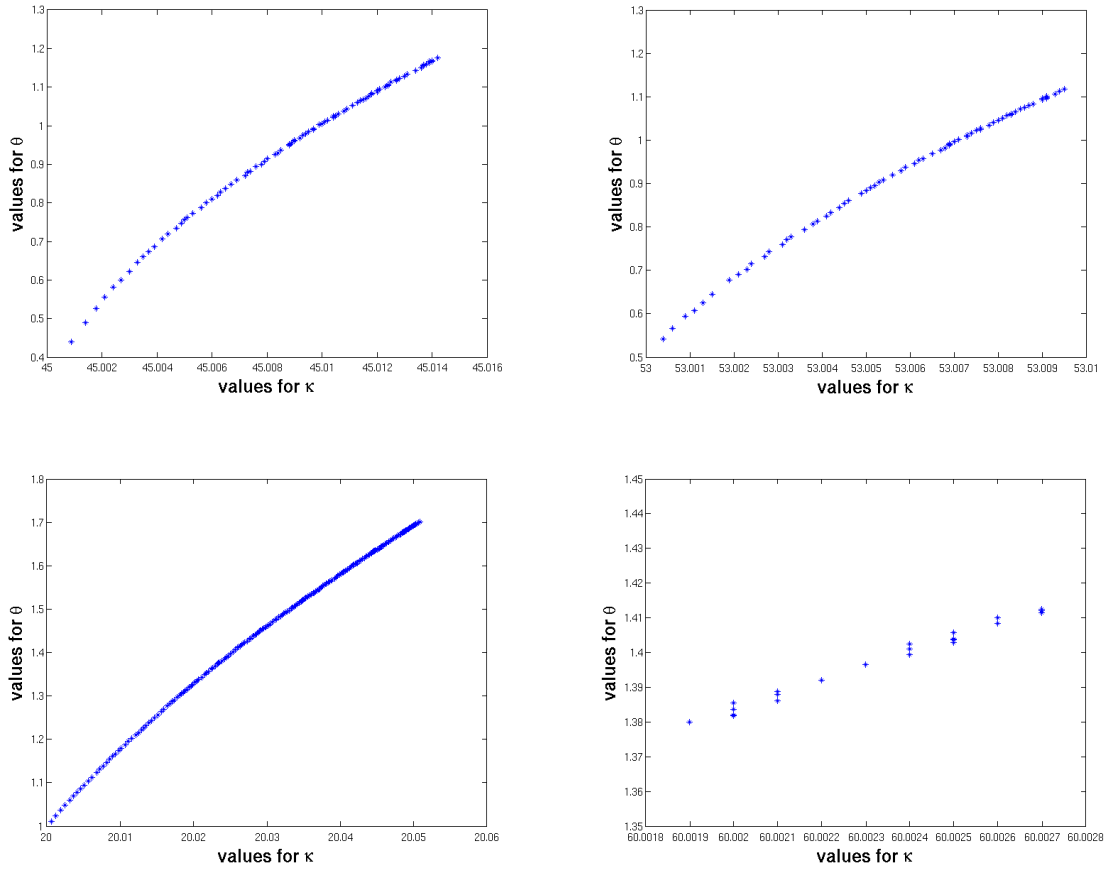



Figure 3.53: The convergence results for the four sets of starting values of the gradient ascent for sheets

### 3.5.2 Parameters for ribbons

The same procedure was followed in the case of ribbons. Ten ribbon data shapes were used for the estimation of the hyperparameters of the Gamma distribution. The gradient ascent was run for four different sets of starting values whilst $\epsilon$ was chosen to be $\epsilon = 0.0025$. For each set of starting values the algorithm was run 10 separate times and it converged either after the maximum number of iterations was achieved or after the value the Euclidean distance $d = \sqrt{d\kappa^2 + d\theta^2}$ was smaller than a threshold which was chosen to be equal to 0.0002. The following table presents the data acquired:

| Starting values | Mean convergent values |
|---|---|
| $\kappa_0 = 6.5,\ \theta_0 = 0.4$ | $\kappa = 6.731 \pm 5 \times 10^{-3}$ , $\theta = 1.693 \pm 1.3 \times 10^{-4}$ |
| $\kappa_0 = 7,\ \theta_0 = 1.5$ | $\kappa = 7.039 \pm 3 \times 10^{-3}$ , $\theta = 1.622 \pm 9 \times 10^{-3}$ |
| $\kappa_0 = 2,\ \theta_0 = 0.1$ | $\kappa = 3.41 \pm 2 \times 10^{-3}$ , $\theta = 2.93 \pm 1 \times 10^{-3}$ |
| $\kappa_0 = 20,\ \theta_0 = 5$ | $\kappa = 19.999 \pm 4 \times 1^{-5}$ , $\theta = 4.996 \pm 1 \times 10^{-3}$ |

The following graphs present the results for the four different sets of starting values, each evaluated for one run.

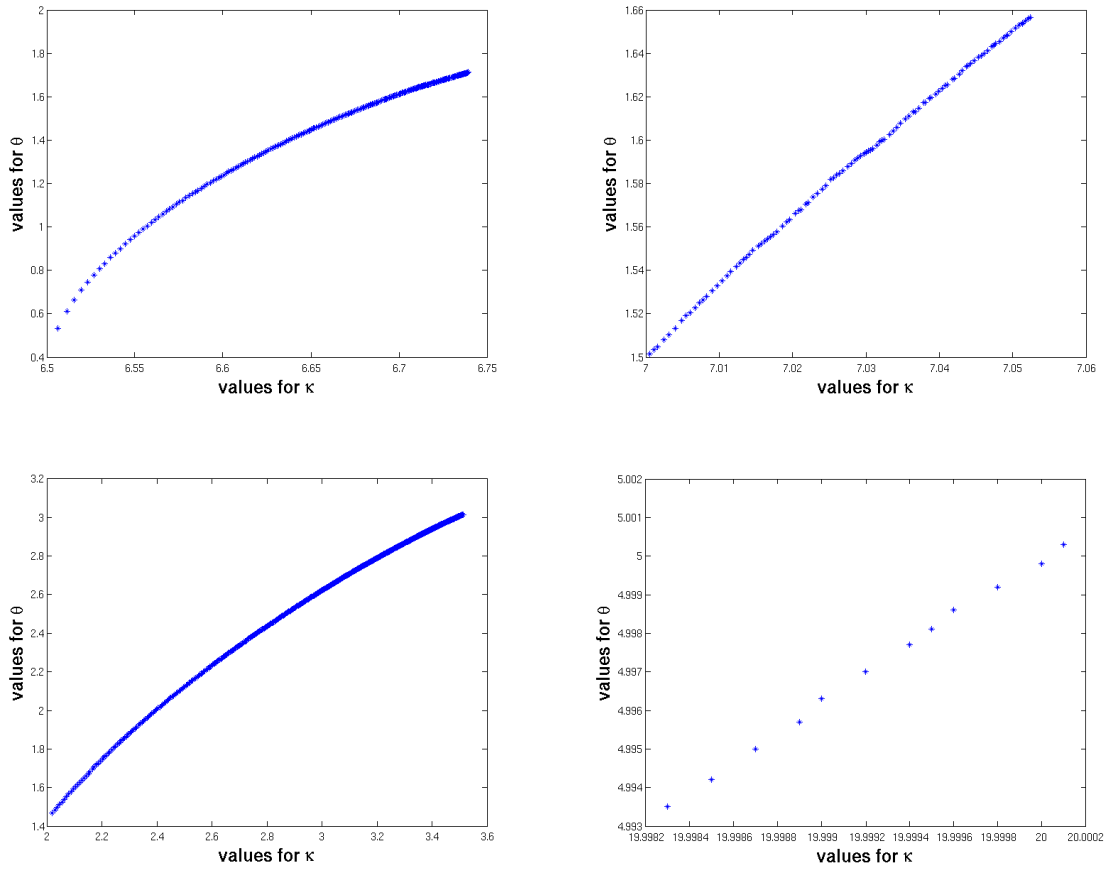
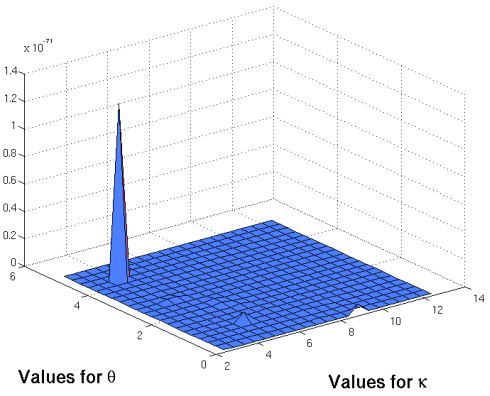



Figure 3.54: The convergence results for the four sets of starting values of the gradient ascent for ribbons

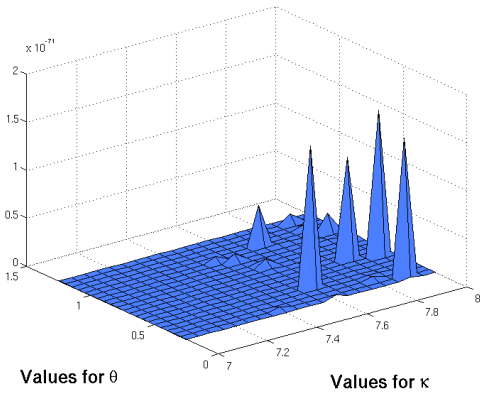
### 3.5.3 Discussion of the results

It is clear from the tables above that the gradient ascent algorithm does not converge to the desired values of $\kappa$ and $\theta$ for both ribbons and sheets and we now discuss the

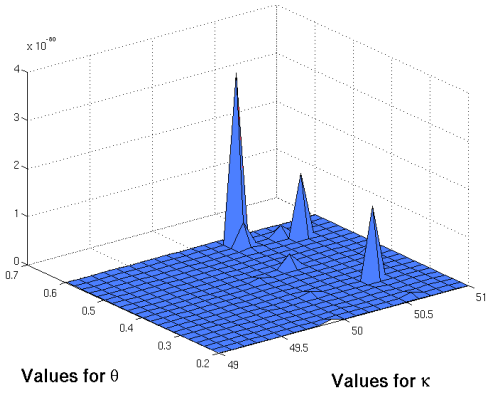
reasons behind this behaviour. Firstly, the threshold for convergence was chosen at the outset of the experiment, and one may ask whether reducing this parameter may improve the experimental outcome. However, as with any application of a basic gradient ascent procedure, the form of the function being maximised is of great importance. Although the algorithm may find a stationary point of the function in question, it is not possible to guarantee that this will not be a local (rather than global) maximum of the function. In particular, it is often necessary to carry out a number of different simulations with different starting values $\kappa_0$ and $\theta_0$. As presented in the tables above for both sheets and ribbons, the values at which the parameters converged were different for each set of starting values. This suggests that the gradient ascent is becoming "trapped" in one of several local maxima of the likelihood surface which in turn implies that the surface itself may be quite uneven and unsmooth. A way to overcome this obstacle would be to examine the value of the likelihood at the convergent point and choose the set of values for which the convergent likelihood was the highest. However, for three out of the four sets of starting values the values of the likelihood at the convergent points are comparable which makes the choice even harder; the convergent value of the likelihood of the fourth set was too small to be considered. This motivates us to consider a scan of the parameter region in order to identify the rough form of the likelihood within the domain in question (note that the nature of Monte-Carlo integration means that for each simulation the likelihood surface will differ, perhaps substantially). This was achieved by evaluating the likelihood on a $(20 \times 20)$ grid on the region $[2.5, 12.5] \times [0.1, 5.1]$ for ribbons and the result of this simulation can be seen in figure (3.55a). Figure (3.55b) shows the likelihood evaluated on a $(20 \times 20)$ grid on the region $[7, 8] \times [0.25, 1.25]$. Figure (3.55c) and figure (3.55d) both show the evaluation of the likelihood on a $(20 \times 20)$ grid on the region $[49, 51] \times [0.2, 0.6]$.



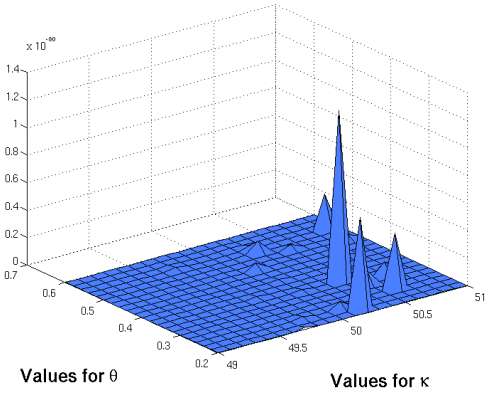

(a) Likelihood surface for 10 ribbons




(b) Likelihood surface for 10 ribbons




(c) Likelihood surface for 10 sheets

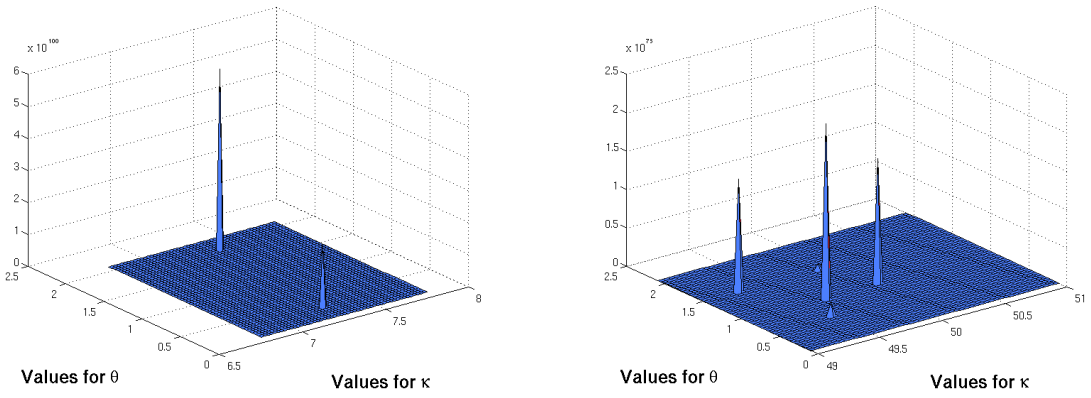



(d) Likelihood surface for 10 sheets

Figure 3.55: Likelihood surfaces

The plot (3.55a) shows an extreme spike in the likelihood around the point $k = 4$, $\theta = 4.35$, but is otherwise comparatively flat. However, the gradient ascent algorithm may not be initialised close to this peak, in which case the form of the surface makes it relatively hard to find. The plot (3.55b) shows an big spike in the likelihood around the point $k = 7.95$, $\theta = 0.6$, where lots of stationary points can be seen, in any of which the algorithm could get stuck. Similarly for sheets in figures (3.55c) and (3.55d), the big spikes can be found at the points $k = 50.3$, $\theta = 0.5$ and $k = 50.4$, $\theta = 0.34$ respectively however in a similar fashion there are multiple stationary points at which the gradient ascent could get trapped. For both ribbons and sheets, we restricted attention to the regions $[6.8, 7.8] \times [0.1, 2.1]$ and $[49, 50] \times [0.2, 0.6]$ and scanning the likelihood surface more closely on a $(50 \times 50)$

grid the following plots show the results in more detail:




(a) Likelihood surface for 10 ribbons

(b) Likelihood surface for 10 sheets

Figure 3.56: Likelihood surfaces

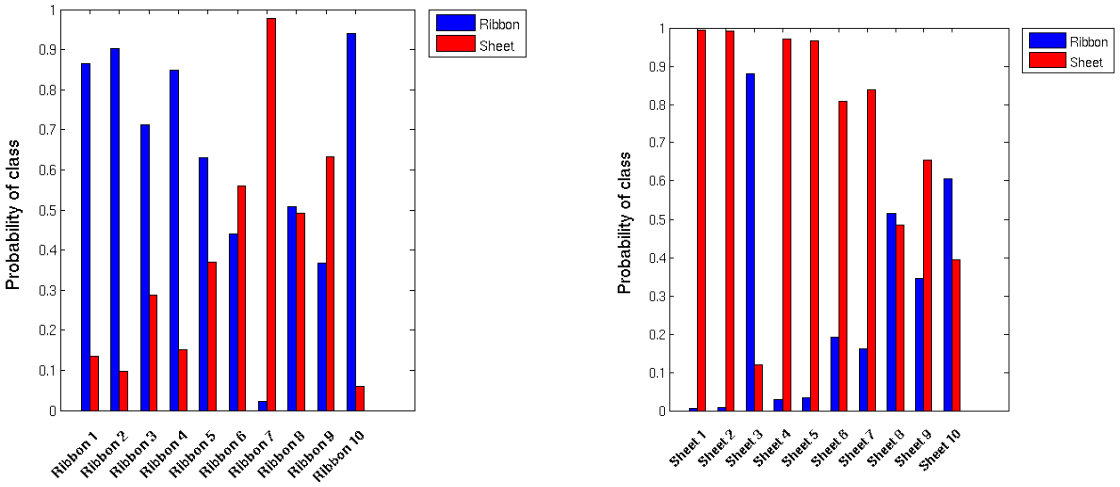
In both plots there is a huge spike at the points $k = 7.48$, $\theta = 0.86$ and $k = 49.64$, $\theta = 0.94$. However, as we mentioned above the gradient ascent algorithm was either not initialised close to these peaks (which are relatively close to the true, expected values) or the algorithm was trapped on a local maximum and hence for different starting values, it converged to a different point. This is likely what is happening in tables (3.5.2) and (3.5.1) and we suggest that it is responsible for the less positive results presented there. Although the results were not as encouraging as expected, we proceeded to the classification of generated sand bodies using the "known" values of the hyperparameters. We discuss this in the next section.

### 3.5.4 Classification results

As we discussed in the previous section, the results produced by the gradient ascent were unsatisfactory and hence the learning of the parameters wasn't fruitful. This would mean than the classification of the sand bodies using the learned parameters wasn't possible. In absence of these data and wanting to evaluate the efficacy of the algorithm in classifying sand body data shapes, we performed classification by using the known parameters. This means that for each classification run, the data

shapes would be compared against example shapes that have been generated with aspect ratios coming from Gamma distributions with the same parameters as the ones that generated the data shapes themselves. Hence the Monte Carlo integration over shape curves would be evaluated by comparing the data shapes to example shapes that their aspect ratios have been generated by $\Gamma(7.5, 1)$ for ribbons and $\Gamma(50, 0.4)$ for sheets. For the following classification results, the sampling Monte Carlo iterations and the Monte Carlo iterations over the curves were both fixed to be 20, the minimum number that is needed for the confidence results to stabilise. For the following runs we use the same values of the regulators as in the Kimia and the alphabet case. The values of the regulators were chosen to be: $B = 10^5$, $D = 10^5$, $\alpha = 1.5$ and $\zeta = 0.1$. The variance of the generalised Gaussian of the diffeomorphisms (2.3.7) was chosen to be $\sigma_s = 1.5$.

In the next figure we present the classification results for 10 ribbons and 10 sheets that were generated under the same parameters. For their simulation we used 30 points and the standard deviation of the observed noise was $\sigma = 0.4 \times 10^{-2}$. For both ribbons and sheets, the success rate was found to be 70% with the average confidence level 77% for ribbons and 88% for sheets. The lowest classification level was 52% for ribbons and 68% for sheets whilst the highest was 95% and 100% respectively. Although the number of the points is considered to be quite low, the confidence levels and the success rates for both classes are particularly high.

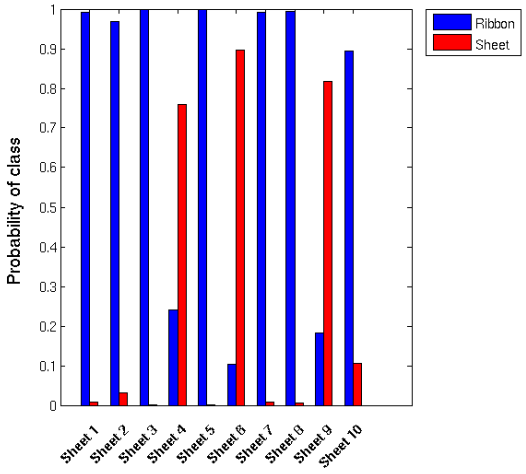



(a) Classification results for 10 ribbons

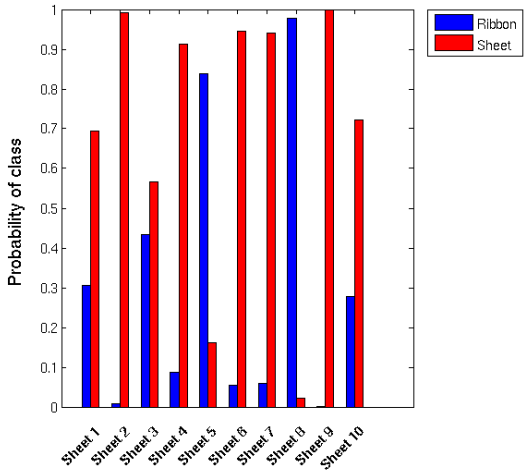
(b) Classification results for 10 sheets

Figure 3.57: Classification results for two classes

In the next figure we present the classification results for 10 ribbons and 10 sheets that were generated with 40 points and the standard deviation of the observed noise was $\sigma = 0.5 \times 10^{-2}$. For ribbons, the success rate was 70%, the lowest classification level was 90% whilst the average classification level was 97%. For sheets, the success rate was 80% with the lowest classification level 58% and the highest 100%. The average classification level for sheets was found to be 84%. Again, although the number of the points is considered to be quite low and the variance of the noise relatively big, the confidence levels and the success rates for both classes are particularly high.



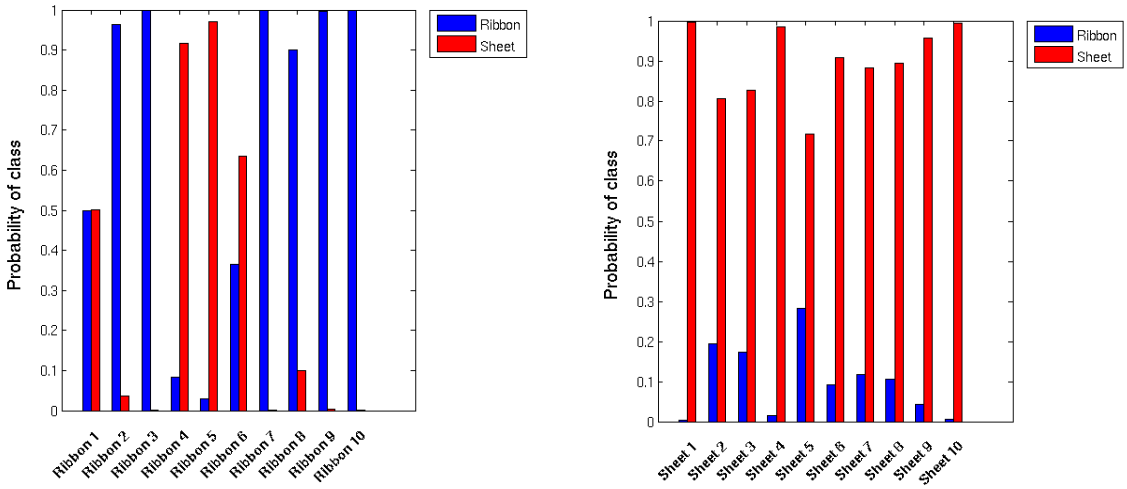

(a) Classification results for 10 ribbons




(b) Classification results for 10 sheets

Figure 3.58: Classification results for two classes

In the next figure we present the classification results for 10 ribbons and 10 sheets that were generated with 50 points and the standard deviation of the observed noise was $\sigma = 0.8 \times 10^{-2}$ which is considered to be high. Although for ribbons, the success rate was 60%, the lowest classification level was 90% whilst the average classification level was 97%. For sheets, the success rate was 100% with the lowest classification level 70% and the average classification level being 89%. The noise variance is extremely high, however the classification levels are high too and the algorithm seems to distinguish the differences between classes consistently.

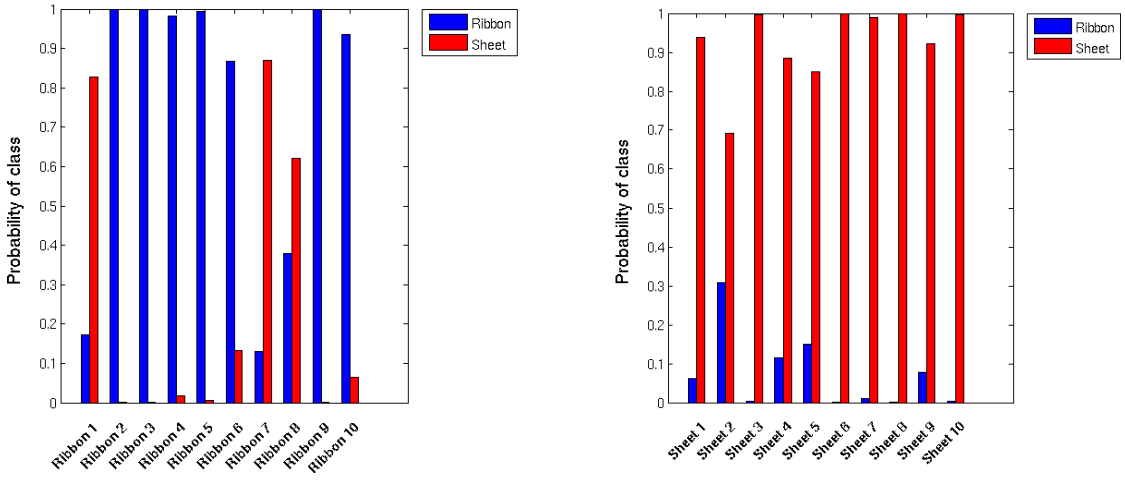



(a) Classification results for 10 ribbons

(b) Classification results for 10 sheets

Figure 3.59: Classification results for two classes

In the next figure we present the classification results for 10 ribbons and 10 sheets that were generated with 60 points and the standard deviation of the observed noise was $\sigma = 0.7 \times 10^{-2}$ which is considered quite high. For ribbons, the success rate was 70%, the lowest classification level was 90% whilst the average classification level was 96%. For sheets, the success rate was 100% with the lowest classification level 70% and the average classification level being 99%.

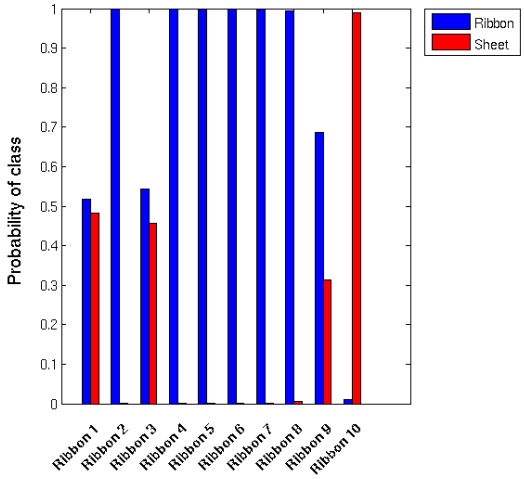



(a) Classification results for 10 ribbons

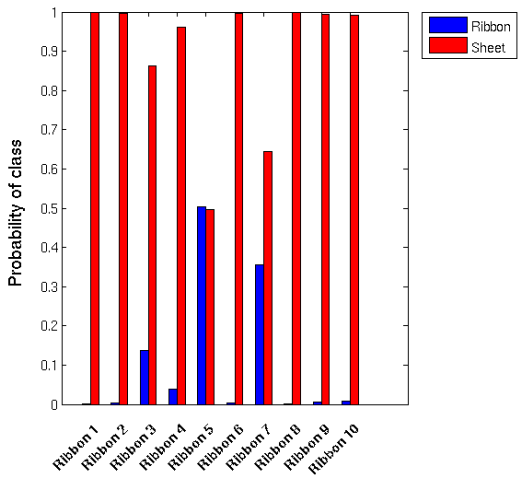
(b) Classification results for 10 sheets

Figure 3.60: Classification results for two classes

In the next figure we present the classification results for 10 ribbons and 10 sheets that were generated with 80 points and the standard deviation of the observed noise was $\sigma = 0.9 \times 10^{-2}$. For ribbons, the success rate was 90%, the lowest classification level was 70% whilst the average classification level was 85%. For sheets, the success rate was 90% with the lowest classification level 65% and the average classification level being 93%. One can see that the the classification rates and levels are extremely high and it seems that the high number of points contemplates the high noise added to the points.



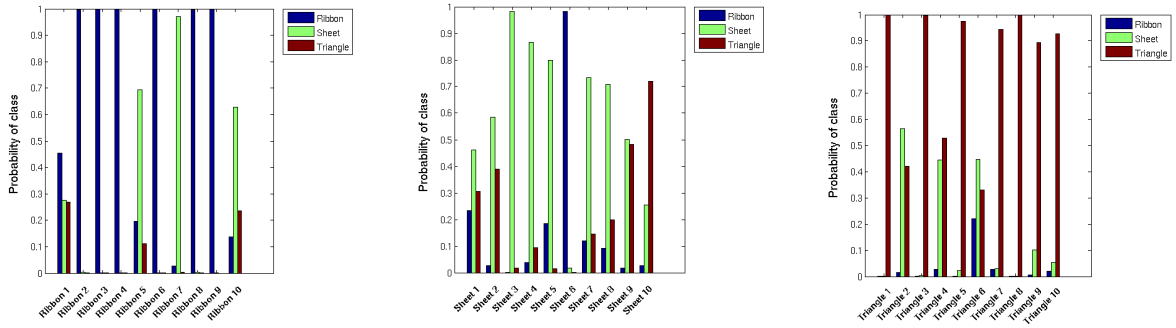

(a) Classification results for 10 ribbons




(b) Classification results for 10 sheets

Figure 3.61: Classification results for two classes

To test the accuracy of the algorithm not only for these two classes, we compared ribbons and sheets against a third class which was chosen to be triangles. The simulated triangles were generated to be isosceles, with their height equal to 1 and their base equal to $\gamma \sim \Gamma(\kappa, \theta)$, with the hyperparameters to be: $\kappa = 60$, $\theta = \frac{2}{3}$. In the next figure we present the classification results for 10 ribbons, 10 sheets and 10 triangles that were generated with 40 points and the standard deviation of the observed noise was $\sigma = 0.8 \times 10^{-2}$. For ribbons, the success rate was 70%, the lowest classification level was 45% whilst the average classification level was 82%. For sheets, the success rate was 80% with the lowest classification level 48% and the average classification level being 76%. For triangles, the success rate was 80% with the lowest classification level 52% and the average classification level 78%. One notices that the algorithm recognises the classes as distinct which is reflected by the high success rates and even higher classification levels.



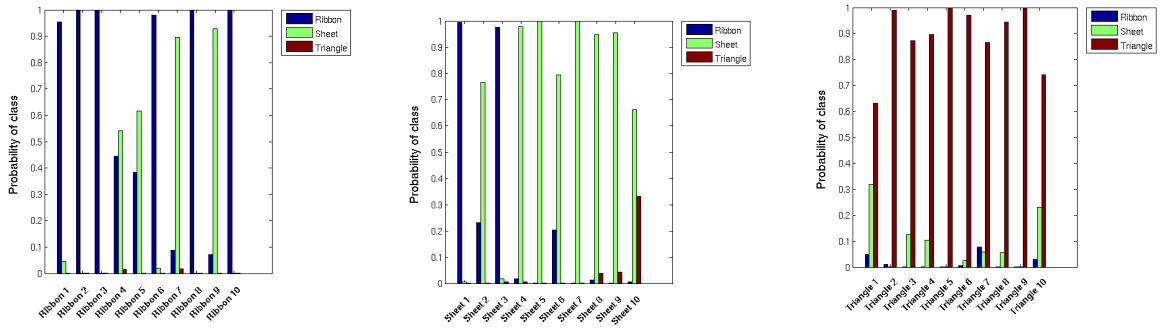

(a) Classification results for 10 ribbons

(b) Classification results for 10 sheets

(c) Classification results for 10 triangles

Figure 3.62: Classification results for three classes

In the next figure we present the classification results for 10 ribbons, 10 sheets and 10 triangles that were generated with 50 points and the standard deviation of the observed noise was $\sigma = 0.6 \times 10^{-2}$. For ribbons, the success rate was 60%, the lowest classification level was 95% whilst the average classification level was 98%. For sheets, the success rate was 80% with the lowest classification level 65% and the average classification level being 88%. For triangles, the success rate was 100% with the lowest classification level 62% and the average classification level 89%.




(a) Classification results for 10 ribbons

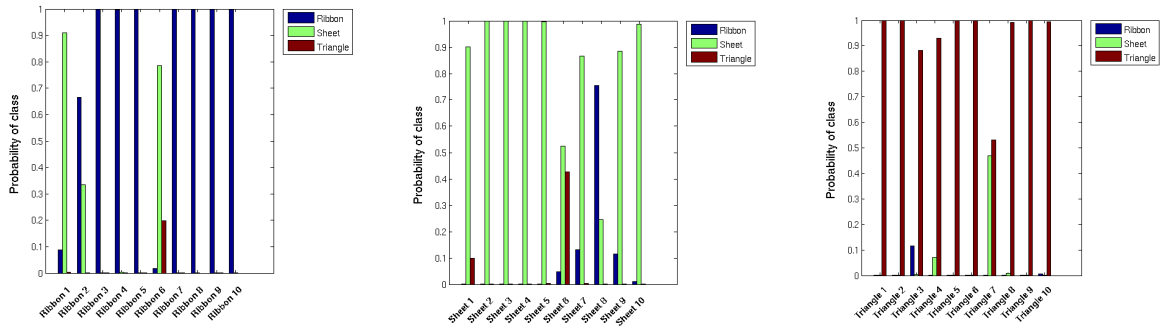
(b) Classification results for 10 sheets

(c) Classification results for 10 triangles

Figure 3.63: Classification results for three classes

In the next figure we present the classification results for 10 ribbons, 10 sheets and 10 triangles that were generated with 70 points and the standard deviation of

the observed noise was $\sigma = 0.8 \times 10^{-2}$. For ribbons, the success rate was 80%, the lowest classification level was 68% whilst the average classification level was 95%. For sheets, the success rate was 90% with the lowest classification level 50% and the average classification level being 90%. For triangles, the success rate was 100% with the lowest classification level 55% and the average classification level 93%.



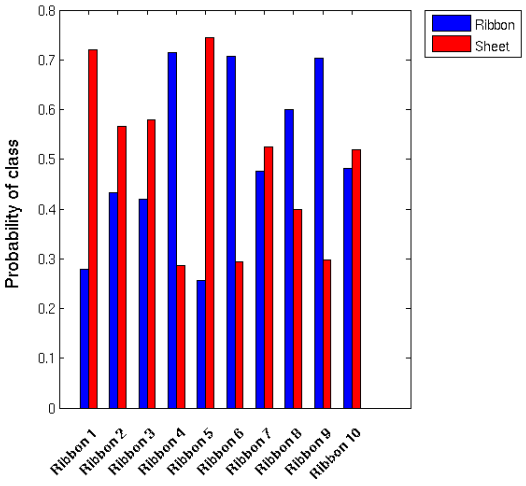

(a) Classification results for 10 ribbons

(b) Classification results for 10 sheets

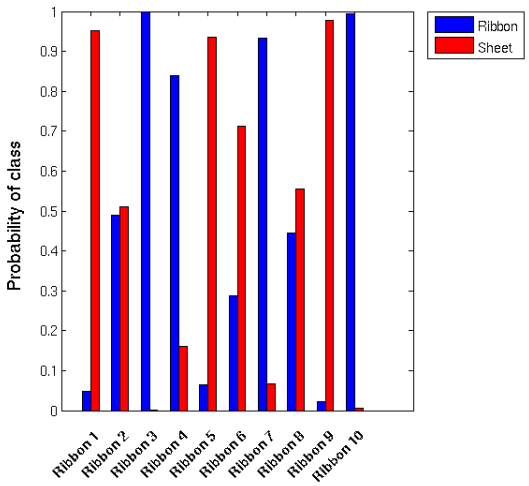
(c) Classification results for 10 triangles

Figure 3.64: Classification results for three classes

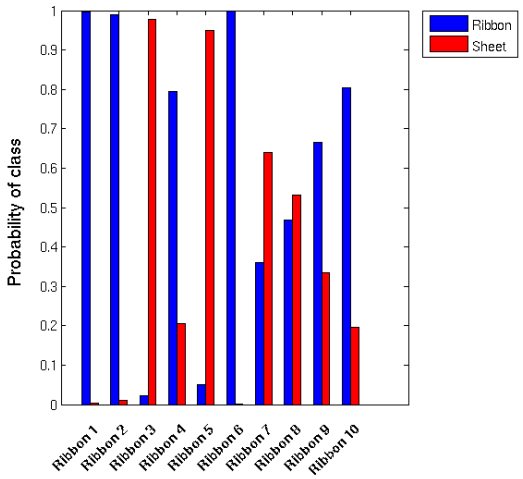
To investigate the classification levels for different numbers of points we classified data shapes from the same class and with stable noise as the number of points increased. The experiment was run 10 times with the number of points ranging from 10 to 100 and the noise being $\sigma = 0.2 \times 10^{-2}$. The following 5 graphs present the results of 5 runs for the class of ribbons. One notices that in the case of 10 and 20 points the 50 and 60 percent of the data shapes are misclassified whereas for 30 points the success rate is 60 percent and the classification levels at least 60 percent. For 50 points the classification levels approach 70 percent and for more than 60 points the confidence is more than 90 percent. The last graph presents the time in hours for a single run as a function of points which shows that for a large number of points the algorithm is computationally expensive.



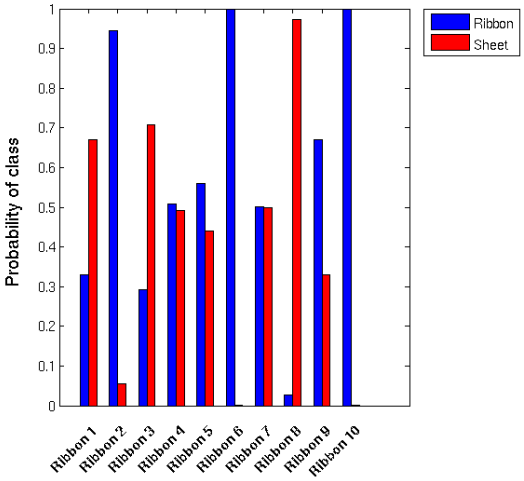

(a) Classification results for 10 points



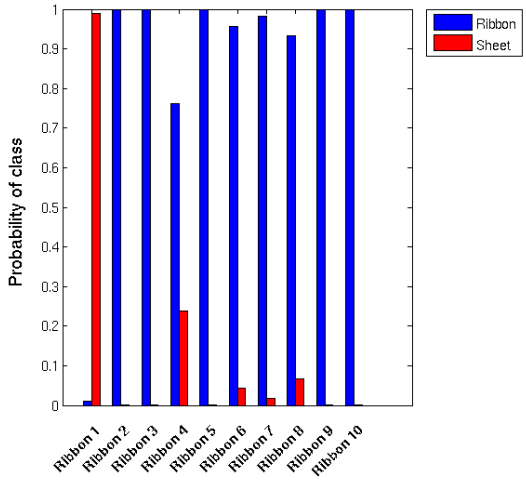

(b) Classification results for 20 points



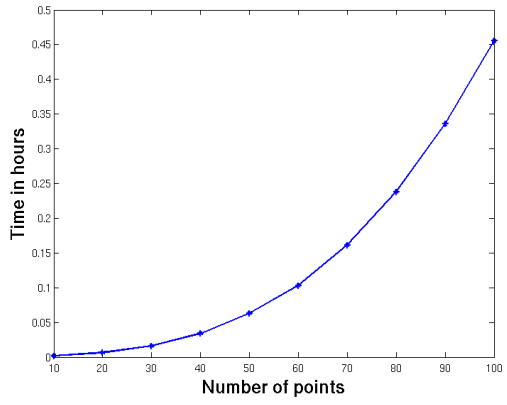

(c) Classification results for 30 points




(d) Classification results for 40 points



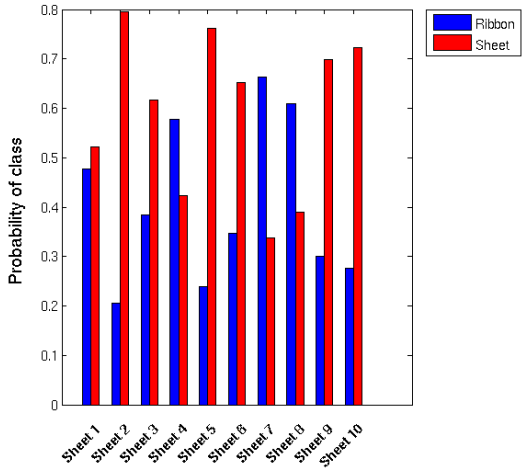

(e) Classification results for 50 points



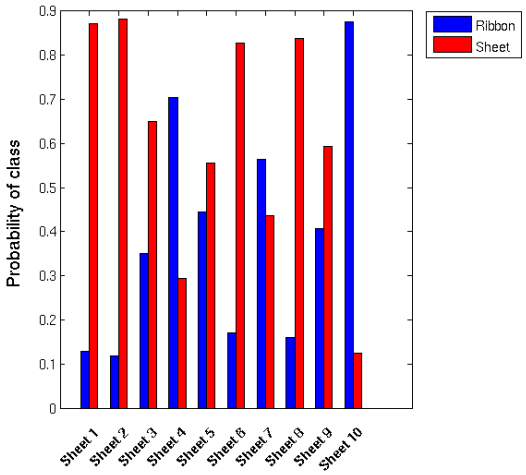

(f) Computational time as a function of the number of points

Figure 3.65: Classification results for 10 different ribbons

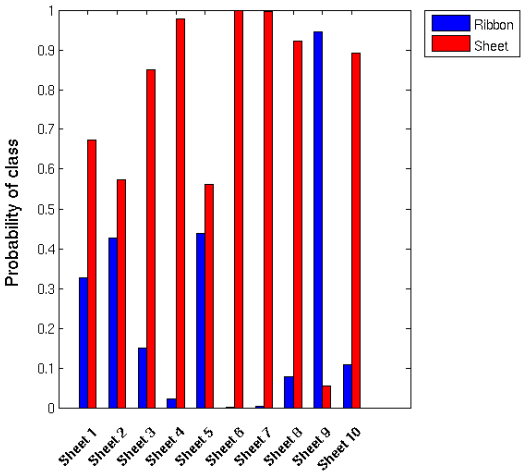
The following experiment was run 10 times with the number of points ranging from 10 to 100 and the noise being $\sigma = 0.2 \times 10^{-2}$. The next 5 graphs present the results of 5 runs for the class of "sheets." One notices that in the case of 10 and 20 points 70 percent of the data shapes are correctly classified whereas for 30 points 90% is correctly classified and for 40 points 70% is correctly classified. For 50 points or more the classification is definite. The last graph presents the time in hours for a single run as a function of points.



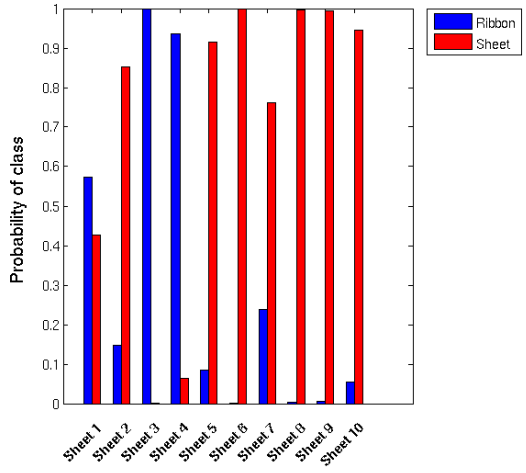

(a) Classification results for 10 points



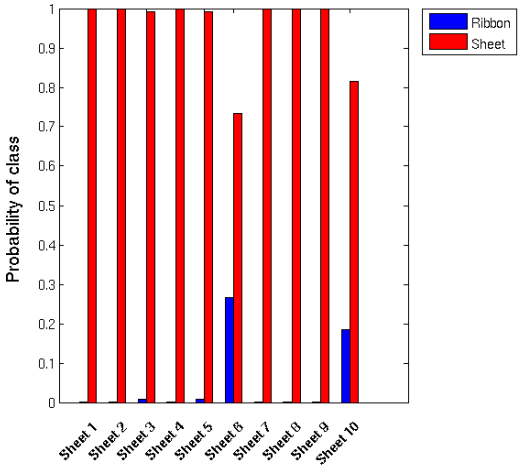

(b) Classification results for 20 points



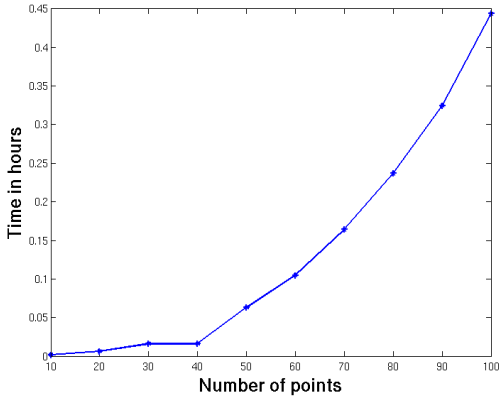

(c) Classification results for 30 points




(d) Classification results for 40 points




(e) Classification results for 50 points




(f) Computational time as a function of the number of points

Figure 3.66: Classification results for 10 different sheets

Overall, we performed 10 runs for 30 ribbons, 30 sheets and 30 triangles each. For the class of ribbons, the average success rate was $67\% \pm 3\%$ and the average classification level was $90\% \pm 2\%$. That means that 7 out of 10 were classified correctly with a classification confidence of 90%. In the case of sheets, the average success rate was $80\% \pm 3\%$ with an average classification confidence of $82\% \pm 2\%$. For triangles, the average success rate was found to be $87\% \pm 3\%$ with the average classification confidence to be $91\% \pm 1\%$. It is also noticeable that when the number of the points is small (10 or 20) the misclassification rate is 50 or 60% but when the number of points increases to 30 or more, then the success rate is more than 60% and the confidence level in most cases is higher than 80 percent. The high success rates and confidence levels, show that our proposed algorithm is a very powerful tool for the classification of geological sand bodies and not only these specific shapes. One of the big advantages of our proposed algorithm is that it captures more geometrical information than other classification methods such as the width-to-thickness ratio. We compare such classification methods with our own classification method in chapter [4].

## 3.6 Concluding remarks

In this chapter we have evaluated the efficiency and efficacy of our proposed algorithm. We have evaluated the success results and the classification levels that our algorithm produces with the help of three different databases.

Firstly, we tested our algorithm with the help of the Kimia database. Since the evaluation of some of the integrals involve Monte Carlo integration we have examined the number of the iterations needed for the stabilisation of the algorithm and we have concluded that 20 iterations for the integration over samplings are sufficient for its stabilisation. Using this number of minimum Monte Carlo iterations we have proceeded to the classification of randomly generated shapes of the Kimia database where we have found that the average classification confidence was $\hat{\mu} = 59\% \pm 7\%$ and the average success rate was $80\% \pm 5\%$. To evaluate the impact of the Gaussian noise on the classification results we tested how the algorithm behaves when the

noise increases. The Gaussian noise was increased by $0.15 \times 10^{-2}$ for each of the data shapes and we noted that the classification levels drop from 25% to 20% as the standard deviation of the added noise increases from $0.2 \times 10^{-2}$ to $1.4 \times 10^{-2}$. As the noise $\sigma$ increases the classification levels drop by 5%. This is a behaviour that one would expect.

Secondly, we have repeated the same experiments with the alphabet database where we have concluded that 20 Monte Carlo iterations over samplings are sufficient for the stabilisation of the algorithm. Classifying randomly simulated letters we have found that the average success rate was $\hat{\mu} = 73\% \pm 6\%$ and the average classification level was $77\% \pm 5\%$. As with the Kimia database, we tested how the algorithm behaves in the case of the letter database when the noise increases. Adding noise in increments of $0.2 \times 10^{-2}$ and increments of $0.5 \times 10^{-2}$ we noticed that the classification results drop from 90% to 77% as the standard deviation of the added noise increases from $0.2 \times 10^{-2}$ to $1.6 \times 10^{-2}$. The classification levels presented in figure (3.23b) drop from 80% to 74% as the standard deviation of the added noise increases from $0.5 \times 10^{-2}$ to $2.5 \times 10^{-2}$.

In the final section of this chapter we have presented the geological sand bodies database. We discussed some of the geological definitions and the existing geological classification schemes. We suggested that our method is more rigorous, quantitative and complete since it encapsulates information that define the geometrical nature of each shape. Having this in mind, we attempted to perform supervised learning and estimate the parameters of a given sand body data set (a simulated one due to the absence of a real one). This required us to maximise our likelihood function over the parameters we wanted to estimate which in turn introduced difficulties. The scores could not be evaluated in a closed form and for this we had to employ an optimisation algorithm (gradient ascent) which due to the unsmoothness of the likelihood surface was trapped to local maxima and could not converge. However, although the results from the learning were unsatisfactory, we performed classification with the known parameters of each class assuming that these would be returned by the gradient ascent in any other case. The classification results returned were very high with the

average success rate of ribbons being $67\% \pm 3\%$ and the average classification level was $90\% \pm 2\%$. For the class of sheets, the average success rate was $80\% \pm 3\%$ with an average classification confidence of $82\% \pm 2\%$. The above results prove that our proposed algorithm is a very powerful tool which is successful more than 80% of the times. One of the reasons for these high results is the fact that our observation model captures the geometrical information of each of the shapes and "explains" the formation of the data themselves in contrast to the current classification methods that are simplistic and don't employ geometrical information per se.

In the next chapter, we evaluate similar methods to the width-to-thickness ratio in the case of the Kimia and the alphabet databases. We compare the classification results returned from these methods to the ones produced by our proposed algorithm.

# Chapter 4

# Experimental results using EM

## 4.1 Introduction

In this chapter we discuss the experimental results acquired when using the Expectation-Maximisation algorithm for both classification and the identification of clusters of shapes in the dataset. The Expectation-Maximisation (EM) algorithm is used as Maximum Likelihood Estimator (MLE) or Maximum a Posteriori (MAP) estimator of the parameters of an underlying distribution from a given data set when the data has missing values or when the data set is incomplete [157, 158]. The EM algorithm is usually used for two main applications. One application is to data sets with missing values which were induced during the observation. The second application is when the optimisation of the likelihood function is intractable but it can be simplified when we assume the existence of latent variables which without we have an incomplete data set.

As a Maximum Likelihood Estimator, the EM algorithm has quite broad applications but the most widely used is the mixture-density parameter estimation problem. The most common mixture-density that EM is applied to is the Gaussian mixture models (GMM). GMM are superpositions i.e. linear combinations of a number of Gaussian distributions with adjusted means and covariances as well as mixing coefficients. In section [4.1.1] we present the derivation of the EM in the case of GMM and in section [4.2] we present a way of using the EM algorithm when

irregularities such as singularities are present in the data. In section [4.3] we discuss classification results of the Kimia and the alphabet database using the EM algorithm. In particular, we use feature vectors of the data to infer whether there exists clustering based on the features and based on these results we classify new data. This is a way to compare the results given by the classification algorithm presented in chapter [2]. In section [4.5] we present an adaptation of the EM algorithm in the case of the sand body database. For this adaptation instead of using a mixture of Gaussians we utilise a mixture of the observation model we presented in chapter [2]. Finally, section [4.6] discusses the concluding remarks of this chapter.

### 4.1.1 Derivation of EM algorithm for Gaussian mixtures

We assume that we are given $K$ multivariate Gaussian distributions $N(\mu_k, \Sigma_k)$ with $k = 1, ..., K$. Then the linear combination of $K$ Gaussians can be formulated as probabilistic models known as mixture distributions [159]. A mixture of Gaussians is a distribution that draws with probability $\pi_k$ from the $k$-th component and is given by

$$f(y|\Theta) = \sum_{k=1}^{K} \pi_k N(y|\mu_k, \Sigma_k) \tag{4.1.1}$$

where $\Theta = (\pi_1, ..., \pi_{K-1}, \mu_1, .., \mu_K, \Sigma_1, ..., \Sigma_K)$ is the vector of parameters we would like to estimate. Here, each of the $y_i$ is a $n \times D$ vector, each of the $\mu_k$ is $D \times 1$ vector and each of the $\Sigma_k$ is a $D \times D$ matrix. Also, $N(y|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_k)^T \Sigma^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_k)\right\}$ which is the normal probability distribution with mean $\boldsymbol{\mu}_k$ and covariance $\Sigma_k$ evaluated at $y$.

Given some data (in our case data shapes) $y_i$, with $i = 1, ..., n$, we wish to obtain an estimator $\hat{\Theta}$ of the parameters $\Theta$. The values of the estimators can be found with the help of the EM algorithm. These estimates are based on some starting values $\Theta_0$ which are found by the EM when it alternates between the E-step and the M-step until convergence is reached. We now describe the derivation of the EM algorithm for the case of Gaussian mixture models.

### 4.1.2 Complete likelihood

Given some data shapes $y_i$, with $i = 1, ..., n$, we wish to obtain an estimator $\hat{\Theta}$ of the parameters $\Theta = (\pi_1, ..., \pi_{K-1}, \mu_1, .., \mu_k, \Sigma_1, ..., \Sigma_k)$. Let $G$ be a random vector which draws a class $k \in \{1...K\}$. We know that $\mathbb{P}(G = k) = \pi_k$. We here denote:

$$f_{ik} = \mathbb{P}(y|G = k) = N(y|\mu_k, \Sigma_k)$$

We also know that the joint probability of $y$ and $G$ is:

$$\mathbb{P}(y_i, G = k) = \mathbb{P}(y_i|G = k)\mathbb{P}(G = k) = f_{ik}\pi_k \tag{4.1.2}$$

We now assume that for an observation $y_i$, the value of $G$ is known so that we know in which of the $K$ components, the $i$-th observation belongs to. To express this knowledge we utilise an indicator variable:

$$G_{ik} = \begin{cases} 1 & \text{if observation } i \text{ is belongs to component k,} \\ 0 & \text{otherwise.} \end{cases}$$

By combining all the above, the complete data $(y_i, G_{i1}, ..., G_{iK})$ is given by:

$$\mathbb{P}(y_i, G_{i1}, ..., G_{iK}) = \prod_{k=1}^{K} (f_{ik}\pi_k)^{G_{ik}} \tag{4.1.3}$$

We can then form the corresponding likelihood function, which is also called the complete likelihood in the following way:

$$\mathcal{L}(\Theta|y_1, ..., y_n) = \prod_{i=1}^{n} \prod_{k=1}^{K} (\pi_k f_{ik})^{G_{ik}} \tag{4.1.4}$$

The log-likelihood is thus formed as:

$$\mathbb{L} = \log(\mathcal{L}(\Theta|y_1, ..., y_n)) = \sum_i \sum_k (G_{ik} \log \pi_k + G_{ik} \log f_{ik}) \tag{4.1.5}$$

As mentioned previously, to estimate the parameter described by $\Theta$ the EM algorithm alternates between the E-step and the M-step. We now evaluate both steps particularly for the case of Gaussian mixtures.

### E-step

One notices that in expression (4.1.5), the values of $G_{ik}$ are unknown; this is a condition required by the formation of the complete likelihood. We can thus replace them by their conditional expectations so that:

$$W_{ik} \equiv \mathbb{E}(G_{ik}|y_i) = \mathbb{P}(G_{ik} = 1|y_i) = \mathbb{P}(G = k|y_i) \tag{4.1.6}$$

Then by using Bayes' theorem one has:

$$W_{ik} = \mathbb{P}(G = k|y_i) = \frac{\mathbb{P}(G = k)\mathbb{P}(y_i|G = k)}{\sum_l \mathbb{P}(G = l)\mathbb{P}(y_i|G = l)} = \frac{\pi_k f_{ik}}{\sum_l \pi_l f_{il}} \tag{4.1.7}$$

which are the membership probabilities
$W_{ik} = \mathbb{P}(\text{observation } i \text{ belongs to component } k)$ or as they are called in the Bayesian framework the responsibilities i.e. the responsibility that the $k$-th component takes for explaining the observation $y_i$). Substituting this back to the log-likelihood (4.1.5) one has:

$$\mathbb{L} = \sum_i \sum_k (W_{ik} \log \pi_k + W_{ik} \log f_{ik}) \tag{4.1.8}$$

which is the expression that we will maximise over the parameters that we want to estimate.

### M-step

Having the responsibilities evaluated from the E-step we can now obtain the estimates of the parameters $\Theta = \{\mu_j, \sigma_j, \pi_j\}$. This can be done by setting the derivatives of the log-likelihood (4.1.8) with respect to the parameters equal to zero. We now present these derivatives for the case of Gaussian mixtures. We will firstly maximise (4.1.8) with respect to the mixing coefficients $\pi_k$. Here, one must take into account that the mixing coefficients $\pi_k$ are subject to the constraint $\sum_{k=1}^{k=K} \pi_k = 1$ which requires them to sum up to one. This can be achieved by using a Lagrange multiplier and by maximising the following:

$$\begin{aligned}\frac{\partial}{\partial\ \pi_j}\left[\mathbb{L} - \lambda\sum_k \pi_k + \lambda\right] &= 0\\ \frac{\partial}{\partial\ \pi_j}\mathbb{L} - \lambda\sum_k \delta_{kj} &= 0\\ \frac{\partial}{\partial\ \pi_j}\mathbb{L} - \lambda &= 0\end{aligned} \tag{4.1.9}$$

The above instructs us to calculate the derivative of the log-likelihood with respect to $\pi_j$:

$$\begin{aligned}&\frac{d}{d\ \pi_j}\left[\sum_{i,k} W_{ik}\log\pi_k + W_{ik}\log f_{ik}\right] =\\ &= \sum_{i,k} W_{ik}\ \frac{d}{d\ \pi_j}\ \log\pi_k + (\log\pi_k + \log f_{ik})\ \frac{d}{d\ \pi_j}\ W_{ik}\\ &= \sum_{i,k} W_{ik}\ \frac{d}{d\ \pi_j}\ \log\pi_k\end{aligned} \tag{4.1.10}$$

We should note here, that the derivative of the responsibilities $W_{ik}$ with respect to any of the parameters is zero since $W_{ik}$ is the conditional expectation of the class labels that we acquired from the E-step and hence it is a constant. Thus, the derivative of expression (4.1.10) differentiates to:

$$\sum_{i,k} W_{ik}\frac{d}{d\ \pi_j}\log\pi_k = \sum_{i,k}\frac{W_{ik}}{\pi_k}\delta_{kj} = \sum_i \frac{W_{ij}}{\pi_j} \tag{4.1.11}$$

so overall the maximisation over the mixing coefficients using the Lagrange multiplier of expression (4.1.9) is:

$$\begin{aligned}\sum_i \frac{W_{ij}}{\pi_j} - \lambda &= 0\\ \sum_i W_{ij} &= \lambda\pi_j\\ \sum_{i,j} W_{ij} &= \lambda\\ \Rightarrow \lambda &= n\end{aligned} \tag{4.1.12}$$

Finally, substituting this back into the top line the estimated mixing coefficients are:

$$\boxed{\hat{\pi}_j = \frac{1}{n}\sum_i W_{ij}}$$

which is the average posterior probability for component $j$.

The derivation of the estimators for $\mu_k$ and $\Sigma_k$ are quite lengthy and complicated so we only present the final results. For their explicit derivation one can refer to [158]. The estimators for the remaining parameters are thus:

$$\boxed{\hat{\mu}_k = \frac{\sum_{i=1}^{n} W_{ik} y_i}{\sum_{i=1}^{n} W_{ik}}}$$

$$\boxed{\hat{\Sigma}_k = \frac{1}{\sum_{i=1}^{n} W_{ik}} \sum_{i=1}^{n} \sum_{k=1}^{K} W_{ik}(\boldsymbol{y}_i - \boldsymbol{\mu}_k)(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T}$$

The iterations between the E-step and the M-step are continuing until convergence is reached which was proven in [157, 160]. It is worth it here, emphasising some of the problems one encounters by the use of the EM algorithm associated with the maximisation of the parameters in the Gaussian mixture case. To illustrate the point, consider a Gaussian mixture with all components having equal covariance matrices $\Sigma_k = \sigma_k^2 \mathbf{I}$, with $\mathbf{I}$ the identity matrix. Assuming that one of the components, say the $i$-th component, has its mean equal to one of the data points i.e. $\mu_i = y_n$ for some value of $n$, then this data point contributes to the likelihood:

$$N(y_n|\mu_i, \sigma_i^2\mathbf{I}) = \frac{1}{(2\pi)^{1/2}\frac{1}{\sigma_i^2}} \tag{4.1.13}$$

Taking the limit $\sigma_i \to 0$ then this term tends to infinity which causes the likelihood to diverge. One could say that the maximisation of the log likelihood function is an ill posed problem because such singularities are unavoidable whenever one or more of the Gaussian components collapses to a single point. There are several ways of alleviating the problem. One example is to use certain heuristics e.g. detecting the singularities and resetting the mean of the Gaussian at a random value and resetting the covariance to some big value. In the Bayesian framework the problem

is alleviated by performing MAP and hence including a prior distribution over the components [158]. However, this is not aligned with the philosophy of the EM algorithm, a purely Maximum Likelihood Estimator algorithm. A different way to treat singularities is the nonparametric maximum likelihood (NPML) which utilises the help of an Aitchinson-Aitken kernel. We discuss this approach in the next section since it is the approach taken for the experimental results of the current chapter.

## 4.2 A version of EM-based NPML

Nonparametric maximum likelihood is a tool usually used in the case of fitting generalised linear models with random effects. The term nonparametric refers to the case where there is no parametric specification of the random effect distribution. In NPML usually the marginal likelihood can be approximated by a finite mixture model of which the model parameters can be calculated by the EM algorithm.

We employ the use of a version of the EM-based NPML [161, 162] to avoid the so called likelihood spikes [158, 163] which are caused by the Gaussian components collapsing to a single point and in turn cause the likelihood to diverge. Likelihood spikes is a common phenomenon when unequal variances are used for the Gaussian mixtures. This allows the components to have independent variances which can freely vary at any values. The problem can be modified with the use of a smoothing component for the mixtures' variances which employs a discrete kernel [164] which we now describe.

Suppose we want to fit a Gaussian mixture with unequal variances $\sigma_k^2$ with $k = 1, ..., K$ and we want to employ the smoothing of the components. The smoothing is performed by the following discrete kernel:

$$w(x, y|\lambda) = \begin{cases} \lambda & \text{if } y = x, \\ \frac{1-\lambda}{K-1} & \text{if } y \neq x. \end{cases} \tag{4.2.14}$$

with $1/K \leq \lambda \leq 1$. Here, $x$ and $y$ denote the class memberships i.e. the component index and thus range from 1 to $K$. The kernel assigns the smoothing parameter

equal to $\lambda$ when the running index equals the component index. When the indices are unequal then the kernel assigns the smoothing parameter equal to $\frac{1-\lambda}{K-1}$. Setting the smoothing parameter $\lambda = 1/K$ corresponds to the maximum smoothing possible which is equivalent to the case of equal variances. When $\lambda = 1$ then all the variances are decoupled and calculated within the components so that all components have independent variances.

The EM algorithm is implemented as in any other case; it alternates between the E-step and the M-step by evaluating the estimators for the parameters $\pi_k, \mu_k, \sigma_k$. The only difference is the kernel which is used to update the estimated variances which are set equal to:

$$\Sigma^{new} = w(x, y|\lambda)\ \Sigma^{old} \tag{4.2.15}$$

$$\begin{pmatrix} \Sigma_1 \\ \Sigma_2 \\ \vdots \\ \Sigma_k \end{pmatrix}^{new} = w(x, y|\lambda) \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \\ \vdots \\ \Sigma_k \end{pmatrix}^{old} \tag{4.2.16}$$

with $\Sigma^{new}$ the adapted vector of the variances and $\Sigma^{old}$ the variances as estimated by the M-step and presented in section [4.1.2]. The above expressions show that in order to avoid the likelihood spikes, the adapted variances are a linear combination of all the component variances as estimated in the M-step. When $\lambda \approx 1$ then this adds to each variance a small $\epsilon$-correction, with $\epsilon \sim \frac{1-\lambda}{K-1}$, and forces the components to be dependent. The use of the kernel imposes this inter-component connection that is needed so that if one of the variances is close to singular then it is forced to artificially "increase" by a small amount and avoid the divergence of the likelihood.

It is worth mentioning here that these singularities provide another example of the severe over-fitting that can occur in a maximum likelihood approach. One can argue that these singularities do not occur in a Bayesian approach. In this approach, the EM algorithm is used to find the Maximum a Posteriori estimate instead of the maximum likelihood for the model with a prior $\mathbb{P}(\Sigma)$ defined over the variance. In

this case, the E-step remains the same whereas the difference is on the M-step which maximises a slightly different quantity. Suitable choices for the particular prior will remove such pathological situations. However, as we mentioned previously, this is not aligned with the philosophy of the EM algorithm, a purely Maximum Likelihood Estimator algorithm.

In the next two sections we describe experimental results that were acquired by the use of the EM algorithm for the Kimia and the alphabet database. Due to the variations of the different existent classes in our data, in our first attempts of running the EM algorithm for these two databases we encountered problems. Such singularities were present in our data and the likelihood was divergent. For this reason, for all the following results we employed the EM-based NPML which allowed us to alleviate the divergences and conclude with our classification results.

## 4.3 Adaptation of EM on the Kimia and alphabet database

As mentioned in chapter[1], there are classification schemes that are based on special features of the shape. In this case, the characterisation of a class of shapes and its differentiation from other classes can be done in terms of some of its properties (also called features or shape descriptors). Shape descriptors are used as a measure of similarity between shapes represented by their features. Usually simple geometrical features such as area and perimeter are used to describe the shapes however such features usually fail to describe shapes with small differences. In other words there can be more than one classes that can be described by the same features. Our proposed classification algorithm from chapter [2] classifies purely based on the geometrical properties of the observed shapes. In the next section, we discuss how we used the EM algorithm to infer the existence of clusters of data based on their features. We estimate their properties and then perform classification based on these. Finally, we compare the results of this classification procedure to the ones acquired by the algorithm presented in chapter [2].

### 4.3.1 EM on Kimia database

In this section we present the results obtained by the EM algorithm for the Kimia database. For this task, we assume that we have at our disposal two data sets: a training and a test data set. We assume that each shape of either of the two data bases can be described by a feature vector of some extracted shape descriptors. We assume that the description of each of the classes is sufficient by these feature vectors. We also assume that each of the classes' features represents a cluster or that classes with similar features will be combined in a single cluster. Each of the clusters can thus be sufficiently represented as a multidimensional Gaussian distribution of which the mean and the covariance matrix we will learn by using the EM algorithm. We can then use this information to classify new, unobserved data from the test data set in the inferred classes.

For this task, we assumed that the training data set is comprised by the idealised example shapes of the Kimia database. For the generation of the example shapes we extracted the boundaries in the way described in section [3.2.1]. The test data for this task were generated as follows: a random number of Kimia data shapes were generated in the way described in section [3.2.2]. For all the data shapes the feature properties of shape factor, roundness, convexity and solidity were extracted. These features were used as the new, unobserved values based on which the data shapes would be classified. The class labels associated to each of the shapes were retained to enable us to evaluate the results of the EM algorithm by a comparison to its class assignments.

The feature extraction for both the training and the test data set was done in the following way: for all 256 Kimia shapes we extracted the perimeter, area, convex hull perimeter, convex hull area, shape factor, roundness, convexity and solidity. In particular, the convex hull of a shape is defined as the smallest convex set that contains the shape i.e. a minimum bounding polygon. The shape factor is defined as $4\pi\frac{\text{Area}}{\text{perimeter}^2}$, the roundness is defined as $4\pi\frac{\text{Area}}{\text{convex perimeter}^2}$, the convexity as $\frac{\text{convex perimeter}}{\text{perimeter}}$, and the solidity as $\frac{\text{area}}{\text{convex area}}$ [165, 166]. The extracted features of each shape form its feature vector and the collection of all feature vectors can

be used for the inference of existent cluster. However, the final feature vector was only comprised by the shape factor, roundness, convexity and solidity since the other properties differ in dimensionality whereas the chosen four properties are all dimensionless. Since each data shape is represented by these four properties, we assume that each class of shapes can be represented by a four-dimensional Gaussian which is characterised by a certain mean and covariance matrix. The EM algorithm was used to estimate the values of the mean and the covariance matrix of each of the clusters.

One of the drawbacks of the EM algorithm is the fact that the number of components $K$ must be known a priori. Since we have a priori knowledge of the number of classes present in the data, for the first run we chose the number of components to be the same as the number of the existing classes which is $K = 21$; in the Bayesian framework this would be expressed as imposing a delta function as a prior on the number of the present classes namely $\delta_{K,21}$. The algorithm was allowed to run until convergence with an appropriate threshold and was found to be maximised for $\lambda = 0.941$ (see equations (4.2.14) and (4.2.16)). Since the data were simulated, it is possible for us to check how well the EM algorithm worked. One would expect, that the distinct types of shape would lead to unique clusters. For example we would expect all "hands" to form a single cluster and to be differentiated from "tools."

To evaluate the efficiency of the clustering produced by the algorithm we examined the final values of the $W_{ik}$. We split the $W_{ik}$ in parts so as to compare the responsibilities of the shapes on a class by class basis. For each of the 21 classes of simulated data we evaluated the mode assignment determined by the maxima of the $W_{jk}$ for those $j$ associated to the given class. This mode was then taken as the class label for that cluster. For example Table [4.1] shows some partial data highlighting how the assignment was compared to the data label. Specifically, all of the shapes in the top two rows were known to have been simulated from the class of spectacles. The most frequent value of $k$ which maximised the responsibilities of these shapes was $k = 8$. However, as can be seen not all of the shapes in this subset were labelled in class 8 so we then evaluated the percentage of shapes classified in

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Data label** | 11 | 8 | 8 | 8 | 15 | 11 | 11 | 11 | 11 | 8 | 8 | 8 | 11 | 8... |
| **Assignment** | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8... |
| **Data label** | 6 | 6 | 6 | 6 | 6 | 10 | 10 | 10 | 10 | 6 | 10 | 6 | 6 | 4... |
| **Assignment** | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6... |
| **Data label** | 13 | 13 | 13 | 13 | 4 | 4 | 4 | 9 | 13 | 4 | 4 | 14 | 14 | 14... |
| **Assignment** | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13... |

Table 4.1: The MAP assignment of data shapes, $y_i$, into classes, determined by the maximum over $k$ of $W_{ik}$. Each row represents a subset of the shapes belonging to a single class (Spectacles, Tools, Hands). The assignment label is fixed by the mode over the data labels in each class.

each subset mode. Across all shapes, this was found to be 65 percent; this means that on average, 65 percent of the shapes were classified into the mode of the class memberships.

Having the estimators of the mixing coefficients, the mean and the covariance matrix that the EM returned for each of the clusters we then classified new, unobserved data which were obtained from the test data set. This was an important task since it allows us to examine how well the EM-based approach can classify new shapes at the end of its learning period. For this we generated 1000 data shapes, in the way described in section [3.2.2], coming from random classes of the Kimia database and extracted the four features for each one of them. For each of the 1000 shapes we performed MAP to evaluate in which cluster it is classified. This was then compared to the class labels supplied by the EM algorithm in its determination of clusters. For example, the shapes on the top row of Table [4.1] were spectacles; the EM algorithm assigned most of the data from this class into class 8; a correct classification of future data shapes generated from the spectacles class would be into class 8. In a way similar to the MAP performed in chapter [3] we evaluated:

$$\mathbb{P}(C_i|y) \propto \mathbb{P}(y|C)\mathbb{P}(C) \tag{4.3.17}$$

where in this case the observation model is a four dimensional Gaussian distribution. The average success rate of this classification procedure is: $62.9\% \pm 4.1\%$. However, this approach did uncover a problem. Rather than providing us with 21 unique class assignments, the EM algorithm returned class modes totalling only 13. To investigate further we repeated the experiment with the EM initiated to $K = 13$ components. Repeating the same process as before, the percentage of shapes classified in the subset modes was found to be 76 percent which shows an increase of the result found for 21 components. We then performed classification of 1000 data shapes which were randomly generated from the classes of the Kimia database. Using MAP classification for each of the data shapes we found that the average success rate for this classification was $72.7\% \pm 4.5\%$. Although this increases the success (at least as measured by the procedure outlined above) of the algorithm, it is not in agreement with our expectation of 21 distinct classes. We will return later on to explore whether our more geometric approach can offer a better outcome.

### 4.3.2 EM on alphabet database

For completion, we repeated the above experiment for the alphabet database we first introduced in chapter [3]. As with the Kimia database, for all 156 letters we extracted the feature vectors comprising of the shape factor, roundness, convexity and solidity. Having a priori knowledge that the existent classes of letters are 26, we imposed a delta prior on the number of classes $\delta_{K,26}$ and we run the EM algorithm with $K = 26$ components. To evaluate the efficiency of the clustering that EM returned, we evaluated the class modes as in the Kimia database. The percentage of shapes classified in their subset mode was found to be 55 percent. We then evaluated the success rate of the returned clustering. We generated 1000 data shapes coming from random letter classes and performed MAP. The success rate of the classification was $33.1\% \pm 5.6\%$. However, as in the case of the Kimia database, the class memberships returned from the EM algorithm were not all unique. We repeated the experiment with the number of components equal to the number of unique class modes determined by the EM algorithm which was initiated with $K = 17$ components. Repeating the same process as before, the percentage of shapes

classified in the subset mode was found to be 60 percent which shows a slight increase of the previous result with 26 components. We then classified 1000 data shapes which were randomly generated from all letter classes. Classifying through the MAP procedure we found that the average success rate for this classification was $37.3\% \pm 5.4\%$.

### 4.3.3 Discussion of the results

The purpose of introducing the model which is the subject of this thesis is that we had hoped it would yield a more accurate classifier. We must therefore compare the results found above – which use older and more established techniques – with the classification rate achieved using our new technique which we presented in chapter [3]. First we consider the Kimia database. The success rate of 80% that was produced with our method is slightly better than the 62.9% and 72.7% noted above. The numbers are comparable (the $K = 13$ result is within experimental error of the result arising with our new approach) however we must recall the approach developed above did not lead to an EM algorithm which correctly separates the training data into the correct number of distinct classes. This is a huge drawback of the an EM-based approach based on the information non-preserving [35] feature extraction. Similarly, in the case of the alphabet data set, our work in chapter [3] led to a success rate of 73% which is indeed much better than the rates of 33% and 37% found with $K = 26$ components and $K = 17$ components respectively.

In contrast to the classification based on our geometrical method presented in chapter [2], there is big difference between the success rates of the Kimia database and the letter database when using the EM approach. We suggest that such behaviour can be understood as signaling that the features extracted for the letters show far greater similarity than the shapes of the Kimia database. Indeed, many of the alphabet letters are fairly similar (for example C and G or B and D) whereas the Kimia shapes show much greater diversity. This goes somewhere to explaining why fewer mixtures than expected were present when the algorithm had decided on its clusters.

It is probably not too surprising that the approach used above is not able to separate the training data into the classes that we anticipated, since much of the geometrical information is being lost. No distinguishing data related to the shapes' boundaries or curvature is being used in either the EM algorithm or in classification. Instead these data are being diluted as they are combined into the chosen features. The approach of chapter [3] prefers to retain much more of this information which then plays an active role in building the likelihood function. Furthermore, our model easily encompass shapes which are generated from parameters drawn from a given distribution, such as our description of the aspect ratios of sand bodies which follow a $\Gamma$-distribution.

To investigate the applicability of our proposed model further it seems appropriate to develop it in the context of unsupervised learning. In this way we hope to find out whether or not our new marginalised likelihood can be used to discover clusters, to learn their parameters and eventually to classify new data into these classes. For this reason we now turn to the development of an EM procedure based upon our new likelihood. Our hope is that with this approach, the algorithm will converge to parameter estimations which separate out the training data into the expected clusters and eventually be used to recognise differences between real world shapes (such as, for example, geological sand bodies).

## 4.4 Adaptation of the EM for sand bodies

The adapted EM algorithm is an attempt to find clusters in the data by using our observation model of section [2.6] as the mixture model. In any other case, one could use a mixture of Gaussian or Gamma distributions to find the clusters that describe the data by their properties, for example the mean and the covariance matrix of the Gaussian distributions or the shape and scale parameter of the Gamma distributions. Clusters can be identified by the properties that described them and then new observations can be classified to their respective clusters according to their properties. The question we are faced with is: can we describe clusters by their underlying geometry and then classify new observations according to how similar they

are to the clusters? The answer to that is that one would have to use a mixture of distributions that reflects the belief that clusters can be identified geometrically. For this reason, we assume that clusters of shapes can be described by their underlying geometry and the identification of them will be done with a mixture of distributions that describe the likelihood of a certain shape belonging to a particular class.

We assume that we are given $K$ distributions which in our case can be seen as the assumed existent classes in which the data shapes can be partitioned to. Each of the distributions can be described by the likelihood $\mathbb{P}(y|C_k)$ and as we have seen in the case of the sand bodies each sand body class can be described by its own parameters $\{\kappa, \theta\}$ that define its aspect ratio. A finite mixture is a distribution which draws with probability $\pi_k$ from the $k$-th likelihood distribution. The density of a finite likelihood mixture is given by:

$$\mathbb{P}(\boldsymbol{y}|\Theta) = \sum_{k=1}^{K} \pi_k \mathbb{P}(\boldsymbol{y}|C_k) \tag{4.4.18}$$

with $k = 1, ..., K$ the number of mixture components, $\Theta = \{\pi_1, ..., \pi_{K-1}$ $, \kappa_1, ..., \kappa_K, \theta_1, ..., \theta_K\}$ the vector of the parameters. Since we want to investigate the clustering of the classes based on their underlying geometry, the likelihood $\mathbb{P}(y|C_k)$ is the marginalised likelihood presented in expression (3.2.1) which has been marginalised over the nuisance parameters and the similarity transformations have been integrated out:

$$\begin{aligned}\mathbb{P}(y|C_k) &= \sum_{b\in\mathcal{B}} \int \mathscr{D}\beta \; \mathscr{D}s \; \mathscr{D}g \; \mathscr{D}\sigma \; \mathbb{P}(y|b,\beta,s,g,\sigma)\mathbb{P}(b)\mathbb{P}(s)\mathbb{P}(g)\mathbb{P}(\sigma)\mathbb{P}(\beta|C_k) \\ &= \sum_{b} \int \mathscr{D}\gamma \; \mathscr{D}s \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \Gamma(\gamma;\kappa_k,\theta_k)\mathbb{P}(s)\end{aligned} \tag{4.4.19}$$

which is the probability that a given data shape $y$ comes from a class which can be described by its aspect ratio $\gamma$ and its parameters $\{\kappa_k, \theta_k\}$ which is captured by the prior distribution $\Gamma(\gamma; \kappa_k, \theta_k)$. One should note here that $\pi_K = 1 - \sum_{k=1}^{K-1} \pi_k$. In the next section, we discuss the derivation of the EM algorithm in the case of the finite likelihood mixture.

## 4.5 Derivation of EM algorithm for finite likelihood mixture

To obtain an estimator $\hat{\Theta}$ of the parameters $\Theta$ we use the EM algorithm. The EM algorithm alternates between the E-step and the M-step until convergence is reached. We now describe the derivation of the EM algorithm for the finite likelihood mixture.

### 4.5.1 Complete likelihood

In this section, we construct the complete likelihood for our mixture model as we did in section (4.1.2). The derivation of the EM algorithm for the E-step of our adapted version is the same as in the Gaussian mixture case so for its calculation one can refer to (4.1.2). However, the difference is in the model we employ so we will here denote:

$$\begin{aligned} f_{ik} = \mathbb{P}(y|C_k) &= \sum_{b\epsilon\mathcal{B}} \int \mathscr{D}\beta \; \mathscr{D}s \; \mathscr{D}g \; \mathscr{D}\sigma \; \mathbb{P}(y|b,\beta,s,g,\sigma)\mathbb{P}(b)\mathbb{P}(s)\mathbb{P}(g)\mathbb{P}(\sigma)\mathbb{P}(\beta|C_k) \\ &= \sum_{b} \int \mathscr{D}\gamma \; \mathscr{D}s \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \Gamma(\gamma;\kappa_k,\theta_k)\mathbb{P}(s) \end{aligned} \tag{4.5.20}$$

In our adapted version of EM, the difference lies in derivation of the M-step which we present in the next section. Before that, we remind the readers that the expression that needs to be maximised in the M-step is:

$$\mathbb{L} = \sum_i \sum_k W_{ik} \log \pi_k + W_{ik} \log f_{ik} \tag{4.5.21}$$

**M-step**

Having the responsibilities $W_{ik}$ evaluated from the E-step we can now obtain the estimates of the parameters $\Theta = \{\kappa_j, \theta_j, \pi_j\}$. This can be done by setting the derivatives of the log-likelihood (4.5.21) with respect to the parameters equal to zero. We

firstly maximise (4.5.21) with respect to the mixing coefficients $\pi_k$. Since the maximisation of these coefficients is independent of $f_{ik}$, the result is exactly the same as in the case of Gaussian mixtures which we present here:

$$\boxed{\hat{\pi}_j = \frac{1}{n}\sum_i W_{ij}}$$

To obtain the estimators for $\{\kappa_j, \theta_j\}$ we set the derivatives of the log-likelihood (4.5.21) with respect to these parameters equal to zero. We start with the derivative of the log-likelihood with respect to $\kappa_j$:

$$\begin{aligned}\frac{\partial}{\partial\ \kappa_j}\mathbb{L} &= \sum_i \sum_k (\log \pi_k + \log f_{ik}) \frac{\partial}{\partial\ \kappa_j} W_{ik} + W_{ik} \frac{\partial}{\partial\ \kappa_j} \log f_{ik} \\ &= \sum_i \sum_k W_{ik} \frac{\partial}{\partial\ \kappa_j} \log f_{ik} \\ &= \sum_i \sum_k W_{ik} \frac{\frac{\partial}{\partial\ \kappa_j} f_{ik}}{f_{ik}} \end{aligned} \tag{4.5.22}$$

We should again note here, that the derivative of the responsibilities $W_{ik}$ with respect to any of the parameters is zero since $W_{ik}$ is the conditional expectation of the class labels that we acquired from the E-step and hence it is considered to be constant. One notices that in the above calculation the term that gets differentiated with respect to $\kappa_j$ is the observation model $f_{ik}$. We will now examine the evaluation of this derivative and then substitute it back to expression (4.5.22). As we mentioned in chapter [3], the planar curves of the sand bodies are specified by their aspect ratios so for computational reasons we substitute: $\mathscr{D}\beta\ \mathbb{P}(\beta) \to d\gamma\ \Gamma(\gamma;\kappa,\theta)$ with $\Gamma(\gamma;\kappa,\theta) = \frac{\gamma^{\kappa-1}\ e^{-\frac{\gamma}{\theta}}}{\theta^{\kappa}\Gamma(\kappa)}$. Having this in mind, the partial derivative of the likelihood with parameters $\{\kappa_k, \theta_k\}$ with respect to $\kappa_j$ is thus:

$$\begin{aligned}\frac{\partial\ f_{ik}}{\partial\ \kappa_j} &= \frac{\partial}{\partial\ \kappa_j}\left(\sum_b \int \mathscr{D}\beta\ \mathscr{D}s\ \mathbb{P}(y_i|b,\beta,s)\mathbb{P}(\beta)\mathbb{P}(s)\right) \\ &= \frac{\partial}{\partial\ \kappa_j}\left(\sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \mathbb{P}(y_i|b,\beta,s)\Gamma(\gamma;\kappa_k,\theta_k)\mathbb{P}(s)\right) \\ &= \frac{\partial}{\partial\ \kappa_j}\left(\sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \Gamma(\gamma;\kappa_k,\theta_k)\mathbb{P}(s)\right)\end{aligned} \tag{4.5.23}$$

However, the above expression has an implicit $\kappa$ dependence in the likelihood but has an explicit $\kappa$ dependence only through the $\Gamma$ prior on the aspect ratio. We have seen in chapter [3], section [3.5] that the derivative of a $\Gamma$ distribution with respect to $\kappa$ is:

$$\frac{\partial}{\partial\ \kappa}\Gamma(\gamma;\kappa,\theta) = \Gamma(\gamma;\kappa,\theta)\left(\log(\gamma) - \log(\theta) - \log(\psi(\kappa))\right) \tag{4.5.24}$$

Since expression (4.5.23) is only dependent on the derivative of $\kappa$ through the $\Gamma$ prior we have:

$$\begin{aligned}
\frac{\partial\ f_{ik}}{\partial\ \kappa_j} &= \frac{\partial}{\partial\ \kappa_j}\left(\sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \Gamma(\gamma;\kappa_k,\theta_k)\right)\mathbb{P}(s) \\
&= \sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \frac{\partial}{\partial\ \kappa_j}\Gamma(\gamma;\kappa_k,\theta_k)\mathbb{P}(s) \\
&= \sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \mathbb{P}(s)\times \\
&\qquad \delta_{kj}\quad \Gamma(\gamma;\kappa_k,\theta_k)\left(\log(\gamma) - \log(\theta_k) - \log(\psi(\kappa_k))\right) \\
&= \delta_{kj}F_{ik}^{(\kappa)}
\end{aligned} \tag{4.5.25}$$

where we have substituted:

$$\begin{aligned}
F_{ik}^{(\kappa)} = \sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ &\left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \mathbb{P}(s)\times \\
&\Gamma(\gamma;\kappa_k,\theta_k)\left(\log(\gamma) - \log(\theta_k) - \log(\psi(\kappa_k))\right)
\end{aligned} \tag{4.5.26}$$

and the upper index denotes the parameter the derivative was found with respect with. The above enters expression (4.5.22) in the following way:

$$\begin{aligned}\frac{\partial}{\partial\ \kappa_j}\mathbb{L} &= \sum_i \sum_k W_{ik} \frac{\frac{\partial}{\partial\ \kappa_j} f_{ik}}{f_{ik}} \\ &= \sum_i \sum_k W_{ik} \frac{\delta_{kj} F_{ik}^{(\kappa)}}{f_{ik}} \\ &= \sum_i W_{ij} \frac{F_{ij}^{(\kappa)}}{f_{ij}} \end{aligned} \tag{4.5.27}$$

$$\boxed{\frac{\partial}{\partial\ \kappa_j}\mathbb{L} = \sum_i W_{ij} \frac{F_{ij}^{(\kappa)}}{f_{ij}}}$$

which is the derivative of the log-likelihood with respect to $\kappa_j$. In order to find the estimator for $\kappa_j$ we have to set the above expression equal to zero. Before that, we will evaluate the derivative of the log-likelihood with respect to $\theta_j$ and then evaluate both estimators. We now evaluate the derivative of the log-likelihood (4.5.21) with respect to $\theta_j$:

$$\begin{aligned}\frac{\partial}{\partial\ \theta_j}\mathbb{L} &= \sum_i \sum_k (\log \pi_k + \log f_{ik}) \frac{\partial}{\partial\ \theta_j} W_{ik} + W_{ik} \frac{\partial}{\partial\ \theta_j} \log f_{ik} \\ &= \sum_i \sum_k W_{ik} \frac{\partial}{\partial\ \theta_j} \log f_{ik} \\ &= \sum_i \sum_k W_{ik} \frac{\frac{\partial}{\partial\ \theta_j} f_{ik}}{f_{ik}} \end{aligned} \tag{4.5.28}$$

Once again, we see that the the term that gets differentiated with respect to $\theta_j$ is $f_{ik}$. We will now examine the evaluation of this derivative and then substitute it back to expression (4.5.28). However, as with $\kappa$, the likelihood has an implicit $\theta$ dependence in $f_{ik}$ but has an explicit $\theta$ dependence only through the $\Gamma$ prior on the aspect ratio. We have seen in chapter [3], section [3.5] that the derivative of a $\Gamma$ distribution with respect to $\theta$ is:

$$\frac{\partial}{\partial\ \theta} \left(\Gamma(\gamma; \kappa, \theta)\right) = \Gamma(\gamma; \kappa, \theta) \left(\frac{\gamma}{\theta^2} - \kappa\theta^{-1}\right) \tag{4.5.29}$$

Thus, the derivative of $f_{ik}$ with respect to $\theta$ is:

$$\begin{aligned}
\frac{\partial\ f_{ik}}{\partial\ \theta_j} &= \frac{\partial}{\partial\ \theta_j}\left(\sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \Gamma(\gamma;\kappa_k,\theta_k)\mathbb{P}(s)\right) \\
&= \sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \frac{\partial}{\partial\ \theta_j}\Gamma(\gamma;\kappa_k,\theta_k)\mathbb{P}(s) \\
&= \sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \mathbb{P}(s)\times \\
&\qquad\qquad \delta_{kj}\Gamma(\gamma;\kappa_k,\theta_k)\left(\frac{\gamma}{\theta_k^2} - \kappa_k\theta_k^{-1}\right) \\
&= \delta_{kj}F_{ik}^{(\theta)}
\end{aligned} \tag{4.5.30}$$

where we have substituted:

$$\begin{aligned}
F_{ik}^{(\theta)} = \sum_b \int \mathscr{D}\gamma\ \mathscr{D}s\ \left[\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{y})} - \frac{B^2\tilde{n}^2\left|\widetilde{\mathrm{Cov}(\boldsymbol{v},\boldsymbol{y})}\right|^2}{(B^2\tilde{n}\widetilde{\mathrm{Var}(\boldsymbol{v})}+1)} + 2\zeta\right]^{-n-\alpha} \mathbb{P}(s)\times \\
\Gamma(\gamma;\kappa_k,\theta_k)\left(\frac{\gamma}{\theta_k^2} - \kappa_k\theta_k^{-1}\right)
\end{aligned} \tag{4.5.31}$$

and the upper index denotes the parameter the derivative was found with respect with. Thus, the derivative of the log-likelihood in expression (4.5.28) with respect to $\theta_j$ is:

$$\begin{aligned}
\frac{\partial}{\partial\ \theta_j}\mathbb{L} &= \sum_i\sum_k W_{ik}\frac{\partial}{\partial\ \theta_j}\log f_{ik} \\
&= \sum_i\sum_k W_{ik}\frac{\frac{\partial}{\partial\ \theta_j}f_{ik}}{f_{ik}} \\
&= \sum_i\sum_k W_{ik}\frac{\delta_{kj}F_{ik}^{(\theta)}}{f_{ik}} \\
&= \sum_i W_{ij}\frac{F_{ij}^{(\theta)}}{f_{ij}}
\end{aligned} \tag{4.5.32}$$

$$\boxed{\frac{\partial}{\partial\ \theta_j}\mathbb{L} = \sum_i W_{ij}\frac{F_{ij}^{(\theta)}}{f_{ij}}}$$

Having the derivatives of the log-likelihood, we can now find the estimators for the parameters $\{\kappa_j, \theta_j\}$ over which we want to maximise the log likelihood. For the maximisation of the parameters, one would have to set the derivatives in expressions (4.5.27) and (4.5.32) equal to zero. One notices that the expressions are quite complicated and thus there is no closed formed solution for the evaluation of the maximum in both cases. However, the solution for this is to utilise an optimisation algorithm such as the gradient ascent which has an equivalent effect. Finding the maximum of a function by setting its derivative to zero is equivalent to finding the maximum via gradient ascent. The only difference though is that that by using the gradient ascent, one must have some knowledge of the parameters since the algorithm needs a set of initial values. Another difficulty of using the gradient ascent is the fact that the algorithm is sensitive to these initial values and it can easily get trapped in a local maximum if the likelihood surface is uneven and unsmooth, something that makes the finding of the global maximum a tedious job.

In a similar fashion as it was discussed in chapter [3], section [3.5], the evaluation of the maximum via the gradient ascent will be done in the following way:

$$\{\kappa_{n+1}, \theta_{n+1}\} = \{\kappa_n, \theta_n\} + \epsilon \, \nabla_{\kappa,\theta} \, \mathbb{L} \tag{4.5.33}$$

where $\nabla_{\kappa,\theta} \, \mathbb{L}$ is the derivative of the log-likelihood as was evaluated in expressions (4.5.27) and (4.5.32). We have seen in chapter [3], section [3.5] that the evaluation of $F_{ik}^{\kappa}$ and $F_{ik}^{\theta}$ can be achieved by using Monte Carlo integration.

We now describe the results we acquired for our adapted version of the EM algorithm for the estimation of the hyperparameters of the Gamma distributions. To evaluate the algorithm in more detail we created a third class of shapes that would be used to make the differences between classes even more distinct and distinguishable. The third class of shapes was chosen to be triangles that were generated to be isosceles, with its height equal to 1 and its base equal to $\gamma \sim \Gamma(\kappa, \theta)$. For this class, we chose the hyperparameters to be: $\kappa = 60$, $\theta = \frac{2}{3}$. For each run of the EM algorithm, we simulated 15 shapes of which the first five were ribbons with their aspect ratios generated by $\Gamma(\gamma; 7.5, 1)$, the next five were sheets with their aspect

| | Ribbons | Sheets | Triangles |
|---|---|---|---|
| $\pi_K$ | 0.9017 | 0.0353 | 0.0630 |
| $\kappa_K$ | 4.8748 | 48.8974 | 59.2832 |
| $\theta_K$ | 5.6898 | 24.4812 | 22.1817 |

| | Ribbons | Sheets | Triangles |
|---|---|---|---|
| $\pi_K$ | 0.8491 | 0.1353 | 0.0157 |
| $\kappa_K$ | 4.3567 | 49.5937 | 59.2898 |
| $\theta_K$ | 4.5306 | 12.7205 | 16.9104 |

| | Ribbons | Sheets | Triangles |
|---|---|---|---|
| $\pi_K$ | 0.9647 | 0.0159 | 0.0194 |
| $\kappa_K$ | 3.4118 | 47.6232 | 58.9301 |
| $\theta_K$ | 5.6326 | 46.1990 | 711.7743 |

| | Ribbons | Sheets | Triangles |
|---|---|---|---|
| $\pi_K$ | 0.9176 | 0.0112 | 0.0712 |
| $\kappa_K$ | 5.1487 | 48.4691 | 57.6410 |
| $\theta_K$ | 5.5970 | 43.1922 | 13.2641 |

Table 4.2: The results of the parameters as estimated by the EM algorithm for four different runs.

ratios generated by $\Gamma(\gamma; 50, 0.4)$ and the last five were triangles with their base generated by $\Gamma(\gamma; 60, 2/3)$. We then initiated the EM algorithm for some starting values of the parameters $\kappa$ and $\theta$ and since we have a priori knowledge on the number of the components we imposed a delta prior over their number namely $\delta_{K,3}$. In this adapted version of the EM, the M-step is calculated by the gradient ascent of which the value of $\epsilon$ of expression (4.5.33) was chosen to be $\epsilon = 1$. The M-step was evaluated either after its maximum number of iterations was achieved or after the value the Euclidean distance $d = \sqrt{d\kappa^2 + d\theta^2}$ was smaller than a threshold which was chosen to be equal to 0.005. The following tables present the data acquired for 4 different runs of the EM algorithm with the starting values of the parameters being $[\pi_1, \pi_2, \pi_3] = [1/3, 1/3, 1/3]$, $[\kappa_1, \kappa_2, \kappa_3] = [6, 49, 58]$, $[\theta_1, \theta_2, \theta_3] = [0.4, 0.3, 0.5]$:

Table [4.2] illustrates the failure of the EM algorithm to estimate the true values of the hyperparameters as well as the mixing coefficients for 4 different runs. As we discuss in the next section, this is problem that is introduced by the unsmoothness and unevenness of the likelihood surface. This is the reason for the gradient ascent and effectively the EM algorithm to not converge to the expected values of the hyperparameters. The estimated responsibilities were equally disappointing since

the estimated values of the parameters were not even close to the expected ones. We discuss this in the next section.

### 4.5.2 Discussion of the results

As can be seen from the results presented above, the gradient ascent and hence the EM algorithm have not converged to the anticipated values. Although for four parameters it is not possible to include a plot of the likelihood hypersurface, we already have experience of some of its properties. We have seen in section [3.5] that it displays a very prominent peak (which is what we are searching for with the gradient ascent algorithm) but suffers from having many stationary points away from this point. For this reason it is very likely that the algorithm will become stuck at one of the local maxima, rather than finishing at the global maximum. For the EM algorithm to work properly the maximisation step needs to correctly find the values of the parameters which give the greatest value of the likelihood based on the currently computed assignments of the mixing coefficients.

We suggest that it is this unfortunate behaviour of the likelihood hypersurface which is responsible for the failure of the EM-algorithm to learn the class parameters accurately. This is the same problem that was encountered in section [3.5] where we discussed the learning of the parameters for labelled data. This is a severe obstacle which needs to be overcome in order to apply the model we propose to the problem of parameter estimation. The use of gradient ascent was forced upon us by little hope that the vanishing of (4.5.32) and (4.5.27) could be solved in closed form for $\theta j$ and $\kappa_j$. Progress in an analytic solution to this problem, or a numerical approach which overcomes the drawbacks of the gradient ascent algorithm, would be very welcome in future work. There are several algorithms which could be used for the maximisation of the likelihood with respect to the parameters of interest. We briefly refer to the simulated annealing [167, 168] and the stochastic gradient ascent [169, 170]. The former is an adaptation of the Metropolis-Hasting [171, 172] algorithm but is generally slow. The later is a hill-climbing algorithm which avoids the evaluation of the gradient for the whole training set but rather evaluates it for

one sample (or small number of samples). The stochastic nature of the algorithm makes it able to avoid getting stuck on local extrema. We leave these algorithms for future consideration as an extension of the analysis presented in this thesis.

Although we were unlucky with the exploration of the EM for a finite likelihood mixtures, we had however applied our model in chapter [3] to the problem of classification when such parameters are known, where we have had much success. We believe that this shows that our new approach has the potential to become a powerful alternative approach to shape classification making use of much more geometrical information than previous formalisms.

## 4.6 Concluding remarks

We have investigated the results of a different classification method to the one we proposed in chapter [2]. In section [4.3] we assumed that any data shape can be represented by a feature vector that it is comprised by certain geometrical properties. We assumed that each of the classes of shapes of the Kimia and the alphabet database, can be represented by a multivariate Gaussian distribution of which the mean and covariance matrix we estimated with the help of the EM algorithm. To avoid divergences and singularities generated by the covariances collapsing to a single point during the estimation, we used the EM-based non-parametric maximum likelihood algorithm which utilises a smoothing kernel. We then used the estimated parameters for classification of new, unobserved data that we simulated for this purpose. For the Kimia database, we found that the classification results are slightly better with our presented method of chapter [2] but are comparable to the method introduced in this chapter since they are within experimental error of the result arising with our new approach. In the case of the alphabet data set, our work in chapter [3] led to a much better success rate almost 40% better than the classification method presented in this chapter.

We concluded that the results of the approach presented in this chapter are not surprising since most of the geometrical information is lost by the feature extraction.

The approach of chapter [3] retains much more of this information which plays role in the building the likelihood function. Furthermore, our proposed method reflects the fact that we can include prior information about the data shapes (for example the description of the aspect ratios of sand bodies which follow a Γ-distribution).

For the investigation of the applicability of our proposed model we developed it in the context of unsupervised learning. We developed an EM procedure based upon our new likelihood which would find clusters in the data by using our observation model of section [2.6] as the mixture model. We were anticipating that with this approach, the algorithm would converge to parameter estimations which separate out the training data into the expected clusters and eventually recognise differences between real world shapes. Our adapted version of the EM algorithm was used for the estimation of the hyperparameters of the Gamma distributions that generate the shape curves of each class of shapes. For experimental purposes we generated an extra class of triangles of which the shape curves were also generated from a Gamma distribution. Our goal was to use the estimated values for the classification of new shapes. However, the evaluation of the M-step didn't allow us to calculate the scores of the likelihood in a closed form and thus an optimisation algorithm had to be employed for this task. The M-step of the EM was evaluated by using the gradient ascent algorithm and, due to this fact, the EM didn't converge to the anticipated values. This was a somehow expected behaviour because, as we saw in section [3.5] the use of gradient ascent is ill-used and ill-behaved since the likelihood hypersurface is probably uneven and unsmooth. A solution that overcomes this problem would be the use of an analytic solution or a numerical approach.

# Chapter 5

# The three dimensional case

## 5.1 Introduction

For the past four chapters we have treated shapes as continuous planar curves i.e. a collection of points in $\mathbb{R}^2$. The classification of the planar curves was achieved by the maximisation of the posterior probability $\mathbb{P}(C|\boldsymbol{y})$. For the classification of the sand bodies' data set, we assumed that information about the three dimensional point clouds such as their corresponding paleodirections and dip angles were available to us. This information was sufficient to know the plane on which to project the three dimensional point cloud in order to acquire the two dimensional planar shapes which would constitute the data set for classification purposes. One of the questions that rose during this research was what happens in the case that the paleodirection and even more the dip angle of a sand body data set is not available. In particular, in the absence of the dip angle one wouldn't know the projection plane of the three dimensional point cloud and would have to estimate it as an extra parameter. In section [5.2] we discuss the solution to this problem in case that information about the true projection plane was unavailable. This gives rise to the problem of three-dimensional classification which we discuss that can be treated using the Bayesian paradigm and solved in the same fashion as in the two-dimensional case. We present how the algorithm of chapter [2] can be upgraded for three dimensional case and we focus on the integration of similarity transformations as we discussed in section [2.8]. Although in chapter [2], we integrated over all similarity transformations

we didn't have the same luck in the three-dimensional case and we only achieved integration over translations and rotations. In section [5.3], we present the results of the integration over three dimensional translations which are similar to the two-dimensional case; they differ in the normalisation constants. Integrating over three-dimensional rotations is a complicated problem for which we had to choose their representation so that integration is more straight forward. In section [5.4] we discuss the chosen representation of three-dimensional rotations which is quaternions. We present the properties of quaternions and how they can be used for the desired integration and in section [5.5] we present the results of the integration over the three-dimensional rotations using quaternions. Finally, section [5.6] presents the concluding remarks of this chapter.

## 5.2 Classification of three-dimensional shapes

A problem we encountered during this research due to the absence of real geological data, was the fact that the paleodirection or the dip angle wouldn't always be available to us. One could argue that could be a real situation that a geologist may encounter whilst gathering data from the field. These two parameters are vital for the classification of sand bodies since they define the projection plane of the three-dimensional point cloud. The solution that we came up with was to treat the sand body as a complete three dimensional object, which is to be classified by comparison to non-planar example shapes.

For the simulation of example shapes we could begin with an idealised planar sand body, $x$, assumed to be cut perpendicular to the paleoflow. Adding isotropic Gaussian white noise each point is perturbed to $y_i = x_i + \nu_i$ where the $\{\nu\} \sim N(\underline{0}, \Sigma)$ and $\Sigma = \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix}$. To then compare this to the data shape we follow our previous construction and consider all shapes that are related to this by rigid similarity transformations. This requires the integration over the group of rotations in three dimensions, $SO(3)$. As in the case of planar curves and following the Bayesian paradigm the posterior probability of a class is:

$$\begin{aligned}\mathbb{P}(C|\boldsymbol{y}) &\propto \mathbb{P}(\boldsymbol{y}|C)\mathbb{P}(C)\\ &\propto \sum_{b\in\mathcal{B}}\int \mathscr{D}\beta\ \mathscr{D}s\ \mathscr{D}g\ d\sigma\ \mathbb{P}(\boldsymbol{y}|b,\beta,s,g,\sigma)\mathbb{P}(\beta|C)\mathbb{P}(s)\mathbb{P}(g)\mathbb{P}(\sigma)\mathbb{P}(C)\end{aligned} \quad (5.2.1)$$

To maximise the posterior probability $\tilde{C} = \text{argmax}_C\mathbb{P}(C|\boldsymbol{w})$ and perform the desired classification we marginalise the likelihood over the same nuisance parameters and utilise the same probability models that we described in chapter [2] for the case of planar curves. To perform the classification we marginalise over the nuisance parameters that give rise to a particular shape: similarity transformations $g$ which are namely translations $\boldsymbol{t}$, rotations R and also bijections $b$, curves $\beta$ and samplings $s$. Note that for the three dimensional case we don't take into account scalings $a$ for reasons explained in the end of this chapter. Hence of particular interest as in the case of planar curves is the observation model which, by returning to our previous notation now is:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,g,\sigma) &= \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}|\boldsymbol{y_{b_i}} - g\circ\boldsymbol{\beta}(s(b_i^{-1}))|^2\right)\\ &= \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}|\boldsymbol{y_{b_i}} - \mathrm{R}\boldsymbol{\beta}(s(b_i^{-1})) - \boldsymbol{t}|^2\right)\\ &= \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}|\boldsymbol{y_{b_i}} - \mathrm{R}\boldsymbol{v_i} - \boldsymbol{t}|^2\right)\end{aligned} \quad (5.2.2)$$

where we have substituted $\boldsymbol{v} = \boldsymbol{\beta}(s(b_i^{-1}))$ for simplicity and in this case $\boldsymbol{y}$, $\boldsymbol{t}$ and R in $\mathbb{R}^3$. For the calculation of the Maximum a Posteriori approximation we need to calculate the integrated likelihood which is described by model (5.2.2). We will integrate over all nuisance parameters in the same way we did for the two-dimensional case in chapter [2]. It is worth mentioning here, that for the integration of both translations and rotations we didn't make use of Jeffreys prior. In the next sections, we describe the integration over these parameters in detail.

## 5.3 Integration of translations $t$

As we have seen in chapter [2], the integration over translation is quite straightforward. Since we are in $\mathbb{R}^3$ we have a single data shape $\boldsymbol{y} = (y_1, y_2, y_3)$ and translations $\boldsymbol{t} = (t_1, t_2, t_3)$. We have:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,\mathrm{R},\sigma) &= \frac{1}{(2\pi\sigma)^{3n/2}} \iiint d^3\boldsymbol{t}\ \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}|\boldsymbol{y_{b_i}} - \mathrm{R}\boldsymbol{v_i} - \boldsymbol{t}|^2\right) \\ &= \frac{1}{Z}\iiint d^3\boldsymbol{t}\ \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}|\boldsymbol{Y_i} - \boldsymbol{t}|^2\right) \end{aligned} \tag{5.3.3}$$

where we have defined $\boldsymbol{Y_i} = \boldsymbol{y_{b_i}} - \mathrm{R}\boldsymbol{v_i}$ and $\frac{1}{Z} = \frac{1}{(2\pi\sigma)^{3n/2}}$. For the integration over translations we have:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,\mathrm{R},\sigma) &= \frac{1}{Z}\iiint d^3\boldsymbol{t}\ \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}|\boldsymbol{Y_i} - \boldsymbol{t}|^2\right) \\ &= \frac{1}{Z}\left(\frac{2\pi\sigma^2}{n}\right)^{3/2}\exp\left(\frac{n\,|\sum_i^N \boldsymbol{Y_i}|^2}{2\sigma^2} - \frac{\sum_i^N|\boldsymbol{Y_i}|^2}{2\sigma^2}\right) \end{aligned} \tag{5.3.4}$$

Notice that the exponent of expression (5.3.4) is the variance of $\boldsymbol{Y}$, hence we can write:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,\mathrm{R},\sigma) &= \frac{1}{Z}\left(\frac{2\pi\sigma^2}{n}\right)^{3/2}\exp\left(-\frac{n}{2\sigma^2}\mathrm{Var}[\boldsymbol{Y}]\right) \\ &= \frac{1}{Z}\left(\frac{2\pi\sigma^2}{n}\right)^{3/2}\exp\left(-\frac{n}{2\sigma^2}[\overline{\boldsymbol{Y}^2} - \overline{\boldsymbol{Y}}^2]\right) \\ &= \frac{1}{Z}\left(\frac{2\pi\sigma^2}{n}\right)^{3/2}\exp\left(-\frac{n}{2\sigma^2}\left[\sum_i|\boldsymbol{Y_i}|^2 - \frac{1}{n}\sum_i\sum_j\boldsymbol{Y_i}\boldsymbol{Y_j}\right]\right) \end{aligned} \tag{5.3.5}$$

This result is in the same form as in the case of integration of two dimensional translations with a flat prior with different normalisation constants. To integrate expression (5.3.5) over rotations we need to choose the appropriate representation of three-dimensional rotations. The space over three-dimensional rotations is quite complicated since the integration domain is a 3-ball i.e. rotations about $\theta, \phi, r$. To perform such integration one should also choose the appropriate measure; usually one chooses the left and right invariant Haar measure. This was one of the most difficult

challenges of the task. There were quite a few attempts over the representation of rotations and the calculation of the induced measure. The most, relatively, simple representation was the one we describe in the next section, that of quaternions.

## 5.4 Quaternions

The integration of rotations for two-dimensional shapes is a quite straight-forward calculation since the space of two dimensional rotations is compact. In this case, we chose to represent rotations parametrised by $\theta$ in the complex plane as $\mathrm{R} = e^{i\theta}$ so that a complex arithmetic is used for a geometric operation. The step to three-dimensional shapes is quite large since the rotation group become that of $SO(3)$. Since the three dimensional integration over rotations is fairly more complicated we chose to represent them by the three-dimensional equivalent complex arithmetic which is quaternions. Quaternions are a four-dimensional algebra and the quaternionic space is defined as: $\mathbb{H} = \{a + bi + cj + dk : a, b, c, d \in \mathbb{R}\}$ with $i^2$, $j^2$ and $k^2$ are equal to $-1$ and $ij = k = -ji$, $jk = i = -kj$, $ki = j = -ik$ with $i, j, k$ the three special unit imaginary quaternions.

Quaternions are represented by a scalar part (we will call this body) and a vector part (we will call this the soul): $v_o + v_1 i + v_2 j + v_3 k = (v_0, \underline{\boldsymbol{v}})$ with $\underline{\boldsymbol{v}} = (v_1, v_2, v_3)$. The product of quaternions is found to be: $(v_0, \underline{\boldsymbol{v}})(w_o, \underline{\boldsymbol{w}}) = (v_0 w_0 - \underline{\boldsymbol{v}} \cdot \underline{\boldsymbol{w}}, v_0 \underline{\boldsymbol{w}} + w_0 \underline{\boldsymbol{v}} + \underline{\boldsymbol{v}} \times \underline{\boldsymbol{w}})$. The length of a quaternion $v = (v_0, \underline{v})$ is defined by its norm which is defined, as with complex numbers, as the square root of the product of the quaternion by its conjugate $v^* = (v_0, -\underline{v})$. This is: $|v| = \sqrt{vv^*} = \sqrt{v_0^2 + v_1^2 + v_2^2 + v_3^2}$.

Consider the three-dimensional space as purely quaternionic so that: $\mathbb{R}^3 = \{xi + yj + zk\}$, $x, y, z \in \mathbb{R}$. Just like complex numbers, three-dimensional rotations are done by using **unit** quaternions like for example $\cos\theta + i\sin\theta$, $\cos\theta + j\sin\theta$, $\cos\theta + k\sin\theta$ by analogy to Euler's formula. However, $i, j, k$ are just three special unit imaginary quaternions and one can construct much more unit quaternions than these. Let a unit vector be $\boldsymbol{u} = u_1 i + u_2 j + u_3 k$, then $\cos\phi + \boldsymbol{u}\sin\phi$ is also a unit quaternion which by analogy to Euler's formula can be written as $e^{\boldsymbol{u}\phi}$. Unit quaternions form a

special group which is: $\mathrm{Spin}(3) = \{q \in \mathbb{H} : |q|^2 = 1\}$ and has the special property that is isomorphic to $\mathrm{SU}(2) \cong \mathrm{S}^3$. To understand rotations by quaternions we present the following theorem.

***Theorem* 1** *If* $\boldsymbol{u}$ *is a unit vector and* $\boldsymbol{v}$ *is any vector, the expression* $e^{\boldsymbol{u}\phi}\boldsymbol{v}e^{-\boldsymbol{u}\phi}$. *gives the result of rotating* $\boldsymbol{v}$ *about the axis parallel to* $\boldsymbol{u}$ *by* $2\phi$ *degrees.*

Thus, any three-dimensional rotation in the quaternionic representation can be written as $\mathrm{R} = q\boldsymbol{v}\bar{q}$ with the help of a unit vector that is multiplied by the left by the quaternion we want to rotate by and to the right by its conjugate. For the proof of the above theorem refer to [173, 174].

An important point about quaternions, in contrary to complex numbers, is the fact that they do not commute. The basis quaternions anti-commute and they provide a representation of $SU(2)$ so $[i, j] = 2k$ etc. For two arbitrary quaternions the value of the commutator is $[y, q] = 2\ \underline{\boldsymbol{y}} \times \underline{\boldsymbol{q}}$, with $y, q$ quaternions and $\underline{\boldsymbol{y}}, \underline{\boldsymbol{q}}$ their vectorial parts. This is something we take into account for the calculations to follow.

To proceed with the calculation of the integral over rotations of expression (5.3.5) we need to utilise the quaternionic representation of rotations and write it into its quaternionic equivalent. Let $y = (0, y_1, y_2, y_3)$, $q = (q_o, q_1, q_2, q_3)$, $v = (0, v_1, v_2, v_3)$ the quaternionic expressions for a data shape $y$, a rotation by $q$ and an idealised example curve $v$. Note that $y$ and $v$ contain only the vectorial quaternionic part since they are in $\mathbb{R}^3$ and $q$ is the quaternion by which we want to rotate.

## 5.5 Integration of rotations

Although we have found the simplest representation of three-dimensional rotations, we only need to take into account a small subset of the quaternionic space since rotations are represented only by unit quaternions. For this reason, we choose to integrate over the full quaternionic space $\mathbb{R}^4$ and impose a constraint that takes into account only unit quaternions. Since unit quaternions live on the surface of the unit 3-sphere, we impose the constraint $\delta(|q|^2 - 1) = \delta(qq^* - 1)$ where $q^*$ is the

quaternionic conjugate of $q$. This $\delta$ function is invariant under the action of $SU(2)$ on the parameters since rotations do not change the length of the quaternion. Before we continue with the integration over rotations we will bring expression (5.3.5) in an appropriate form. In particular, we work with the exponent:

$$
\begin{aligned}
&-\frac{1}{2\sigma^2}\left(\sum_i |\boldsymbol{Y_i}|^2 - \frac{1}{n}\sum_i\sum_j \boldsymbol{Y_i}\boldsymbol{Y_j}\right) = \\
&= -\left(\frac{1}{2\sigma^2}\right)\sum_i (\boldsymbol{y_i} - \mathrm{R}\boldsymbol{v_i})^{\mathrm{T}}(\boldsymbol{y_i} - \mathrm{R}\boldsymbol{v_i}) - \frac{1}{n}\sum_{ij}(\boldsymbol{y_i} - \mathrm{R}\boldsymbol{v_i})^{\mathrm{T}}(\boldsymbol{y_j} - \mathrm{R}\boldsymbol{v_j}) \\
&= -\left(\frac{1}{2\sigma^2}\right)\Bigg[\sum_i \boldsymbol{y_i}^{\mathrm{T}}\boldsymbol{y_i} - \sum_i \boldsymbol{y_i}^{\mathrm{T}}\mathrm{R}\boldsymbol{v_i} - \sum_i (\mathrm{R}\boldsymbol{v_i})^{\mathrm{T}}\boldsymbol{y_i} + \sum_i (\mathrm{R}\boldsymbol{v_i})^{\mathrm{T}}\mathrm{R}\boldsymbol{v_i} \\
&\qquad -\frac{1}{n}\sum_{ij}\boldsymbol{y_i}^{\mathrm{T}}\boldsymbol{y_j} + \frac{1}{n}\sum_{ij}\boldsymbol{y_i}^{\mathrm{T}}\mathrm{R}\boldsymbol{v_j} + \frac{1}{n}\sum_{ij}(\mathrm{R}\boldsymbol{v_i})^{\mathrm{T}}\boldsymbol{y_j} - \frac{1}{n}\sum_{ij}(\mathrm{R}\boldsymbol{v_i})^{\mathrm{T}}\mathrm{R}\boldsymbol{v_j}\Bigg] \\
&= \left(\frac{1}{2\sigma^2}\right)\Big[-n\ \overline{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}} + n\ \overline{\boldsymbol{y}^{\mathrm{T}}\mathrm{R}\boldsymbol{y}} + n\ \overline{\boldsymbol{v}^{\mathrm{T}}\mathrm{R}^{\mathrm{T}}\boldsymbol{y}} - n\ \overline{\boldsymbol{v}^{\mathrm{T}}\mathrm{R}^{\mathrm{T}}\mathrm{R}\boldsymbol{v}} \\
&\qquad +n\ \overline{\boldsymbol{y}}^2 - n\ \overline{\boldsymbol{y}}^{\mathrm{T}}\mathrm{R}\overline{\boldsymbol{v}} - n\ \overline{\boldsymbol{v}}^{\mathrm{T}}\mathrm{R}^{\mathrm{T}}\overline{\boldsymbol{y}} + n\ \overline{\boldsymbol{y}}^2\Big] \\
&= \left(\frac{1}{2\sigma^2}\right)\Big[-n\ (\overline{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}} - \overline{\boldsymbol{y}}^2) - n\ (\overline{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}} - \overline{\boldsymbol{v}}^2) \\
&\qquad +n\ \left(\overline{\boldsymbol{v}^{\mathrm{T}}\mathrm{R}^{\mathrm{T}}\boldsymbol{y}} - \overline{\boldsymbol{v}}^{\mathrm{T}}\mathrm{R}^{\mathrm{T}}\overline{\boldsymbol{y}}\right) + n\ \left(\overline{\boldsymbol{y}^{\mathrm{T}}\mathrm{R}\boldsymbol{v}} - \overline{\boldsymbol{y}}^{\mathrm{T}}\mathrm{R}\overline{\boldsymbol{v}}\right)\Big] \\
&= \left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\boldsymbol{y}) - n\ \mathrm{Var}(\boldsymbol{v})] + n\ \left[\overline{(\boldsymbol{y}-\bar{\boldsymbol{y}})^{\mathrm{T}}\mathrm{R}(\boldsymbol{v}-\bar{\boldsymbol{v}}) + (\boldsymbol{v}-\bar{\boldsymbol{v}})^{\mathrm{T}}\mathrm{R}^{\mathrm{T}}(\boldsymbol{y}-\bar{\boldsymbol{y}})}\right]
\end{aligned}
\tag{5.5.6}
$$

We make a change of variables so that $\hat{\boldsymbol{y}} = \boldsymbol{y} - \bar{\boldsymbol{y}}$ and $\hat{\boldsymbol{v}} = \boldsymbol{v} - \bar{\boldsymbol{v}}$ and expression (5.5.6) now is:

$$
\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\hat{\boldsymbol{y}}) - n\ \mathrm{Var}(\hat{\boldsymbol{v}})] + n\ \left[\overline{\hat{\boldsymbol{y}}^{\mathrm{T}}\mathrm{R}\hat{\boldsymbol{v}} + (\mathrm{R}\hat{\boldsymbol{v}})^{\mathrm{T}}\hat{\boldsymbol{y}}}\right] \tag{5.5.7}
$$

This is the exponent of expression (5.3.5). Substituting the exponent back, expression (5.3.5) becomes:

$$\begin{aligned}\mathbb{P}(y|b,\beta,s,\mathrm{R},\sigma) &= \frac{1}{Z}\left(\frac{2\pi\sigma^2}{n}\right)^{3/2} \times \\ &\exp\left(\frac{1}{2\sigma^2}\left[-n\ \mathrm{Var}(\hat{\boldsymbol{y}}) - n\ \mathrm{Var}(\hat{\boldsymbol{v}})\right] + n\ \left[\overline{\hat{\boldsymbol{y}}^{\mathrm{T}}\mathrm{R}\hat{\boldsymbol{v}} + (\mathrm{R}\hat{\boldsymbol{v}})^{\mathrm{T}}\hat{\boldsymbol{y}}}\right]\right) \\ &= \frac{1}{Z}\exp\left(\frac{1}{2\sigma^2}\left[-n\ \mathrm{Var}(\hat{\boldsymbol{y}}) - n\ \mathrm{Var}(\hat{\boldsymbol{v}})\right] + n\ \left[\overline{\hat{\boldsymbol{y}}^{\mathrm{T}}\mathrm{R}\hat{\boldsymbol{v}} + (\mathrm{R}\hat{\boldsymbol{v}})^{\mathrm{T}}\hat{\boldsymbol{y}}}\right]\right)\end{aligned} \tag{5.5.8}$$

where we have absorbed all the constants into the normalisation coefficient. To be able to integrate the above expression with respect to rotations, we will have to do some work and extract the quaternionic dependence of the exponent. Since the variance of $y$ and the variance of $v$ are independent of quaternions we will only work with the second term of the exponent. The expression in the exponent is a real number so the final term of expression (5.5) must be a real number too. We also make use of the following quaternionic property that for any $q$ and $t$, we have: $qt^* + tq^* = 2\ \underline{\boldsymbol{q}} \cdot \underline{\boldsymbol{t}}$ with $\underline{\boldsymbol{q}}, \underline{\boldsymbol{t}}$ the souls of the two quaternions. The first term in the square brackets becomes:

$$\begin{aligned}\overline{\hat{\boldsymbol{y}}^{\mathrm{T}}\mathrm{R}\hat{\boldsymbol{v}}} = \frac{1}{2}\left(\overline{\hat{\boldsymbol{y}}^{\mathrm{T}}(\mathrm{R}\hat{\boldsymbol{v}})^* + (\mathrm{R}\hat{\boldsymbol{v}})^{\mathrm{T}}\hat{\boldsymbol{y}}^*}\right) &= \frac{1}{2}\left(\overline{\hat{\boldsymbol{y}}^{\mathrm{T}}(q\hat{\boldsymbol{v}}q^*)^* + q\hat{\boldsymbol{v}}^{\mathrm{T}}q^*\hat{\boldsymbol{y}}^*}\right) \\ &= \frac{1}{2}\left(\overline{\hat{\boldsymbol{y}}^{\mathrm{T}}(q\hat{\boldsymbol{v}}^*q^*) + q\hat{\boldsymbol{v}}^{\mathrm{T}}q^*\hat{\boldsymbol{y}}^*}\right)\end{aligned} \tag{5.5.9}$$

We know that the commutator for two quaternions is $[q,t] = 2\ \underline{\boldsymbol{q}} \times \underline{\boldsymbol{t}}$, i.e. the cross product of their souls. This gives us: $\hat{\boldsymbol{y}}^{\mathrm{T}}q = q\hat{\boldsymbol{y}}^{\mathrm{T}} + 2\ \underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \times \underline{\boldsymbol{q}}$ and $q^*\hat{\boldsymbol{y}}^* = \hat{\boldsymbol{y}}^*q^* - 2\ \underline{\hat{\boldsymbol{y}}}^* \times \underline{\boldsymbol{q}}^*$. We now work with expression (5.5.9) and we remove the expectational overbars to make the calculation easier for the reader. Then expression (5.5.9) is:

$$\begin{aligned}&(q\hat{\boldsymbol{y}}^{\mathrm{T}} + 2\ \underline{\hat{\boldsymbol{y}}}^{T} \times \underline{\boldsymbol{q}})\hat{\boldsymbol{v}}^*q^* + q\hat{\boldsymbol{v}}^{\mathrm{T}}(\hat{\boldsymbol{y}}^*q^* - 2\ \underline{\hat{\boldsymbol{y}}}^* \times \underline{\boldsymbol{q}}^*) = \\ &q(\hat{\boldsymbol{y}}^{\mathrm{T}}\hat{\boldsymbol{v}}^* + \hat{\boldsymbol{v}}^{\mathrm{T}}\hat{\boldsymbol{y}}^*)q^* + 2\ (\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \times \underline{\boldsymbol{q}})\hat{\boldsymbol{v}}^*q^* - 2\ q\hat{\boldsymbol{v}}^{\mathrm{T}}(\underline{\hat{\boldsymbol{y}}}^* \times \underline{\boldsymbol{q}}^*)\end{aligned} \tag{5.5.10}$$

We know that $(\hat{\boldsymbol{y}}^{\mathrm{T}}\hat{\boldsymbol{v}}^* + \hat{\boldsymbol{v}}^{\mathrm{T}}\hat{\boldsymbol{y}}^*) = 2\ \underline{\hat{\boldsymbol{y}}} \cdot \underline{\hat{\boldsymbol{v}}}, \in \mathbb{R}$. This expression must be real and it only has a body but not a soul; we transform the other terms to bring them in the desired form. Making use of the following properties:

$$\hat{\boldsymbol{v}}^* q^* = q\hat{\boldsymbol{v}} - 2\ \underline{\boldsymbol{q}} \times \underline{\hat{\boldsymbol{v}}} - 2q_o\underline{\hat{\boldsymbol{v}}} \tag{5.5.11}$$

$$q\underline{\boldsymbol{v}} = -\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{v}}} + q_o\underline{\hat{\boldsymbol{v}}} + \underline{\boldsymbol{q}} \times \underline{\hat{\boldsymbol{v}}} \tag{5.5.12}$$

$$\underline{\hat{\boldsymbol{v}}}^* \times \underline{\boldsymbol{q}}^* = \underline{\hat{\boldsymbol{v}}} \times \underline{\boldsymbol{q}} \tag{5.5.13}$$

expression (5.5.10) becomes:

$$\begin{aligned} q(2\ \underline{\hat{\boldsymbol{y}}} \cdot \underline{\hat{\boldsymbol{v}}})q^* + 2\Big[-\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{v}}}\left(\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \times \underline{\boldsymbol{q}}\right) - \left(\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \times \underline{\boldsymbol{q}}\right)\left(\underline{\boldsymbol{q}} \times \underline{\hat{\boldsymbol{y}}}\right) - q_o(\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \times \underline{\boldsymbol{q}})\underline{\hat{\boldsymbol{b}}})\Big] \\ + 2\Big[\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}(\underline{\hat{\boldsymbol{y}}} \times \underline{\boldsymbol{q}}) - q_o\hat{\boldsymbol{v}}^{\mathrm{T}}(\underline{\hat{\boldsymbol{y}}} \times \underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}} \times \underline{\hat{\boldsymbol{v}}}^{\mathrm{T}})(\underline{\hat{\boldsymbol{y}}} \times \underline{\boldsymbol{q}})\Big] \end{aligned} \tag{5.5.14}$$

Since the final result must be real we keep only the parts that are real from expression (5.5.14) and drop any parts that are purely quaternionic. The first term is real since it can be written as: $2\ (\underline{\hat{\boldsymbol{y}}} \cdot \underline{\hat{\boldsymbol{v}}})qq^* = 2\ \underline{\hat{\boldsymbol{y}}} \cdot \underline{\hat{\boldsymbol{v}}}$ and is kept. The term $-\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{v}}}\left(\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \times \underline{\boldsymbol{q}}\right)$ from the first square bracket will be dropped since it is bodyless; the same stands for the term in the second bracket: $\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}(\underline{\hat{\boldsymbol{y}}} \times \underline{\boldsymbol{q}})$. Thus, expression (5.5.14) now becomes:

$$\begin{aligned} &2\ \underline{\hat{\boldsymbol{y}}} \cdot \underline{\hat{\boldsymbol{v}}} + 2\Big[\left(\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \times \underline{\boldsymbol{q}}\right) \cdot \left(\underline{\boldsymbol{q}} \times \underline{\hat{\boldsymbol{v}}}\right) + q_o(\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \times \underline{\boldsymbol{q}}) \cdot \underline{\hat{\boldsymbol{b}}})\Big] + 2\ \Big[(\underline{\boldsymbol{q}} \times \underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}) \cdot (\underline{\hat{\boldsymbol{y}}} \times \underline{\boldsymbol{q}}) + q_o\underline{\boldsymbol{q}} \cdot (\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \times \underline{\hat{\boldsymbol{y}}})\Big] \\ &= 2\ \underline{\hat{\boldsymbol{y}}} \cdot \underline{\hat{\boldsymbol{v}}} + 2\Big[(\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \cdot \underline{\boldsymbol{q}})(\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{v}}}) - (\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \cdot \underline{\hat{\boldsymbol{v}}})(\underline{\boldsymbol{q}} \cdot \underline{\boldsymbol{q}}) + q_o\underline{\boldsymbol{q}} \cdot (\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}^{\mathrm{T}})\Big] \\ &\qquad + 2\ \Big[(\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{y}}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \cdot \underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}} \cdot \underline{\boldsymbol{q}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \cdot \underline{\hat{\boldsymbol{y}}}) + q_o\underline{\boldsymbol{q}} \cdot (\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \times \underline{\hat{\boldsymbol{y}}})\Big] = \\ &= 2\ \underline{\hat{\boldsymbol{y}}} \cdot \underline{\hat{\boldsymbol{v}}} + 4\Big[(\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{y}}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \cdot \underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}} \cdot \underline{\boldsymbol{q}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \cdot \underline{\hat{\boldsymbol{y}}}) + q_o\underline{\boldsymbol{q}} \cdot (\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \times \underline{\hat{\boldsymbol{y}}})\Big] \end{aligned} \tag{5.5.15}$$

We know take an overall factor of a $\frac{1}{2}$ as we should from expression (5.5.9), expression (5.5.15) becomes:

$$\underline{\hat{\boldsymbol{y}}} \cdot \underline{\hat{\boldsymbol{v}}} + 2\ \Big[(\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{y}}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \cdot \underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}} \cdot \underline{\boldsymbol{q}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \cdot \underline{\hat{\boldsymbol{y}}}) + q_o\underline{\boldsymbol{q}} \cdot (\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \times \underline{\hat{\boldsymbol{y}}})\Big] \tag{5.5.16}$$

which constitutes the first term in square brackets of expression (5.5). We now work on the second term of the exponent in expression (5.5). By using the same analysis as above, the term becomes:

$$
\begin{aligned}
(\mathrm{R}\hat{\boldsymbol{v}})^{\mathrm{T}}\hat{\boldsymbol{y}}^{*} &= 2\ \underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}} + 2\Big[(\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}}\cdot\underline{\boldsymbol{q}})(\underline{\boldsymbol{q}}\cdot\underline{\hat{\boldsymbol{v}}}) - (\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}}\cdot\underline{\hat{\boldsymbol{v}}})(\underline{\boldsymbol{q}}\cdot\underline{\boldsymbol{q}}) + q_o\underline{\boldsymbol{q}}\cdot(\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}})\Big] \\
&\quad + 2\ \Big[(\underline{\boldsymbol{q}}\cdot\underline{\hat{\boldsymbol{y}}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}}\cdot\underline{\boldsymbol{q}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\hat{\boldsymbol{y}}}) + q_o\underline{\boldsymbol{q}}\cdot(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\times\underline{\hat{\boldsymbol{y}}})\Big] = \\
&= 2\ \underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}} + 4\ \Big[(\underline{\boldsymbol{q}}\cdot\underline{\hat{\boldsymbol{y}}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}}\cdot\underline{\boldsymbol{q}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\hat{\boldsymbol{y}}}) + q_o\underline{\boldsymbol{q}}\cdot(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\times\underline{\hat{\boldsymbol{y}}})\Big] \qquad (5.5.17)
\end{aligned}
$$

Taking an overall factor of a $\frac{1}{2}$ as with the first term, expression (5.5.17) becomes:

$$
\underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}} + 2\ \Big[(\underline{\boldsymbol{q}}\cdot\underline{\hat{\boldsymbol{y}}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}}\cdot\underline{\boldsymbol{q}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\hat{\boldsymbol{y}}}) + q_o\underline{\boldsymbol{q}}\cdot(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\times\underline{\hat{\boldsymbol{y}}})\Big] \qquad (5.5.18)
$$

which constitutes the second term in square brackets of expression (5.5). Finally, replacing the expectational overbars and replacing the terms (5.5.16) and (5.5.18) to the exponent of expression (5.5), the exponent becomes:

$$
\begin{aligned}
&\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\boldsymbol{y}) - n\ \mathrm{Var}(\boldsymbol{v})] \\
&\quad + 2n\ \left[\overline{\underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}} + 2\ \Big((\underline{\boldsymbol{q}}\cdot\underline{\hat{\boldsymbol{y}}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}}\cdot\underline{\boldsymbol{q}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\hat{\boldsymbol{y}}}) + q_o\underline{\boldsymbol{q}}\cdot(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\times\underline{\hat{\boldsymbol{y}}})\Big)}\right] \qquad (5.5.19)
\end{aligned}
$$

We extracted the quaternionic dependence from the exponent of the marginalised likelihood in (5.3.5) and we can now perform the integration over the quaternionic space with respect to rotations imposing the $\delta$-function for the unit quaternions.

$$
\begin{aligned}
&\mathbb{P}(y|b,\beta,s,\sigma) = \\
&= \frac{1}{Z}\int d^4q\ \delta(qq^{*}-1)\ \exp\left(\frac{|\sum_i^N \boldsymbol{Y_i}|^2}{2n\sigma^2} - \frac{\sum_i^N |\boldsymbol{Y_i}|^2}{2\sigma^2}\right) = \qquad (5.5.20) \\
&= \frac{1}{Z}\exp\left[\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\hat{\boldsymbol{y}}) - n\ \mathrm{Var}(\hat{\boldsymbol{v}})] + 2n\ \overline{\underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}}}\right]\times \\
&\int d^4q\ \delta(qq^{*}-1)\exp\left(4n\ \left[\overline{\Big((\underline{\boldsymbol{q}}\cdot\underline{\hat{\boldsymbol{y}}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}}\cdot\underline{\boldsymbol{q}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\cdot\underline{\hat{\boldsymbol{y}}}) + q_o\underline{\boldsymbol{q}}\cdot(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}}\times\underline{\hat{\boldsymbol{y}}})\Big)}\right]\right) \qquad (5.5.21)
\end{aligned}
$$

For the convenience of the reader, we will ignore the constant terms and the normalisation constants in front of the rotational integral and study how the integration over quaternions will be carried out. To perform the integration we replace the

$\delta$-function restriction by its Fourier equivalent which introduces a second integral. That is:

$$\delta(qq^* - 1) = \frac{1}{2\pi} \int dk' \ \exp\big(ik'(|q|^2 - 1)\big) \tag{5.5.22}$$

The result of integrating over $k$ will then supply the Haar measure on the space of unit quaternions and restrict our parameters $q$ to this surface. We will discuss this in more detail presently – see eq. (5.5.24). Now, expression (5.5.21) becomes:

$$\frac{1}{2\pi} \iint dk' \ d^4q \ \exp\big(ik'(|q|^2 - 1)\big) \times \\ \exp\left(4n \ \left[\overline{\Big((\underline{\boldsymbol{q}} \cdot \underline{\hat{\boldsymbol{y}}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \cdot \underline{\boldsymbol{q}}) - (\underline{\boldsymbol{q}} \cdot \underline{\boldsymbol{q}})(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \cdot \underline{\hat{\boldsymbol{y}}}) + q_o \underline{\boldsymbol{q}} \cdot (\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \times \underline{\hat{\boldsymbol{y}}})\Big)}\right]\right) \tag{5.5.23}$$

One notices that the integrand can be written in the form of a Gaussian distribution, so that the exponent can be expressed as $q^{\mathrm{T}} M(k)\ q$, where $M_{ij}(k) = ik'\delta_{ij} + \delta_{oi}(\overline{\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \times \underline{\hat{\boldsymbol{y}}}})_i + (1 - \delta_{oi})(1 - \delta_{oi})\left[\overline{(\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \otimes \underline{\hat{\boldsymbol{v}}})}_{ij} - \delta ij \overline{(\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \underline{\hat{\boldsymbol{y}}})}\right]$ is the $4 \times 4$ matrix of the $q$ components. We will now write the matrix $M(k)$ explicitly:

$$M(k) = \left[\begin{array}{c|c} \frac{ik'}{4} & \overline{\underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \times \underline{\hat{\boldsymbol{y}}}} \\ \hline \underline{0} & \frac{ik'}{4}\mathbb{1} + \overline{\underline{\hat{\boldsymbol{y}}}^{\mathrm{T}} \otimes \underline{\hat{\boldsymbol{v}}} - \underline{\hat{\boldsymbol{v}}}^{\mathrm{T}} \underline{\hat{\boldsymbol{y}}}} \end{array}\right]$$

We will substitute $\frac{ik'}{4} = ik$ and we will symmetrise the matrix since in the case of Gaussian distributions it should be a symmetric, positive definite covariance matrix. The symmetrised matrix $M$ is:

$$M(k) = \left[\begin{array}{c|ccc} ik & \frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_x & \frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_y & \frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_z \\ \hline \frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_x & ik + \overline{\hat{v}_1\hat{y}_1 - \underline{\hat{\boldsymbol{v}}} \cdot \underline{\hat{\boldsymbol{y}}}} & \frac{1}{2}(\overline{\hat{y}_1\hat{v}_2 + \hat{y}_2\hat{v}_1}) & \frac{1}{2}(\overline{\hat{y}_1\hat{v}_3 + \hat{y}_3\hat{v}_1}) \\ \frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_y & \frac{1}{2}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2}) & ik + \overline{\hat{v}_2\hat{y}_2 - \underline{\hat{\boldsymbol{v}}} \cdot \underline{\hat{\boldsymbol{y}}}} & \frac{1}{2}(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2}) \\ \frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_z & \frac{1}{2}(\overline{\hat{y}_3\hat{v}_1 + \hat{y}_1\hat{v}_3}) & \frac{1}{2}(\overline{\hat{y}_3\hat{v}_2 + \hat{y}_2\hat{v}_3}) & ik + \overline{\hat{v}_3\hat{y}_3 - \underline{\hat{\boldsymbol{v}}} \cdot \underline{\hat{\boldsymbol{y}}}} \end{array}\right]$$

where $(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_x$ , $(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_y$, $(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_z$ are the $x, y, z$ components of the cross product of $\boldsymbol{y}$ and $\boldsymbol{v}$. Now, expression (5.5.23) can be written as:

$$\frac{1}{2\pi} \iint dk \; d^4q \; \exp\left(-ik'\right) \; \exp\Big(4n \; \left[q^{\mathrm{T}} M(k) \; q\right]\Big) \tag{5.5.24}$$

At this point we return to discuss the integral over the quaternionic parameters that generate the $SO(3)$ rotations. We have chosen to use the $\delta$-function to enforce the constraint that these quaternions are of unit length. In analogy to our discussion of the two-dimensional rotations, we suggest here an alternative approach advocated by Wood [127] which has the benefit of offering another perspective on how our expression above does not favour one rotation over another.

By diagonalising $M(0)$ we can rewrite expression (5.5.24) as in [127] in the form:

$$\int_{S^3} \exp\left(\sum_i \lambda_i \tilde{q}_i^2\right) \; d[\tilde{q}] \tag{5.5.25}$$

Here, the $\tilde{q}_i$ generate rotations in $SO(3)$ which will be uniformly distributed if and only if the $\tilde{q}$ are uniform on a unit hemisphere in $\mathbb{R}^4$. Choosing the usual uniform measure on $S^3$ for $d\tilde{q}$ induces the Haar measure on the space of rotations. This is what (5.5.24) represents only we have chosen to integrate over all quaternions and to impose the constraint through a $\delta$-function. However, an equally valid alternative would be to follow [127] in changing variables to four-dimensionsional spherical polars for which the Jacobian of the transformation would provide the measure on these variables after which we would set the radial component equal to 1 and integrate over the remaining angular variables.

This has been done in [127] in the calculation for the normalisation constant of the Bingham distribution, although his final answer is left in integral form. In this thesis we explore the determination of (5.5.24) as it stands which requires us to find the determinant of the matrix $M$ rather than explicit expression for its eigenvalues. The relationship between (5.5.24) and (5.5.25), however, shows that our choice of

measure on the quaternions is unbiased. This in fact shows that our choice of measure on $\mathbb{R}^4$ induces the correct measure on $SO(3)$.

Returning on our choice of representation and expression (5.5.24), we assume that the eigenvalues of matrix $M$ are negative and thus the evaluation of the quaternionic integral of this multivariate Gaussian distribution is:

$$\frac{1}{2\pi}\int dk\ \exp(-ik')\ \frac{4\ n\ \pi^2}{\sqrt{\det(M)}} \tag{5.5.26}$$

and the marginalised likelihood of expression (5.5.21) is:

$$\mathbb{P}(y|b,\beta,s,\sigma) = \frac{1}{Z}\exp\left[\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\hat{\boldsymbol{y}}) - n\ \mathrm{Var}(\hat{\boldsymbol{v}})] + 2n\ \overline{\underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}}}\right]\times \frac{1}{2\pi}\int dk\ \exp(-ik')\ \frac{4\ n\ \pi^2}{\sqrt{\det(M)}} \tag{5.5.27}$$

The eigenvalues of the matrix $M(k)$ came in pairs of whom one pair was negative and one pair was positive. This means that the integral of the Gaussian form diverges for some values of the data $\boldsymbol{y}$. However, we can still perform the integration with the help of analytic continuation. Analytic continuation is a technique for the extension of the domain of a given function; it is also used to define values where the function is divergent for example in the case of the Gamma function. In our case we can do this because the domain of integration is compact and the original integrand is finite. The extra integration over $k$ was introduced artificially and we only need to find a region of the input data for which this integral converges and the final result must take the same form for any input data. We proceed by calculating the integral for the region of the parameter space for which the eigenvalues are negative and extend this answer to the rest of that space. We will calculate the determinant by expanding it by its first column and write it in terms of a finite power series in $k$ but because the calculations are extremely complicated we will firstly write the four sub-determinants and their results.

Of particular interest is the first sub-determinant because it contributes to the maximum power of the power series of $k$. To make the calculation of the whole

determinant easier we choose to firstly change our coordinate system and align the $z$ component of the $(\overline{\hat{\underline{\boldsymbol{v}}} \times \hat{\underline{\boldsymbol{y}}}})_z$ with the $z$ axis i.e. all the other components of the cross product are zero. Our goal is to write the result in terms of invariant quantities so that the coordinate system plays no role. We evaluate the first sub-determinant of

$$M_1 = ik \times \begin{bmatrix} ik + \overline{\hat{v_1}\hat{y_1} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}} & \frac{1}{2}(\overline{\hat{y_1}\hat{v_2} + \hat{y_2}\hat{v_1}}) & \frac{1}{2}(\overline{\hat{y_1}\hat{v_3} + \hat{y_3}\hat{v_1}}) \\ \frac{1}{2}(\overline{\hat{y_2}\hat{v_1} + \hat{y_1}\hat{v_2}}) & +ik + \overline{\hat{v_2}\hat{y_2} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}} & \frac{1}{2}(\overline{\hat{y_2}\hat{v_3} + \hat{y_3}\hat{v_2}}) \\ \frac{1}{2}(\overline{\hat{y_3}\hat{v_1} + \hat{y_1}\hat{v_3}}) & \frac{1}{2}(\overline{\hat{y_3}\hat{v_2} + \hat{y_2}\hat{v_3}}) & ik + \overline{\hat{v_3}\hat{y_3} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}} \end{bmatrix}$$

Expanding the sub-determinant, we have:

$$\begin{aligned} \det(M_1) = k^4 + 2\ i\ k^3\ \overline{\hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}} + \\ + k^2 \Big[ \frac{1}{4}(\overline{\hat{y_2}\hat{v_3} + \hat{y_3}\hat{v_2}})^2 + \frac{1}{4}(\overline{\hat{y_1}\hat{v_2} + \hat{y_2}\hat{v_1}})^2 + \frac{1}{4}(\overline{\hat{y_1}\hat{v_3} + \hat{y_3}\hat{v_1}})^2 \\ -(\overline{\hat{y_2}\hat{v_2} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}})(\overline{\hat{y_3}\hat{v_3} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}}) + (\overline{\hat{y_1}\hat{v_1} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}})(\overline{\hat{y_1}\hat{v_1} + \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}}) \Big] \\ ik \Big[ -\frac{1}{4}(\overline{\hat{y_2}\hat{v_3} + \hat{y_3}\hat{v_2}})^2(\overline{\hat{y_1}\hat{v_1} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}}) - \frac{1}{4}(\overline{\hat{y_2}\hat{v_1} + \hat{y_1}\hat{v_2}})^2(\overline{\hat{y_3}\hat{v_3} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}}) \\ -\frac{1}{4}(\overline{\hat{y_3}\hat{v_1} + \hat{y_1}\hat{v_3}})^2(\overline{\hat{y_2}\hat{v_2} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}}) \\ +\frac{1}{4}(\overline{\hat{y_3}\hat{v_1} + \hat{y_1}\hat{v_3}})(\overline{\hat{y_2}\hat{v_1} + \hat{y_1}\hat{v_2}})(\overline{\hat{y_3}\hat{v_2} + \hat{y_2}\hat{v_3}}) \\ +(\overline{\hat{y_1}\hat{v_1} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}})(\overline{\hat{y_2}\hat{v_2} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}})(\overline{\hat{y_3}\hat{v_3} - \hat{\underline{\boldsymbol{v}}} \cdot \hat{\underline{\boldsymbol{y}}}}) \Big] \end{aligned} \tag{5.5.28}$$

In the same way, we are going to express all the sub-determinants as finite power series of $k$ and then combine the results to form the final result of the determinant of $M$. The second sub-determinant with respect to the second term of the first column of $M$ is:

$$M_2 = -\tfrac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x \times \begin{bmatrix} \tfrac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x & \tfrac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_y & \tfrac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_z \\ \tfrac{1}{2}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2}) & ik + \overline{\hat{v}_2\hat{y}_2 - \hat{\underline{v}} \cdot \hat{\underline{y}}} & \tfrac{1}{2}(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2}) \\ \tfrac{1}{2}(\overline{\hat{y}_3\hat{v}_1 + \hat{y}_1\hat{v}_3}) & \tfrac{1}{2}(\overline{\hat{y}_3\hat{v}_2 + \hat{y}_2\hat{v}_3}) & ik + \overline{\hat{v}_3\hat{y}_3 - \hat{\underline{v}} \cdot \hat{\underline{y}}} \end{bmatrix}$$

Evaluating the second sub-determinant, we have:

$$\begin{aligned} \det(M_2) = \frac{1}{4}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x^2 \; k^2 + ik \Bigg[ & \frac{1}{4}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x^2 (\overline{\hat{y}_1\hat{v}_1 + \hat{\underline{v}} \cdot \hat{\underline{y}}}) + \frac{1}{8}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x (\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_y (\overline{\hat{y}_1\hat{v}_2 + \hat{y}_1\hat{v}_2}) \\ & + \frac{1}{8}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x (\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_z (\overline{\hat{y}_1\hat{v}_3 + \hat{y}_3\hat{v}_1}) \Bigg] \\ & - \frac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x^2 \left[ \frac{1}{2}(\overline{\hat{y}_2\hat{v}_2 - \hat{\underline{v}} \cdot \hat{\underline{y}}})(\overline{\hat{y}_3\hat{v}_3 - \hat{\underline{v}} \cdot \hat{\underline{y}}}) - \frac{1}{8}(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2})^2 \right] \\ & - \frac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x (\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_y \left[ -\frac{1}{4}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2})(\overline{\hat{y}_3\hat{v}_3 - \hat{\underline{v}} \cdot \hat{\underline{y}}}) \right. \\ & \left. + \frac{1}{8}(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2})(\overline{\hat{y}_1\hat{v}_3 + \hat{y}_3\hat{v}_1}) \right] \\ & - \frac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x (\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_z \left[ -\frac{1}{4}(\overline{\hat{y}_3\hat{v}_1 + \hat{y}_1\hat{v}_3})(\overline{\hat{y}_2\hat{v}_2 - \hat{\underline{v}} \cdot \hat{\underline{y}}}) \right. \\ & \left. + \frac{1}{8}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2})(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2}) \right] \end{aligned} \tag{5.5.29}$$

The last two sub-determinant are related by cyclicity to $M_2$. Thus, the third sub-determinant is:

$$M_3 = \tfrac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_y \times \begin{bmatrix} \tfrac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_x & \tfrac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_y & \tfrac{1}{2}(\overline{\hat{\underline{v}} \times \hat{\underline{y}}})_z \\ ik + \overline{\hat{v}_1\hat{y}_1 - \hat{\underline{v}} \cdot \hat{\underline{y}}} & \tfrac{1}{2}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2}) & \tfrac{1}{2}(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2}) \\ \tfrac{1}{2}(\overline{\hat{y}_3\hat{v}_1 + \hat{y}_1\hat{v}_3}) & \tfrac{1}{2}(\overline{\hat{y}_3\hat{v}_2 + \hat{y}_2\hat{v}_3}) & ik + \overline{\hat{v}_3\hat{y}_3 - \hat{\underline{v}} \cdot \hat{\underline{y}}} \end{bmatrix}$$

Expanding this sub-determinant, we have:

$$\begin{aligned}\det(M_3) = \frac{1}{4}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y^2\ k^2 + ik\Bigg[&\frac{1}{8}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y^2(\overline{\hat{y}_2\hat{v}_2 + \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}}) + \frac{1}{8}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_x(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y(\overline{\hat{y}_1\hat{v}_2 + \hat{y}_1\hat{v}_2})\\ &+\frac{1}{8}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2})\Bigg]\\ &+\frac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y^2\Bigg[-\frac{1}{2}(\overline{\hat{y}_1\hat{v}_1 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}})(\overline{\hat{y}_3\hat{v}_3 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}}) + \frac{1}{8}(\overline{\hat{y}_1\hat{v}_3 + \hat{y}_3\hat{v}_1})^2\Bigg]\\ &+\frac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_x(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y\Bigg[\frac{1}{4}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2})(\overline{\hat{y}_3\hat{v}_3 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}})\\ &\qquad-\frac{1}{8}(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2})(\overline{\hat{y}_1\hat{v}_3 + \hat{y}_3\hat{v}_1})\Bigg]\\ &+\frac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z\Bigg[\frac{1}{4}(\overline{\hat{y}_3\hat{v}_2 + \hat{y}_2\hat{v}_3})(\overline{\hat{y}_1\hat{v}_1 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}})\\ &\qquad-\frac{1}{8}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2})(\overline{\hat{y}_1\hat{v}_3 + \hat{y}_3\hat{v}_1})\Bigg]\end{aligned} \tag{5.5.30}$$

The last sub-determinant of the matrix $M$ is:

$$M_4 = -\tfrac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z \times \begin{bmatrix} \frac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_x & \frac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y & \frac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z \\ ik + \overline{\hat{v}_1\hat{y}_1 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}} & \frac{1}{2}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2}) & \frac{1}{2}(\overline{\hat{y}_1\hat{v}_3 + \hat{y}_3\hat{v}_1}) \\ \frac{1}{2}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2}) & ik + \overline{\hat{v}_2\hat{y}_2 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}} & \frac{1}{2}(\overline{\hat{y}_3\hat{v}_2 + \hat{y}_2\hat{v}_3}) \end{bmatrix}$$

The evaluation of the sub-determinant is:

$$\begin{aligned}\det(M_4) = \frac{1}{4}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z^2\ k^2 + ik\Bigg[&\frac{1}{4}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z^2(\overline{\hat{y}_3\hat{v}_3 + \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}}) + \frac{1}{8}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_x(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z(\overline{\hat{y}_1\hat{v}_3 + \hat{y}_3\hat{v}_1})\\ &+\frac{1}{8}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2})\Bigg]\\ &-\frac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z^2\Bigg[\frac{1}{2}(\overline{\hat{y}_1\hat{v}_1 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}})(\overline{\hat{y}_2\hat{v}_2 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}}) - \frac{1}{8}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2})^2\Bigg]\\ &-\frac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_y(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z\Bigg[-\frac{1}{4}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2})(\overline{\hat{y}_3\hat{v}_3 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}})\\ &\qquad+\frac{1}{8}(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2})(\overline{\hat{y}_1\hat{v}_3 + \hat{y}_3\hat{v}_1})\Bigg]\\ &-\frac{1}{2}(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_x(\overline{\hat{\underline{\boldsymbol{v}}}\times\hat{\underline{\boldsymbol{y}}}})_z\Bigg[-\frac{1}{4}(\overline{\hat{y}_3\hat{v}_1 + \hat{y}_1\hat{v}_3})(\overline{\hat{y}_2\hat{v}_2 - \hat{\underline{\boldsymbol{v}}}\cdot\hat{\underline{\boldsymbol{y}}}})\\ &\qquad+\frac{1}{8}(\overline{\hat{y}_2\hat{v}_1 + \hat{y}_1\hat{v}_2})(\overline{\hat{y}_2\hat{v}_3 + \hat{y}_3\hat{v}_2})\Bigg]\end{aligned} \tag{5.5.31}$$

We will write the overall result of the expansion of the determinant of the matrix $M(k)$ as a finite series of $k$. We now collect all the powers of $k$ and in order to

make the derivation of the calculation easier we make the following substitution: $z_{ij} = \frac{1}{2}(\hat{y}_i\hat{v}_j + \hat{y}_j\hat{v}_i)$ and $z^2 = \underline{\hat{\boldsymbol{v}}} \cdot \underline{\hat{\boldsymbol{y}}}$. We start with the highest terms of the expansion:

$$k^4 + 2\ i\ k^3\ \ \overline{\underline{\hat{\boldsymbol{v}}} \cdot \underline{\hat{\boldsymbol{y}}}} \tag{5.5.32}$$

We follow with the $k^2$ terms:

$$k^2\Big[\frac{1}{4}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_x^2 + \frac{1}{4}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_y^2 + \frac{1}{4}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_z^2 + \frac{1}{4}z_{23}^2 + \frac{1}{4}z_{12}^2 + \frac{1}{4}z_{13}^2 - z_{22}z_{33} + z_{22}z^2 + z_{33}z^2 - z^4 + z_{11}^2 + z_{11}z^2 - z^2z_{11} - z^4\Big] \tag{5.5.33}$$

Making use of the property that $z_{ii}z_{jj} - \frac{1}{4}z_{ij}^2 = z_{ii}z_{jj} - \frac{1}{4}z_{[ij]} - v_iy_jv_jy_i =$ $= -\frac{1}{4}(v \times y)_{ij} + \frac{1}{2}(v \times v)_{ij}(y \times y)_{ij}$, where $z_{[ij]} = z_{ij} - z_{ji}$ is the antisymmetrisation over the indices of $z$ and thus expression (5.5.33) becomes:

$$k^2\left[-|\mathrm{Cov}(v,y)|^2 + \frac{1}{2}|\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}}|^2 - \frac{1}{2}\sum_{ij}(\overline{\hat{\boldsymbol{v}}_{\boldsymbol{i}} \times \hat{\boldsymbol{v}}_{\boldsymbol{j}}}) \cdot (\overline{\hat{\boldsymbol{y}}_{\boldsymbol{i}} \times \hat{\boldsymbol{y}}_{\boldsymbol{j}}})\right] \tag{5.5.34}$$

which has now been written in a manifestly rotationally invariant way. We now examine the first order terms, i.e. all the terms with respect to $k$:

$$\begin{aligned} ik\ \Big[&\frac{1}{4}z^2\Big[(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_x^2 + (\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_y^2 + (\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_z^2\Big] \\ &-z^2\left(z_{11}z_{22} - \frac{1}{4}z_{12}^2 + z_{22}z_{33} - \frac{1}{4}z_{23}^2 + z_{11}z_{33} - \frac{1}{4}z_{13}^2\right) \\ &+\frac{1}{4}z_{13}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_x(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_z + \frac{1}{4}z_{23}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_y(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_z + \frac{1}{4}z_{12}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_x(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_y + \frac{1}{4}z_{33}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_z \\ &+\frac{1}{4}z_{22}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_y + \frac{1}{4}z_{11}(\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}})_x - \frac{1}{4}z_{11}z_{23}^2 - \frac{1}{4}z_{22}z_{13}^2 - \frac{1}{4}z_{33}z_{12}^2 + \frac{1}{4}z_{12}z_{13}z_{23} + z_{11}z_{22}z_{33}\Big] \end{aligned}$$

A long calculation allows us to write the above in terms of invariant quantities so that expression (5.5.35) becomes:

$$ik\left[\frac{1}{2}|\mathrm{Cov}(\underline{\hat{\boldsymbol{v}}}, \underline{\hat{\boldsymbol{y}}})|^2|\overline{\underline{\hat{\boldsymbol{v}}} \times \underline{\hat{\boldsymbol{y}}}}|^2 - \frac{1}{2}\mathrm{Cov}(\underline{\hat{\boldsymbol{v}}}, \underline{\hat{\boldsymbol{y}}})\sum_{ij}(\overline{\hat{\boldsymbol{v}}_{\boldsymbol{i}} \times \hat{\boldsymbol{v}}_{\boldsymbol{j}}}) \cdot (\overline{\hat{\boldsymbol{y}}_{\boldsymbol{i}} \times \hat{\boldsymbol{y}}_{\boldsymbol{j}}}) + \frac{1}{6}\sum_{ijk}\overline{\hat{\boldsymbol{v}}_{\boldsymbol{i}} \cdot (\hat{\boldsymbol{v}}_{\boldsymbol{j}} \times \hat{\boldsymbol{v}}_{\boldsymbol{k}})\ \hat{\boldsymbol{y}}_{\boldsymbol{i}} \cdot (\hat{\boldsymbol{y}}_{\boldsymbol{j}} \times \hat{\boldsymbol{y}}_{\boldsymbol{k}})}\right] \tag{5.5.35}$$

Lastly, we collect from all the sub-determinants all constant terms with no $k$ dependence:

$$
\begin{aligned}
&-\frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_x(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_z\left[\frac{1}{4}z_{12}z_{23}-\frac{1}{2}z_{13}z_{22}+\frac{1}{2}z^2z_{13}\right]\\
&-\frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_y(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_z\left[\frac{1}{4}z_{13}z_{12}-\frac{1}{2}z_{23}z_{11}+\frac{1}{2}z^2z_{23}\right]\\
&-\frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_x(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_y\left[\frac{1}{4}z_{32}z_{13}-\frac{1}{2}z_{12}z_{33}+\frac{1}{2}z^2z_{12}\right]\\
&-\frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_x^2\left[-\frac{1}{8}(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_x^2+\frac{1}{4}((\overline{\hat{\boldsymbol{v}}_{\boldsymbol{i}}\times\hat{\boldsymbol{v}}_{\boldsymbol{j}}})_x\cdot(\overline{\hat{\boldsymbol{y}}_{\boldsymbol{i}}\times\hat{\boldsymbol{y}}_{\boldsymbol{j}}}))_x+\frac{1}{2}z^2z_{11}\right]\\
&-\frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_y^2\left[-\frac{1}{8}(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_y^2+\frac{1}{4}((\overline{\hat{\boldsymbol{v}}_{\boldsymbol{i}}\times\hat{\boldsymbol{v}}_{\boldsymbol{j}}})_y\cdot(\overline{\hat{\boldsymbol{y}}_{\boldsymbol{i}}\times\hat{\boldsymbol{y}}_{\boldsymbol{j}}}))_y+\frac{1}{2}z^2z_{22}\right]\\
&-\frac{1}{2}(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_z^2\left[-\frac{1}{8}(\overline{\underline{\hat{\boldsymbol{v}}}\times\underline{\hat{\boldsymbol{y}}}})_z^2+\frac{1}{4}((\overline{\hat{\boldsymbol{v}}_{\boldsymbol{i}}\times\hat{\boldsymbol{v}}_{\boldsymbol{j}}})_z\cdot(\overline{\hat{\boldsymbol{y}}_{\boldsymbol{i}}\times\hat{\boldsymbol{y}}_{\boldsymbol{j}}}))_z+\frac{1}{2}z^2z_{33}\right]
\end{aligned}
\tag{5.5.36}
$$

Writing the above in terms of invariant quantities, we have:

$$
\begin{aligned}
-\frac{1}{4}\mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})\ \left[\hat{\boldsymbol{v}}\cdot(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}})\ \hat{\boldsymbol{y}}\cdot(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}})\right]+\frac{1}{16}|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2\cdot|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2\\
-\frac{1}{8}\left[(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}})\cdot(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\right]\ \left[(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}})\cdot(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\right]
\end{aligned}
\tag{5.5.37}
$$

Collecting all invariant terms and all powers of $k$, the final result of the determinant of the matrix $M$ is:

$$
\begin{aligned}
\det(M)=k^4-2\ \mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})\ ik^3-k^2\left[|\mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})|^2-\frac{1}{2}|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2+\frac{1}{2}(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})\right]\\
+\frac{1}{2}\ ik\ \left[\mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})\ |\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2-\mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})+\frac{1}{3}\overline{\hat{\boldsymbol{v}}\cdot(\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}})\ \hat{\boldsymbol{y}}\cdot(\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}})}\right]\\
+\frac{1}{16}|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^4-\frac{1}{8}(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}})\cdot(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\ (\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})-\frac{1}{4}\ \mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})\ \overline{\hat{\boldsymbol{v}}\cdot(\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}})\ \hat{\boldsymbol{y}}\cdot(\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}})}
\end{aligned}
\tag{5.5.38}
$$

Thus, we can now evaluate the integral over the quaternionic space, which from expression (5.5.27) is:

$$
\begin{aligned}
\mathbb{P}(y|b,\beta,s,\sigma)=\frac{1}{Z}\exp\left[\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\hat{\boldsymbol{y}})-n\ \mathrm{Var}(\hat{\boldsymbol{v}})]+2n\ \overline{\underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}}}\right]\times\\
\frac{1}{2\pi}\int dk\ \exp(-ik')\ \frac{4\ n\ \pi^2}{\sqrt{\det(M)}}
\end{aligned}
\tag{5.5.39}
$$

with $\det(M)$ the evaluated determinant of expression (5.5.38) which has $k$ dependence and is invariant to rotations of both $y$ and $v$ since it has been written in a manifestly rotationally invariant way. It is common practice to evaluate integrals of this form by contour integration. For this we would have to promote $k$ to the complex plane and choose an appropriate path in the $k$-plane. The residue theorem is a tool for evaluating such integrals and requires finding the poles of the function i.e. the points at which the function diverges as $\frac{1}{k-k_0}$. The presence of the square root in the denominator however instead of turning points at which the denominator of (5.5.39) vanishes into poles, it turns them into branch cuts making the integration over these extremely difficult. The expression of the determinant is not a perfect square and thus contour integration cannot be of help and the integral over $k$ cannot be done analytically. One of our attempts was the evaluation of the integral by a Laplace approximation. To do so, we would have to bring expression (5.5.39) into the following form:

$$\frac{1}{2\pi}\int dk\ \exp(-ik')\ \frac{4\ n\ \pi^2}{\sqrt{\det(M)}} = 2\ n\ \pi\ \int dk\ \ \exp\left(-ik' - \frac{1}{2}\log(\det M(k))\right)$$
$$\approx \exp\left(-4ik_0 - \frac{1}{2}\log(\det M(k_0))\right)\sqrt{\frac{2\pi}{|(4ik_0 + \frac{1}{2}\log(\det M(k_0)))''|}} \tag{5.5.40}$$

where we would have to evaluate the roots $k_0$ of the denominator and the second derivative of the exponent. Although the roots were intractable and impossible to be found in a closed form one could always evaluate them numerically. However, we came up with a more realistic solution which was the "analytic" evaluation by Taylor expanding the square root of the determinant. This is the approach that we present next.

To Taylor expand the square root of the determinant, we write it in the form $\sqrt{det(M)} = \sqrt{C^2 + D}$ so that:

$$\frac{1}{\sqrt{\det(M)}} = \frac{1}{\sqrt{C^2(k)+D(k)}} = \frac{1}{C(k)\sqrt{1+D(k)/C^2(k)}}$$
$$= \frac{1}{C(k)}\left[1-\frac{1}{2}\frac{D(k)}{C^2(k)}\right] = \frac{1}{C(k)} - \frac{1}{2}\frac{D(k)}{C^3(k)} + .... \tag{5.5.41}$$

so that expression (5.5.39) becomes:

$$\mathbb{P}(y|b,\beta,s,\sigma) = \frac{1}{Z}\exp\left[\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\hat{\boldsymbol{y}}) - n\ \mathrm{Var}(\hat{\boldsymbol{v}})] + 2n\ \underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}}\right]\times$$
$$\frac{4\ n\ \pi^2}{2\pi}\int dk\ \left[\frac{\exp(-ik')}{C(k)} - \frac{1}{2}\frac{D(k)\ \exp(-ik')}{C^3(k)} + ....\right] \tag{5.5.42}$$

For the Taylor expansion of the determinant, we complete the square of the result in (5.5.38) and subtract all extra terms that have arisen from completing the square. We now write expression (5.5.38) as:

$$C^2(k) = k^4 - 2\ \mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})\ ik^3 - k^2\left[|\mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})|^2 - \frac{1}{2}|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2 + \frac{1}{2}(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})\right]$$
$$+\frac{1}{2}\ ik\ \left[\mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})\ |\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2 - \mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})\right]$$
$$+\frac{1}{16}|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^4 + \frac{1}{4}\left[(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})\right]^2 - \frac{1}{8}|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}}) =$$
$$= (k^2 - ikA + B)^2 \tag{5.5.43}$$

with $A = \mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})$ and $B = \frac{1}{2}|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2 - \frac{1}{2}(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})$. We now subtract/add the extra terms of the completion. For $D$ now, we have:

$$D(k) = \frac{1}{6}\ ik\ \overline{\hat{\boldsymbol{v}}\cdot(\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}})}\ \overline{\hat{\boldsymbol{y}}\cdot(\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}})} - \frac{1}{16}\left[(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})\right]^2 + \frac{1}{8}|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2\left[(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})\right]$$
$$-\frac{1}{8}(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}})\cdot(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\ (\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}}) - \frac{1}{4}\ \mathrm{Cov}(\hat{\boldsymbol{y}},\hat{\boldsymbol{v}})\ \overline{\hat{\boldsymbol{v}}\cdot(\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}})\ \hat{\boldsymbol{y}}\cdot(\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}})} \tag{5.5.44}$$

For calculational simplicity and to illustrate the idea, we choose to only include terms up to second order of the Taylor expansion. Having the expression written

as such, we will be able to carry out the integral with respect to $k$ by employing contour integration techniques. To do so, we choose the appropriate contour for our integration. Before we proceed, we introduce some aspects of contour integration. Contour integration allows us to carry out integrals in the complex plain around a chosen contour. The evaluation of the integrals is done by summing the values of the residues inside the contour which according to the Residue Theorem can be calculated by:

$$\int_C f(z)\ dz = 2\ \pi\ i\ \sum_{z_o} \text{Res}(z \to z_o) \tag{5.5.45}$$

where $\text{Res}(f) = a_{-1}$ of the Laurent expansion of $f$ about the point $z = z_o$ which is also called a **pole** and $a_{-1}$ is the coefficient of $(z - z_o)^{-1}$ of the Laurent expansion. Contour integrals are calculated by enclosing the poles inside the contour and then summing their residues. The residue at a pole of order $n$ is calculated by:

$$\text{Res} = \frac{1}{(n-1)!}\frac{d^{n-1}}{dz^{n-1}}\ (z - z_0)^n\ f(z)\ \Big|_{z=z_o} \tag{5.5.46}$$

Now that our integral is in an appropriate form for contour integration we need to find, according the Residue Theorem, the poles i.e. the roots of the denominators of expression (5.5.42). Firstly we evaluate the roots of expression (5.5.43). The roots are:

$$k_{o,1} = \frac{iA \pm i\sqrt{A^2 - 4B}}{2} = \frac{i\ \text{Cov}^2(\hat{\boldsymbol{v}}, \hat{\boldsymbol{y}}) \pm i\sqrt{\text{Cov}^2(\hat{\boldsymbol{v}}, \hat{\boldsymbol{y}}) - |\overline{\hat{\boldsymbol{v}} \times \hat{\boldsymbol{y}}}|^2 + \overline{(\hat{\boldsymbol{v}} \times \hat{\boldsymbol{v}}) \cdot (\hat{\boldsymbol{y}} \times \hat{\boldsymbol{y}})}}}{2} \tag{5.5.47}$$

We would now have to evaluate the poles of the second term of the contour integral in (5.5.42), however we notice that the poles are the same as with the first term but of order 3. We now evaluate the contour integral and we choose to close the contour on the lower half plane so that the integral converges as $k \to -i\infty$, then $-ik \to -\infty$. Its evaluation is then:

$$\int dk \left[ \frac{\exp(-ik')}{C(k)} - \frac{1}{2}\frac{D(k)\ \exp(-ik')}{C^3(k)} + .... \right] = \frac{e^{-ik_1}}{(k_1 - k_0)} - \frac{1}{2}\frac{d^2}{dk^2}\ \frac{D(k)\ (e^{-ik})}{(k-k_0)^3}\Big|_{k=k_1} \tag{5.5.48}$$

We now evaluate the derivatives involved in the evaluation of the contour integral of expression (5.5.48). The first derivative of the second term of expression (5.5.48) is:

$$\begin{aligned} \frac{d}{dk}\left[ \frac{D(k)\ (e^{-ik})}{(k-k_0)^3} \right] &= \frac{d}{dk}\left[D(k)\ (e^{-ik})\right]\frac{1}{(k-k_0)^3} - \frac{3\ D(k)\ (e^{-ik})}{(k-k_0)^4} = \\ &= \left[-i\ D(k)\ (e^{-ik}) + e^{-ik}\ D'(k)\right]\frac{1}{(k-k_0)^3} - \frac{3\ D(k)\ (e^{-ik})}{(k-k_0)^4} \end{aligned} \tag{5.5.49}$$

with $D'(k) = \frac{1}{6}\ i\ \overline{\hat{\boldsymbol{v}} \cdot (\hat{\boldsymbol{v}} \times \hat{\boldsymbol{v}})\ \hat{\boldsymbol{y}}(\hat{\boldsymbol{y}} \times \hat{\boldsymbol{y}})}$

The second derivative of the second term in (5.5.48) is:

$$\begin{aligned} &\frac{d}{dk}\left[\left[-i\ D(k)\ (e^{-ik}) + e^{-ik}\ D'(k)\right]\frac{1}{(k-k_0)^3} - \frac{3\ D(k)\ (e^{-ik})}{(k-k_0)^4}\right] = \\ &= \frac{-3}{(k-k_1)^4}\left[-i\ D(k)\ (e^{-ik}) + e^{-ik}\ D'(k)\right] + \frac{1}{(k-k_0)^3}\left[-\ D(k)\ (e^{-ik}) - 2\ i\ e^{-ik}\ D'(k)\right] \\ &+ \frac{12\ D(k)\ (e^{-ik})}{(k-k_0)^5} - \frac{3}{(k-k_0)^4}\left[-i\ D(k)\ (e^{-ik}) + e^{-ik}\ D'(k)\right] \end{aligned} \tag{5.5.50}$$

Overall, we combine expressions (5.5.49) and (5.5.50) of expression (5.5.48) and form the integral over the quaternionic space and $k$. The derivatives will have to be evaluated at $k = k_1$. The final integral over rotations evaluates at this order in the Taylor expansion as:

$$
\begin{aligned}
\mathbb{P}(y|b,\beta,s,\sigma) &= \frac{1}{Z}\exp\left[\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\hat{\boldsymbol{y}}) - n\ \mathrm{Var}(\hat{\boldsymbol{v}})] + 2n\ \overline{\underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}}}\right]\times \\
&\qquad \frac{1}{2\pi}\int dk\ \exp(-ik')\frac{4\ \pi\ n^2}{\sqrt{det(M)}} \\
&= \frac{1}{Z}\exp\left[\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\hat{\boldsymbol{y}}) - n\ \mathrm{Var}(\hat{\boldsymbol{v}})] + 2n\ \overline{\underline{\hat{\boldsymbol{y}}}\cdot\underline{\hat{\boldsymbol{v}}}}\right]\times \\
&\qquad \frac{4\ n\ \pi^2}{2\pi}\int dk\ \left[\frac{\exp(-ik')}{C(k)} - \frac{1}{2}\frac{D(k)\ \exp(-ik')}{C^3(k)} + ....\right] \\
&= \frac{2\ n\ \pi}{Z}\exp\left[\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\boldsymbol{y}) - n\ \mathrm{Var}(\boldsymbol{v})] + 2n\ \mathrm{Cov}(\hat{\boldsymbol{v}},\hat{\boldsymbol{y}})\right]\times \\
&\quad \left[\frac{i\ e^{-ik_1}}{(k_1-k_0)} + \frac{3}{2}\frac{1}{(k_1-k_0)^4}\left(-i\ D(k)\ (e^{-ik_1}) + e^{-ik_1}\ D'(k)\right)\right. \\
&\quad + \frac{1}{(k_1-k_0)^3}\left(-\ D(k)\ (e^{-ik_1}) - 2\ i\ e^{-ik_1}\ D'(k)\right) \\
&\quad \left. + \frac{12\ D(k)\ (e^{-ik_1})}{(k_1-k_0)^5} - \frac{3}{(k_1-k_0)^4}\left(-i\ D(k)\ (e^{-ik_1}) + e^{-ik_1}\ D'(k)\right)\right]
\end{aligned}
\tag{5.5.51}
$$

However we notice that $k_1 - k_0 = -\sqrt{A^2 - 2B}$, where $A = \mathrm{Cov}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{v}})$ and $B = \frac{1}{2}|\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{y}}}|^2 - \frac{1}{2}(\overline{\hat{\boldsymbol{v}}\times\hat{\boldsymbol{v}}})\cdot(\overline{\hat{\boldsymbol{y}}\times\hat{\boldsymbol{y}}})$. We make the substitution in the result above so that:

$$
\begin{aligned}
\mathbb{P}(y|b,\beta,s,\sigma) &= \frac{2\ n\ \pi}{Z}\exp\left[\left(\frac{1}{2\sigma^2}\right)[-n\ \mathrm{Var}(\boldsymbol{y}) - n\ \mathrm{Var}(\boldsymbol{v})] + 2n\ \mathrm{Cov}(\hat{\boldsymbol{v}},\hat{\boldsymbol{y}})\right]\times \\
&\quad \left[\frac{i\ e^{-ik_1}}{(-\sqrt{A^2-2B})} + \frac{3}{2}\frac{1}{(-\sqrt{A^2-2B})^4}\left(-i\ D(k)\ (e^{-ik_1}) + e^{-ik_1}\ D'(k)\right)\right. \\
&\quad + \frac{1}{(-\sqrt{A^2-2B})^3}\left(-\ D(k)\ (e^{-ik_1}) - 2\ i\ e^{-ik_1}\ D'(k)\right) \\
&\quad \left. + \frac{12\ D(k)\ (e^{-ik_1})}{(-\sqrt{A^2-2B})^5} - \frac{3}{(-\sqrt{A^2-2B})^4}\left(-i\ D(k)\ (e^{-ik_1}) + e^{-ik_1}\ D'(k)\right)\right]
\end{aligned}
\tag{5.5.52}
$$

with the following being:

$$D(k) = \frac{1}{6}\, ik\, \overline{\hat{\boldsymbol{v}} \cdot (\hat{\boldsymbol{v}} \times \hat{\boldsymbol{v}})}\; \overline{\hat{\boldsymbol{y}} \cdot (\hat{\boldsymbol{y}} \times \hat{\boldsymbol{y}})} - \frac{1}{16}\left[(\overline{\hat{\boldsymbol{v}} \times \hat{\boldsymbol{v}}}) \cdot (\overline{\hat{\boldsymbol{y}} \times \hat{\boldsymbol{y}}})\right]^2 + \frac{1}{8}|\overline{\hat{\boldsymbol{v}} \times \hat{\boldsymbol{y}}}|^2 \left[(\overline{\hat{\boldsymbol{v}} \times \hat{\boldsymbol{v}}}) \cdot (\overline{\hat{\boldsymbol{y}} \times \hat{\boldsymbol{y}}})\right] - \frac{1}{8}(\overline{\hat{\boldsymbol{v}} \times \hat{\boldsymbol{y}}}) \cdot (\overline{\hat{\boldsymbol{v}} \times \hat{\boldsymbol{v}}})\; (\overline{\hat{\boldsymbol{v}} \times \hat{\boldsymbol{y}}}) \cdot (\overline{\hat{\boldsymbol{y}} \times \hat{\boldsymbol{y}}}) - \frac{1}{4}\, \mathrm{Cov}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{v}})\; \overline{\hat{\boldsymbol{v}} \cdot (\hat{\boldsymbol{v}} \times \hat{\boldsymbol{y}})\; \hat{\boldsymbol{y}} \cdot (\hat{\boldsymbol{v}} \times \hat{\boldsymbol{y}})} \tag{5.5.53}$$

$$D'(k) = \frac{1}{6}\, i\, \overline{\hat{\boldsymbol{v}} \cdot (\hat{\boldsymbol{v}} \times \hat{\boldsymbol{v}})\; \hat{\boldsymbol{y}} (\hat{\boldsymbol{y}} \times \hat{\boldsymbol{y}})} \tag{5.5.54}$$

$$k_1 = \frac{i\, \mathrm{Cov}^2(\hat{\boldsymbol{v}}, \hat{\boldsymbol{y}}) - i\sqrt{\mathrm{Cov}^2(\hat{\boldsymbol{v}}, \hat{\boldsymbol{y}}) - |\overline{\hat{\boldsymbol{v}} \times \hat{\boldsymbol{y}}}|^2 + \overline{(\hat{\boldsymbol{v}} \times \hat{\boldsymbol{v}}) \cdot (\hat{\boldsymbol{y}} \times \hat{\boldsymbol{y}})}}}{2} \tag{5.5.55}$$

The result above constitutes the final result of the integration of likelihood with respect to both translations and rotations. This result would then be used for the approximation of the complete likelihood as:

$$\mathbb{P}(y|C) = \propto \sum_{b \in \mathcal{B}} \int \; \mathscr{D}\beta\; \mathscr{D}s\; d\sigma\; \mathbb{P}(y|b, \beta, s, \sigma)\mathbb{P}(\beta|C)\mathbb{P}(s)\mathbb{P}(\sigma)\mathbb{P}(C) \tag{5.5.56}$$

by evaluating the remaining integrals with respect to bijections, curves (which in this case would be two dimensional surfaces), samplings and the noise parameter $\sigma$. Had we taken scalings $a$ into consideration for the similarity transformations, we would also have to integrate over scalings as well. Forming the approximation of the likelihood, we could then then perform MAP and classify three dimensional shapes into their respective categories.

## 5.6 Concluding remarks

To summarise, the above constitutes the final result of the integration over translations and rotations of the marginalised likelihood. The space of three dimensional rotations is extremely complicated and so is the result of the integration over it. Due to these complications, our attempts over the analytic calculation of the integrals of similarity transformations have stopped and we weren't able to complete the Maximum a Posteriori of expression (5.5.56). This is one of the reasons we did

not include scalings in our initial formulation of the three-dimensional observation model in (5.2.2), only because including one extra parameter would complicate the calculations even more. However, being able to integrate over three-dimensional rotations is a very interesting result on its own although we did not reach our desired result of integrating all similarity transformations. It is worth mentioning here, that although we achieved to integrate over three-dimensional rotations, the result of the integration is just an approximation since we have Taylor expanded the integrand. This approximation depends on the data $y$ since in their turn depend on the terms $C(k)$ and $D(k)$ of which we don't know the magnitude. We suggest that this could constitute future expansion of the current work.

# Chapter 6

# Discussion, open questions and conclusion

In this chapter we give an overview of the results presented in this work, we discuss the computational results and present some open questions and suggestions for future work. We have only initiated a study of classification techniques especially in the case of three dimensions. Since this work has not been completed properly there is still much to consider for future work. We can only be encouraged by the results and hope in further success in the future.

In chapter [2] of this work we have presented previous work that has been done in the classification of continuous, planar shapes by Srivastava and Jermyn [12]. In their work, they perform classification by maximising the posterior probability of a class given the observed data $\mathbb{P}(C|y)$ under the assumption that the likelihood $\mathbb{P}(y|C)$ can be broken down into components. The process involves the marginalisation of the likelihood with respect to nuisance parameters that are involved into the data formation process imposing prior distributions on these parameters; however, the marginalisation of the likelihood introduces complex integrals and sums over the nuisance parameters. In [12], the integrations and summations of the marginalised likelihood and hence the maximisation of the posterior are evaluated by using approximation algorithms. In this work, we have presented a way of evaluating some of the integrals in a closed form; in particular, we integrated over similarity transforma-

tions namely translations, scalings and rotations. To evaluate this five dimensional integral we imposed a Jeffreys prior that reflect our ignorance on the space of the parameters we integrate over. The result of the integration however was found to be divergent due to the nature of the likelihood but also due to the fact that Jeffreys prior was improper. To alleviate the problem that the integration introduced we regularised Jeffreys prior. This meant that we made an appropriate choice of combined priors that both reflected our beliefs about the parameters over which we imposed them but also the end result of the integration was of the same form as the result of the integration against Jeffreys prior. To interchange between the results one needs to remove the regulators by taking their limit to appropriate values. The regularisation allowed us to firstly overcome the divergences and secondly to arrive to a closed form solution that could be used as a computational algorithm for the classification of observed data shapes.

In chapter [3] we have presented the experimental results of the algorithm proposed in chapter [2]. To evaluate the confidence results and success rates of the algorithm we simulated data coming from three different databases: the Kimia, the alphabet and the sand body database. For each of the databases we examined the confidence results the algorithm gives when varying different parameters such as the Monte Carlo iterations of the saplings or the Monte Carlo iterations of the curves. For the Kimia database we found that for 10 runs of 10 shapes each, the average classification level was $\hat{\mu} = 59\% \pm 7\%$ with the average success rate being more than $80\% \pm 5\%$. We also concluded that as soon as the number of points increases to more than 50 the confidence levels become almost 90 percent. For the alphabet database, we repeated the experiment for the same number of runs, and found that the average classification level was $\hat{\mu} = 77\% \pm 5\%$ with the average success rate $73\% \pm 6\%$. We also concluded that when the number of points increases to more than 50 the classification levels are more than 80%. We have also identified that the algorithm is sensitive in the presence of too few points or too high noise however it seems that a high number of points can contemplate the high noise. The high computational cost is also one of the drawbacks of the algorithm since for the minimum value of sampling iterations and more than 50 points the computational time can take more

than 5.5 hours. In the last part of this chapter we made use of the sand body database. To perform classification of sand bodies, we tries to learn the parameters of the models through a set of training data. The learning was done via the maximisation of the log likelihood with respect to the parameters we wanted to learn. Due to the nature of the log likelihood for the maximisation we had to employ an optimisation algorithm for which case we chose the gradient ascent. However, the gradient ascent was trapped in several local maxima which implied that the likelihood surface was quite uneven and unsmooth. Due to the unsatisfactory results of the learning of the parameters, to test the classification efficacy of the algorithm in the case of the sand bodies we used the parameters by which we assume that the data shapes are generated. The classification results were the following: in the case of ribbons the average success rate was $67\% \pm 3\%$ with the average confidence levels $90\% \pm 2\%$. In the case of sheets, the average success rate was $80\% \pm 3\%$ with an average classification confidence of $82\% \pm 2\%$. Lastly, for the third simulated class of triangles, the average success rate was found to be $87\% \pm 3\%$ with the average classification confidence to be $91\% \pm 1\%$. One can conclude that the behaviour of our algorithm is extremely satisfactory since the classification levels and the success rates are more than 80%. This illustrates the importance of the fact that our algorithm incorporates the geometry of each of the shapes since this is the feature that purely characterises them. The results also illustrate that our algorithm is a very powerful tool for the classification of geological sand bodies and considering the experimental results presented in chapter [4], where we present a classification method similar in nature to the width-to-thickness ratio, we conclude that our algorithm excels current geological classification methods.

In chapter [4], we compared our classification method to the classification of the data shapes when only using a small subset of features that explain each class of shapes. For each shape we extracted the following features: shape factor, roundness, convexity and solidity. These four features constituted the feature vector of each of the shapes. Assuming that each of the classes can be described by a multidimensional Gaussian so that the distribution of the data can explained as a mixture of multidimensional Gaussians. By using the Expectation-Maximisation (EM) al-

gorithm we inferred the number of existing clusters in the data and the statistical properties that describe each class. Based on these properties we classified new, unobserved data in the classes as they were decided by the EM. The average success rate of this process in the case of the Kimia database was $62.9\% \pm 4.1\%$ for 21 components and $72.7\% \pm 4.5\%$ for 13 components. In the case of the alphabet database, the success rates were $33.1\% \pm 5.6\%$ for 26 components and $37.3\% \pm 5.4\%$ for 17 components. Although the success rates of this classification method seem high and comparable to the results we acquired with our proposed method, in the Kimia case, it fails to capture the variability of each individual class since through this one cannot distinguish the differences between different classes since they are found to be described by the same properties. In the case of sand bodies, we suggested an adaptation of the EM algorithm which employs our method and assumes that the available classes can be described by a mixture of our observational model. However, the scores could not be maximised in a closed form and thus once again an optimisation algorithm had to be utilised. The gradient ascent algorithm returned the same unsatisfactory results as in chapter [4] due to the unevenness of the likelihood hypersurface.

The final chapter, chapter [5], was an attempt to extend the work of chapter [2] in three dimensions. In particular, we tried to perform classification through MAP of a class which as in chapter [2] could be approximated via the marginalisation of the likelihood over the nuisance parameters that take part in the data formation process. To follow the steps of the two dimensional case, we tried to integrate over similarity transformations. The integration of translations has brought similar results as the two dimensional case. The difficulty was the integration of three dimensional rotations. To perform the integration we chose to represent rotations by unit quaternions and performed the integration over the isomorphic $\mathbb{R}^4$ by imposing a delta function that only encounters the unit quaternions that live on the surface of the unit 3-ball. Although the integral over quaternions could be evaluated up to a point, the final step was impossible due to the form of the integrand which prohibited the use of contour integration. However, we came up with a realistic solution which was the "analytic" evaluation by a Taylor expansion of the integrand which allowed

us to perform the desired contour integration and reach the final result over three dimensional rotations.

An overall note and conclusion is that the proposed algorithm is a powerful tool for the classification of planar shapes and gives extremely accurate results had certain parameters chosen to be above a certain threshold. A problem that could be examined in the future is the maximisation of the likelihood with respect to some of the parameters in the case of the sand body database. A suggestion would be the employment of other optimisation algorithms or the scanning of the likelihood surface more closely or in a different way. With regards to the results of the final chapter, although we have gone far with the evaluation of the integrals over translations and rotations and the result is interesting on its own, we have not completed the integration over all similarity transformations and our attempts have stopped before we completed the classification process. Classification of three dimensional shapes is an interesting branch of shape analysis and has recently expanded quite rapidly. There are many open questions left from this problem. Firstly, one could encounter the integration of scalings of three-dimensional shapes. Furthermore, a more interesting problem is to tackle the sampling of two dimensional surface in the same manner as in the case of two dimensions. That would suggest the use of a similar integration technique and a similar generalised Gaussian prior that "favours" even samplings of surfaces. In addition, an intriguing problem to solve would be the summation of bijections for the three-dimensional case if one bears in mind the complexity this part has even in the two-dimensional case. The problem we posed in chapter [5] remains open and incomplete and thus we believe that it is one of our future works and expansions of the present thesis in hope that will offer to the problem of classification of three-dimensional shapes since it is a problem of great importance and of great interest in the present days.

# Appendix A

### A.0.1 Expansion of the determinant for Fisher information matrix

In this section we will calculate the determinant of Fisher's information matrix. The matrix is:

$$\mathcal{I}(\phi) = \begin{array}{c} \\ \sigma \\ a \\ c_x \\ c_y \\ \theta \end{array} \begin{array}{c} \begin{array}{ccccc} \sigma & a & c_x & c_y & \theta \end{array} \\ \begin{pmatrix} -\frac{4n}{\sigma^2} & 0 & 0 & 0 & 0 \\ 0 & -\frac{\sum_i |\boldsymbol{v_i}|^2}{\sigma^2} & \frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & \frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & 0 \\ 0 & \frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & -\frac{n}{\sigma^2} & 0 & -a\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} \\ 0 & \frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & 0 & -\frac{n}{\sigma^2} & a\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} \\ 0 & 0 & -a\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & a\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & -a^2\frac{\sum_i |\boldsymbol{v_i}|^2}{\sigma^2} \end{pmatrix} \end{array}$$

where we have substituted: $\boldsymbol{v_i}^{y'} = (\sin\theta \boldsymbol{v_i}^x + \cos\theta \boldsymbol{v_i}^y)$ and $\boldsymbol{v_i}^{x'} = (\cos\theta \boldsymbol{v_i}^x - \sin\theta \boldsymbol{v_i}^y$. Expanding by the first row the determinant of the above matrix is:

$$-\frac{4n}{\sigma^2}\begin{vmatrix} -\frac{\sum_i |\boldsymbol{v_i}|^2}{\sigma^2} & \frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & \frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & 0 \\ \frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & -\frac{n}{\sigma^2} & 0 & -a\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} \\ \frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & 0 & -\frac{n}{\sigma^2} & a\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} \\ 0 & -a\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & a\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & -a^2\frac{\sum_i |\boldsymbol{v_i}|^2}{\sigma^2} \end{vmatrix} \tag{A.0.1}$$

Expanding the determinant by the first row, the first expanded term will be:

$$\left(-\frac{4n}{\sigma^2}\right)\left(-\frac{\sum_i |\boldsymbol{v_i}|^2}{\sigma^2}\right)\begin{vmatrix} -\frac{n}{\sigma^2} & 0 & -a\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} \\ 0 & -\frac{n}{\sigma^2} & a\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} \\ -a\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & a\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & -a^2\frac{\sum_i |\boldsymbol{v_i}|^2}{\sigma^2} \end{vmatrix} =$$

$$= \frac{4n\sum_i |\boldsymbol{v_i}|^2}{\sigma^4}\left\{-\frac{n}{\sigma^2}\left[\frac{a^2 n}{\sigma^4}\sum_j |\boldsymbol{v_j}|^2 - \frac{a^2}{\sigma^4}\sum_j \boldsymbol{v_j}^{x'}\sum_k \boldsymbol{v_k}^{x'}\right] - \frac{a}{\sigma^2}\sum_j \boldsymbol{v_j}^{y'}\left[-\frac{an}{\sigma^4}\sum_k \boldsymbol{v_k}^{k'}\right]\right\}$$

$$= \frac{4n\sum_i |\boldsymbol{v_i}|^2}{\sigma^4}\left\{-\frac{a^2 n^2}{\sigma^2}\left[\sum_j |\boldsymbol{v_j}|^2 - \frac{1}{n}\sum_j\sum_k \boldsymbol{v_j}^{x'}\boldsymbol{v_k}^{x'}\right] - \frac{a^2 n}{\sigma^6}\sum_j\sum_k \boldsymbol{v_j}^{y'}\boldsymbol{v_k}^{y'}\right\} =$$

$$= \frac{4n\sum_i |\boldsymbol{v_i}|^2}{\sigma^4}\left\{-\frac{a^2 n^2}{\sigma^6}\left[\sum_j |\boldsymbol{v_j}|^2 - \frac{1}{n}\left[\sum_j\sum k\boldsymbol{v_j}^{x'}\boldsymbol{v_k}^{x'} + \sum_j\sum_k \boldsymbol{v_j}^{y'}\boldsymbol{v_k}^{y'}\right]\right]\right\} =$$

$$= -\frac{4n\ a^2 n^2\sum_i |\boldsymbol{v_i}|^2}{\sigma^{10}}\left[\underbrace{\sum_j |\boldsymbol{v_j}|^2 - \frac{1}{n}\sum_j\sum_k \boldsymbol{v'_j}\boldsymbol{v'_k}}_{K}\right] \tag{A.0.2}$$

The second expanded term will be:

$$\left(-\frac{4n}{\sigma^2}\right)\left(-\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2}\right)\begin{vmatrix} \frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & 0 & -a\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} \\ \frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & -\frac{n}{\sigma^2} & a\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} \\ 0 & a\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & -a^2\frac{\sum_i |\boldsymbol{v_i}|^2}{\sigma^2} \end{vmatrix} =$$

$$= \frac{4n\sum_i \boldsymbol{v_i}^{x'}}{\sigma^4}\left\{\frac{\sum_j \boldsymbol{v_j^{x'}}}{\sigma^2}\left[\frac{a^2 n\sum_k |\boldsymbol{v_k}|^2}{\sigma^4} - \frac{a^2}{\sigma^4}\sum_k \boldsymbol{v_k}^{x'}\sum_l \boldsymbol{v_l}^{x'}\right]\right.$$
$$\left. - \frac{a\sum_l \boldsymbol{v_l}^{y'}}{\sigma^2}\left[\frac{a}{\sigma^4}\sum_k \boldsymbol{v_k}^{y'}\sum_l \boldsymbol{v_l}^{x'}\right]\right\}$$

$$= \frac{4n\sum_i \boldsymbol{v_i}^{x'}}{\sigma^4}\left\{\frac{a^2 n\sum_j \boldsymbol{v_j}^{x'}}{\sigma^6}\left[\sum_k |\boldsymbol{v_k}|^2 - \frac{1}{n}\sum_k\sum_l \boldsymbol{v_k}^{x'}\boldsymbol{v_l}^{x'}\right]\right.$$
$$\left. - \frac{a^2\sum_l \boldsymbol{v_l}^{y'}}{\sigma^6}\left(\sum_k \boldsymbol{v_k}^{y'}\sum_l \boldsymbol{v_l}^{x'}\right)\right\}$$

$$= \frac{4n\sum_i \boldsymbol{v_i}^{x'}}{\sigma^4}\left\{\frac{a^2 n\sum_j \boldsymbol{v_j}^{x'}}{\sigma^6}\left[\sum_k |\boldsymbol{v_k}|^2 - \frac{1}{n}\left(\sum_k\sum_l \boldsymbol{v_k}^{x'}\boldsymbol{v_l}^{x'} + \boldsymbol{v_k}^{y'}\boldsymbol{v_l}^{y'}\right)\right]\right\} =$$

$$= \frac{4n\ a^2 n\sum_i\sum_j \boldsymbol{v_i}^{x'}\boldsymbol{v_j}^{x'}}{\sigma^{10}}\left[\underbrace{\sum_k |\boldsymbol{v_k}|^2 - \frac{1}{n}\sum_k\sum_l \boldsymbol{v'_k}\cdot\boldsymbol{v'_l}}_{K}\right] \tag{A.0.3}$$

Expanding the determinant and its third term:

$$\left(-\frac{4n}{\sigma^2}\right)\left(\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2}\right)\begin{vmatrix} \frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} & -\frac{n}{\sigma^2} & -a\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} \\ \frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & 0 & a\frac{\sum_i \boldsymbol{v_i}^{x'}}{\sigma^2} \\ 0 & -a\frac{\sum_i \boldsymbol{v_i}^{y'}}{\sigma^2} & -a^2\frac{\sum_i |\boldsymbol{v_i}|^2}{\sigma^2} \end{vmatrix} =$$

$$= -\frac{4n\sum_i \boldsymbol{v_i}^{y'}}{\sigma^4}\left\{\frac{\sum_k \boldsymbol{v_k}^{x'}}{\sigma^4}\left[\frac{a^2}{\sigma^4}\sum_j \boldsymbol{v_j}^{y'}\sum_l \boldsymbol{v_l}^{x'}\right] - \frac{\sum_j \boldsymbol{v_j}^{y'}}{\sigma^2}\left[\frac{a^2 n}{\sigma^4}\sum_k |\boldsymbol{v_k}|^2 - \frac{a^2}{\sigma^4}\sum_k \boldsymbol{v_k}^{y'}\sum_l \boldsymbol{v_l}^{y'}\right]\right\} =$$

$$= -\frac{4n\sum_i \boldsymbol{v_i}^{y'}}{\sigma^4}\left\{-\frac{a^2 n}{\sigma^6}\sum_j \boldsymbol{v_j}^{y'}\left[\sum_k |\boldsymbol{v_k}|^2 - \frac{1}{n}\sum_k \boldsymbol{v_k}^{y'}\boldsymbol{v_l}^{y'}\right] + \frac{a^2}{\sigma^6}\sum_k \boldsymbol{v_k}^{x'}\sum_j \boldsymbol{v_j}^{y'}\sum_l \boldsymbol{v_l}^{x'}\right\} =$$

$$= -\frac{4n\sum_i \boldsymbol{v_i}^{y'}}{\sigma^4}\left\{-\frac{a^2 n}{\sigma^6}\sum_j \boldsymbol{v_j}^{y'}\left[\sum_k |\boldsymbol{v_k}|^2 - \frac{1}{n}\sum_k\sum_l \left(\boldsymbol{v_k}^{x'}\boldsymbol{v_l}^{x'} + \boldsymbol{v_k}^{y'}\boldsymbol{v_l}^{y'}\right)\right]\right\} =$$

$$= \frac{4n\ a^2 n}{\sigma^{10}}\sum_i\sum_j \boldsymbol{v_i}^{y'}\boldsymbol{v_j}^{y'}\left[\underbrace{\sum_k |\boldsymbol{v_k}|^2 - \frac{1}{n}\sum_k\sum_l \boldsymbol{v'_k}\cdot\boldsymbol{v'_l}}_{K}\right] \tag{A.0.4}$$

Combining terms one,two and three the complete expansion of the determinant is:

$$-\frac{4n\ a^2 n^2}{\sigma^{10}}\sum_i |\boldsymbol{v_i}|^2\cdot K + \frac{2(n-3)a^2 n}{\sigma^{10}}\sum_i\sum_j \boldsymbol{v_i}^{x'}\boldsymbol{v_j}^{x'}\cdot K + \frac{4n\ a^2 n}{\sigma^{10}}\sum_i\sum j\boldsymbol{v_i}^{y'}\boldsymbol{v_j}^{y'}\cdot K =$$

$$= -\frac{4n\ a^2 n^2}{\sigma^{10}}\cdot K\left[\sum_i |\boldsymbol{v_i}|^2 - \frac{1}{n}\sum_i\sum_j \left(\boldsymbol{v_i}^{x'}\boldsymbol{v_j}^{x'} + \boldsymbol{v_i}^{y'}\boldsymbol{v_j}^{y'}\right)\right] =$$

$$= -\frac{4n\ a^2 n^2}{\sigma^{10}}\cdot K\left[\sum_i |\boldsymbol{v_i}|^2 - \frac{1}{n}\sum_i\sum_j \boldsymbol{v'_i}\boldsymbol{v'_j}\right] =$$

$$= -\frac{4n\ a^2 n^2}{\sigma^{10}}\left[\sum_k |\boldsymbol{v_k}|^2 - \frac{1}{n}\sum_k\sum_l \boldsymbol{v'_k}\boldsymbol{v'_l}\right]\left[\sum_i |\boldsymbol{v_i}|^2 - \frac{1}{n}\sum_i\sum_j \boldsymbol{v'_i}\boldsymbol{v'_j}\right] \tag{A.0.5}$$

Here, we introduce a new quantity as in chapter [2]:

$$\tilde{\mathrm{Var}}(\boldsymbol{v}) = \frac{1}{\tilde{n}}\left[\sum_i |\boldsymbol{v_i}|^2 - \frac{1}{\tilde{n}}\sum_i\sum_j \boldsymbol{v'_i}\cdot\boldsymbol{v'_j}\right] \tag{A.0.6}$$

so that equation (A.0.5) becomes:

$$-\frac{4n\ a^2n^2}{\sigma^{10}}\mathrm{Var}^2(\boldsymbol{v}) \tag{A.0.7}$$

The overall Jeffrey's prior is then the square root of the above expression:

$$\mathrm{J} \propto \frac{\sqrt{4n}\ a\ n\mathrm{Var}(\boldsymbol{v})}{\sigma^5} \tag{A.0.8}$$

Taking into account only the parameters we are interested in and ignoring all the constants, Jeffreys prior is proportional to:

$$\mathrm{J} \propto \frac{a\ \mathrm{Var}(\boldsymbol{v})}{\sigma^5} \tag{A.0.9}$$

where $\mathrm{Var}(\boldsymbol{v}) = \mathrm{Var}(\beta_i(s(b^{-1}))$.

### A.0.2 Laplace's approximation

Laplace's method is a relatively simple idea that is used to approximate integrals of the form $\int_a^b \exp(Mf(x))$. We usually treat the integrand as an unnormalised probability density and we assume that $f$ is maximised at a point $x_o$ which is not an endpoint in the integration interval and that $f''(x_o) < 0$. We can then Taylor expand the function $f$ as follows:

$$f(x_o) = f(x_o) + f'(x_o)(x - x_o) + \frac{1}{2}f''(x_o)(x - x_o)^2 + \mathcal{O}((x - x_o)^3) \tag{A.0.10}$$

We assumed that $f$ is maximised at $x_o$ and because the maximum is not an endpoint, then $f'(x_o)$ vanishes at $x_o$. Thus, the function can be approximated up to its quadratic term as:

$$f(x_o) \simeq f(x_o) + \frac{1}{2}\left|f''(x_o)\right|(x - x_o)^2 \tag{A.0.11}$$

The integral that we want to approximate can then be written as:

$$\int_a^b \exp(Mf(x)) \simeq \exp(Mf(x_o)) \int_a^b \exp\left(-\frac{M}{2}|f''(x_o)|(x-x_o)^2\right) \tag{A.0.12}$$

One notices that this integral is now Gaussian if we take the integral bounds to infinity. Doing so, the desired integral can be approximated by:

$$\begin{aligned}\int_a^b \exp(Mf(x)) &\simeq \exp(Mf(x_o)) \int_a^b \exp\left(-\frac{M}{2}|f''(x_o)|(x-x_o)^2\right) \\ &\simeq \exp(Mf(x_o)) \int_{-\infty}^{\infty} \exp\left(-\frac{M}{2}|f''(x_o)|(x-x_o)^2\right) \\ &\simeq \exp(Mf(x_o))\sqrt{\frac{2\pi}{M|f''(x_o)|}}\end{aligned} \tag{A.0.13}$$

This is the Laplace approximation which can be generalised for more than one dimensions. Physicists also call this approximation the saddle-point approximation.

# Bibliography


[1] H. Huang, F. Makedon, and R. McColl, "High dimensional statistical shape model for medical image analysis," pp. 1541–1544, 2008.

[2] O. Faugeras, J. Janin, F. Cazals, and P. Kornprobst, *Modeling in Computational Biology and Biomedicine: A Multidisciplinary Endeavor*. Springer Science & Business Media, 2012.

[3] P. Steffen, B. Voss, M. Rehmsmeier, J. Reeder, and R. Giegerich, "Rnashapes: an integrated rna analysis package based on abstract shapes," *Bioinformatics*, vol. 22, no. 4, pp. 500–503, 2006.

[4] E. d. A. M. Filho, L. Monteiro, and S. dos Reis, "Skull shape and size divergence in dolphins of the genus sotalia: A tridimensional morphometric analysis," *Journal of Mammalogy*, vol. 83, no. 1, pp. 125–134, 2002.

[5] F. L. Bookstein, "Landmark methods for forms without landmarks: morphometrics of group differences in outline shape," *Medical Image Analysis*, vol. 1, no. 3, pp. 225–243, 1997.

[6] J. Mantas and A. Heaton, "Handwritten character recognition by parallel labelling and shape analysis," *Pattern Recognition Letters*, vol. 1, no. 5–6, pp. 465–468, 1983.

[7] J. Rocha and T. Pavlidis, "A shape analysis model with applications to a character recognition system," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 393–404, 1994.

[8] I. Dryden, "Statistical shape analysis in archaeology," in *Spatial statistics in archaeology workshop*, 2000.

[9] I. Saragusti, A. Karasik, I. Sharon, and U. Smilansky, "Quantitative analysis of shapes attributes based on contours and section profiles in artifact analysis," *Journal of Archaeological Science*, vol. 32, no. 6, pp. 841–853, 2005.

[10] R. Corruccini, "Shape in morphometrics: Comparative analyses," *American Journal of Physical Anthropology*, vol. 73, no. 3, pp. 289–303, 1987.

[11] D. Kendall, "Shape manifolds, procrustean metrics, and complex projective spaces," *Bulletin of the London Mathematical Society*, vol. 16, no. 2, pp. 81–121, 1984.

[12] A. Srivastava and I. Jermyn, "Looking for shapes in two-dimensional cluttered point clouds.," *Pattern Analysis and Machine Inttelligence, IEEE Transactions*, vol. 31, no. 9, pp. 1616–1629, 2009.

[13] K. B.B., "Kimia database." Available at `http://www.lems.brown.edu/~dmc/`.

[14] I. Dryden and K. Mardia, *Statistical Shape Analysis*. West Sussex: Wiley, 1998.

[15] C. Small, *The Statistical Theory of Shape*. New York, NY, USA: Springer Verlag, 1996.

[16] L. d. F. D. Costa and R. M. Cesar, Jr., *Shape Analysis and Classification: Theory and Practice*. Boca Raton, FL, USA: CRC Press, Inc., 1st ed., 2000.

[17] V. Torre and T. Poggio, "On edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 147–163, 1984.

[18] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev*, pp. 183–193, 1954.

[19] S. Wold, "Spline functions in data analysis," *Technometrics*, vol. 16, no. 1, pp. 1–11, 1974.

[20] T. P. Wallace and P. A. Wintz, "An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors," *Computer Graphics and Image Processing*, vol. 13, no. 2, pp. 99–126, 1980.

[21] D. Marr, "Early processing of visual information," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 275, no. 942, pp. 483–519, 1976.

[22] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[23] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, pp. 2–14, Jan. 1986.

[24] J. Ponce and M. Brady, *Toward a surface primal sketch.* Springer, 1987.

[25] G. Matheron, *Random Sets and Integral Geometry.* Wiley, 1975.

[26] J. Serra, "Introduction to mathematical morphology," *Comput. Vision Graph. Image Process.*, vol. 35, pp. 283–305, Sept. 1986.

[27] R. Kirsch, "Computer determination of the constituent structure of biological images," *Computer and Biomedical research*, vol. 4, pp. 315–328, 19871.

[28] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 674–693, July 1989.

[29] D. Geiger and A. Yuille, "A common framework for image segmentation.," *International Journal of Computer Vision*, vol. 6, no. 3, pp. 227–243, 1991.

[30] C. Chen, J. Lee, and Y. Sun, "Wavelet transformation for gray-level corner detection," *Pattern Recognition*, vol. 28, no. 6, pp. 853–861, 1995.

[31] L. A. Iverson and S. W. Zucker, "Logical/linear operators for image curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 982–996, Oct. 1995.

[32] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 113–125, Feb. 1998.

[33] A. Watson, "The cortex transform: Rapid computation of simulated neural images," *Comput. Vision Graph. Image Process.*, vol. 39, pp. 311–327, Sept. 1987.

[34] A. Bovik, M. Clarke, and W. Geisler, "Multichannel Texture Analysis Using Localized Spatial Filters," vol. 12, pp. 55–73, 1990.

[35] T. Pavlidis, "Waveform segmentation through functional approximation," *Computers, IEEE Transactions on*, vol. C-22, pp. 689–697, July 1973.

[36] R. Jain, R. Kasturi, and B. Schunck, *Machine Vision*. New York, NY, USA: McGraw-Hill, Inc., 1995.

[37] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 4–37, Jan. 2000.

[38] N. Kiryati and D. Maydan, "Calculating geometric properties from fourier representation," *Pattern Recogn.*, vol. 22, pp. 469–475, Sept. 1989.

[39] S. Marshall, "Review of shape coding techniques.," *Image Vision Comput.*, vol. 7, no. 4, pp. 281–294, 1989.

[40] F. L. Bookstein, "Landmark methods for forms without landmarks: localizing group differences in outline shape.," *Medical Image Analysis*, vol. 1, no. 3, pp. 255–244, 1997.

[41] M. Fischler and H. Wolf, "Locating perceptually salient points on planar curves.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 2, pp. 113–129, 1994.

[42] B. J. Super, "Fast correspondence-based system for shape retrieval," *Pattern Recognition Letters*, vol. 25, no. 2, pp. 217–225, 2004.

[43] J. Zhang, X. Zhang, H. Krim, and G. Walter, "Object representation and recognition in shape spaces," *Pattern Recognition*, vol. 36, no. 5, pp. 1143–1154, 2003.

[44] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 509–522, Apr. 2002.

[45] G. McNeill and S. Vijayakumar, "2d shape classification and retrieval," 2005.

[46] P. D. Sampson, F. L. Bookstein, F. H. Sheehan, and E. L. Bolson, "Eigenshape analysis of left ventricular outlines from contrast ventriculograms," in *Advances in morphometrics*, pp. 211–233, Springer, 1996.

[47] S. Wang, T. Kubota, and T. Richardson, "Shape correspondence through landmark sliding," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–143, IEEE, 2004.

[48] C. Chang, S. M. Hwang, and D. J. Buehrer, "A shape recognition scheme based on relative distances of feature points from the centroid," *Pattern Recognition*, vol. 24, no. 11, pp. 1053–1063, 1991.

[49] D. W. Thompson, *On growth and form.* Cambridge: Cambridge University Press, 1917.

[50] F. L. Bookstein, "Size and shape spaces for landmark data in two dimensions (with discussion)," *Statistical Science*, vol. 1, pp. 181–242, 1986.

[51] J. T. Kent and K. V. Mardia, "Shape, procrustes tangent projections and bilateral symmetry.," *Biometrika*, vol. 88, pp. 469–485, 2001.

[52] H. Le and D. G. Kendall, "The riemannian structure of euclidean shape spaces: A novel environment for statistics," *The Annals of Statistics*, vol. 21, pp. 1225–1271, 09 1993.

[53] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[54] T. F. Cootes and C. J. Taylor, "Active shape models – "smart snakes"," in *BMVC92*, pp. 266–275, Springer, 1992.

[55] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "The use of active shape models for locating structures in medical images," in *Information Processing in Medical Imaging*, pp. 33–47, Springer, 1993.

[56] T. F. Cootes, C. J. Taylor, and A. Lanitis, "Active shape models: Evaluation of a multi-resolution method for improving image search.," in *BMVC* (E. R. Hancock, ed.), pp. 1–10, BMVA Press, 1994.

[57] C. Kervrann and F. Heitz, "A hierarchical statistical framework for the segmentation of deformable objects in image sequences," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pp. 724–728, IEEE, 1994.

[58] K. F. Lai and R. T. Chin, "Deformable contours: Modeling and extraction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 11, pp. 1084–1090, 1995.

[59] A. Pentland and S. Sclaroff, "Closed-form solutions for physically based shape modeling and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 715–729, July 1991.

[60] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models;their training and application," *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[61] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[62] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[63] A. P. Pentland, "Automatic extraction of deformable part models," *International Journal of Computer Vision*, vol. 4, no. 2, pp. 107–126, 1990.

[64] T. Cootes, E. R. Baldock, and J. Graham, "An introduction to active shape models," *Image processing and analysis*, pp. 223–248, 2000.

[65] U. Grenander and M. Miller, *Pattern Theory: From Representation to Inference*. New York, NY, USA: Oxford University Press, Inc., 2007.

[66] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *Int. J. Comput. Vision*, vol. 61, pp. 139–157, Feb. 2005.

[67] Y. Amit, U. Grenander, and M. Piccioni, "Structural image restoration through deformable templates," *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 376–387, 1991.

[68] M. I. Miller, G. E. Christensen, Y. Amit, and U. Grenander, "Mathematical textbook of deformable neuroanatomies," *Proceedings of the National Academy of Sciences*, vol. 90, no. 24, pp. 11944–11948, 1993.

[69] A. Trouvé, "Diffeomorphisms groups and pattern matching in image analysis," *International Journal of Computer Vision*, vol. 28, no. 3, pp. 213–221, 1998.

[70] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International journal of computer vision*, vol. 8, no. 2, pp. 99–111, 1992.

[71] U. Grenander and M. I. Miller, "Computational anatomy: An emerging discipline," *Quarterly of applied mathematics*, vol. 56, no. 4, pp. 617–694, 1998.

[72] A. Srivastava, P. Turaga, and S. Kurtek, "On advances in differential-geometric approaches for 2d and 3d shape analyses and activity recognition," *Image and Vision Computing*, vol. 30, no. 6, pp. 398–416, 2012.

[73] A. Hobolth, J. Kent, and I. Dryden, "On the relation between edge and vertex modelling in shape analysis," *Scandinavian Journal of Statistics*, vol. 29, no. 3, pp. 355–374, 2002.

[74] J. Kent and K. V. Mardia, "Shape, procrustes tangent projections and bilateral symmetry," *Biometrika*, vol. 92, no. 1, pp. 249–249, 2005.

[75] A. Hill and C. J. Taylor, "A method of non-rigid correspondence for automaticlandmark identification.," in *BMVC*, pp. 1–10, 1996.

[76] G. Heitz, G. Elidan, B. Packer, and D. Koller, "Shape-based object localization for descriptive classification," *International journal of computer vision*, vol. 84, no. 1, pp. 40–62, 2009.

[77] J. G. Raudseps, "Some aspects of the tangent-angle vs. arc length representation of contours," tech. rep., DTIC Document, 1965.

[78] R. M. Rangayyan, D. Guliato, J. D. De Carvalho, and S. A. Santiago, "Feature extraction from the turning angle function for the classification of contours of breast tumors," Citeseer.

[79] C. T. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," *Computers, IEEE Transactions on*, vol. 100, no. 3, pp. 269–281, 1972.

[80] J. R. Bennett and J. S. Mac Donald, "On the measurement of curvature in a quantized environment," *IEEE Trans. Comput.*, vol. 24, pp. 803–820, Aug. 1975.

[81] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 3, pp. 209–216, 1991.

[82] L. Younes, "Computable elastic distances between shapes," *SIAM J. of Applied Math*, pp. 565–586, 1998.

[83] L. Younes, "Optimal matching between shapes via elastic deformations," *Image and Vision Computing*, vol. 17, no. 5, pp. 381–389, 1999.

[84] E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi, "Analysis of planar shapes using geodesic paths on shape spaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 3, pp. 372–383, 2004.

[85] A. Srivastava, S. Joshi, W. Mio, and X. Liu, "Statistical shape analysis: Clustering, learning and testing.," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 590–602, 2005.

[86] A. Backes, D. Casanova, and O. Bruno, "A complex network-based approach for boundary shape analysis," *Pattern Recognition*, vol. 42, no. 1, pp. 54–67, 2009.

[87] F. R. Schmidt, M. Clausen, and D. Cremers, "Shape matching by variational computation of geodesics on a manifold," vol. 4174, pp. 142–151, Springer, 2006.

[88] P. Minchor and D. Mumford, "Riemannian geometries on spaces of plane curves," *J. European Math. Soc.*, vol. 8, pp. 1–48, 2006.

[89] A. Yezzi and A. Mennucci, "Metrics in the space of curves," *ArXiv Mathematics e-prints*, Dec. 2004.

[90] G. Sundaramoorthi, A. Mennucci, S. Soatto, and A. J. Yezzi, "A new geometric metric in the space of curves, and applications to tracking deforming objects by prediction and filtering," *SIAM J. Imaging Sciences*, January 2011.

[91] E. Sharon and D. Mumford, "2d-shape analysis using conformal mapping," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 55–75, 2006.

[92] W. Mio, A. Srivastava, and S. Joshi, "On shape of plane elastic curves," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 307–324, 2007.

[93] W. Mio and A. Srivastava, "Elastic-string models for representation and analysis of planar shapes," in *Computer Vision and Pattern Recognition, 2004.*

*CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–10, IEEE, 2004.

[94] J. Shah, "An h2 riemannian metric on the space of planar curves modulo similitudes," *Adv. Appl. Math.*, vol. 51, pp. 483–506, Sept. 2013.

[95] S. H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn, "A novel representation for riemannian analysis of elastic curves in rn," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–7, IEEE, 2007.

[96] S. H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn, "Removing shape-preserving transformations in square-root elastic (sre) framework for shape analysis of curves," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 387–398, Springer, 2007.

[97] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn, "Shape analysis of elastic curves in euclidean spaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 7, pp. 1415–1428, 2011.

[98] P. W. Michor, D. Mumford, J. Shah, and L. Younes, "A metric on shape space with explicit geodesics," *arXiv preprint arXiv:0706.4299*, 2007.

[99] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[100] Z. Barutcuoglu and C. DeCoro, "Hierarchical shape classification using Bayesian aggregation," in *Shape Modeling International*, June 2006.

[101] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, pp. 121–167, June 1998.

[102] C. Goodall and K. Mardia, "The noncentral bartlett decompositions and shape densities," *Journal of Multivariate Analysis*, vol. 40, no. 1, pp. 94–108, 1992.

[103] C. Goodall, "Procrustes methods in the statistical analysis of shape," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 285–339, 1991.

[104] J. R. Hurley and R. B. Cattell, "The procrustes program: Producing direct rotation to test a hypothesized factor structure," *Behavioral Science*, vol. 7, no. 2, pp. 258–262, 1962.

[105] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 9, pp. 850–863, 1993.

[106] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for computer vision.," *Online report*, 2001.

[107] N. Duta, A. Jain, and M. P. Dubuisson-Jolly, "Learning 2d shape models," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, IEEE, 1999.

[108] W. Kristof and B. Wingersky, "A generalization of the orthogonal procrustes rotation procedure to more than two matrices.," in *Proceedings of the Annual Convention of the American Psychological Association*, American Psychological Association, 1971.

[109] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[110] C. Goodall, "Procrustes methods in the statistical analysis of shape," *Jour. Royal Statistical Society*, vol. Series B, no. 53, pp. 285–339, 1991.

[111] C. Goodall and A. Bose, "Models and procrustes methods for the analysis of shape differences," in *Proceedings of the 19th INTERFACE Symposium*, vol. 91, Fairfax Station, Interface Foundation, 1987.

[112] K. Grove and H. Karcher, "How to conjugatec 1-close group actions," *Mathematische Zeitschrift*, vol. 132, no. 1, pp. 11–20, 1973.

[113] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Communications on pure and applied mathematics*, vol. 30, no. 5, pp. 509–541, 1977.

[114] H. Karcher, "Riemannian Center of Mass and so called karcher mean," *ArXiv e-prints*, July 2014.

[115] A. Srivastava, W. Mio, E. Klassen, and S. Joshi, "Geometric analysis of continuous, planar shapes," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 341–356, Springer, 2003.

[116] A. Srivastava and I. H. Jermyn, "Bayesian classification of shapes hidden in point cloud data," in *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th*, pp. 359–364, IEEE, 2009.

[117] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[118] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society of Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[119] T. Sebastian, P. Klein, and B. Kimia, "On aligning curves," *IEEE TPAMI*, vol. 25, no. 1, pp. 116–124, 2003.

[120] A. Srivastava, I. Jermyn, and S. Joshi, "Riemannian analysis of probability density functions with applications in vision," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[121] S. J. Maybank, "The fisher-rao metric for projective transformations of the line.," *International Journal of Computer Vision*, vol. 63, no. 3, pp. 191–206, 2005.

[122] A. Peter and A. Rangarajan, "A new closed-form information metric for shape analysis," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*, pp. 249–256, Springer, 2006.

[123] A. Peter and A. Rangarajan, "Shape analysis using the fisher-rao riemannian metric: Unifying shape representation and deformation," in *Biomedical Imag-*

*ing: Nano to Macro, 2006. 3rd IEEE International Symposium on*, pp. 1164–1167, IEEE, 2006.

[124] N. Cencov, *Statistical Decision Rules and Optimal Inferences Translations of Mathematical MMonographs*. AMS, 1982.

[125] H. Jeffreys, "An invariant form for the prior probability in estimation problems," vol. 186, pp. 453–461, 1946.

[126] G. Box and G. Tiaon, *Bayesian Inference in Statistical Analysis*. Hoboken, NJ: Wiley, 1992.

[127] A. Wood, "Estimation of the concentration parameters of the Fisher matrix distribution on SO(3) and the Bingham distribution on $S_q, q \geqslant 2$," *Australian Journal of Statistics*, vol. 35, no. 1, pp. 69–79, 1993.

[128] M. Prentice, "A distribution-free method of interval estimation for unsigned directional data," *Biometrika*, vol. 71, no. 1, pp. 147–154, 1984.

[129] I. S. Gradshteyn and I. M. Ryzhik, *Table of integrals, series, and products*. seventh ed., 2007.

[130] D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge: Cambridge University Press, 2003.

[131] S. Weinzierl, "Introduction to Monte Carlo methods," 2000.

[132] C. Commander, "A survey if the quadratic assignment problem, with applications," *Morehead Electronic Journal of Applicable Mathematics*, 2005.

[133] S. Sahni and T. Gonzalez, "P-complete approximation problems," *J. ACM*, vol. 23, no. 3, pp. 555–565, 1976.

[134] G. Finke, R. Bunkard, and F. Rendl, "Quadratic assignment problems," *Annals of Discrete Mathematics*, vol. 31, pp. 61–82, 1987.

[135] T. Sebastian, P. Klein, and B. Kimia, "On aligning curves," *IEEE TPAMI*, vol. 25, no. 1, pp. 116–124, 2003.

[136] R. Gonzalez, R. Woods, and S. Eddins, *Digital Image Processing Using MATLAB*. Pearson Prentice Hall, 2004.

[137] D. Doria, "Moore neighbor tracing," 08 2011.

[138] P. Reddy, V. Amarnadh, and B. M., "Evaluation of stopping criterion in contour tracing algorithms," *International Journal of Computer Science and Information Technologies*, vol. 3, no. 3, pp. 3888–3894, 2012.

[139] P. Kearey, *The Penguin Dictionary of Geology*. Penguin, 2001.

[140] P. Potter, "Sand bodies and sedimentary environments: A review," *Am. Assoc. Petroleum Geologists Bull.*, vol. 51, no. 3, pp. 337–365, 1967.

[141] M. Gibling, "Width and thickness of fluvial channel bodies and valley fills in the geological record: A literature compilation and classification," *Journal of Sedimentary Research*, vol. 76, pp. 731–770, 2006.

[142] J. Schieber, "Paleoflow patterns and macroscopic sedimentary features in the late devonian chattanooga shale of tennessee: Differences between the western and eastern appalachian basin," 1994.

[143] J. Hirst, "Variations in alluvial architecture across the oligo-miocene huesca fluvial system, ebro basin, spain, the three-dimensional facies architecture of terrigenous clastic sediments and its implications for hydrocarbon discovery and recovery edited by andrew d. miall and noel tyler," *Society for Sedimentary Geology, Concepts in Sedimentology and Paleontology*, vol. 3, pp. 111–121, 1991.

[144] J. L. Rich, "Shoestring sands of eastern kansas," *Am. Assoc. Petroleum Geologists Bull.*, vol. 7, pp. 103–113, 1923.

[145] P. Potter, "Late palaezoic sandstones of the illinois basin," *Rep. Invest. Illinois St. geol. Surv.*, vol. 217, p. 92, 1963.

[146] P. Krynine, "The megascopic study and field classification of sedimentary rocks," *The Journal of Geology*, vol. 56, pp. 130–156, 1948.

[147] A. McGugan, "Occurrence and persistence of thin sellf deposits of uniform lithology," *Geological Society of America Bulletin*, vol. 76, pp. 125–130, 1965.

[148] J. Collinson, "Vertical sequence and sand body shape in alluvial sequences," *Fluvial Sedimentology, Memoir 5*, pp. 577–586, 1977.

[149] M. Moody-Stuart, "High and low sinuosity stream deposits, with examples from the devonian of spitsbergen," *J. Sediment. Petrol.*, vol. 36, pp. 1102–1117, 1966.

[150] P. Friend, M. Slater, and R. Williams, "Vertical and lateral building of river sandstone bodies, ebro basin, spain," *Journal of the Geological Society*, vol. 136, pp. 39–46, 1979.

[151] J. Allen, "The classification of cross-stratified units with notes on their origin," *Sedimentology*, vol. 2, pp. 93–114, 1963.

[152] S. Schum, "The shape of alluvial channels in relation to sediment type. erosion and sedimentation in a semiarid environment," *U.S. Geological Survey, Professional Paper*, vol. 352–B, pp. 17–30, 1960.

[153] C. Atkinson, *Comparative sequences of ancient fluviatile deposits in the Tertiary, South Pyrenean Basin Northern Spain.* PhD thesis, 1983.

[154] G. Nadon, "The genesis and recognition of anastomosed fluvial deposits and: data from the st. mary river formation, southwest alberta, canada," *Journal of Sedimentary Research*, vol. 64, pp. 451–463, 1994.

[155] J. Bridge and R. Tye, "Interpreting the dimensions of ancient fluvial channel bars, channels, and channel belts from wireline-logs and cores," *AAPG Bulletin*, vol. 84, no. 8, pp. 1205–1228, 2000.

[156] J. Almhana, Z. Liu, V. Choulakian, and R. McGorman, "A recursive algorithm for gamma mixture models," in *Communications, 2006. ICC'06. IEEE International Conference on*, vol. 1, pp. 197–202, IEEE, 2006.

[157] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[158] C. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[159] G. J. McLachlan and K. E. Basford, "Mixture models: Inference and applications to clustering," *Applied Statistics*, 1988.

[160] C. F. J. Wu, "On the convergence properties of the em algorithm," *Ann. Statist.*, vol. 11, pp. 95–103, 03 1983.

[161] J. Einbeck and J. Hinde, "A note on npml estimation for exponential family regression models with unspecified dispersion parameter.," *Austrian journal of statistics.*, vol. 35, pp. 233–243, June 2006.

[162] J. Einbeck, R. Darnell, and J. Hinde, *npmlreg: Nonparametric maximum likelihood estimation for random effect models*, 2014. R package version 0.46-0.

[163] C. Biernacki and S. Chrtien, "Degeneracy in the maximum likelihood estimation of univariate gaussian mixtures with em," *Statistics and Probability Letters*, vol. 61, no. 4, pp. 373–382, 2003.

[164] J. Aitchison and C. G. Aitken, "Multivariate binary discrimination by the kernel method," *Biometrika*, vol. 63, no. 3, pp. 413–420, 1976.

[165] J. Russ, *Image Processing Handbook*. Boca Raton, FL, USA: CRC Press, Inc., 4th ed., 2002.

[166] Y. Mingqiang, K. Kpalma, and J. Ronsin, "A survey of shape feature extraction techniques.," *Peng-Yeng Yin. Pattern Recognition, IN-TECH*, pp. 43–90, 2008.

[167] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *SCIENCE*, vol. 220, no. 4598, pp. 671–680, 1983.

[168] V. Černỳ, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of optimization theory and applications*, vol. 45, no. 1, pp. 41–51, 1985.

[169] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," in *Advances in Neural Information Processing Systems 23*, pp. 2595–2603, Curran Associates, Inc., 2010.

[170] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," *arXiv preprint arXiv:1212.1824*, 2012.

[171] N. Metropolis, A. Rosenbluth, M. N. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[172] W. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[173] J. Conway and D. Smith, *On quaternions and octonions: their geometry, arithmetic and symmetry*. AK Peters, 2003.

[174] J. Kuipers, *Quaternions and rotation sequences: a primer with applications to orbits, aerospace and virual reality*. Princeton University Press, 2002.