Immaculate Depth Perception: Recovering 3D Scene Information via Depth Completion and Prediction

AMIR ATAPOUR-ABARGHOUEI

Immaculate Depth Perception: Recovering 3D Scene Information via Depth Completion and Prediction

Amir Atapour Abarghouei

A thesis presented for the degree of
Doctor of Philosophy at Durham University

Department of Computer science
Durham University
United Kingdom

16th October 2019

Immaculate Depth Perception: Recovering 3D Scene Information via Depth Completion and Prediction

Amir Atapour Abarghouei

Submitted for the degree of Doctor of Philosophy

Even though obtaining three-dimensional (3D) information has received significant attention in scene capture systems in recent years, there are currently numerous challenges in scene depth estimation, which is one of the fundamental components of any 3D vision system focusing on RGB-D images. This has led to the creation of specific areas of research where the goal is to estimate complete scene depth or fill the missing 3D information post capture. In many downstream applications, incomplete scene depth is of limited value, and thus techniques are required to *fill the holes* that exist in terms of missing depth information. An analogous problem exists within the scope of scene filling post object removal in the same context. Although considerable research has resulted in notable progress in the synthetic expansion or reconstruction of missing colour scene information in both statistical and structural forms, work on the plausible completion of missing scene depth is contrastingly limited. In this thesis, we present various methods capable of performing the depth completion process required to achieve high quality scene depth post capture. Two novel methods capable of preforming object removal in an RGB-D image and, at the same time, filling the naturally-occurring holes within the depth image are proposed inspired by seminal approaches towards exemplar-based RGB image inpainting and texture synthesis. Another proposed approach takes advantage of the recent advances in semantic segmentation and a set of carefully designed hole cases to carry out object-wise depth completion in real time. Using the significant progress made in generative models, we then move on to a learning-based approach that utilizes a convolutional neural network trained on synthetic RGB-D images in a supervised framework using the Discrete Cosine Transform, adversarial training and domain adaptation to complete large missing portions of depth images. The representation learning capabilities of the network is evaluated by adapting the network to perform the somewhat similar task of monocular depth estimation, with outstanding results. Based on the success of the adapted monocular depth estimation model, we then propose two monocular depth estimation techniques, also trained on synthetic data, that can generate hole-free depth information from a single RGB image, circumnavigating the need for depth completion and refinement altogether. One of the approaches makes use of style transfer as a form of domain adaptation, and the other uses a recurrent model, a series of complex skip connections and adversarial training in a multi-task framework to generate temporally homogeneous depth outputs based on an input of a sequence of RGB images.

Declaration

The work in this thesis is based on research carried out within the Department of Computer Science at Durham University, UK. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all the author's own work unless referenced to the contrary in the text.

Acknowledgements

I would like to express my sincerest gratitudes to my supervisor, Prof. Toby Breckon, who gave me an opportunity when no one else would and supported me in ways no one else could. All the time and effort he has put in towards my success can never be paid back, but he has my undying appreciation for all he has done.

The role of my peers, colleagues and friends cannot be ignored as they provided me with the invaluable support and the necessary occasional distractions any mere mortal facing constant academic deadlines may come to need, and for those, I am forever grateful.

My parents, sister and late grandmother have always been there for me through the years with their unwavering support and dedication, even when I was not there to reciprocate. Their wise counsel and helping hands have elevated me thus far and I earnestly hope they will continue to do so for many years to come. I cannot thank them enough for I would not be who and where I am today without their many sacrifices.

And most of all, I am immensely grateful to the love of my life, Ellie, who has given up on her own hopes and dreams so I could chase mine. She has always borne my burdens, endured my flaws, comforted my aches, put up with what scant few would, and shone a light on my otherwise bleak world. And it is with great shame that I have nought to give back, but to dedicate not just this meagre thesis but my entire life in the hope that I might bring her a fraction of the happiness she deserves.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The world is visually diverse, irregular and contrastingly structured at the same time. Three-dimensional scenes containing depth information are highly applicable within visual systems such as autonomous driving, augmented reality, environment modelling and alike. This thesis is directed towards answering two major research questions within the context of three-dimensional scene understanding, namely depth completion and monocular depth estimation.

Three-dimensional scene understanding is gaining an ever-increasing applicability and importance due to its wide-spread uses in real-world scenarios, including areas such as interactive entertainment, future

Figure 1.1: Image of Durham Cathedral and the estimated depth (approach presented in Chapter 6.1).

vehicle autonomy, environment modelling, security surveillance and future manufacturing in technologies. Despite extensive work on 3D sensing of late [1–5], a number of limitations pertaining to environmental conditions, inter-object occlusion and sensor capabilities constrain fully-effective scene depth capture [1, 6]. Such issues often result in missing and invalid segments within a captured depth image, hence limiting its utilisation in various applications. As a consequence, significant research needs to be focused specifically on developing techniques to complete missing scene depth to increase the quality of the depth information for better applicability.

A major portion of this thesis is therefore dedicated to different approaches towards achieving the goal of scene depth completion using conventional computer vision techniques (Chapter 3 and Chapter 4) and learning based methods (Chapter 5).

On the other hand, while depth completion can be a useful process for creating full dense depth for various downstream vision-based application, learning-based monocular depth estimation techniques can be an invaluable tool that can provide hole-free scene depth in a cheap and efficient manner, completely removing the need for any depth completion in the process. In essence, monocular depth estimation is the ability to estimate complete scene depth from a single RGB image used as the input to the process. An example of the results of such an approach (presented in Chapter 6.1) is illustrated in Figure 1.1. As a result, we also propose two efficient and effective monocular depth estimation approaches capable of predicting full dense scene depth based on a single RGB image (Chapter 6). In short, the work in thesis is aimed at providing an answer to the following questions:

- Can we provide the entire scene depth captured through imperfect and flawed means by completing the missing regions of the acquired depth in a post-processing stage, considering different application requirements such as accuracy, efficiency, underlying scene geometry, surface detail and texture and alike?
- Can we obtain the entire scene depth without any need for post processing from a single RGB image by learning about content, context and other cues present within the scene?

1.1 Motivation

Creating a fully-realized three-dimensional environment has been a long-standing challenge in computer vision, which has lead to various techniques to represent and visualize 3D scenes [7–11]. Depth images are one of the most popular methods of representing the depth component of a scene, which can easily and efficiently be processed and used, similar to any other image.

Depending on how the depth image is captured, however, there are numerous possible challenges associated with the depth image. While high-end depth sensing technologies, including light field cameras and

Figure 1.2: Examples of depth acquired via stereo correspondence. we can see an example of a colour image (RGB), its corresponding depth (D) image (disparity calculated via [12]) and a hole mask (H) indicating missing depth values.

LiDAR, exist that are capable of capturing accurate scene depth with relatively fewer anomalies (missing or invalid depth, undesirable artefacts and depth inhomogeneity) compared to consumer devices, they remain expensive and difficult to operate in terms of size, weight and power. As a result, both industry and academia have gravitated toward more easily accessible technologies such as stereo correspondence [13–15], structured light [2, 16–18] and time-of-flight cameras [19–21].

Stereo imaging as a passive scene acquisition method has long been used as a reliable source of depth sensing, but not without certain issues. Although stereo correspondence is better equipped than structured light and time-of-flight cameras to estimate depth where highly granular texture is present, smoothing still occurs. Additionally slightest mis-calibration or issues in the setup and synchronization can lead to invalid or missing depth information. Moreover, missing depth (holes) are often observed in the scene depth where absence of camera overlap, featureless surfaces, sparse information for a scene object such as shrubbery, unclear object boundaries, very distant objects and alike are present. Such issues can be seen in Figure 1.2, where we see the left colour image (RGB), the estimated depth (D) and a binary mask (H) marking where depth holes are (in black).

Structured light devices and time-of-flight cameras are active range sensors and while they can suffer from mis-calibration issues, they are more-widely utilised for a variety of purposes due to their low-cost availability in the commercial market with factory calibration settings [22–25].

Despite this, structured light sensors are subject to a wide range of issues including but not limited to over-saturation due to ambient light [22], external active illumination source interference [23, 24], active light path error caused by reflective surfaces, occlusion, fronto-parallel angle of the object to the sensor [25, 26], erroneous light pattern detection in dynamic scenes [25] and others.

Similarly, time-of-flight cameras have their own flaws that lead to invalid or missing depth, noise and other additional artefacts, such as depth error caused by light scattering or semi-transparent surfaces [27, 28], external illumination interference [29], depth offset for non-reflective objects [30] and alike [25].

It must be noted that not all of such issues will lead to missing depth information (holes), but invalid depth and noise are essentially detriments in practice and are best handled through removal and subsequent filling. Figure 1.3 depicts examples of depth images obtained using a structured light device (left) and a time-of-flight camera (right).

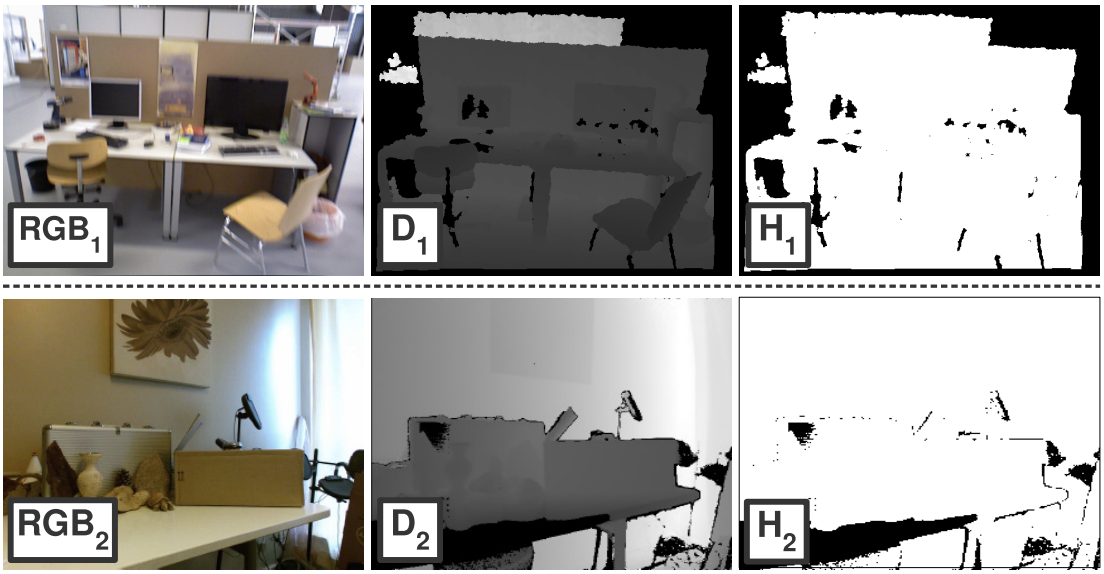

Figure 1.3: Examples of depth acquired via a structured light device (left) and time-of-flight camera (right). we can see examples of colour images (RGB), corresponding depth images (D) and hole masks (H) indicating missing depth values.

While many 3D computer vision applications continue to move forward as they cope with the issues caused by depth holes, performance can be improved in many respects if accurate hole-free depth information is readily available for processing, hence the creation of the entire literature on depth completion. Our aim is to contribute to the state-of-the-art research conducted on this subject matter by proposing depth completion methods, which can rectify many of the issues commonly found in depth images in post precessing and creating alternative depth estimation techniques that effectively generate depth without the need for any refinement.

1.2 Thesis Contributions

The work presented in this thesis contributes to the state of the art in the following areas:

- Adapting conventional exemplar-based RGB image inpainting and texture synthesis techniques to perform accurate and plausible object removal and depth completion in RGB-D images (Chapter 3).
- Proposing an object-wise real-time non-parametric strategy for depth completion built on a scene segmentation prior that preserves relief texture within scene depth (Chapter 4).

- Creating a generative model based on a convolutional neural network using the Earth Mover's distance to complete depth via the Discrete Cosine Transform based on a synthetic training corpus with predicted depth holes and domain adaptation. The model is capable of learning better semantics and context as illustrated by superior sharp and artefact-free qualitative outputs when performing monocular depth estimation (Chapter 5).

- Presenting two novel state-of-the-art monocular depth estimation techniques trained on synthetic images capable of producing depth from a single RGB image. One approach utilises image style transfer as a novel form of domain adaptation and the other is capable of maintaining temporal homogeneity within video sequences (Chapter 6).

1.3 Publications

The work undertaken as part of this thesis have been published or are under review in the following:

- **A. Atapour-Abarghouei** and T. P. Breckon, 'Dealing with Missing Depth: Recent Advances in Depth Image Completion and Estimation.' RGB-D Image Analysis and Processing. Springer, 2019.

- **A. Atapour-Abarghouei** and T. P. Breckon, 'A Comparative Review of Plausible Hole Filling Strategies in the Context of Scene Depth Image Completion.' In Journal of Computers & Graphics, 2018, 72: pp. 39-58 [31].

- **A. Atapour-Abarghouei** and T. P. Breckon, 'Veritatem Dies Aperit - Temporally Consistent Depth Prediction Enabled by a Multi-Task Geometric and Semantic Scene Understanding Approach.' In the IEEE Conference on Computer Vision and Pattern Recognition, 2019. [32].

- **A. Atapour-Abarghouei** and T. P. Breckon. 'Real-Time Monocular Depth Estimation using Synthetic Data with Domain Adaptation via Image Style Transfer.' In the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2800-2810, 2018 [33].

- **A. Atapour-Abarghouei**, S. Akcay, G. P. de La Garanderie and T. P. Breckon, 'Generative Adversarial Framework for Depth Filling via Wasserstein Metric, Cosine Transform and Domain Transfer.' In Journal of Pattern Recognition, pp. 232-244, 2019 [34].

- **A. Atapour-Abarghouei** and T. P. Breckon. 'Extended Patch Prioritization for Depth Filling within Constrained Exemplar-based RGB-D Image Completion.' In International Conference on Image Analysis and Recognition, pp. 306-314, 2018 [35].

- **A. Atapour-Abarghouei** and T. P. Breckon. 'DepthComp: Real-Time Depth Image Completion based on Prior Semantic Scene Segmentation.' In British Machine Vision Conference, pp. 1-13, 2017 [36].

- **A. Atapour-Abarghouei**, G. P. de La Garanderie and T. P. Breckon, 'Back to Butterworth - a Fourier Basis for 3D Surface Relief Hole Filling within RGB-D Imagery.' In International Conference on Pattern Recognition, pp. 2813-2818, 2016 [37].

1.4 Scope

In this thesis, we will cover the details of the various techniques proposed to perform existing scene depth completion and refinement and monocular depth estimation. However, it is important to note that the work is limited by several factors, such as the availability of data and hardware.

A major portion of the work done for this thesis involves training high-capacity deep neural network capable of performing depth completion and estimation. The training of such networks requires large quantities of high-quality hole-free depth data, which does not exist. Depth captured through stereo correspondence, structured light, or time-of-flight devices often contain corruptions and missing regions (as explained in Section 1.1), thus requiring depth completion in the first place and while technologies such as LiDAR and laser scanning can produce better quality scene depth, albeit sparse, are expensive and time-consuming, making them intractable in real-world applications.

Consequently, we have taken advantage of synthetic data captured form virtual environments to train our models. Despite the large corpus of images (hundreds of thousand) gathered from graphically rendered virtual environments [38, 39], the lack of variability can negatively affect the training, ignoring the issue domain shift, which can be partly rectified using various domain adaptation techniques utilised in Chapters 5 and 6. Given larger and more varied datasets, natural or synthetic, the work can be expanded to function better in real-world applications, but that is beyond the scope of this work.

Additionally, other specific factors such as lack of test data and plausibility metrics can hinder rigorous evaluations, especially when it comes to the performance of an object removal approach, but we have attempted to carry out experiments using existing data and commonly-used metrics.

1.5 Thesis Structure

In Chapter 2, we present a review of the existing literature on all areas relevant to the work done for this thesis. This includes a comprehensive survey of the depth completion literature, recent advances made in

monocular depth estimation, image style transfer, domain adaptation, the use of temporal information in learning-based approaches and brief overview of the progress made in semantic segmentation.

Chapter 3 contains a discussion on the use of exemplar-based inpainting within the depth modality and two techniques are proposed that are capable of performing object removal in RGB-D images and completing naturally occurring holes within depth images. The proposed methods are evaluated against well-known commonly-used comparators within the literature.

In Chapter 4, a very efficient depth filling approach is presented that takes advantage of the fast-growing area of semantic segmentation to provide a platform for an object-wise approach capable of completing a depth image in three passes over the image. Evaluations further confirm the long-standing trade-off between efficiency and accuracy.

From Chapter 5, we will move into the territory of machine learning, convolutional neural networks and generative models. A leaning-based approach trained on a dataset consisting of corresponding pairs of synthetic RGB and depth images captured from a virtual environment enables us to fill large holes within depth images in a semantically meaningful way. This has been impossible in more traditional approaches as entire objects may need to be synthesised from scratch if they happen to fall inside the boundaries of the missing regions.

Chapter 6 explores the potentials of monocular depth estimation as an alternative to more conventional depth estimation techniques. We propose two monocular depth estimation techniques that are capable of generating high-quality depth images as their outputs hence removing the need for any depth completion or refinement as a post-processing step.

Chapter 7 will finally conclude the thesis by providing a summery of the techniques and their results presented in the previous chapters and discussing their implications and impacts in real-world applications. Possible directions for future work is also investigated and discussed.

As the primary focus of this thesis revolves around a computer vision problem with a significant visual component, we recommend the readers view this document in its digital form so the images can be better observed.

Chapter 2

Literature Review

Despite extensive work on 3D sensing of late [1–5], a number of limitations pertaining to environmental conditions, inter-object occlusion and sensor capabilities (as explained in Section 1.1) constrain fully-effective scene depth capture [1, 6]. As a result, significant research has been focused specifically on developing techniques to complete missing scene depth to increase the quality of the depth information for better applicability.

Although many have attempted to use traditional texture synthesis and structural image completion techniques, (in whole or in part) to address the problem of scene depth completion, challenges remain in terms of efficiency, depth continuity, surface relief and local feature preservation that have hindered flawless operation against high expectations of plausibility [40–45]. This chapter aims to present a review of prior work in the domain focusing on current state-of-the-art capabilities, shortcomings and future challenges. Although within the completion modality, the main focus of this study is on *scene depth completion*, a summary of the most influential approaches within colour image completion and synthesis are additionally presented to support this agenda.

Moreover, recent progress in the area of monocular depth estimation [33, 46–48] has lead to a cheap and innovative alternative to completely replace other more expensive and performance-limited depth sensing approaches such as stereo correspondence [13], structure from motion [49, 50] and depth from shading and light diffusion [51, 52], among others. Apart from computationally intensive demands and careful calibration requirements, these conventional depth sensing techniques suffer from a variety of quality issues including depth inhomogeneity, missing or invalid values and alike, which is why the need for depth completion and refinement in post precessing arises in the first place.

As a result, generating complete scene depth from a single image using a learning-based approach can be of significant value. Consequently, a portion of this chapter is dedicated to covering the state-of-the-art

monocular depth estimation techniques capable of producing complete depth which would eliminate any need for depth completion or refinement.

In this vein, this chapter presents a description of a number of most relevant state-of-the-art colour image completion techniques (Section 2.1) and a taxonomy of recent advances in scene depth completion covering aspects of problem formulation, spatial consistency, temporal continuity and others (Section 2.2). Furthermore, a brief outline of the existing literature on the recent advances in monocular depth estimation (Section 2.3), a history of the use of temporal information in various computer vision and scene understanding tasks (Section 2.6) and a short description of the relevant literature on domain adaptation (Section 2.4), image style transfer (Section 2.5), optical flow (Section 2.7) and semantic segmentation (Section 2.8) are also presented to provide a brief background for the contributions of this work.

Part of the material used in this chapter is based on the following papers, created during the work done for this thesis:

- **A. Atapour-Abarghouei** and T. P. Breckon, 'Dealing with Missing Depth: Recent Advances in Depth Image Completion and Estimation.' RGB-D Image Analysis and Processing. Springer, 2019 **(Under Review)**.
- **A. Atapour-Abarghouei** and T. P. Breckon, 'A Comparative Review of Plausible Hole Filling Strategies in the Context of Scene Depth Image Completion.' In Journal of Computers & Graphics, 2018, 72: pp. 39-58 [31].

2.1 RGB Image Completion

A long-standing and analogous challenge to the depth completion problem has been to present an efficient and robust approach to completing a colour image after a selected object or region is removed or alternatively to create a plausible synthesis of the image over a larger spatial area. Texture synthesis and image inpainting have both been widely studied due to their applicability in computer vision problems [53–60], such as occlusion filling, object removal, recovering missing data after transmission, image restoration and alike. Texture synthesis approaches are most effective against the challenges of generating or expanding pure texture patterns [53–55], while most inpainting methods mainly focus on pure structure and underlying geometry of what should be filled within a constrained target region [56–58]. However, there are a number of completion approaches that take advantage of the best of both by combining texture synthesis and structural inpainting in various ways [59, 60].

A synopsis of the original texture synthesis problem is slightly different from the concept of image completion relevant to this study. It can be defined as follows: given a small sample of a given texture

image, generate a larger similar-looking texture region without visible artefacts of repetition within the texture pattern of the larger region [53, 61–64]. Such a task can be relevant to plausible completion in the constrained target region (hole filling) case across both colour and depth.

However, most of the notable texture synthesis methods from the literature [54, 61–64] cannot be used for image completion. Whilst the method proposed in [53] is well-suited for constrained texture synthesis in the general sense, leading to significant influence within the image completion genre [60, 63, 65], its performance shortcomings in terms of speed (exhaustive search) and robustness (degenerate output) limit direct applicability to this wider context of use.

As for structure-based inpainting approaches, in most early techniques (focusing on the geometry of the shapes in the image), the isophotes, which are lines where the intensity value is the same, are continued smoothly into the target area that is to be inpainted. However, most methods in this group overlook one of the most important image components which plays a significant role in what the observer senses as reality: high fidelity (spatial frequency) texture. As a result, subsequent inpainting techniques began to incorporate the significant body of work from the related domain of texture synthesis into their inpainting techniques [60, 66, 67], which resulted in exemplar-based image completion methods.

With their focus on structure rather than texture, Bertalmio et al. [58] attempt to solve the problem of inpainting in a pioneering work by making use of higher order Partial Differential Equations and anisotropic diffusion to propagate pixel values along isophote directions. After consulting various experts on scene composition in the artistic sense, they created a general set of inpainting principles that have henceforth become widely-used standard guidelines for how inpainting algorithms should function, which remain highly relevant even in depth completion cases:

- After the inpainting process is completed, the inpainted target region must be consistent with the known region of the image to preserve global continuity.
- The structures present within the known region must be propagated and linked into the target region.
- The structures formed within the target region must be filled with colour consistent with what is in the known region.
- Texture is added into the target region after the structures are filled.

These principles are used in an iterative approach to fill the target region [58] and mimic the principles of generalised 3D object completion as identified in [9] with reference to the psychological literature on human visual perception. The direction of the propagation is calculated by estimating the image gradient after going through a ninety-degree rotation and improvements are subsequently made in each

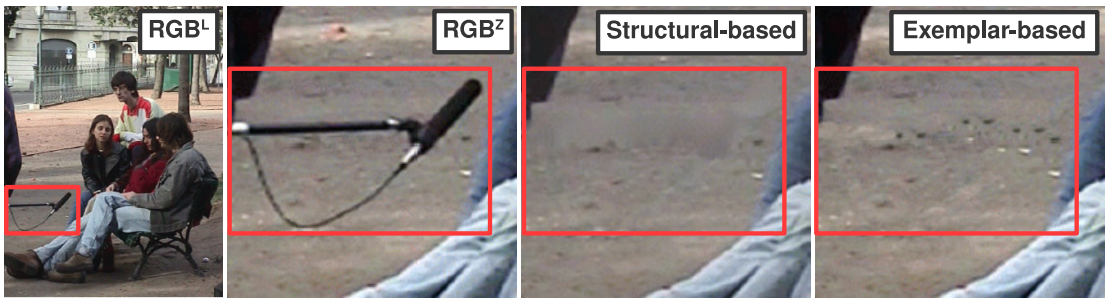

Figure 2.1: Results of the structural-based inpainting method of [58] compared to the exemplar-based approach of [67]. The microphone has been removed and inpainted compared to [67], the texture in the result of [58] is not accurate, leading to a perception of blurring. We can see the large original image (L) and an image zoomed in on the microphone (Z).

step by continuing the isophotes to generate structures, which are later filled with the appropriate colours sampled from the known regions.

While the approach [58] works well for small areas or smooth and untextured background regions, inpainting was by no means a solved problem and in the presence of fine texture, the approach fails to generate satisfactory results due to the fact that it mainly focuses on structure rather than structure and texture both. As seen in Figure 2.1, when compared to the approach in [67], the foreground microphone has been removed and inpainted but the texture in the result of [58] is not accurate, which creates an artificial perception of blurring.

Improved inpainting approaches began to emerge based on a range of techniques including fast marching method [68], Total Variational (TV) models [69–71] and exemplar-based methods that focus on 'synthesising' fine texture in the target region along with propagating structure [60, 65, 67, 72].

The notable work in [60], which is often used within the context of depth completion as well [73, 74] follows traditional exemplar-based texture synthesis methods by prioritising the order of filling based on the strength of the gradient along the target region boundary. Although there have previously been attempts to complete images via exemplar-based synthesis [65, 72], they are all lacking in either structure propagation or defining an explicit filling order that could prevent the introduction of blurring or distortion in shapes and structures. The approach in [60] makes use of the idea that exemplar-based methods are not only well-suited for two-dimensional texture but also capable of propagating isophotes and linear structures. An example of the results of this method is seen in Figure 2.2, where we see that plausible water texture has been synthesised in the target region after the person is removed from the original image. However, even though the algorithm is able to deal with texture and linear structure, it cannot handle curved structures and is highly dependent on the existence of similar pixel neighbourhoods in the

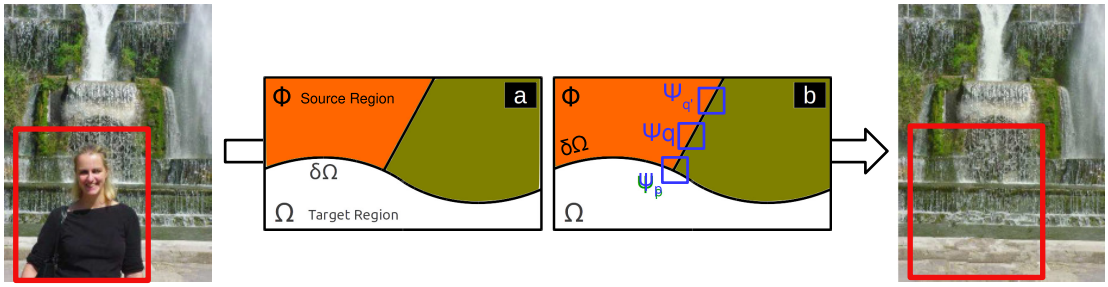

Figure 2.2: The overall outline and results of the method in the exemplar-based approach of [60].

sample region for plausible completion. Additionally, the approach heavily relies on the existence of fine texture to prioritise patches and can fail when dealing with large objects in more smooth depth images (Figure 2.3 - left).

While exemplar-based inpainting (sampling and copying patches from the known regions of the image) have been proven successful in many respects, there are limitations regarding the amount of samples available but more importantly, performance is significantly degraded when dealing with scenes that are not of a fronto-parallel view, which can create issues such as perspective handling within the completion process. Some methods have been proposed to combat this issue [75, 76] by including the transformed version of patches in the sample search space. The transformation can include rotation, scale, gain and bias colour adjustments. Although this can solve the problem of perspective and view angle and improve the performance of the completion process, it exponentially increases the size of the search space from 2 degrees of freedom per output pixel or patch to 8 (i.e. equivalent to a homography transformation) or more (photometric variations, e.g. bias/gain of intensity channels). Not only is the efficiency and speed of the process thus affected but also an elevated probability of taking a *local optimum* as the result can be expected.

Since then, many other image completion techniques [66, 78–86] have been proposed that are capable of completing large portions of an image successfully. For instance, exemplar-based inpainting has been formulated as a metric labelling problem [78], solved using simulated annealing, where the outputs are

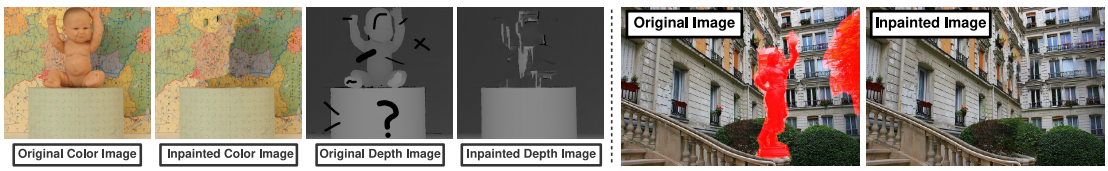

Figure 2.3: **Left:** an example of the results and process of exemplar-based inpainting [60]. **Right:** an example of the results of [77].

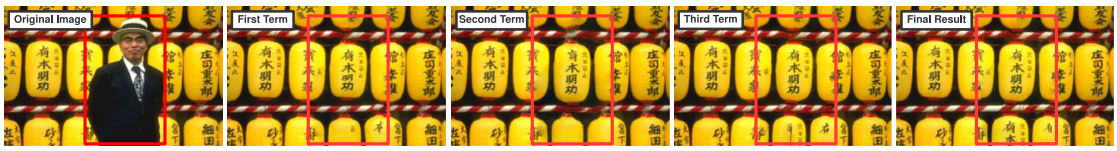

Figure 2.4: An example of the results of [86]. The figure demonstrates the results of the approach when different terms within the function are only used during the optimisation process.

evaluated using a support vector machine regressor [87] that measures distortions in the image from scene statistics of locally normalised luminance coefficients and their products.

Image completion has also been accomplished using schemes similar to energy minimisation methods [79–81], Markov Random Field models with labels assigned to patches [82] using belief propagation via priority scheduling, models represented as an optimal graph labelling problem, where the shift-map represents the selected label, solved by graph cuts [83] and combinations of diffusion methods with exemplar-based filling [66] based on a Gaussian image pyramid [88].

The work in [86] takes advantage of a variational model where the benefits of other commonly-used inpainting methods are combined by introducing three energy terms to create a more robust approach. Their first energy term represents texture synthesis as introduced in [53], the variational formulation of which was suggested in [89, 90]. The second energy term, discrete Laplacian operator applied to the target region, is used for propagation and diffusion and their third energy term maintains coherence inspired by [91, 92] by pushing for the similarity of the patches corresponding to neighbouring pixels. As seen in Figure 2.4 (middle lantern at the bottom row), the full approach functions more effectively than when individual energy terms corresponding to texture synthesis, propagation and diffusion and spatial coherence are applied to the input alone.

The approach in [77] builds upon [80] by adding the mid-level scene understanding constraints of translational regularity and planar perspective to guide the image completion process. Perspective parameters of detected planes are first estimated and then scale-invariant feature (SIFT) matching [93] is used to specify translational regularities in each plane. The completion process is guided by incorporating the detected perspective planes and the translational regularities into the prior probabilities of the search space. Although the results are satisfactory and plausible even for large target regions (Figure 2.3 - right), the computations required to add the search constraints make it relatively expensive.

There have also been various works of research based on the notion of using external images to complete the target region of an image. The approach in [94] extracts semantically similar images from a gigantic database of photographs and fills the target region by copying a region from a semantically valid image (Figure 2.5 - left). Similarly, [95] uses an external dataset query [96] by performing a viewpoint invariant image search. The obtained results are geometrically and photometrically registered with the input image

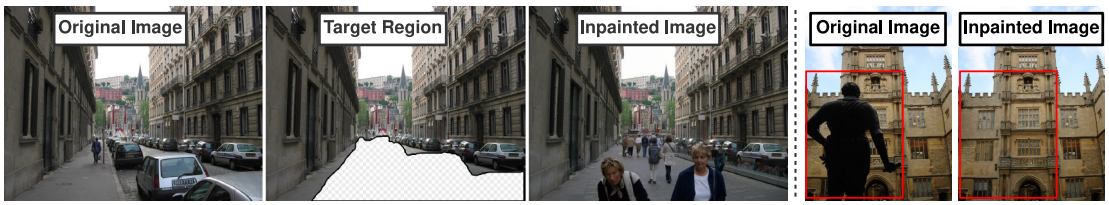

Figure 2.5: **Left:** result of exemplar-based inpainting using an external database [94]. **Right:** an example of the results of [95].

using multiple homographies and a global affine transformation and the seams between target and known regions are dealt with using Markov random field optimisation (Figure 2.5 - right).

Deep neural networks have recently revolutionised the state of the art in many computer vision tasks such as image stylisation [97–100], super-resolution [101, 102] and colourisation [103]. Image completion has also seen its fair share of progress using such techniques. In [104], an approach is proposed that is capable of predicting missing regions in an RGB image via adversarial training of a generative model [105]. In a related work, [106] utilises an analogous framework with similar loss functions to map the input image with missing or corrupted regions to a latent vector, which in turn is passed through their generator network that recovers the target content. The approach in [107] proposes a joint optimisation framework composed of two separate networks, a content encoder, based on [104], which is tasked to preserve contextual structures within the image and a texture network, which enforces similarity of the fine texture within and without the target region using neural patches [108]. The model is capable of completing higher resolution images than [104, 106] but at the cost of greater inference time since the final output is not achievable via a single forward pass through the network.

More recently, significantly better results have been achieved using [109], which improves on the model in [104] by introducing global and local discriminators as adversarial loss components. The global discriminator assesses whether the completed image is coherent as a whole, while the local discriminator concentrates on small areas within the target region to enforce local consistency. Similarly, [110] trains a fully-convolutional neural network capable of not only synthesising geometric image structures but also explicitly using image features surrounding the target region as references during training to make better predictions.

Not unlike most RGB image inpainting techniques, these learning-based image completion approaches are well capable of generating perceptually plausible outputs despite the significant corruption applied to the input. However, when it comes to depth, they are incapable of producing high-quality outputs due in part to the significantly higher number of target regions (holes) both large and small over the smoother surfaces in depth images. Examples of these novel approaches applied to depth images is seen in Figure

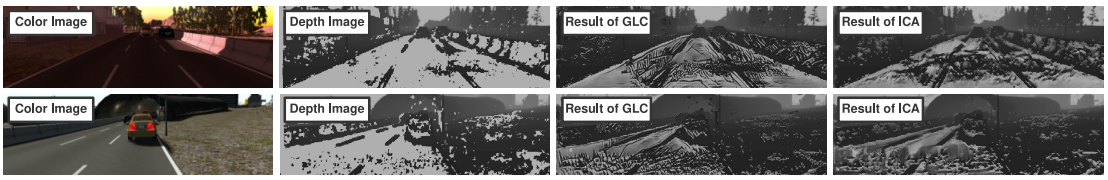

Figure 2.6: Results of global and local completion (GLC) [109] compared to inpainting with contextual attention (ICA) [110] applied to depth images.

2.6, which indicates how ineffective learning-based RGB image inpainting approaches can be within the depth modality, which is why depth completion techniques focusing on filling depth holes are of utmost importance.

The contents of this section certainly do not represent the entirety of the literature on colour image completion and are not the focus of this study. As such, since wide-expanding surveys already exists on the issues of texture synthesis and inpainting within the context of colour images [111–115], we will not delve any further into the subject and only refer to techniques directly pertinent to depth images.

2.2 Depth Completion

One of the most important steps in addressing any problem, such as that of depth completion, is to focus on how the problem can be formulated. Numerous research works have attempted to solve the depth completion problem by concentrating on different challenges within the domain. In the following, a general overview of the most common formulations of the depth completion problem is presented before moving on to discussions on a taxonomy of the depth completion literature (Section 2.2.2).

2.2.1 Problem Formulation

Creatively reformulating an ill-posed problem such as scene depth completion and inpainting will lead to solutions that can fulfil particular required elements pertaining to certain situations, including time, computation, accuracy and alike. In this section, we will discuss some of the most common ways in which depth completion has been posed and solved as a problem along with the effects each reformulation can have on the results.

Anisotropic Diffusion

Formulating the image completion and de-noising problem as anisotropic diffusion [116] has been a long-standing and successful technique in the field of colour image inpainting [58, 70, 86, 117]. As such,

diffusion-based solutions have also entered the realm of depth completion, since the smoothing and edge-preserving qualities of the diffusion-based depth completion output is desirable in certain downstream applications such as localisation and mapping [118, 119].

Anisotropic diffusion is a non-linear partial differential equation scheme [116] with edge-preserving smoothing qualities. As a space-variant transformation of an input image, it generates a family of smoothed parametrised images, each of which corresponds with a filter that depends on the local statistics of the input image.

More formally, let $I(\cdot, t)$ be a family of parametrised images, then the anisotropic diffusion is:

$$I_t = div(c(x, y, t)\nabla I) = c(x, y, t)\Delta I = \nabla c \cdot \nabla I, \tag{2.1}$$

where div is the divergence operator, ∇ and Δ denote the gradient and Laplacian operators respectively and $c(x, y, t)$ is the diffusion coefficient, which can be a constant or a function of the image gradient.

Equation 2.1 can be discretised using a 4-neighbourhood scheme, as in [120] where the colour image is used to guide the diffusion in an iterative process. In this approach [120], the depth image is completed at a low resolution and the ensuing iterative colour-guided anisotropic diffusion within the up-sampling steps corrects the depth image (see an example of the results of the approach [120] in Figure 2.9).

Another example of the use of diffusion in depth completion can be seen in [121]. The approach attempts to fill depth holes by extracting the edges from the accompanying colour image captured from a structured-light device. Subsequently, different diffusion algorithms are applied to smooth and edge regions. The separation of these regions before the diffusion process is performed based on the observation that surfaces which need to be smooth in the depth may be textured in the colour image and object boundaries within the depth image can be missed during the colour edge extraction process due to the low contrast in the colour image.

Using diffusion methods, the resulting completed depth image can be smooth in the presence of flat planes with sharp edges. While smooth surfaces and strong edges and object boundaries can be very desirable traits in a depth image, the implementation requires discretization and will bring forth numerical stability issues and is computationally expensive. The longer run-time of diffusion-based methods makes them intractable for real-time applications.

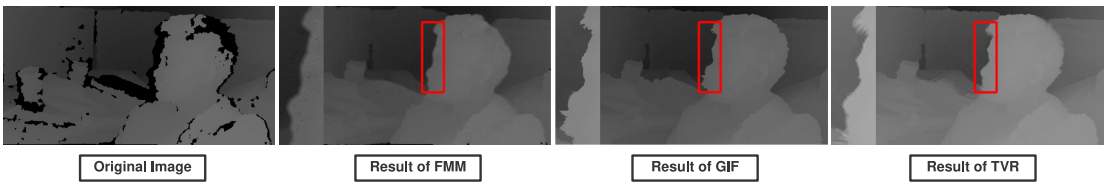

Figure 2.7: An example of the results of depth completion using energy minimisation with TV regularisation (TVR) [122] compared to the fast marching method based inpainting (FMM) [68] and guided inpainting and filtering (GIF) [41]. The energy function assumes that in small local neighbourhoods, depth and colour values are linearly correlated.

Energy Minimisation

Following the success of energy minimisation used within the colour image completion framework [79–81], the technique has been used in various depth completion approaches. The foundations of an energy minimisation approach stem from certain assumptions made about the colour and/or depth image, based on which an energy function is designed. The function is subsequently optimised, completing and enhancing the original image based on the criteria set within the different terms added to said energy function.

Depth completion approaches using energy minimisation are mostly accurate and produce plausible results but more importantly, the capability of these approaches to focus on specific features within the image based on the terms added to the energy function is highly advantageous.

For instance, the energy function in [123] incorporates the characteristics of a depth image acquired via a structured light device into the completion process. The noise model of the capture device and structure information are taken into account using terms added to the energy function, performing the regularisation during the minimisation process.

The approach in [122] assumes a linear correlation between depth and colour values within small local neighbourhoods. In this approach [122], an additional regularisation term based on [124] enforces sparsity in vertical and horizontal gradients of the depth image, resulting in more crisp object boundaries with less noise. An example of the results of [122] is seen in Figure 2.7. The work of [125] includes a data term that favours pixels surrounding hole boundaries and a smoothing prior that encourage flat and smooth surfaces within the depth image. This is very advantageous in terms of geometry and structure of the scene following the design of the energy function, even though important information such as relief and texture is lost in the output.

Designing an energy function based on the characteristics of the input and the requirements of the output can be very beneficial, as the function can be modified or regularised to produce desirable outputs based on the specific application of the resulting depth image. However, the optimisation process can come with implementation difficulties, numerical instabilities and computationally intensive necessities.

Figure 2.8: Examples representing challenges involving depth textures (captured using Microsoft Kinect v2). When captured from close proximity, depth values of highly-textured objects are missing due to the short camera distance (left). The same objects captured from a distance are smooth with little granular relief (right).

Exemplar-based Filling

One of the most important challenges involving the depth completion problem is related to texture, which not unlike the related work in colour image completion can be solved by copying and pasting textured patches from the known regions of the image (exemplar-based image completion). However, there can be major pitfalls with using an exemplar-based technique used for colour image completion for a depth image.

While texture and relief are very important in many modern computer vision tasks, most active depth sensing devices do not cope well with texture, which is why missing and invalid depth values in close views of highly-textured regions is commonplace. In Figure 2.8 (left), we can see that an attempt to capture the depth of a highly-textured object from a close distance fails, due to the fact that most active depth sensors (in this case a time-of-flight camera) cannot deal with objects *too close to the camera*. On the other hand, if more distance is allowed between the camera and the textured object to resolve this issue, the resulting depth image is more smooth and shape-based than the equivalent colour image (Figure 2.8 - right). We can see in Figure 2.8 (left) that the colour (RGB) image contains highly-textured objects close to the capture device, while all the depth values in these objects are missing in the depth (D) image, because of the close proximity of the camera to the objects in the scene. Figure 2.8 (right) indicates that the same objects captured from a distance, while still visibly textured in the colour image (representing the human perception of relief), are far smoother in the depth image. Passively obtained depth is normally better textured than depth information acquired through active devices but the amount of captured texture and fine relief is still not comparable to the relief perceived from the scene by a human observer (Figure 1.2).

Furthermore, simply assuming that a depth image is just a gray-scale image with no texture [126] is a significant oversight and ignores the many potentials an accurate and textured depth image can have.

Even though there have been many attempts to directly use structure-based or exemplar-based *RGB image completion* approaches for depth completion [68, 73, 74], particular factors create obstacles. As mentioned earlier, a depth image is not as visibly textured as a colour image of the same scene. Therefore, when a structural inpainting technique is being used to propagate the shapes and structures into the target regions, identifying the points at which the propagation must be terminated is challenging. There is little texture present and in many cases, object boundaries lie within or adjacent to the holes, which makes detecting them extremely challenging.

Formulating depth completion as an exemplar-based completion problem based on specific depth image characteristics can rectify many of these issues but is not without its own challenges. Lack of colour texture on a smooth surface which leads to unified depth can confuse an exemplar-based approach to a great degree. As seen in Figure 2.3 (left), the notable exemplar-based inpainting method of [60] is capable of filling the target region post object removal from the colour image in a reasonably plausible way due to the existence of colour texture in the background (Figure 2.3 - left) but within the context of depth, when colour texture is removed and uniform depth of a flat plane is all that is left, results are not nearly as impressive (Figure 2.3 - left). Please note that the goal is to remove an object (the baby) from both the colour and depth images and plausibly complete the remaining hole post removal and at the same time fill the existing holes in the depth image (represented by black markings on the depth image).

Moreover, attempting to replicate texture usually requires copying pixels or entire patches from the known regions of the image and using them to fill the holes. One drawback stems from the fact that there may not be enough useful information in the known region to sample from, which is a very common problem in completing depth images if they are not of a fronto-parallel view, which does occur with colour image completion as well. However, as mentioned before, the problem can be solved for RGB images by including the transformed version of the patches by varying rotation, scale, shear, aspect ratio, keystone corrections, gain and bias colour adjustments and other photometric transformations in the search space when trying to find similar patches to sample from. This will exponentially increases the size of the search space and effect the efficiency and accuracy but is still a solution to the problem, nonetheless. However, with depth images, there may be scenes in which no suitable patch can be found to fill a specific part of the hole even if the search space contains all possible transformed patches from the input.

Imagine Figure 2.9 (right) contains the outline of a scene. As we go deeper into the image, the intensity values in the colour channels of the image may change. However, the hue remains the same, while the illumination changes. Therefore, by transforming patches sampled from the known region (outside the black rectangle in Figure 2.9 - right) using homographic and photometric transformations like illumination, suitable samples can be found that can fill the target region.

On the other hand, assume Figure 2.9 (right) represents an *ideally* accurate depth image where the depth continuously varies from pixel to pixel as we move deeper into the image (i.e. in a row of pixels on the

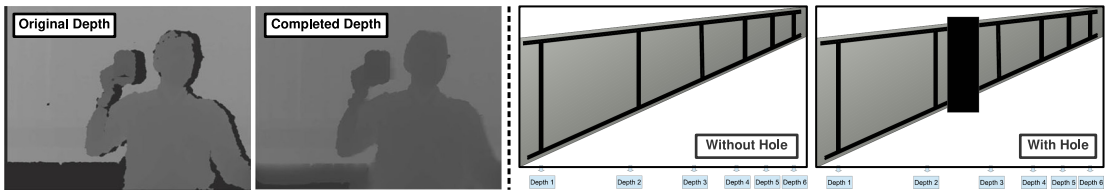

Figure 2.9: **Left:** the results of the method in [120]. The approach is an anisotropic diffusion-based method with real-time capabilities. **Right:** a simple virtual image used to describe the issues of inpainting methods applied to depth images.

fence, no two pixels have the same depth value). In a scenario like this, neither homography transformation nor any commonly-used photometric variation can guarantee that patches exist in the resulting search space that can be used to accurately fill the target region. Essentially, the 3D depth variation of the scene is captured within the 2D topology of the depth image but exemplar-based completion, following a 2D paradigm, will inherently fail in such an *ideal* depth image.

It should be noted, however, that the depth images captured using current 3D sensing technology are not *ideal* and in practice, patches that fit the criteria required to fill the target are often found in the depth images obtained through the currently existing technology but this does not guarantee that an exemplar-based image completion solution will always fill depth images successfully as it does within colour images. That said, there are specific depth completion techniques that still take advantage of classic inpainting approaches such as [60] and [68], which have been commonly employed, with or without additional improvements, for depth value filling [41, 68, 85, 126, 127]. In this vein, some of the work presented in this thesis (Chapter 3) is also focused on completing depth images within an exemplar-based framework.

The work in [128] attempts to perform object removal in multi-view images with an extracted depth image and uses both structure propagation and structure-guided completion to fill the images, which results in better geometric and structural coherence. The target region is completed in one of a set of multi-view photographs casually taken in a scene. The obtained images are first used to estimate depth via Structure from Motion (SfM). Structure propagation and structure-guided completion are employed to create the final results after an initial colour and depth completion step. The individual steps of this algorithm use the colour image completion method of [80] and the patch based exemplar approach of [76] to generate results. The approach is relatively costly due to the fact that each of the three steps require an independent form of image completion.

On the other hand, [129] proposes a colour and depth inpainting method using a segmentation-based approach in stereo images. The approach makes use of the fact that parts of the removed region in one stereo image may still be visible in the other and tries to complete both images via 3D warping. The

process subsequently involves completing both colour and depth via depth-assisted texture synthesis, a modified version of the well-known exemplar-based filling technique in [60]. However, in cases where stereo or multi-camera views are not available, as in many active 3D sensing devices such as time-of-flight (ToF) cameras but missing depth data is abundant (e.g. Figure 2.8), other completion approaches not dependent on stereo or multi-view images have to be used to fill the naturally occurring holes in depth images. Furthermore, this method has no built-in mechanism to handle large structures and geometric structures are not accounted for.

Another occasion where exemplar-based depth completion is often used is in Depth Image-Based Rendering techniques (DIBR). This is an extension to Image-Based Rendering (IBR) that tries to create a novel *virtual* view from a set of 'real' views, with the added benefit of having depth information available. The images are normally warped and combined to create new synthetic views [130] but the greatest part of the challenge is to deal with the newly exposed holes that are created after the warping. There have been attempts to solve the depth image issues using exemplar-based image completion techniques such as the one proposed in [60]. Daribo and Saito [73] directly utilise this method in their approach to DIBR. The work of [74] has modified said method to complete stereo-vision generated disparity maps, where the information from the complementary disparity is used to fill the missing information.

Solving the depth completion problem using an exemplar-based framework has the potential to produce outputs in which structural continuity within the scene is preserved and granular relief texture is accurately and consistently replicated in the missing depth regions. However, if the scene depth is not captured from a fronto-parallel view, there is no guarantee that correct depth values can be predicted for the missing regions even if the patches sampled within the exemplar-based completion approach undergo different transformations.

Matrix Completion

Matrix completion has recently emerged as an interesting formulation of the image completion problem, especially since it has been observed [131] that similar patches in an RGB-D image lie in a low-dimensional subspace and can be approximated by a matrix with a low rank. The approach in [131] presents a linear algebraic method for low-rank matrix completion-based depth image enhancement to simultaneously remove noise and complete depth images using the corresponding RGB images, even if they contain heavily visible noise. In order to accomplish simultaneous de-noising and completion, the low-rank subspace constraint is enforced on a matrix with RGB-D patches via incomplete factorisation, which results in capturing the potentially scene-dependent image structures both in the depth and colour space.

The rank differs from patch to patch depending on the image structures, so a method is proposed to automatically estimate a rank number based on the data. Figure 2.10 demonstrates the performance

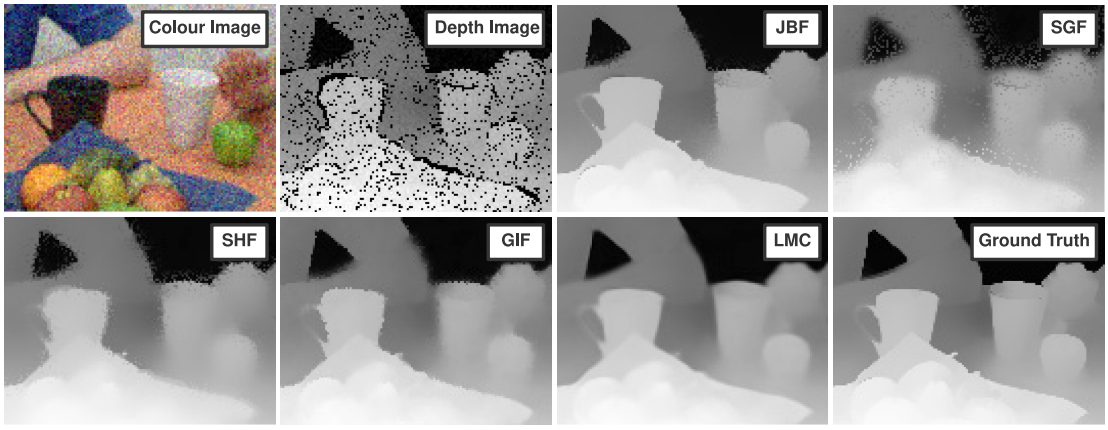

Figure 2.10: Demonstrating the results of the matrix completion technique of [131] using low-rank operations (denoted by LMC) compared to joint bilateral filtering (JBF) [132], structure guided fusion (SGF) [133], spatio-temporal hole filling (SHF) [45] and guided inpainting and filtering (GIF) [41].

capabilities of this approach compared to other depth completion methods, such as joint bilateral filtering (JBF) [132], structure guided fusion (SGF) [133], spatio-temporal hole filling (SHF) [45] and guided inpainting and filtering (GIF) [41]. This approach [131] generates particularly impressive results in that the input RGB image is very noisy (Figure 2.10 - Colour Image). Before the comparisons, a de-noising method [134] is applied to the noisy RGB image used as an input by the comparators.

The work in [135] points out, however, that the low-rank assumption does not fully take advantage of the characteristics of depth images. Sparse gradient regularisation can naively penalise non-zero gradients within the image but based on statistical observations, it is demonstrated that despite most pixels having zero gradients, there is still a relatively significant number of pixels with gradients of 1. Therefore, a low-gradient regularisation scheme is proposed in which the penalty for gradient 1 is reduced while non-zero gradients are penalised to allow for gradual changes within the depth image. This regularisation approach is subsequently integrated with the low-rank regularisation for depth completion.

Image Translation

More recently, with the advent of deep neural networks, many image generation problems such as RGB inpainting [104, 106, 107, 109, 110] are essentially formulated as an image-to-image translation problem using a mapping function approximated by a deep network directly supervised on ground truth samples. However, as seen in Figure 2.6, networks designed to complete RGB images might not work well when it comes to depth. A significant obstacle to creating a neural network trained to complete scene depth is the lack of hole-free ground truth depth images available.

To overcome this problem, [136] creates a dataset of RGB-D images based on available surface meshes reconstructed from multi-view RGB-D scans of large environments [137]. Reconstructed meshes from different camera poses are rendered, which produces a supply of complete RGB-D images. This data is subsequently utilised to train a network that produces dense surface normals and occlusion boundaries. The outputs are then combined with raw depth data provided by a consumer RGB-D sensor to predict all depth pixels including those missing (holes).

The work presented in Chapter 5 of this thesis also proposes a solution to the problem of depth completion formulated as an image translation problem addressed via a mapping function approximated by a supervised model trained on synthetic RGB-D images.

Discussion: The problem of depth image completion, being an inherently ill-posed one, can of course be formulated in a variety of ways, including but not limited to diffusion, energy minimisation, exemplar-based completion and alike. Reformulating the depth completion problem results in a variety of solutions that generate completed depth images with different qualities appropriate for the application for which the depth information is intended. Additionally, there is great potential in attempting to complete an image using a learning-based approach that is capable of understanding the scene intricacies, objects and their spatial relationships.

In the upcoming sections, depth completion strategies are categorised based on three different characterisations: their dependence on the accompanying colour image (which may not always be available), the main objective focus that an approach attempts to fulfil (associated with the principles of inpainting outlined by [58] explained in Section 2.1) and the type of information used within the scene to complete missing depth.

2.2.2 Information Domain Used for Depth Completion

There are three general types of approaches commonly used to deal with holes in depth images obtained using active or passive 3D capture methods based on the domain of information used to carry out the completion process. An approach may only use the spatial information locally contained within the depth and potentially the accompanying colour image, temporal information extracted from a sequence used to complete or homogenise the scene depth, or even a combination of both in various ways. A brief overview of the approaches penalising these types of input information is presented in this section. Furthermore, Table 2.1 provides a short summary of the advantages and the disadvantages of all the categories. Please note that the listed advantages and disadvantages for a given class of approaches in the table obviously vary in degree and strength for different methods in that category and are generalised to be more comprehensive. Figure 2.11 provides a general overview of depth completion techniques categorised based on their input dependencies and required information domain.

Categories	Subcategories	Advantages	Disadvantages	Examples
Spatial-Based Methods	Filtering, Interpolation, Extrapolation	• Simple implementation. • Potential to be fast and efficient. • Potential to provide a clean image.	• Edges and boundaries may be smoothed. • Undesirable depth discontinuities may stay. • May fail in filling large holes.	[36, 40, 42, 138–146]
	Inpainting-based	• Effective in smooth regions. • Edges are not smoothed by mistake.	• Artefacts may be added around boundaries and discontinuities. • Not very fast and efficient.	[41, 73, 74, 120, 121, 126, 133]
	Reconstruction-based	• Higher levels of accuracy in both edge and smooth regions. • No artefacts around boundaries and edges.	• Mostly require guidance from colour image. • Complicated implementation process.	[122, 123, 147–150]
Temporal-Based Methods		• Not limited by sampling constraints. • Texture can be preserved more accurately. • Can maintain depth consistency in a sequence.	• Delays are noticeable in presenting results. • Mostly suited for off-line applications. • Incapable of completing a single image.	[44, 151–154]
Spatio-temporal-Based Methods		• Can avoid jagging and blurring usual in spatial-based methods. • Potential to be more efficient than temporal-based methods.	• Delays still exist in on-line applications. • Cannot complete a single depth effectively.	[45, 132, 155–158]

Table 2.1: Advantages and disadvantages of different categories of depth completion methods.

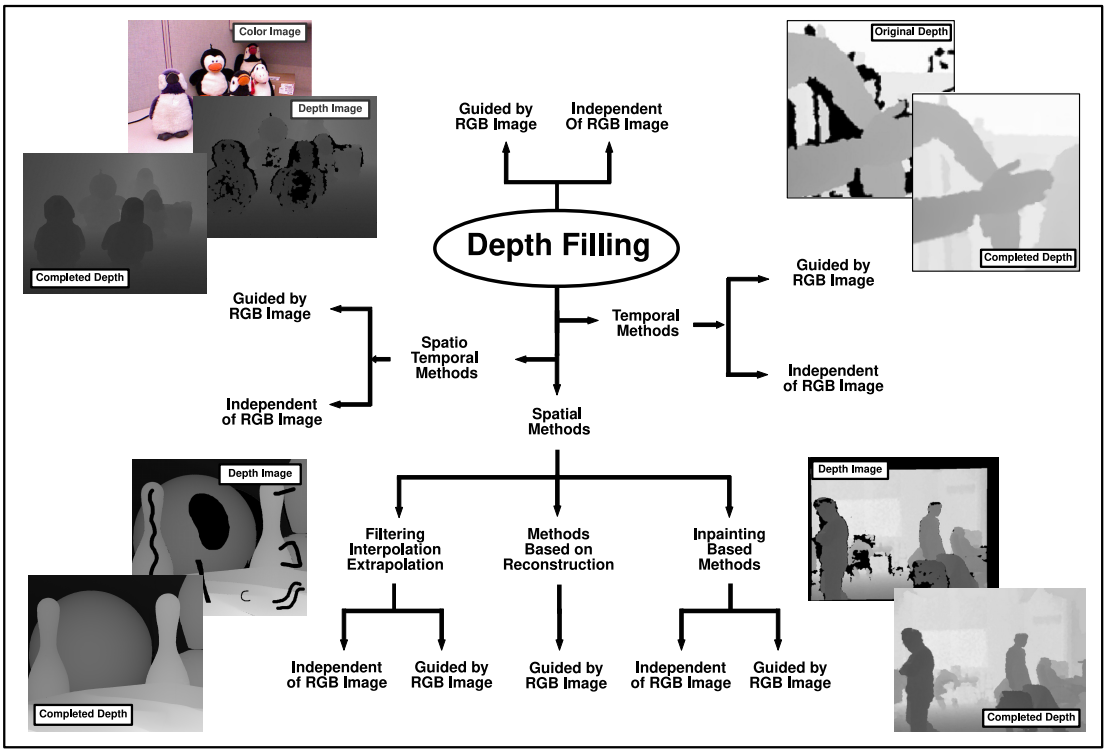

Figure 2.11: A diagrammatic taxonomy of depth completion based on inputs and information domain used during the completion process.

Spatial-based Depth Completion

The methods in the first group of depth completion approaches use the neighbouring pixel values and other information available in a single depth image to complete any missing or invalid data in the depth image. There are also several approaches that take advantage of the information available in the colour image of the same scene to fill the missing data in the depth image.

Even though there are clear limitations to using this type of approach, such as a possible lack of sufficient information contained within the image which can be used to complete a particular hole region, there are many important advantages. For instance, when temporal and motion information is taken into consideration in depth completion, completing one frame in a video requires processing multiple consecutive frames around it and so either the processing has to be done off-line or if real-time results are needed, the results of each frame will appear with a delay. However, if there is no dependence on other frames, with an efficient spatial-based method, real-time results can be generated without any delay.

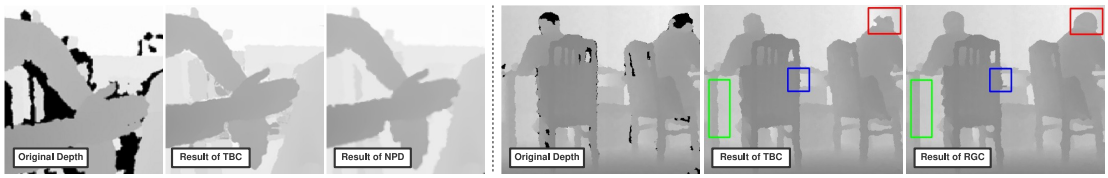

Figure 2.12: **Left:** example of the results of neighbouring pixel distribution (NPD) approach [138] compared to temporal based completion (TBC) of [44]. **Right:** example of the results of the region growing completion (RGC) technique in [139] compared to temporal based completion (TBC) [44]. The depth is filled using region growing based on the accompanying colour image.

Numerous spatial-based methods exist within the literature, the majority of which can be categorised in three different classes: methods that rely upon filtering, interpolation and extrapolation techniques, inpainting-based methods and finally, reconstruction-based methods. Examples of the seminal works in these areas are presented in Table 2.1.

Filtering, Interpolation and Extrapolation

The easiest, yet not always the best, solution to the depth completion problem is applying a filter to the depth data. Some common filters of choice would be the median filter [159] or the Gaussian filter [160] but with their use comes significant blurring and loss of texture and edge detail. As mentioned earlier, there are specific filters that have edge-preserving qualities, such as the bilateral filter [161] and non-local filter [162]. However, these filters will not only preserve edges at object boundaries but the undesirable depth discontinuities caused by depth sensing issues as well. There is also a possibility of distortion in non-hole regions.

As with most depth images obtained via structured light devices, stereo correspondence and so forth, there is a secondary colour or gray-scale image. The visual information present in this accompanying image can be employed to improve the accuracy of the depth image within or near object boundaries (Table 2.2). It has also been utilised to reduce the noise in depth images that is generated by up-sampling procedures [43, 163–166], where the goal is to increase the sharpness, accuracy and the resolution of the depth image. Moreover, it can also be used to assist filtering approaches, as seen in methods such as joint-bilateral filtering [167], joint trilateral filtering [168] and alike.

A fast and non-approximate linear-time guided filtering method is proposed in [169], the output of which is generated based on the contents of a guidance image. It can transfer the structures of the guidance image into the output and has edge-preserving qualities like the bilateral filter but can perform even better near object boundaries and edges by avoiding reversal artefacts. Due to its efficiency and performance, it has been used as the basis for several depth completion methods [41, 148].

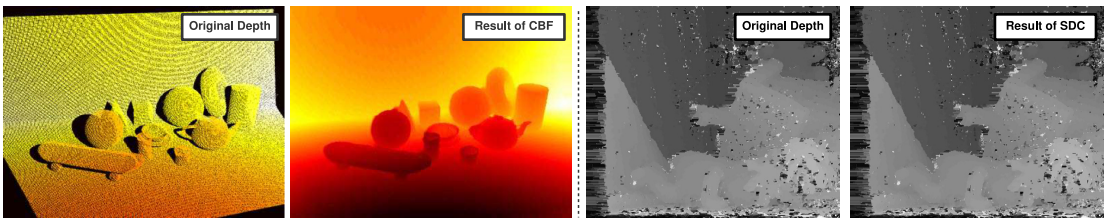

Figure 2.13: **Left:** example of depth completion using cross bilateral filtering (CBF) [170]. **Right:** example of depth enhancement via [146]. Noise is removed across object boundaries via a slope depth compensation (SDC) filter [146].

The approach in [138] completes individual depth holes based on the depth distribution of their neighbouring pixels after labelling each hole and dilating each labelled hole to get the value of the surrounding pixels. Cross-bilateral filtering is subsequently used to refine the results. In Figure 2.12 (left), the results are compared with the temporal based method in [44], which will be reviewed subsequently.
In [139], invalid and missing depth information is detected and filled using a region growing technique based on the accompanying colour image. To increase the overall accuracy, joint bilateral filtering is utilised. Once again, since the detection and filling of invalid depth values depends on the colour image, in regions where the colour values do not match the depth values, validity of the filled hole is questionable even though the results seem plausible and without visible defects. Figure 2.12 (right) demonstrates how the method can fill depth holes without adding artefacts or blurring.
In the method proposed in [140], an approach based on weighted mode filtering and a joint histogram of the colour image and the depth image is used. A weight value is calculated according to the colour similarity between the target and neighbouring pixels on the colour image and used for counting each bin on the joint histogram of the depth image. Subsequently, the approach is expanded to include temporal information for a temporally consistent estimate on the depth video. This method is effective against depth values being blurred on the boundaries.
With regards to improving depth images after a novel virtual viewpoint has been created in DIBR (Depth Image-Based Rendering), [141] utilises a simple average filter to fill depth holes. However, to avoid smoothing and blurring the textured regions and edges, an adaptive method that considers edges and directions is used to enhance the accuracy of object boundaries. The approach in [142] makes use of simple filtering but based on a weighted Gaussian filter taking into account the distance to the contours, so as to apply smoothing close to object boundaries but avoid filtering the smooth areas in the depth image. However, in both of these methods, the novel virtual viewpoint is on the same axis as the real view point, which restricts the applicability of the approach.
The work on [171] uses adaptive cross-trilateral median filtering to reduce the noise and inaccuracies commonly found in depth estimates obtained via stereo correspondence. Parameters of the filter are

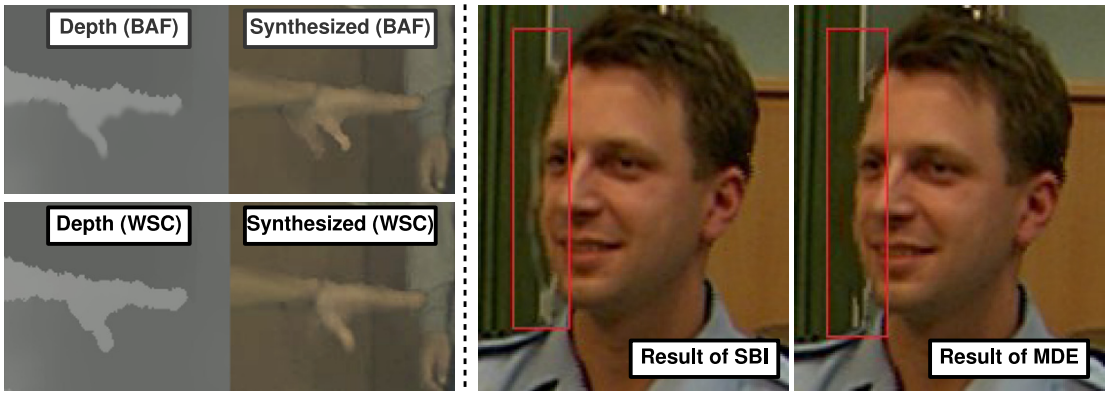

Figure 2.14: **Left:** example of the results of brightness adaptive filter (BAF) [40] compared to watershed segmentation based completion (WSC) [144]. **Right:** example of the result of multi-directional extrapolation (MDE) techniques of [42] compared to structural based inpainting (SBI) [58].

adapted to the local structures and a confidence kernel is employed in selecting the filter weights to reduce the number of mismatches.

In [143], object boundaries are first extracted and then a discontinuity-adaptive smoothing filter is applied based on the distance of the object boundary and the quantity of depth discontinuities. [170] proposes a propagation method, inspired by [172], which makes use of a cross bilateral filter to fill the holes in the warped image. Directional depth information is propagated based on camera calibration to fill the holes caused by disocclusion from 3D warping. Whilst the method produces good results (Figure 2.13 - left), it only accounts for holes caused by transformation and warping.

The technique proposed in [40] handles false contours and noisy artefacts that exist in the depth information estimated through stereo correspondence methods. The approach is based on the use of a joint multilateral filter that consists of kernels measuring proximity of depth samples, similarity between the values of said samples and similarity between the corresponding colour values. The shape of the filter is adaptive to brightness variations. Although the results are promising, there are instances of blurring in the resulting depth images (Figure 2.14 - left).

There are interpolation techniques that fill the holes horizontally or vertically within the boundary of the hole by calculating a normalised distance between opposite points of the border (horizontally or vertically) and interpolating the pixels accordingly. These types of approaches will face obvious problems when the target covers parts of certain structures that are neither horizontal nor vertical. [42] proposes a multi-directional extrapolation technique for completion that uses the neighbouring pixel texture features to estimate the direction in which extrapolation is to take place, rather than using the classic horizontal or vertical directions that create obvious deficiencies in the completed image. Sets of nine directions are

Figure 2.15: Results of the layer labelling approach (LLA) of [173] and spatio-temporal completion (STC) method of [45]. Depth is completed in [173] by assuming the existence of and subsequently labelling different depth layers for foreground and background objects.

proposed to fill the holes so that there is a higher possibility for the completed holes to match the texture or structure of the background and the surrounding objects (Figure 2.14 - right).

The approach proposed in [173] separate the scene into a static background and a number of dynamic foreground objects by assuming different depth layers. As a result, a stochastic framework that combines various RGB-D noise models is proposed to determine the label of each depth layer. In order to fill the missing depth values, joint bilateral filter is used, considering the fact that only the neighbouring pixels that are on the same depth layer contribute to the process of completing the central pixel. Furthermore, not only are missing depth pixels filled, erroneous depth values are corrected by identifying pixels whose values significantly differ from other neighbouring values and refilling them as if they were holes. Figure 2.15 demonstrates the effectiveness of this method compared to the method proposed in [45].

The work in [144] criticizes the use of bilateral and trilateral filters as the major solution used in completing, enhancing and refining depth images in DIBR (Depth Image-Based Rendering) [40, 170] by pointing out that artefacts around edges and object boundaries still exist due to the fact that colour and depth edges are characteristically different. While the method in [144] does not focus on actually filling depth holes, it does attempt to remove and refine the artefacts that can usually be seen in and around filled areas after the holes have been filled using other methods such as [42, 60, 68, 174]. Watershed colour segmentation [175] is used to correct any misalignments and enhance disoccluded regions and sharp depth edges within or without object boundaries by extending the object boundaries in depth images to cover the transitional edge regions of colour images (Figure 2.14 - left). Although the resulting depth images are without any burring, the segmentation adds to the computational cost of the approach.

The approach in [145] uses a joint trilateral filtering method made up of domain, range and depth filters. In this approach, local patch pattern matching is first performed between the image and the depth image and the results are used to tune the parameters of the filter. The range and depth filters are thus adjusted in a way that the edges in the depth image that accurately correspond with the image edges are rewarded and therefore, sharper object boundaries are produced.

Similarly, [146] propose a depth refinement technique that is not specifically designed for depth completion but its elements can certainly be used in completing missing depth regions. Their filter attempts to reduce noise by matching the boundary of an object in the colour image with the boundary of the object in the depth image. Blurring and ringing effects across the boundary of the object are subsequently removed using an additional slope depth compensation filter. The method is not very computationally costly but the authors note that there is always a trade-off between efficiency and accuracy. An example of the results of the method when applied to depth images with large quantities of noise and holes can be seen in Figure 2.13 (right).

In [43], a segmentation-based interpolation technique is proposed to up-sample, refine and enhance depth images. The strategy uses segmentation methods that combine depth and colour information [176, 177] in the presence of texture or segmentation techniques based on graph cuts [178] when the image is not particularly textured to identify the surfaces and objects in the colour image, which are assumed to align with those in the depth image. The low-resolution depth image is later projected on the segmented colour image and interpolation is subsequently performed on the output. This method is highly dependent on the precision of the registration between the colour image and the depth image and the accuracy of the standard segmentation step.

Discussion: Among spatial-based depth completion strategies, filtering, interpolation and extrapolation approaches are of the most used and most efficient methods. Filtering methods are widely used in filling depth holes but most of them have a tendency to blur the image, introduce artefacts around boundaries and produce invalid edges. As seen in many of the aforementioned examples, many researchers try to overcome these issues by combining the filtering techniques with other methods, constraining their filtering elements, or adding post-processing stages to refine the filled data. Although many of these solutions are effective, they tend to diminish one of the most valuable aspects of this type of depth completion, the potential for high computational efficiency.

Interpolation and extrapolation techniques, however, are certainly the most efficient strategies due to their low computational cost and are applicable where real-time results are needed. However, simplistic interpolation methods (bilinear, bicubic and alike) can cause streaking effects and are only capable of filling small holes on flat planes.

Inpainting-Based Approaches

The next category of depth completion approaches are heavily based on traditional inpainting techniques (normally used for colour images, Section 2.1). Although many inpainting-based methods yield promising results (at least more so than many filtering techniques), a majority of them are computationally expensive and can only be used in off-line applications of depth completion.

Figure 2.16: Example of the results of depth inpainting [41]. The approach is an improved fast marching-based [68] method guided by the colour image.

Structure-guided inpainting [71] is used in depth completion but diffusion, which is used to propagate the structures, results in blurring and hence the loss of detail and texture. The method proposed in [60] has been widely used in depth completion. It has been utilised in depth image-based rendering [73] and modified to recover missing data in depth estimates acquired via stereo correspondence [74]. The method in [68] is another popular approach but it does not perform well on depth images and cannot fill large missing regions plausibly.

The technique proposed in [133] attempts to recover the missing depth information using a fusion-based method integrated with a non-local filtering strategy. It is noted that the object boundaries and other stopping points that mark the termination of structure continuation process are not easy to locate in depth images which generally have little or no texture, or the boundaries or stopping points might be in the hole region of the depth image. Therefore, the colour image is used to assist with spotting the boundaries and their corresponding positions in the depth image are estimated according to calibration parameters. Their depth inpainting framework follows the work in [86] that takes advantage of a scheme similar to the non-local means scheme to make more accurate predictions for pixel values based on image textures. To solve the issue of structure propagation termination, a weight function is proposed in the inpainting framework that takes geometric distance, depth similarity and the structure information within the colour image into account.

On the other hand, [41] improves upon the fast marching method-based inpainting proposed by Telea [68] for depth completion. The colour image is essentially used to guide the depth inpainting process. By assuming that the adjacent pixels that have similar colour values have a higher probability of having similar depth values as well, an additional *colour term* is introduced into the weighting function to increase the contribution of the pixels with the same colour. The order of filling is also changed so that the pixels near edges and object boundaries are filled later, in order to produce sharper edges. However, even with all the improvements, this guided depth inpainting method is still not immune to noise and added artefacts around object boundaries (Figure 2.16); therefore, the guided filter proposed in [169] is used in

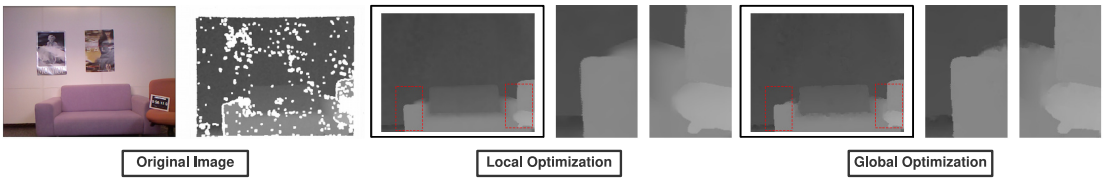

Figure 2.17: Local and global optimisation framework of [123]. The energy function is made up of a fidelity term (generated depth data characteristics) and a regularisation term (joint-bilateral and joint-trilateral kernels). Local filtering can be used instead of global filtering to make parallelization possible.

the post-processing stage to refine the depth image. An example of the results of this widely-acclaimed method with and without the final filtering stage is seen in Figure 2.16.

In [126], an exemplar-based inpainting method to avoid blurring while filling holes in novel views synthesised through depth image-based rendering is introduced. In the two separate stages of warped depth image completion and warped colour image completion, the focus is mainly on depth-assisted colour image completion with texture. The depth image is assumed to be only a gray-scale image with no texture and is therefore filled using any available background information (i.e. depth pixels are filled by being assigned the minimum of the neighbouring values). The assumptions that depth images have no texture, that texture and relief are not of any significant importance in depth images and depth holes can be plausibly filled using neighbouring background depth are obviously not true and lead to ignoring the utter importance of accurate 3D information in the state of the art. As a result, although the inpainting method proposed here to complete newly synthesised views based on depth is reasonable, the depth completion itself is lacking.

The work presented in [121] proposes a colour-assisted depth inpainting method that uses diffusion approaches with different rules for two separated components of a depth image: the edge regions and the smooth regions. It is noted in [121] that the depth edges shrinking or fattening is a common problem seen in the results of depth image inpainting methods. To combat the issue, the concept of a *fluctuating edge region* is introduced, which has an adaptive size and is used in the inpainting process. The big issue is that the mean of the depth values in the fluctuating edge region is used to determine the missing pixels near the boundaries, which does not result in a very accurate representation.

In [120] an anisotropic diffusion-based method is introduced that can have real-time capabilities by means of a GPU. The colour image is used to guide the diffusion in the depth image, which saves computation in the multi-scale pyramid scheme since the colour image does not change. In order to guarantee the alignment of the object boundaries in the colour image and the depth image, anisotropic diffusion is also applied to object boundaries (see results in Figure 2.9 - left).

Discussion: The existing literature supporting inpainting-based depth completion methods is more expansive as such approaches are mostly inspired by colour image completion techniques, which have a

longer history in image processing and computer vision. This class of depth completion approaches are capable of generating plausible outputs, yet not without their own flaws.

Many inpainting-based approaches utilise diffusion techniques and partial differential equations that inherently carry with themselves numerical instabilities and implementation issues. Moreover, efficiency is always a concern when depth completion is needed as pre-processing facet of other applications. Inpainting-based methods [41, 68] need in the order of seconds to process a single image. Although modern hardware and GPU can facilitate a faster performance with such methods, an independent cross-platform application is still more desirable in real-world scenarios.

Reconstruction-Based Methods

Although filtering and inpainting based depth completion techniques can produce reasonable and efficient results, there is a higher possibility of blurring, ringing and added artefacts especially around object boundaries, sharp discontinuities and highly-textured regions. In reconstruction-based methods, missing depth values are predicted using common synthesis approaches. Since a closed-loop strategy is mostly used to resolve the reconstruction coefficients in terms of the minimisation of residuals, higher levels of accuracy can be accomplished in depth completion. There are numerous different models found in the literature that are used to represent the depth completion problem.

In [123, 147] depth completion, specifically depth generated by consumer depth sensors such as Microsoft Kinect, is formulated as an energy minimisation problem, the function of which is made up of a fidelity term that considers the characteristics of consumer device generated depth data and a regularisation term that incorporates the joint-bilateral kernel and the joint-trilateral kernel. The joint-bilateral filter is tuned to incorporate the structure information and the joint-trilateral kernel is adapted to the noise model of consumer device generated depth data. Since the approach is relatively computationally-expensive, local filtering is used to approximate the global optimisation framework in order to make parallelization

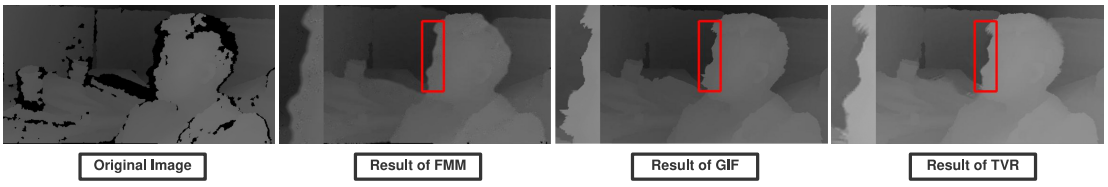

Figure 2.18: Example of the results of depth completion using energy minimisation with TV regularisation (TVR) [122] compared to fast marching method based inpainting (FMM) [68] and guided inpainting and filtering (GIF) [41]. The energy function assumes that in small local neighbourhoods, depth and colour values are linearly correlated.

Figure 2.19: Example of the results of an adaptive colour-guided auto-regressive (CAR) model for depth recovery [148] compared to joint bilateral up-sampling approach of [164].

Figure 2.20: Result of trilateral sparse representation (TSR) based approach in [149] compared to the spatio-temporal completion (STC) method of [45], guided inpainting and filtering (GIF) [41] and the colour-guided auto-regressive (CAR) model in [148].

possible, which brings forth the question of accuracy versus efficiency. A comparison between examples of the results generated through both local and global frameworks is seen in Figure 2.17.

The work in [122] proposes an approach mainly inspired by the work of [179] in image matting. An energy function is designed based on the assumption that in small local neighbourhoods, there is a linear correlation between depth and colour values. In order to remove noise and create sharper object boundaries and edges, a regularisation term originally proposed in [124] is added to the energy function. This added term makes the gradient of the depth image be both horizontally and vertically sparse, which results in less noise and sharper edges. A comparison between the results of this method and inpainting methods in [68] and [41], considered to be very powerful within the literature, is shown in Figure 2.18.

In [148], an adaptive colour-guided auto-regressive model is suggested for depth image recovery. Upon verifying the idea that the auto-regressive (AR) model fits depth images of generic scenes, the problem is formulated as a minimisation of AR prediction errors subject to measurement consistency. Both the local correlation in the original depth image and the non-local similarity in the colour image play a role in creating the AR predictor for each pixel. In order to accomplish more accuracy, a parameter adaptation strategy was designed to increase stability. An example of the results is seen in Figure 2.19.

The authors of [149] build upon their previous work [150] that used a locality regularised representation guided by the colour image to determine the weights from juxtaposed patches to increase the contribution of the most relevant pixels. However, to mend the shortcomings of their previous method, which ignores the effects of geometric distance and position and only concentrates on the impact of locality on coefficient learning, the use of a trilateral constrained sparse representation (SR) is suggested which takes intensity similarity and spatial distance between reference patches and the target on sparsity penalty term and position constraint of central pixel in the target patch on data-fidelity term into account. It should be noted that SR models have been successfully used in stereo vision applications [180–182] for depth estimation, noise removal and reconstruction. However, in depth completion, where the values in the target region are unavailable, reconstruction coefficient learning has to be performed via the accompanying colour image. Figure 2.20 contains a comparison between [149] and the commonly-used depth completion methods of [41, 45, 148].

Discussion: Reconstruction-based methods may be of high complexity, difficult to implement and somewhat computationally expensive but as seen with the aforementioned approaches, they generate more desirable results, without too much blurring or added artefacts. The object boundaries are also estimated more accurately than most other approaches.

Temporal-based Depth Completion

In this section, we discuss a group of approaches that take advantage of the motion and temporal information contained within a sequence of depth images and perhaps additionally the accompanying colour images to fill holes and refine the depth [44, 151].

One of the most commonly-used techniques in the literature is the method proposed in [44] that uses motion information and the difference between the depth values in the current image and those in the consecutive frames to fill the holes by giving the pixels the weighted average values of the corresponding pixels in other frames. Although the results are mostly plausible, one drawback is that the value of the edges of objects cannot be accurately estimated to an acceptable level (Figure 2.12 (left) and Figure 2.17), other than the fact that there is a need for a sequence of depth images and therefore, the holes in a single depth image cannot be filled. Moreover, this is designed to be an off-line approach and cannot be utilised in real-time applications. Also, when the colour information does not correspond with the depth data, the results often contain invalid depth values.

The KinectFusion approach proposed in [152] takes advantage of the depth images from neighbouring frames to complete the missing information during real-time 3D reconstruction. However, camera motion and a static scene are of utmost importance and although the approach is robust, it cannot be utilised for a static view of a scene without any camera motion.

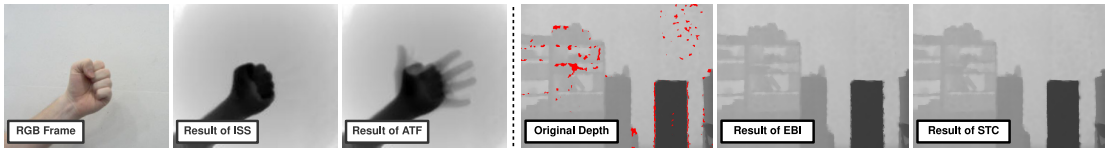

Figure 2.21: **Left:** example of the results of completion based on intrinsic static structure (ISS) [154] compared to adaptive temporal filtering (ATF) [153]. **Right:** example of the results of spatio-temporal completion [45] compared to exemplar-based inpainting (EBI) [60] on still frames.

In [151], holes are grouped into one of two categories: the ones created as a result of occlusion by foreground objects which are assumed to be in motion and the holes created by reflective surfaces and other random factors. Subsequently, the deepest neighbouring values are used to fill pixels according to the groups they are placed in. Even though such assumptions might be true in many real-life scenarios, they are not universal and static objects can be the cause of missing or invalid data in depth images captured via many consumer depth sensors.

The approach in [153] focuses on repairing the inconsistencies and inhomogeneities in depth videos. Depth values of certain objects in one frame sometimes vary from the values of the same objects in a neighbouring frame, while the planar existence of the object has not changed. An adaptive temporal filtering is proposed based on the correspondence between depth and colour sequences. [154] notes that the challenge in detecting and mending temporal inconsistencies in depth videos is due to the dynamic content and outliers. Consequently, the authors propose using the intrinsic static structure, which is initialised by taking the first frame and refined as more frames are available. The depth values are then enhanced by combining the input depth and the intrinsic static structure, the weight of which depends on the probability of the input value belonging to the structure. As seen in Figure 2.21 (left), the approach [154] does not introduce artefacts into the results due to motion delay because temporal consistency is only enforced on static regions, as opposed to the method proposed in [153], which applies temporal filtering to all regions.

Discussion: Temporal-based methods generate reasonable results even when spatial-based approaches are unable to and are necessary when depth consistency and homogeneity is important in a depth sequence, which it often is. On the other hand, the dependency on other frames is a hindrance that causes delays or renders the method only applicable as an off-line approach. Moreover, there are many scenarios where a depth sequence is simply not available but a single depth image still needs to be completed.

Spatio-Temporal Depth Completion

The third group of depth completion approaches combine the elements of the spatial and temporal based methods and attempt to complete the depth using *spatio-temporal* information in depth images [45, 155].

In the method proposed in [155], the process of depth completion is attempted in two stages. First, a *deepest depth image* is generated by combining the spatio-temporal information in the depth image and the colour image and used to complete the missing regions. Subsequently, the filled depth image is enhanced based on the joint information of geometry and colour. To preserve local features of the depth image, filters that are adapted to the features of the colour images are utilised.

In another widely-used method [156], the technique uses an adaptive spatio-temporal approach to fill depth holes utilising bilateral and Kalman filters. The approach is made up of three blocks: an adaptive joint bilateral filter that combines the depth and colour information is used and then random fluctuations of pixel values are subsequently handled by applying an adaptive Kalman filter on each pixel. Finally, an interpolation system uses the stable values in the regions neighbouring the holes provided by the previous blocks and by means of a 2D Gaussian kernel, fills the missing depth values.

On the other hand, in [45], the depth holes are filled using a joint-bilateral filter applied to neighbouring pixels, the weights of which are determined based on visual data, depth information and a temporal consistency map that is created to track the reliability of the depth values near the hole regions. The resulting values are taken into account when filtering successive frames and iterative filtering can ensure increasing accuracy as new samples are acquired and filtered. As seen in Figure 2.21 (right), the results are superior to the ones produced by the exemplar-based inpainting technique proposed in [60].

The approach in [157] employs a joint bilateral filter taking both colour and depth information into account for spatial enhancement. For temporal enhancement, the method takes advantage of block matching applied to the previous and current frame in the colour video to detect stationary objects. Therefore by using block matching, the movement of objects can be predicted by estimating the similarity between the blocks, measured by mean absolute difference, from frame to frame. The method generates a sharper and clearer depth image, as seen in Figure 2.22. However, this method only accounts for the existence of motion and not the length of motion vectors. Therefore, the depth image is stabilised only for stationary objects.

Similarly, [158] uses the temporal sequence and motion to create a moving body detection strategy for occlusion filling. Background differentials and the original images are used to extract the moving bodies and then a 4-neighbour interpolation technique is utilised over the background areas before filling the body areas. The edge can be reasonably preserved but for an interpolation method, the approach is time-consuming.

The work in [132] discusses the improvements made to what can be obtained from a regular video camera alongside a time-of-flight camera. The main focus is on depth up-sampling, colour and depth alignment, etc. However, one of the issues addressed in the work is that of depth completion, which is performed via multi-scale completion technique following the works in [183] and [164]. The output undergoes

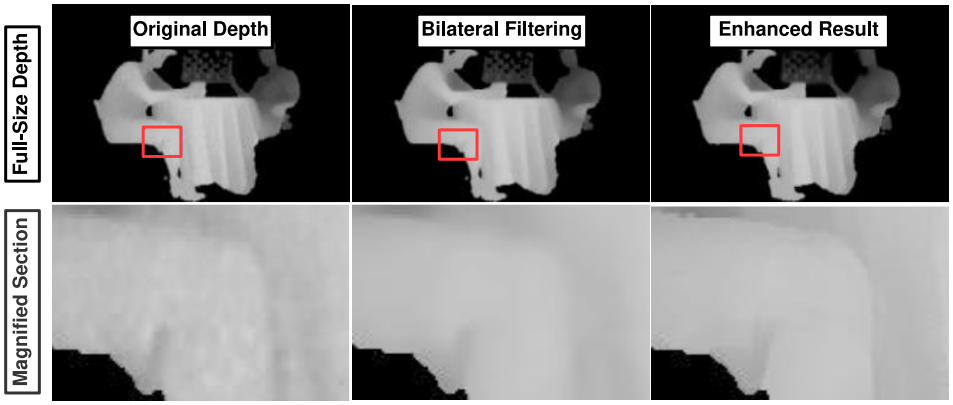

Figure 2.22: Example of the results of [157], in which block matching applied to previous and current colour frames provides temporal enhancement.

joint bilateral filtering and spatio-temporal processing to remove noise by averaging values from several consecutive frames.

The approach in [184] uses a sequence of frames to locate outliers with respect to depth consistency within the frame and utilises an improved and more efficient regression technique using least median of squares (LMedS) [185] to complete missing depth regions and replace outliers with valid depth values. The approach is capable of depth completion and refinement within a sequence of frames but can fail in the presence of invalid depth shared between frames and sudden changes in depth due to fast moving dynamic objects within the scene.

Discussion: Spatio-temporal methods certainly take advantage of the best elements of both spatial-based methods and temporal-based methods but they also inherit the negatives along with the positives. Temporal and motion information can play a part in helping with the blurring, jagging and mismatched object contours that are sometimes created by spatial-based methods. However, they also bring forth the issues of off-line applicability and delay in real-time generation of results.

2.2.3 Use of Secondary Guidance Image

Many modern 3D sensing technologies can provide the user with a depth map and a colour image of the same scene. While the completion process on its own is focused on the depth image, there is valuable information contained within the accompanying colour image that can significantly improve the quality of the results. There are approaches that take advantage of the object boundaries and edges of the colour image to preserve and align the structures in the depth [41, 145, 148]. Even so, it has been pointed out that this can still lead to undesirable artefacts around edges and object boundaries since colour and depth

Input Image Required	Advantages	Disadvantages	Examples of Filling Techniques
Depth and Colour Images	• more processing information • more accurate results	• possible lack of colour input • more computationally intensive	[37], [129], [128], [131], [173], [148], [139], [140] [41], [40], [133], [120], [122], [150], [157], [36]
Depth Image Only	• no dependence on extra inputs • more efficient processing	• less information for processing • lower quality outputs	[138], [141], [142], [171], [170], [143], [42], [144] [126], [44], [151], [152], [45], [132], [184], [68]

Table 2.2: Examples of depth completion approaches categorised according to the type of images required as their input.

edges are characteristically different [144]. Some other approaches have taken to using the colour image as a means to segment the scene before depth completion takes place [43, 129, 144], which can provide the completion process with semantically valid scene objects to sample homogeneous depth information from.

However, despite the advantages the colour information can offer, not all depth acquisition technologies produce an aligned or easily alignable colour image and requirements of the application may not always allow for the additional computation that comes with the colour image processing. In these situations, a depth completion approach that is fully dependent on the colour image as a secondary guidance image may not be desirable.

As such, we provide a simple overview of depth completion approaches by categorising them based on the their use of the colour image to provide guidance for the depth completion process. Table 2.2 presents the aforementioned split over the depth completion techniques commonly used within the literature. Moreover, Figure 2.11 provides a taxonomy of the literature based on the requirements of the approaches in terms of their dependence on a secondary input images and the information domain used for the completion process.

Discussion: Among depth completion approaches, some heavily rely on the view of the scene in colour to guide the depth completion process. While this can positively affect the outcome in terms of quality and consistency, certain limitations ensue. Aside from the colour image not being available at all times, computational requirements can create issues when the application demands light and real-time processing. As seen in Table 2.2 and Figure 2.11, a variety of approaches operate in both spaces, providing a wide range of opportunities to select the appropriate depth completion techniques.

2.2.4 Texture, Boundaries and Smoothing

Four simple rules were proposed in [58] to provide a set of guidelines for generating more plausible and realistic results when attempting to solve the problem of colour image completion (Section 2.1). While not all of these rules apply to depth images (depth images obviously do not contain any colour

Main Focus of Completion Approach	Examples of Completion Techniques
Relief and Object Boundary Preservation	[37], [173], [36], [41], [122], [132], [131], [140], [145], [141], [157]
Accurate Structure and Smooth Surfaces	[126], [74], [73], [45], [138], [139], [68], [133], [143], [125], [146]

Table 2.3: Examples of depth completion approaches categorised according to the main focus of the approach (structure vs. texture and accurate boundaries).

information), preserving texture, relief and clear object boundaries or smoothing can be important factors in a depth completion approach depending on the circumstances under which it needs to operate.

In certain downstream applications, fine-grained texture and relief over surfaces and a clear separation between objects within the depth image is of utmost importance [140], whereas smooth and consistent scene depth [125, 142] can satisfy the requirements of other systems.

It is important to note that preserving fine relief within the depth information of a scene object is a difficult task. Additionally, depth completion is an inherently ill-posed problem. As a result, if texture and relief generation is unnecessarily carried out based on insufficient information, the resulting output can contain more outliers and invalid depth information, which is a hindrance on its own.

Table 2.3 presents a list of depth completion approaches categorised according to their main objectives. Some techniques concentrate on providing very accurate texture and object boundaries, while others generate overly smooth depth in the output with their focus on the structural integrity of the scene depth.

Discussion: The exact characteristics of a depth image depend on its purpose. In certain applications such as object recognition [186, 187] or detection [188], accurate boundaries and relief of an object in the depth image can play an important role in the semantic value of that object within the scene. However, other applications such as localisation and mapping [118, 119] do not require fine texture and relief for each individual scene object accurate structure within the scene depth is sufficient. As seen in Table 2.3, different depth completion techniques exist that can generate complete depth either with fine relief or smoothed object surfaces.

2.3 Monocular Depth Estimation

Over the past few years, research into monocular depth estimation, i.e. predicting complete scene depth from a single RGB image, has significantly escalated [46, 47, 189–192]. Using off-line model training based on ground truth depth data, monocular depth prediction has been made possible [46, 189, 190, 193, 194] sometimes with results surpassing those of more classical depth estimation techniques. Ground truth depth, however, is extremely difficult and expensive to acquire and when it is obtained it is often

sparse and flawed, constraining the practical use of monocular depth estimation in real-world applications. Solutions to this problem of data scarcity include the possibility of using synthetic data containing sharp pixel-perfect scene depth [195] for training or completely dispensing with using ground truth depth and instead utilising a secondary supervisory signal during training which indirectly results in producing the desired depth [47, 191, 192, 196].
As a portion of the work done for this thesis is within this area (Chapter 6), in the following, a brief description of monocular depth estimation techniques within three relevant areas is presented: approaches utilising hand-crafted features based on monocular cues within the RGB input image, approaches based on graphical models and finally techniques using deep neural networks trained in various ways to estimate depth from a single image.

2.3.1 Hand-Crafted Features

While binocular vision is commonly associated with depth perception in humans and machines, estimating depth from a single image based on monocular cues and features is technically possible for both humans and machines, even if the results are not very accurate. Such monocular cues include size considering visual angles, grain and motion parallax. Monocular depth estimation techniques have utilised such features to estimate depth from a single RGB image.
Based on the assumption that the geometric information contained within a scene combined with motion extracted from a sequence can be valuable features for 3D reconstruction, [197] estimates depth based on temporal continuity and geometric perspective. In [198], different cues such as motion, colour and contrast are combined to extract the foreground layer, which is then used to estimate depth. Motion parameters and optical flow are calculated using structure from motion.
In [199, 200], an assumption of ground-vertical geometric structure is used as the basis to construct a basic 3D model from a single image. This is accomplished by labelling the image according to pre-defined geometric classes and subsequently creating a statistical model based on scene orientation. [201] proposes a non-parametric approach based on SIFT Flow, where scene depth is reconstructed from an input RGB image by transferring the depth of multiple similar images and then applying warping and optimising procedures. The work in [202] investigates using semantic scene segmentation results to guide the depth reconstruction process instead of directly predicting depth based on features present in the scene. The work in [189] also takes advantage of combining semantic object labels with depth features to aid in the depth estimation process.
It is important to note that predicting depth based on monocular cues within the scene is not robust enough to deal with complex and cluttered scenes even though approaches using such features have managed to produce promising results when it comes to scenes that contain clear pre-defined features and adhere to simple structural assumptions.

2.3.2 Graphical Models

Within the current literature on monocular depth estimation, there are approaches that take advantage of graphical models to recover scene depth. For instance, [203] introduces a dynamic Bayesian network capable of reconstructing a 3D scene from a monocular image based on the assumption that all scenes contain a *floor-wall* geometry. The model distinguishes said floor-wall boundaries in each column of the image and using the perspective geometry reconstructs a 3D representation of the scene. While the approach produces very promising results, the underlying assumption it is built on (indoor scenes framed by a floor-wall constraint) limits the capabilities of the approach.

The work in [204] utilises a discriminatively-trained Markov Random Field (MRF) and linear regression to estimate depth. The images are segmented into homogeneous regions and the produced patches are used as super-pixels instead of pixels during the depth estimation process. This extended version of the approach [205] utilises the MRF in order to combine planes predicted by the linear model to describe the 3D position and orientation of segmented patches within RGB images. Since depth is predicted locally, the combined output lacks global coherence. Additionally, the model is manually tuned which is a detriment against achieving a learning-based system.

The method proposed in [206] presents cascaded classification models. The approach combines the tasks of scene categorisation, object detection, multi-class image segmentation and, most relevant here, 3D reconstruction by coupling repeated instantiations of the sophisticated off-the-shelf classifiers in order to improve the overall performance at each level.

In [207], monocular depth estimation is formulated as an inference problem in a discrete/continuous Conditional Random Field (CRF) model, in which continuous variables encode the depth information associated with super-pixels from the input RGB image and the discrete ones represent the relationships between the neighbouring super-pixels. Using input images with available ground truth depth, the unary potentials are calculated within a graphical model, in which the discrete/continuous optimisation problem is solved with the aid of particle belief propagation [208, 209].

To better exploit the global structure of the scene, [194] proposes a hierarchical representation of the scene based on a CRF, which is capable of modelling local depth information along with mid-level and global scene structures. Not unlike [207], the model attempts to solve monocular depth estimation as an inference problem in a graphical model in which the edges provide an encoding of the interactions within and across the different layers of the proposed scene hierarchy.

More recently, [210] attempts to perform monocular depth estimation using sparse manual labels for object sizes within a given scene. Utilising these manually estimated object sizes and the geometric relationship between them, a coarse depth image is primarily created. This depth output is subsequently

refined using a CRF that propagates the estimated depth values to generate the final depth image for the scene.

Monocular depth estimation techniques based on graphical models can produce impressive results but despite their excellent generalisation capabilities, deep neural networks generate sharper and more accurate depth images, even though they can be prone to over-fitting and require larger quantities of training data.

2.3.3 Deep Neural Networks

Recent monocular depth estimation techniques using deep convolutional neural networks *directly supervised* using data with ground truth depth images have revolutionised the field by producing highly accurate results. For instance, the approach in [193] utilises a multi-scale network that estimates a coarse global depth image and a second network that locally refines the depth image produced by the first network. The approach is extended in [46] to perform semantic segmentation and surface normal estimation as well as depth prediction.

In the work proposed in [211], a fully-convolutional network is trained to estimate more accurate depth based on efficient feature up-sampling within the network architecture. In the up-sampling procedure, the outputs of four convolutional layers are fused by applying successive up-sampling operations. On the other hand, [212] points to the past successes that CRF-based methods have achieved in monocular depth estimation and presents a deep convolutional neural field model that takes advantage of the capabilities of a continuous CRF. The unary and pairwise potentials of the continuous CRF are learned in a deep network resulting in depth estimation for general scenes with no geometric priors.

The work in [213] trains a supervised model for estimation formulated as a pixel-wise classification task. This reformulation of the problem is made possible by transforming the continuous values in the ground truth depth images into class labels by discretising the values into bins and labelling the bins based on their depth ranges. Solving depth estimation as a classification problem provides the possibility to obtain confidence values for predicted depth in the form of probability distributions. Using the obtained confidence values, an information gain loss is applied that enables selecting predictions that are close to ground-truth values during training.

Similarly, [214] also presents monocular depth estimation as a pixel-wise classification problem. Different side-outputs from the dilated convolutional neural network architecture are fused hierarchically to take advantage of multi-scale depth cues. Finally, soft-weighted-sum inference is used instead of the hard-max inference, which transforms the discretised depth scores to continuous depth values. [215] attempts to solve the commonly-found issue of blurring effects in the results of most monocular depth estimation techniques by fusing features extracted at different scales from a network architecture that

includes a multi-scale feature fusion module and a refinement module trained via an objective function that measures errors in depth, gradients and surface normals.

While these approaches produce consistently more encouraging results than their predecessors, the main draw-back of any directly supervised depth estimation model is its dependence on large quantities of dense ground truth depth images for training. To combat this issue, synthetic depth images have recently received attention in the literature. [195] proposes a framework in which a network takes as its input both synthetic and real-world images and produces modified images which are then passed through a second network trained to perform monocular depth estimation.

While the use of synthetic training data can be a helpful solution to the issue of scarcity of ground truth depth, a new class of *indirectly supervised* monocular depth estimators have emerged that do not require ground truth depth and calculate disparity by reconstructing the corresponding view within a stereo correspondence framework and thus use this view reconstruction as a secondary supervisory signal. For instance, the work in [192] proposes the Deep3D network, which learns to generate the right view from the left image used as the input and in the process, produces an intermediary disparity image. The model is trained on stereo pairs from a dataset of 3D movies to minimise the pixel-wise reconstruction loss of the generated right view compared to the ground truth right view. The desired output is a probabilistic disparity map that is used by a differentiable depth image-based rendering layer in the network architecture. While the results of the approach are very promising, the method is very memory intensive.

The approach in [191] follows a similar framework with a model very similar to an auto-encoder, in which the encoder is trained to estimate depth for the input image (left) by explicitly creating an inverse warp of the output image (right) in the decoder using the estimated depth and the known inter-view displacement, to reconstruct the input image. The technique uses an objective function similar to [192] but is not fully differentiable.

On the other hand, [47] argues that a simple image reconstruction as done in [191, 192] does not produce depth with high enough quality and uses bilinear sampling [218] and a left/right consistency check between the disparities produced relative to both the left and right images incorporated into training to produce better results. Examples of the results of this approach can be seen in Figure 2.23 (LRC). Even though the results are consistently impressive across images from different datasets, pointing to the generalisation capabilities of the approach, blurring effects within the resulting depth images still persist.

In [48], the use of sequences of stereo image pairs is investigated for estimating depth and visual odometry. It is argued that utilising stereo sequences as training data makes the model capable of considering both spatial (between left/right views) and temporal (forward/backward) warp error in its learning process and can constrain scene depth and camera motion to remain within a reasonable scale.

Figure 2.23: Qualitative comparison of depth and ego-motion from video (DEV) [216], estimation based on left/right consistency (LRC) [47] and semi-supervised depth estimation (SSE) [217].

While the approaches that benefit from view synthesis through learning the inter-view displacement and thus the disparity are capable of producing very accurate and consistent results and the required training data is abundant and easily obtainable, there are certain shortcomings. Firstly, the training data must consist of temporally aligned and rectified stereo images and more importantly, in the presence of occluded regions (i.e. groups of pixels that are seen in one image but not the other), disparity calculations fail and meaningless values are generated.

On the other hand, the work in [216] estimates depth and camera motion from video by training depth and pose prediction networks, indirectly supervised via view synthesis. The results are favourable especially since they include ego-motion but the depth outputs are very blurry (as seen in Figure 2.23 - DEV), do not consider occlusions and are dependent on camera parameters. The training in the work of [217] is supervised by sparse ground truth depth and the model is then enforced within a stereo framework via an image alignment loss to output dense depth. This enables the model to take advantage of both direct and indirect training, leading to higher fidelity depth outputs than most other comparators, as demonstrated in Figure 2.23 (SSE).

Within the literature, there are specific metrics that are commonly used to evaluate the performance of monocular depth estimation techniques. Given an estimated depth image d'_p and the corresponding ground truth depth d_p at pixel p with N being the total number of pixels for which valid ground truth and estimated depth exist, the following metrics are often used for performance evaluation in the literature:

- Absolute Relative Error (*Abs. Rel.*) [205]:

$$\frac{1}{N}\sum_{p}\frac{|d_p - d'_p|}{d_p} \tag{2.2}$$

- Squared Relative Error (*Sq. Rel.*) [205]:

$$\frac{1}{N}\sum_{p}\frac{||d_p - d'_p||^2}{d_p} \tag{2.3}$$

- Linear Root Mean Square Error (*RMSE*) [206]:

$$\sqrt{\frac{1}{N}\sum_{p}||d_p - d'_p||^2} \tag{2.4}$$

- Log Scale Invariant RMSE (*RMSE log*) [193]:

$$\sqrt{\frac{1}{N}\sum_{p}||log(d_p) - log(d'_p)||^2} \tag{2.5}$$

- Accuracy under a threshold [189]:

$$max(\frac{d'_p}{d_p}, \frac{d_p}{d'_p}) = \delta < threshold \tag{2.6}$$

Within this thesis (Chapter 6), we will utilise the same metrics to evaluate our monocular depth estimation results.

2.4 Domain Adaptation

In Chapters 5 and 6 of this work, we propose learning-based approaches trained on *synthetically generated* datasets of corresponding RGB and depth images to learn the context and content of the scene. However, due to *dataset bias* [219], a typical model trained on a specific set of data does not necessarily generalise well to other datasets.

In other words, a model trained on *synthetic* data may not perform well on *real-world* data. Therefore, while our depth estimation model may successfully predict the depth for synthetic data, it will not be able to do the same for naturally obtained images, which would make the model utterly useless from a

practical visual sensing perspective. This best epitomises the problem of domain shift, which has received significant attention within the research community and has begotten the expansive literature on domain adaptation.

While a typical solution to the problem of data domain shift is to fine-tune the network on data from the target domain, fitting the large number of parameters in a deep network to a new dataset requires a large amount of data, which can be very time-consuming, expensive, or even practically intractable to collate giving rise to the use of the source data domain in the first place. Given that the objective is to employ a model trained on the source dataset to successfully perform on a target dataset, one strategy is to minimise the distance between the source and target feature distributions [220–226].

Some approaches have taken advantage of Maximum Mean Discrepancy (MMD) which calculates the norm of the distance between the domains to reduce the discrepancy [220, 227, 228], while others have taken to using an adversarial loss which leads to a representation that minimises the domain discrepancy while able to discriminate the source labels easily [221–223, 226]. Although most of the these techniques focus on discriminative models, research on generative tasks has also utilised domain adaptation [224, 229].

Recently [230] proposed that matching the Gram matrices [231] of feature maps, often performed within neural style transfer of images, is theoretically equivalent to minimising the maximum mean discrepancy with the second order polynomial kernel. In Section 2.5, we briefly review neural style transfer and its relevance to our work.

2.5 Image Style Transfer

Image style transfer by means of convolutional neural networks has recently received significant attention within the research community via [97]. Since then, numerous improved and novel approaches have been proposed that can transfer the style of one image onto another (e.g. [98, 230, 232, 233]).

Within the existing literature, style is often represented as a set of Gram matrices [234] that describe the correlations between low-level convolutional features, while content is represented by the raw values of high-level semantic features. Style transfer approaches conventionally extract these representations from a pre-trained *loss network* and use them to quantify style and content losses with respect to the target style and content images. These losses are subsequently combined into a joint objective function. More formally, the content and style losses can be defined as:

$$\mathcal{L}_c = \sum_{i \in \mathcal{C}} \frac{1}{n_i} ||f_i(x) - f_i(c)||_F^2, \tag{2.7}$$

$$\mathcal{L}_s = \sum_{i \in \mathcal{S}} \frac{1}{n_i} ||\mathcal{G}[f_i(x)] - \mathcal{G}[f_i(s)]||_F^2, \tag{2.8}$$

where c, s and x are the content, style and restyled images respectively, f is the *loss network*, $f_i(x)$ is the activation tensor of layer i after passing x through f, n_i is the number of units in layer i, $\mathcal{C}$ and $\mathcal{S}$ are sets containing the indices of the content and style layers, $\mathcal{G}[f_i(x)]$ denotes the Gram matrix of layer i activations of f and $||\cdot||_F$ denotes the Frobenius norm. The overall objective can then be expressed as:

$$\min_x \mathcal{L}_c(x, c) + \lambda \mathcal{L}_s(x, s), \tag{2.9}$$

where λ is a coefficient that determines the relative weights of style and content loss. In the original work in [97], this objective was minimised directly by gradient descent in image space, and although the results of [97] are impressive, its process is very computationally intensive, leading to the emergence of alternative approaches that use neural networks to approximate the global minimum of the objective in a single forward pass. Such approaches [98, 232, 233] utilise neural networks trained to restyle an input image while preserving its content. Although much faster, these approaches are only capable of applying a single style and must be re-trained if a different style is required.

Based on the assumption that there are overlapping characteristics between different styles, [235] builds on the work in [236] and trains one network to apply up to 32 styles using conditional instance normalisation, which sets the mean and standard deviation of each intermediate feature map to different learned values for each style. The approach in [99] generalises this to a fully arbitrary style transfer approach, using a separate network [237] to predict the re-normalisation parameters from the style image. [238] matches the mean and variance statistics of a convolutional encoding of the content image with those of the style image and then decodes them into a restyled image, while [239] concatenates a learned style embedding onto an early convolutional layer in a *style transformer network* similar to [98].

As demonstrated in [230], style transfer can be considered as a distribution alignment process from the content image to the style image [240]. In other words, transferring the style of one image (from the source domain) to another image (from the target domain) is essentially the same as minimising the distance between the source and target distributions (Section 2.4). In Section 6.1, we take advantage of this idea to adapt our data distribution (i.e. real-world images) to our depth estimation model trained on data from a different distribution (i.e. synthetic images).

2.6 Learning from Temporal Information

The temporal information contained within a sequence of frames can be as useful to some spatially solvable computer vision tasks such as image classification and segmentation as it is indispensable to

others, including action recognition, pose estimation, video generation and alike.

Within action recognition [241], temporal learning has been attempted by using a stack of consecutive frames [242] and more successfully, optical flow vectors between consecutive frames as the network input [243, 244]. Feature extraction from a Long Short-Term Memory (LSTM) network [245] and 3D convolutions on entire video volumes [246, 247] have also lead to promising results.

Similarly, pose estimation techniques [248, 249] have also used optical flow between consecutive frames to predict positions in the current frame. In [250, 251], flow vectors are employed to warp predictions backward and forward to align them to the current frame, thus strengthening confidence in the current output. Video prediction is another long-established task [252, 253] heavily dependent on temporal information. Recent work on video prediction includes methods such as decomposing a sequence into motion and content components [254, 255] normally achieved using a convolutional LSTM [256] to decode motion and extracting content from a single frame [255]. Convolutional LSTM have also been employed to generate videos based on unseen captions [257], perform sequential object segmentation [258] and tracking [259], among others [260].

While there are many other active areas of research where temporal continuity is highly advantageous [261–264], in this work, we primarily focus on depth estimation, depth completion and semantic segmentation.

2.7 Optical Flow

Despite the remarkable achievements made by more conventional optical flow approaches [265–268], modern learning-based techniques capable of running on GPU hardware are a noteworthy stride towards better performance. FlowNet [269] initiated the trend of optical flow estimation using convolutional neural networks by exploring end-to-end training of two different architectures. FlowNet 2.0 [270] improves on its predecessor by introducing a network focusing on small movements, a novel learning schedule and a more complex architecture.

In Section 6.2 of this work, we use the technique proposed in [271], which is a light-weight approach that utilises a coarse-to-fine spatial pyramid to learn residual flow at each scale.

2.8 Semantic Segmentation

As one of the fundamental aspects of scene understanding, semantic segmentation has rightly received significant attention in recent years. Whether segmentation is performed on an image region basis [272], on a pixel level [273–275] or both [276], large quantities of annotated training data is required to perform

the task. Within the existing literature, fully-convolutional networks [273], saved pooling indices [277], skip connections [275], multi-path refinement networks [278], spatial pyramid pooling [279], attention modules focusing on scale or channel features [280, 281] and others have been used to carry out accurate pixel level segmentation. To combat the issue of large data requirements, researchers have also taken to using synthetic data [282, 283] and weakly-supervised approaches [276, 284, 285].

The temporal information in videos has also been used to improve either the accuracy or the efficiency of the segmentation process. [286] proposes a spatio-temporal LSTM based on frame-wise features to provide higher accuracy. Semantic labels are propagated in [287] using gated recurrent units. In [288], features from preceding frames are warped using flow fields to reinforce the current frame features.

On the other hand, [289] reuses previous frame features in certain layers in a multi-stage framework to reduce computation and improve efficiency. In [290], an optical flow network [269] is used to propagate features from key frames to the current one. Similarly, [291] uses an adaptive key frame scheduling policy instead of the fixed one in [290] to improve both accuracy and efficiency. Additionally, [292] proposes an adaptive feature propagation module that employs spatially variant convolutions to fuse the frame features, thus further reducing computation.

2.9 Relevance to Contributions

Based on the overview of the current literature on the variety of subjects presented in this chapter, we will outline the novel contributions of this thesis in the following chapters.

The existing body of work on exemplar-based depth completion is covered in Section 2.2.1, noting that such approaches do not perform as well on depth images as they do with colour information. Building on prior work, in Chapter 3, we present two novel exemplar-based completion approaches specifically designed to deal with the characteristics of depth images, leading to more coherent and accurate depth outputs post completion.

Section 2.2.2 of this chapter presents the prior work done on filtering, interpolation and extrapolation based techniques used to perform depth completion. Such techniques can be highly efficient, making them suitable for real-time applications but suffer from numerous issues and undesirable artefacts. In Chapter 4, we propose a very efficient depth completion approach based on prior semantic segmentation (Section 2.8) capable of completing depth images with minimal artefacts in the final output.

Existing work on learning-based depth completion is very limited (Section 2.2.1). However, such approaches are very suitable for completing depth images with very large missing sections, where the spatial information contained within the known regions of the image might be insufficient to carry out plausible completion. we propose a learning-based approach in Chapter 5 that is capable of generating content

for the missing portions of a depth image based on the context and content of the scene learned from a large corpus of synthetic training data. Using adversarial domain adaptation (Section 2.4), the proposed approach is capable of performing well on real-world test data despite only being trained on synthetic images.

As mentioned in Section 2.3, monocular depth estimation can be an effective alternative to more conventional depth sensing approaches, hence potentially removing the need for any post processing including depth completion. In this vein, Chapter 6 offers two novel monocular depth estimation techniques trained on synthetic data. The first utilises image style transfer (Section 2.5) as a domain adaptation approach (Section 2.4), which leads to the superior performance of the proposed method in the presence of previously-unseen real-world data compared to contemporary techniques trained on such data. The second approach presented in Chapter 6 attempts to enforce temporal consistency (Section 2.6) in a multi-task learning framework capable of performing monocular depth estimation (Section 2.3) and semantic segmentation (Section 2.8) at the same time.

Chapter 3

Exemplar-based RGB-D Completion

Exemplar-based image completion (Section 2.2.1) can be very effective in synthesising texture, relief and surface detail within the context of colour image inpainting. However, such an approach can be ineffective within the depth modality as there may not be enough useful information in the known regions of depth to sample from, which is indeed a very common problem if the image is not of a fronto-parallel view.

This issue can be solved for RGB images by including the transformed version of the patches by varying rotation, scale, shear, aspect ratio, keystone corrections, gain and bias colour adjustments and other photometric transformations in the search space when trying to find similar patches to sample from. However, with depth images, there may be scenes in which no suitable patch can be found to fill certain hole regions even if the search space contains all possible transformed patches from the input.

Depth images captured using current 3D sensing technology are not *ideal*, however, and in practice, they can potentially be completed using exemplar-based approaches. In this chapter, we take advantage of the main strength of exemplar-based techniques - their ability to synthesise texture and relief within hole regions while at the same time maintaining structural coherence within the image to perform hole filling and object removal in RGB-D images.

Consequently, this chapter outlines two approaches contributing to RGB-D image completion. Section 3.1 contains an RGB-D completion technique that decomposes the depth into low and high frequency components using the Fourier transform and subsequently completes the low and high frequency components using structural inpainting and a pixel-level exemplar-based texture synthesis method, respectively. Similarly, Section 3.2 presents a constrained exemplar-based completion technique designed specifically to best handle depth images in terms of object boundaries and surface relief.

The material presented in this chapter of the thesis has been published in the following peer-reviewed publications:

Figure 3.1: Comparing the results of the texture synthesis approach (TSM) in [53] and the results of the approach proposed in Section 3.1 with query expansion.

- **A. Atapour-Abarghouei**, G. P. de La Garanderie and T. P. Breckon, 'Back to Butterworth - a Fourier Basis for 3D Surface Relief Hole Filling within RGB-D Imagery.' In Proc. International Conference on Pattern Recognition, 2016 [37].

- **A. Atapour-Abarghouei** and T. P. Breckon. 'Extended Patch Prioritization for Depth Filling within Constrained Exemplar-based RGB-D Image Completion.' In Proc. International Conference on Image Analysis and Recognition, 2018 [35].

3.1 A Fourier Basis for RGB-D Image Completion

In this section, we present an approach designed to complete an RGB-D image after a selected object or region is removed from the image. This is accomplished by combining texture synthesis (which mostly deals with simplistic texture [53–55]) and image inpainting (which works best with underlying image structure [57, 58]). We can similarly apply such an approach to generalised depth image completion for the task of depth completion for images with missing depth regions due to sensing deficiencies (Section 1.1) or dynamic object removal.

Both texture synthesis and structural inpainting techniques have both been widely studied within the existing literature due to their real-life applications in various computer vision tasks such as occlusion filling, object removal, transmission loss recovery, image restoration and alike. Several image inpainting

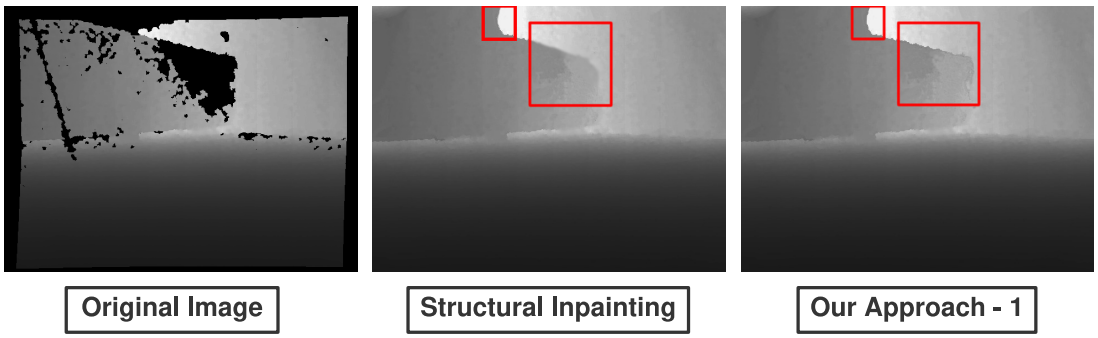

Figure 3.2: Qualitative analysis of the approach proposed in Section 3.1 when applied to depth images compared to the structural inpainting approach of [66].

methods attempt to fill holes by propagating isophotes [70, 117, 293] into the holes using diffusion. However, most of these can create blurring in the target region and the larger the hole, the more blurring is introduced.

Many texture synthesis methods follow the seminal approach proposed in [53], a non-parametric Markov chain synthesis algorithm that uses exhaustive nearest neighbour searching. Although this method has inspired dozens of other texture synthesis techniques, one of its major drawbacks is that it can expand undesirable local features and degenerate texture (i.e. grow 'garbage', as it is called in [53]) or generate verbatim samples of the original image (see example in Figure 3.1). This limits its general applicability within the context of large-region inpainting and the underlying scene structure preservation. Here, we propose using a technique inspired by the query expansion methods in image retrieval [294, 295] where image information from verified parts of the image are used to avoid this phenomenon in synthesising depth relief and fine texture detail.

With an increasing demand for complete 3D scene information [296], the approach presented here can be used not only to consistently remove a specified object from a depth image but also to fill the already existing holes, resulting from limitations in 3D scene capture [297].

The main idea of this proposed approach is to decompose the depth image into two separate images using the Fourier Transform. One contains the high frequency (HF) information or texture and the other consists of the low frequency (LF) information or general structure. The improved texture synthesis technique is then used to fill the high frequency image and the inpainting method in [66] is used to inpaint the low frequency image. The two images are then recombined in the Fourier space to generate the desired depth image.

As such, we apply the Butterworth filter [298] in the frequency domain to separate the high frequency information (texture and relief) from the low frequency information (continuous surface shape). After

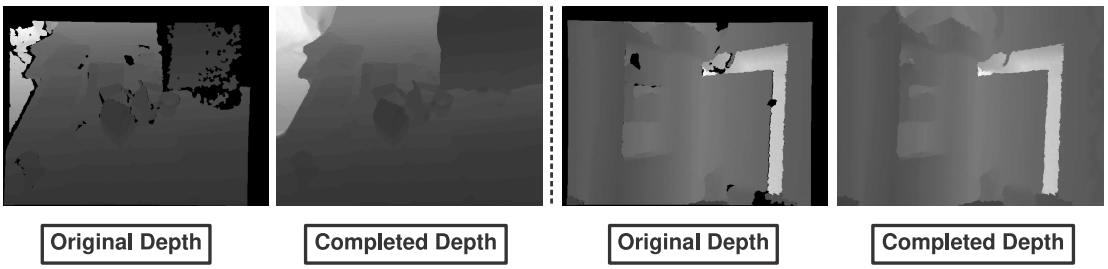

Figure 3.3: Examples of the approach proposed in Section 3.1 used to complete naturally-occurring holes in depth images.

filling the holes in the high frequency image via our improved texture synthesis method and inpainting the low frequency image using the method presented in [66], the two results are subsequently recombined in Fourier space to generate the final depth-filled output. The results demonstrate that the proposed method is effective even upon depth surfaces oblique to the camera and does not introduce any further artefacts into the depth output. The details of the proposed approach are outlined in the next section.

3.1.1 Details of the Approach

Our approach uses three key elements in order to complete a depth image: separation of depth relief detail from the underlying depth shape, the extension of texture synthesis [53] via the integration of standard query expansion for the completion of that relief detail and a structural inpainting technique used to fill the underlying continuous surfaces and structures.

Our approach, therefore, consists of four simple stages (Figure 3.4):

1. Decomposing the depth image into two images via the Fourier Transform - one with only high spatial frequency information (including edges and surface relief) and one with only low spatial frequency information such as continuous surfaces shape and underlying depth gradients.

2. Filling the holes in the high frequency (HF) image via an extended synthesis method in [53].

3. Using structural inpainting [66] to propagate the underlying shape structures in the low frequency (LF) image.

4. Final recombination of the low frequency shape information, obtained from structural inpainting, and the high frequency detail, obtained from the extended texture synthesis in the Fourier space.

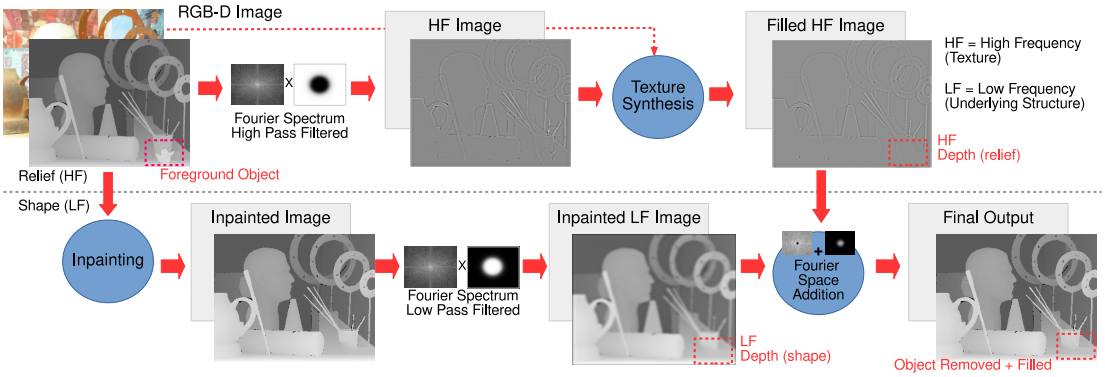

Figure 3.4: An overview of the Fourier-based approach proposed in Section 3.1 for the completion of scene depth via Butterworth low/high pass filtering.

Decomposing the Image for Completion

In order to disentangle the two components of the image from each other, we use the Discrete Fourier Transform [299] to convert the depth image into the frequency domain. By applying a simple high pass filter, the low frequency coefficients are then removed and only the high frequency features of the depth image are left. To avoid introducing any ringing artefacts into the image, the Butterworth high pass filter [298] is used, which is essentially a piecewise continuous circularly symmetric filter and a discrete approximation to the Gaussian as follows:

$$B(x,y) = \frac{1}{1 + (\frac{k}{\sqrt{x^2+y^2}})^{2n}}, \tag{3.1}$$

and the corresponding low pass Butterworth filter defined as follows:

$$B(x,y) = \frac{1}{1 + (\frac{\sqrt{x^2+y^2}}{k})^{2n}}, \tag{3.2}$$

where x and y are pixel positions in the Fourier image, k is the cut-off frequency and n is the order. In our experiments, $k = 5$ and $n = 1$. It is important to note that the higher the value of the cut-off frequency, k, the more image information is included in the high-frequency component and the smaller the value of k is, the more information is contained within the low-frequency component. However, as long as the same values are selected for k and n for both the high and low pass filters, as expected, the overall approach

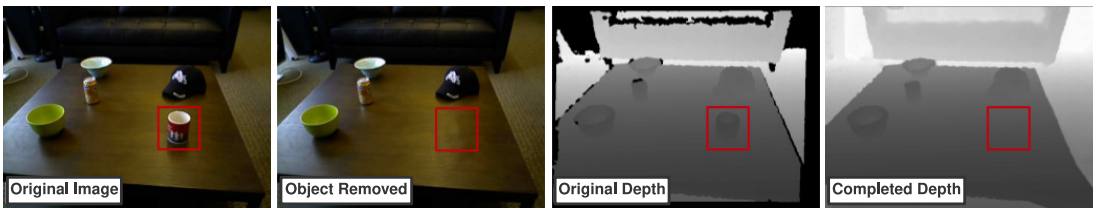

Figure 3.5: Object removed from an RGB-D image captured via a structured light depth sensor (Microsoft Kinect) using the approach proposed in Section 3.1.

is not very sensitive to the selected values as both high and low frequency components are completed separately.

Figure 3.4 (upper) illustrates how the original image is thus transformed into the frequency domain where the high-pass Butterworth filter is applied (Equation 3.1). The resulting high frequency depth content (relief) is then completed via our extended texture synthesis approach, whilst the low frequency surface shape is completed via structural inpainting akin to [66].

Although it is possible to structurally inpaint the low frequency components after the decomposition of the image, experiments demonstrate superior results if the low-pass filter is applied post structural inpainting (Figure 3.4 - lower).

In either case, both the synthesis and inpainting approaches could be equally replaced with an alternate constrained texture synthesis and another structural inpainting method that does not use fronto-parallel translational patches respectively.

After the two images have been filled, they are recombined in the Fourier domain to generate the final image with the missing depth regions plausibly filled. The stages of this process are illustrated in Figure 3.4 for the removal of a foreground object, from which the resulting depth hole is subsequently filled across both the high and low frequency spatial components of the scene.

Texture Synthesis via Query Expansion

In the seminal work in [53], texture is modelled as a Markov Random Field. To generate texture, the neighbourhood of a pixel is considered to be a square window around the pixel. For every unknown pixel to be filled, a $w \times w$ pixel neighbourhood, N_q, is created which is then used to query the known regions of the image to find all sample neighbourhoods, N_s, that are similar. The similarity of the two neighbourhoods is determined by calculating the Gaussian weighted sum of squared difference (SSD) between the two (Equations 3.3 and 3.4):

$$SSD(N_q, N_s) = \sum_{1 \leq i,j \leq w} d[p_{i,j}(N_q) - p_{i,j}(N_s)] \times G(i,j), \tag{3.3}$$

$$G(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{(i-i_0)^2+(j-j_0)^2}{2\sigma^2}}, \tag{3.4}$$

where $d[p_{i,j}(N_q) - p_{i,j}(N_s)]$ is the squared Euclidean difference between the corresponding pixels in neighbourhoods N_q (query neighbourhood) and N_s (sample neighbourhoods), w is the neighbourhood size, G is a two-dimensional Gaussian kernel, σ is the standard deviation and $i_0 = j_0 = \frac{w-1}{2}$ marks the center of the neighbourhood. As this approach is, in essence, a texture synthesis method, the size of the neighbourhood should be carefully selected so it is large enough to contain the smallest indivisible element of texture contained within the scene. While larger values can lead to more visually-plausible outcomes, this comes at a cost of efficiency. We select the neighbourhood size to be $w = 8$, which can lead to promising results such as those seen in Figure 3.1. For an image of size 1390×1110 with 10% of its pixels selected for removal and subsequent completion, the completion process takes 143 minutes with a neighbourhood size of $w = 8$ with same image taking 507 minutes for a neighbourhood size of $w = 16$.

It should be noted that the SSD can be calculated for a one-channel image (as in a single depth image, Figure 3.2 and Figure 3.3), a three-channel image (as in an RGB image, Figure 3.1), or a four-channel image (as in an RGB-D image, Figures 3.5 and 3.7).

All windows N_s that meet the following criterion are placed in a list, from which one is randomly selected to fill the pixel:

$$SSD(N_q, N_s) < (1 + \epsilon) d_{min}, \tag{3.5}$$

where $\epsilon = 0.1$ and d_{min} is the minimum of the SSD between N_s and N_q.

It is here where this approach can fall into a narrow subspace of the overall query space and produce substandard results. To reduce this likelihood, we propose using a technique from the object retrieval literature [294] to improve comparative query results. The solution is to simply expand the original query, issuing new queries and hence avoiding a narrowing of the search space towards degenerate sampling of the query space.

In our proposed approach, after generating the set of neighbourhoods that satisfy the condition in Equation 3.5, instead of randomly choosing one instantaneously, new queries are performed based on each of the neighbourhoods already in the list. The results of the search are subsequently added to the list once again.

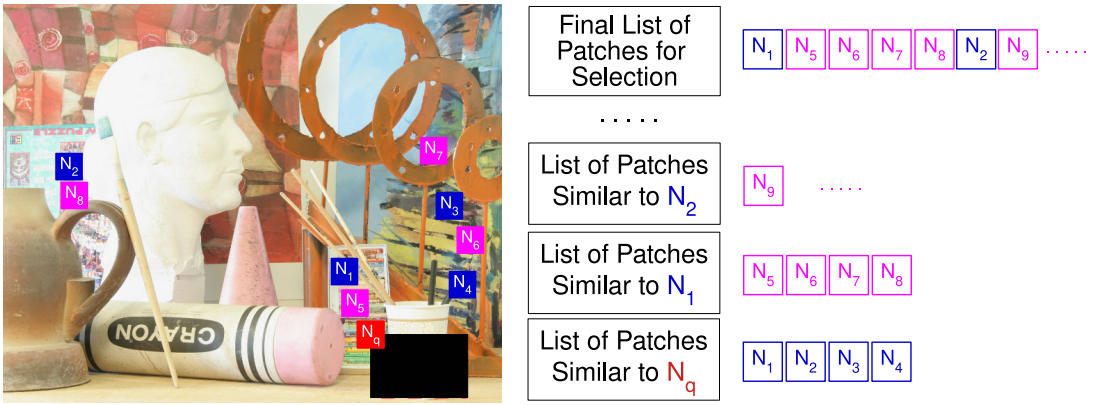

Figure 3.6: An example demonstrating the query expansion process for the approach in Section 3.1.

As seen in Figure 3.6, after the initial list of neighbourhoods is created by finding sample patches from the known regions of the image, the search process can be started again by issuing new queries in search of secondary patches similar to the sample patches already in the list.

This can be repeated several times; however, experiments have empirically shown acceptable results are achieved when the search space is expanded once. By expanding the search for each pixel, even if the neighbouring pixels are synthesised incorrectly, the overall probability of degenerate sampling due to a few bad pixels is greatly reduced. Exemplar comparative results, operating solely on RGB colour at this stage, are shown in Figure 3.1.

Within our framework, this extended texture synthesis is performed on RGB-D imagery with the SSD (Equation 3.3) extended to a four channel $\{R,B,G,D\}$ formulation to operate on both colour (RGB) and high frequency depth (D) pixel values (Figure 3.4 - upper). In general, the concept can be applied to RGB and/or depth information in isolation (Figures 3.1 and 3.2).

Furthermore, this expansion of search space can be applied to any other approach that issues queries in a similar fashion, be it for various types of patch-based or pixel-based texture synthesis techniques or inpainting methods [60, 62, 63].

Structural Inpainting

The variational framework for inpainting proposed in [66] is mainly inspired by the energy function used in [300]. However, in [66], a second term is added to the energy equation, which improves the estimation of the image function.

Let u be the image function $u : \Omega \rightarrow \mathbb{R}$, where Ω denotes the image domain. $O \subset \Omega$ is the target region (hole) and $O^c = \Omega \backslash O$, with $\tilde{O}$ taken as the extended target region, which consists of the centers of the

patches that intersect the target region. As a result, the patches in $\tilde{O}^c$ are completely outside the target region O.

The energy function proposed in [66] measures the consistency between the patches in $\tilde{O}$ and $\tilde{O}^c$ for a similarity weight function $w : \tilde{O} \times \tilde{O}^c \to \mathbb{R}$ (Equation 3.6).

$$E(u, w) = \frac{1}{h}\tilde{F}_w(u) - \int_{\tilde{O}} H_w(x)dx, \tag{3.6}$$

subject to: $\int_{O^c} w(x, \hat{x})d\hat{x} = 1,$

where h is the selectivity parameter of the Gaussian weights,

$$\tilde{F}_w(u) := \int_{\tilde{O}} \int_{\tilde{O}^c} w(x, \hat{x})\epsilon(p_u(x) - p_{\hat{u}}(\hat{x}))d\hat{x}dx, \tag{3.7}$$

and

$$H_w(x) := -\int_{\tilde{O}^c} w(x, \hat{x}) \log w(x, \hat{x})d\hat{x}, \tag{3.8}$$

where ϵ is an error function for image patches and $H_w(x)$ is the entropy of the probability function $w(x, .)$. For further details on the error function ϵ, please see [66].

Minimizing Equation 3.7 will result in patches $p_u(x)$ and $p_{\hat{u}}(\hat{x})$ being similar when the weight $w(x, \hat{x})$ is high. By taking the limit $h \to 0$, the energy is approximated as:

$$E(u, w) \simeq \int_{\tilde{O}} \int_{\tilde{O}^c} w(x, \hat{x})\epsilon(p_u(x) - p_{\hat{u}}(\hat{x}))d\hat{x}dx. \tag{3.9}$$

Therefore, inpainting is now formulated as an optimisation problem:

$$(u^*, w^*) = \arg\min_{u,w} E(u, w), \tag{3.10}$$

subject to: $\int_{\tilde{O}^c} w(x, \hat{x})d\hat{x} = 1 \quad \forall x \in \tilde{O},$

where E is the energy defined in Equation 3.6. The minimization of E is described in [66]. The inpainting is then applied on a Gaussian image pyramid [88]. The results at each scale are up-sampled, starting from the coarse scale and ending with the finest scale and used as initial values in the next scale.

Let there be S scales with A_0 being the size of the image at the finest scale and A_{S-1} at the coarsest. The process is initialised at the coarsest scale:

$$(u^{S-1}, w^{S-1}) = \arg\min_{u,w} E_{a_0}(u, w), \tag{3.11}$$

Figure 3.7: Examples of the results of the approach proposed in Section 3.1 applied to real-world images captured from urban driving scenarios.

where a_0 is the size of the patch and E_{a_0} the corresponding energy. The subsequent up-sampling from one scale to the next is done as per [80].

Here, [66] provides a robust approach for the structural inpainting of low frequency shape information within the scene (Figure 3.4 - lower). However, in general, our overall depth completion proposal remains agnostic to the exact inpainting routine in use for this task and many other inpainting techniques can yield similarly satisfactory or even superior results.

3.2 Constrained Exemplar-based RGB-D Image Completion

Although there have been many attempts to use structure-based or exemplar-based colour image completion approaches for depth hole filling [60, 66, 68, 74], particular factors such as the absence of granular texture, clear object separation and the lack of in-scene transferability of varying depth sub-regions all create notable obstacles not present in the corresponding colour completion case.

In this section, we propose an improved exemplar-based inpainting approach [60] for depth completion that adds additional *boundary* and *texture* terms to aid in determining the priority of the sample patches

used to propagate the structure and texture into the target region (Figure 2.2). High computational demands, commonly associated with such approaches, are also reduced by dynamically constraining the query space based on the location of spatially adjacent sample patch selections. This is demonstrated by providing superior results within a traditional exemplar-based image completion paradigm against other contemporary approaches. In the following, the approach is outlined in detail.

3.2.1 Details of the Approach

In this approach, improvements are made to the framework of the exemplar-based inpainting technique in [60] to create a more suitable and efficient depth completion approach. In the methodology of [60], the target region and its boundary are identified, a patch is selected to be inpainted and the source region is queried to find the best-matching patch via an appropriate error metric (e.g. sum of squared differences). After the candidate patch is found, all the information is updated and the process starts over. An extremely important factor in generating desirable results is the order in which these patches are selected for filling.

In [60], the priority of each patch is given by:

$$P(p) = C(p)D(p), \tag{3.12}$$

where $C(p)$, the *confidence* term and $D(p)$, *data* term, are determined by:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (\mathcal{I} - \Omega)} C(q)}{|\Psi_p|}, \tag{3.13}$$

$$D(p) = \frac{|\nabla \mathcal{I}_p^{\perp} \cdot n_p|}{\alpha}, \tag{3.14}$$

where $|\Psi_p|$ is the area of the selected patch Ψ_p, $\mathcal{I}$ is the image, Ω is the target region, α is the normalisation factor, n_p is a unit vector orthogonal to the target boundary and $\perp$ is the orthogonal operator (Figure 2.2). Before the inpainting begins, the *confidence* term is initialised as:

$$C(p) = \begin{cases} 0, & \forall p \in \Omega \\ 1, & \forall p \in \Omega - \mathcal{I} \end{cases} \tag{3.15}$$

The *confidence* term prioritises patches constrained by more valid depth values (fewer missing neighbours) and the *data* term encourages the filling of patches into which isophotes (lines of equal intensity)

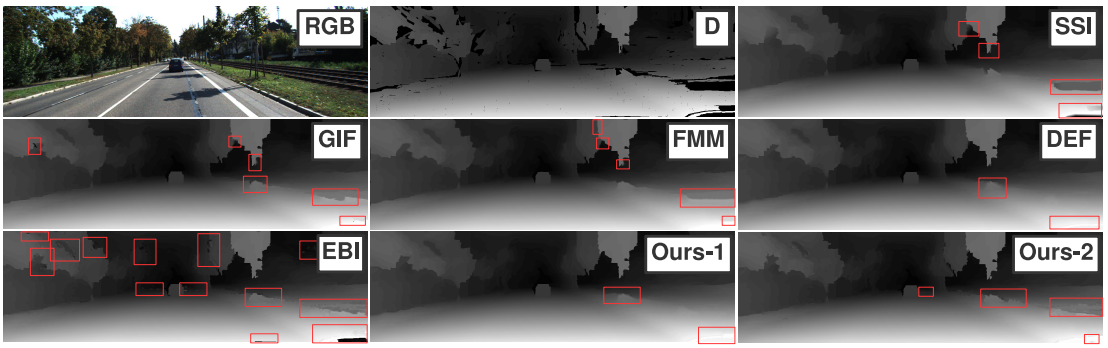

Figure 3.8: An example of the results of the Fourier-based inpainting method proposed in Section 3.1 (Ours-1) and exemplar-based depth completion approach proposed in Section 3.2 (Ours-2) applied to the KITTI dataset [301] compared to the depth inpainting approach using second-order smoothness prior (SSI) [125], guided inpainting and filtering (GIF) [41], the fast marching method based inpainting (FMM) [68], diffusion-based exemplar filling (DEF) [66] and the seminal exemplar based inpainting (EBI) in [60], by which the approach proposed in Section 3.2 is inspired.

flow. This framework creates a balance between these two terms for a more plausible inpainting [60]. However, when completing real-world depth images with large holes covering entire objects, boundaries and isophotes, the information in the accompanying colour image (within RGB-D) can be used to create a suitable depth completion approach.

Here, the *confidence* term is initialised and updated based on the depth image while the *data* term is calculated over the corresponding colour image region (from RGB-D).

To ensure a better flow of dominant linear structures into the target region, a *boundary* term is added based on the colour image:

$$B(p) = \frac{\sum_{q \in \Psi_p \cap (\mathcal{I}-\Omega)} (|G_{x \geq \tau}(q)| + |G_{y \geq \tau}(q)|)}{|\Psi_p|}, \tag{3.16}$$

where $G_{x \geq \tau}$ and $G_{y \geq \tau}$ are strong intensity gradients in the colour image in the x and y directions respectively, with τ being the gradient threshold (e.g. $\tau = 0.7$). This term essentially prioritises patches that contain a larger number of pixels that are part of a significant edge or gradient structure in the colour image. This ensures a better propagation of object boundaries into the target region. As seen in Figure 3.9, the original exemplar-based approach [60] gives equal priority to points A, B and C (Figure 3.9, result of [60]) while the proposed method prioritises points B and C because of the *boundary* term (Figure 3.9, proposed approach), which greatly affects the quality of the results.

Additionally, a *texture* term is introduced to guarantee a better propagation of texture into the target region. Since the colour and depth gradients in certain parts of an image do not always match due to factors such

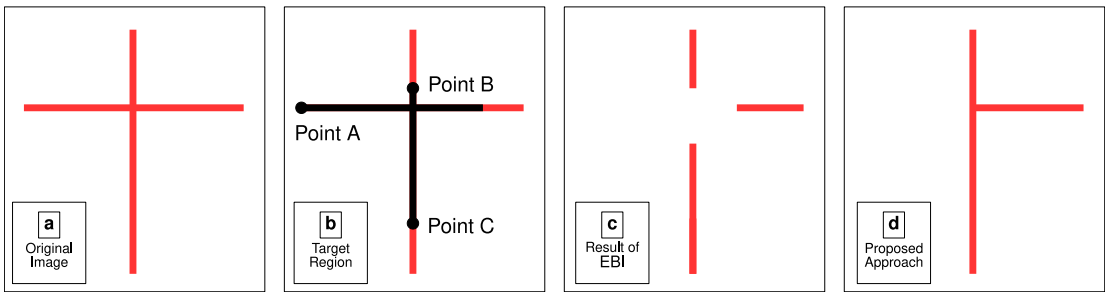

Figure 3.9: A demonstration of the effect of the *boundary* term on a synthetic image. The approach proposed in Section 3.2, which utilises this term creates better results than the seminal exemplar based inpainting (EBI) in [60], which the proposed approach is based on.

as lighting and perspective, colour information is not always a great indicator of texture. However, soft depth gradients always point to texture and relief, even though a depth image might appear smooth to the human eye. The *texture* term, which is applied to the depth image, determines which parts of the image surrounding the target boundary contain texture and encourages the process to fill them earlier to propagate texture in the target region:

$$T(p) = \frac{\sum_{q \in \Psi_p \cap (\mathcal{I}-\Omega)} |G_{x<\tau}(q)| + |G_{y<\tau}(q)|}{|\Psi_p|}, \tag{3.17}$$

where $G_{x<\tau}$ and $G_{y<\tau}$ are slight intensity gradients in the depth image in the x and y directions respectively, with τ being the gradient threshold (e.g. $\tau = 0.3$). Smallest changes in the depth image are identified and taken into account for a better relief texture propagation. As seen in Figure 3.10, in which significant edges and linear structures are hard to find, the proposed method correctly prioritises patches with slight depth changes and functions better than the original approach [60]. After adding the

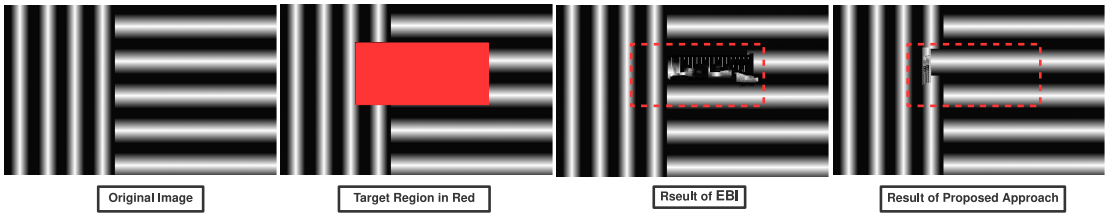

Figure 3.10: A demonstration of the effect of the *texture* term on a synthetic image. The approach proposed in Section 3.2, which utilises said term creates better results than the seminal exemplar based inpainting (EBI) in [60], which the proposed approach is based on.

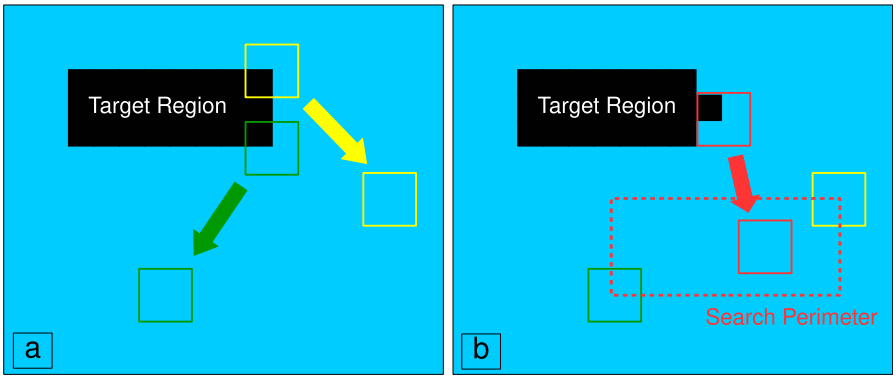

Figure 3.11: A demonstration of constraining the query space to improve the efficiency of the approach proposed in Section 3.2.

two aforementioned terms, the priority evaluation function is transformed to:

$$P(p) = C(p)D(p)B(p)T(P), \tag{3.18}$$

where $C(p)$, $D(p)$, $B(p)$, $T(P)$ are the *confidence* term (based on the depth image), *data* term (based on the colour image), *boundary* (based on the colour image) and the *texture* term (based on the depth image) respectively.

Finally, in most exemplar-based methods [60, 127, 302], the entire source region is queried for candidate patches. However, our analyses show that most suitable candidates for any patch are located close to where the best-matching candidates are found for adjacent patches in previous patch filling iterations. As a result, a dynamic search perimeter is created when sampling candidates for a patch with previously filled neighbours (Figure 3.11). The maximum and minimum of x and y indices of the selected candidates for the previously-filled adjacent patches are used to determine the perimeter. Tests run over 20 different colour and depth image pairs indicate that in 91.2% of queries, the best matching patch is found inside perimeter. Although this can negatively effect the quality of the results for the remaining 8.8% of patches, the efficiency is improved by an average of 31% (with negligible standard deviation), which is significant.

3.3 Experimental Results

In this section, we present the experimental analysis of the exemplar-based approaches proposed in Sections 3.1 and 3.2. We evaluate the techniques both qualitatively and quantitatively against publicly available benchmark data [301, 303].

3.3.1 Qualitative Evaluation

Results evaluating the qualitative efficacy of the approaches proposed in this chapter across both interior and exterior scenes are presented here. For the Fourier-based approach in Section 3.1, these results include depth relief completion (Figures 3.2 and 3.3) and object removal (Figures 3.5 and 3.1), in addition to combined depth completion and object removal in complex exterior scenes in the context of driving scenarios within urban environments (Figure 3.7).

As indicated in Figure 3.2, the method proposed in Section 3.1 for depth completion can be used to fill the existing holes in depth images. The structural inpainting method [66] used alone can create additional artefacts in the image around the edges and is incapable of producing accurate texture, whereas the proposed approach can identify texture and edges more accurately and is capable of producing more visually satisfactory results with sharper and crisper depth relief.

This approach has also been applied to single depth images with missing data. As seen in Figure 3.3, it is capable of filling the holes in the images obtained through active 3D sensors without being affected by any noise or depth blurring as is usually the case in most hole-filling strategies or inpainting techniques applied to depth images [66, 138, 140]. Moreover, even when there are holes in or around the boundaries of narrow objects, our approach functions effectively.

Figure 3.5 demonstrates what results the approach proposed in Section 3.1 can yield when applied to an actively sensed RGB-D image (Microsoft Kinect). An object has been removed from the both the colour and the depth image and the naturally-occurring holes in the depth image have also been filled successfully as seen in Figure 3.5.

Evaluation was also performed using real-world images locally captured using stereo cameras for automotive applications [296, 297]. An example of a major challenge in vision-based automotive systems is being able to remove undesirable dynamic objects while trying to map a static scene through which the automotive system attempts to navigate [296]. Figure 3.7 demonstrates how selected objects, such as other cars or pedestrians, have been removed from both the RGB and disparity images and the existing holes in the disparity maps have been successfully filled.

We also utilise publicly available test images from the KITTI dataset [301] to assess the performance of the approaches. As a well-known benchmark for a variety of computer vision tasks such as stereo correspondence, depth estimation, optical flow, semantic segmentation and others, KITTI [301] offers large quantities of labelled image data for training and testing tailored towards challenging driving scenarios. At a resolution of 1242×375, KITTI images contain cluttered scenes with a large number of objects and components with different texture types which are suitable for testing our approaches in Sections 3.1 and 3.2.

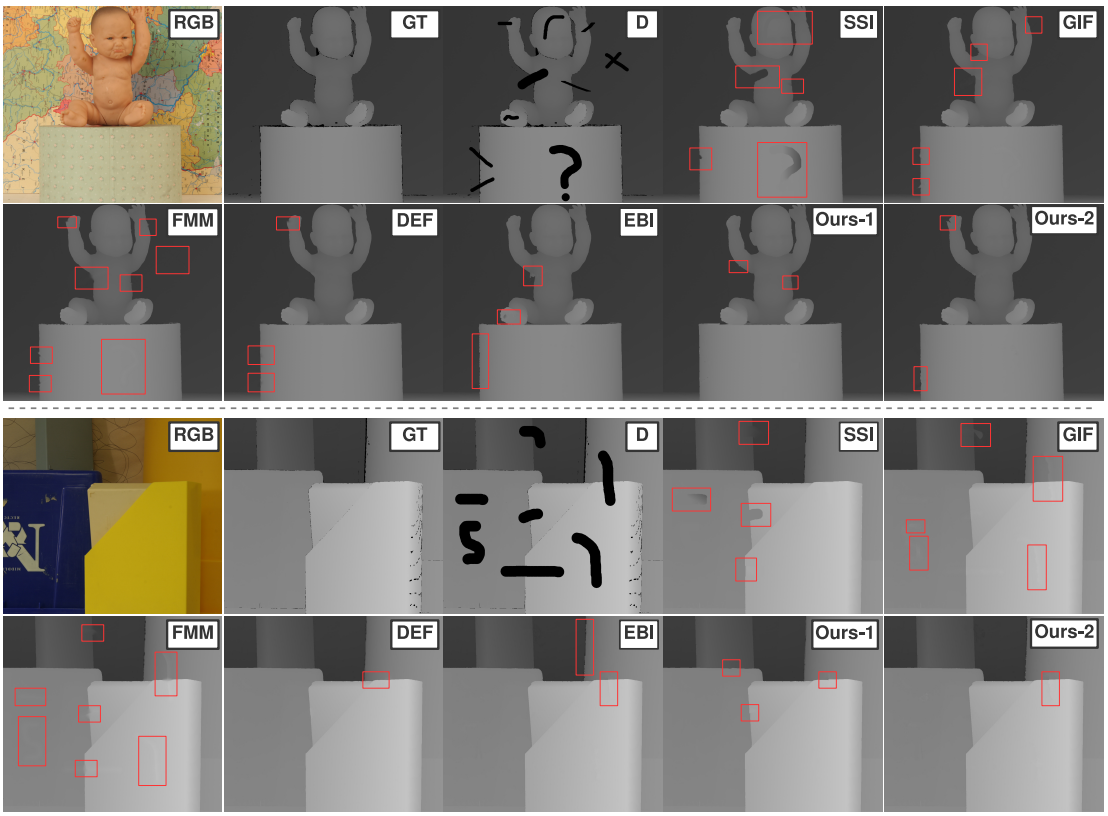

Figure 3.12: Examples of the results of the Fourier-based inpainting method proposed in Section 3.1 (Ours-1) and exemplar-based depth completion approach proposed in Section 3.2 (Ours-2) using the Middlebury dataset [303] compared to depth inpainting approach using second-order smoothness prior (SSI) [125], guided inpainting and filtering (GIF) [41], the fast marching method based inpainting (FMM) [68], diffusion-based exemplar filling (DEF) [66] and the seminal exemplar-based inpainting (EBI) in [60], by which the approach proposed in Section 3.2 is inspired.

Figure 3.8 depicts the results of the two approaches proposed in Sections 3.1 and 3.2 in comparison with the methods in [41, 60, 66, 68, 125]. For this experiment, depth is calculated from the rectified stereo images using [304] with significant disparity speckles filtered out. Both approaches result in sharp images with fewer additional artefacts (Figure 3.8).

Moreover, the Middlebury dataset [303] is used to provide more qualitative evaluation. Figure 3.12 demonstrates that our techniques generate plausible results without significant invalid outliers, blurring, jagging or other artefacts compared to other methods such as [41, 60, 66, 68, 125]. All flaws and artefacts are marked in Figure 3.12.

Method	Plastic [303]			Baby [303]			Bowling [303]		
	RMSE	PBMP	Run-time (s)	RMSE	PBMP	Run-time (s)	RMSE	PBMP	Run-time (s)
GIF [41]	0.7947	0.0331	3.10800	0.6008	0.0095	2.58000	0.9436	0.0412	4.87500
SSI [125]	1.7573	0.0102	42.3600	2.9638	0.0180	41.2000	6.4936	0.0455	71.1200
FMM [68]	0.9580	0.0435	0.9390	0.83490	0.0120	0.79400	1.2422	0.054	1.1190
DEF [66]	0.6952	0.0032	1641.33	0.6755	0.0024	995.250	0.4857	0.0035	1937.47
EBI [60]	0.4081	0.0066	2145.71	0.9053	0.0025	1196.49	0.8733	0.0045	2921.15
Ours - 1	0.8643	0.0023	>20000	0.6238	0.0081	>20000	0.5918	0.0072	>20000
Ours - 2	0.3843	0.0051	1538.16	0.6688	0.0021	879.730	0.7021	0.0037	1606.75

Table 3.1: Numerical evaluation of the Fourier-based inpainting method proposed in Section 3.1 (Ours-1) and the exemplar-based depth completion approach proposed in Section 3.2 (Ours-2) using the Middlebury dataset [303] compared to depth inpainting approach using second-order smoothness prior (SSI) [125], guided inpainting and filtering (GIF) [41], the fast marching method based inpainting (FMM) [68], diffusion-based exemplar filling (DEF) [66] and the seminal exemplar based inpainting (EBI) in [60]. RMSE (Root-Mean-Square Error), PBMP (Percentage of Bad Matching Pixels) and mean run-time are used as metrics to evaluate the performance of the approaches. RMSE measures the Euclidean distance between the results and the ground truth, while PBMP represents the percentage of image pixels whose values differ in the result and the ground truth above a certain acceptable error threshold (1 in this work). PBMP is in the range [0,1].

3.3.2 Quantitative Evaluation

Depth completion is fraught with constant compromises between efficiency and accuracy. The approaches proposed in this chapter best epitomize this notion, as one (Section 3.1) is inefficient but outperforms comparators in terms of plausibility and accuracy, while the other (Section 3.2) offers a more balanced trade-off between accuracy and efficiency. This trade-off is best demonstrated based on the fact that the approach proposed in Section 3.2 outperforms many of the other more efficient approaches [41, 60, 68, 125] while being faster than the approach in Section 3.1 and others [60, 66].

Table 3.1 provides a quantitative evaluation of the proposed approaches against the same comparator set (the guided inpainting and filtering (GIF) [41], the second-order smoothness inpainting (SSI) [125], the fast marching method (FMM) [68], diffusion-based exemplar filling (DEF) [66] and the exemplar-based inpainting (EBI) [60]). As shown in Table 3.1, the approach proposed in Section 3.1 performs very well numerically but is very inefficient. Our approach proposed in Section 3.2, however, is in balance between efficiency and accuracy. While it is more efficient than other exemplar-based methods [60, 66], it has a smaller root-mean-square error and fewer bad pixels than faster comparators [41, 68]. Experiments are performed using a 2.30GHz CPU (Table 3.1).

3.4 Limitations

Even though the Fourier-based completion approach presented in Section 3.1 is capable of plausible object removal from RGB-D images and depth completion, its biggest shortcoming is inefficiency. Completing the high frequency components of an image can take in the order of hours for a high-resolution image since the image is completed one pixel at a time and for every pixel to be filled, the entire image needs to be sampled to find patches suitable to be used in the completion process. Consequently, efficiency has been sacrificed for the sake of accuracy. Consequently, the next logical step would be to propose an alternative approach that is more in balance with respect to the trade-off between accuracy and efficiency.

The exemplar-based approach proposed in Section 3.2 offers a more balanced compromise with respect to efficiency and accuracy. It is not capable of producing final depth outputs as accurately as the approach in Section 3.1 but with a much more reasonable run-time, it can be used in specific off-line applications. It is highly capable of synthesising depth texture and surface relief even though it may fall short when dealing with scenes that are not of a fronto-parallel view.

3.5 Summary

In this chapter, we propose two novel exemplar-based approaches to object removal and depth completion within the context of RGB-D images. The first techniques mainly revolves around on RGB-D image completion, taking advantage of the Fourier space to facilitate plausible hole filling prioritising both high frequency depth detail (relief) and low frequency surface continuation (shape). After using the Discrete Fourier Transform to decompose a depth image into separate high and low frequency component via Butterworth filtering, our proposed technique employs an improved texture synthesis method to fill the high frequency depth detail and a structural inpainting approach to complete the underlying image structures. The two image components are then recombined in the Fourier space to create a depth image, where the holes are plausibly filled and the surface relief and edges preserved. While this approach produces accurate and visually plausible outputs, its inefficiency makes in intractable for use in any real-world application.

Qualitative and quantitative evaluations reveal how this approach is capable of plausibly and accurately completing depth images, outperforming comparators [41, 60, 66, 68, 125] often used for the same objective. Despite its accuracy and attention to surface relief and boundary details, the approach suffers from intensive computational requirements and a slow run-time, leading to the need for a more balanced perspective towards the compromise between accuracy and efficiency.

Consequently, the second proposed approach addresses the problem of depth completion with a focus on a reasonable trade-ff between efficiency and attention to surface (relief) detail accuracy. While exemplar-based methods are mostly used for colour images, their ability to preserve texture in the target region makes them an ideal candidate for depth completion when texture is of importance. In this approach, the priority term that determines the order of patch sampling has been modified to allow for a better propagation of strong linear structures and texture into the target region. Moreover, by constraining the query space, the method performs more efficiently than other exemplar-based approaches. Experiments involving this technique demonstrate that while the efficiency of the proposed method is better than other exemplar-based frameworks, the plausibility and statistical relevance of the depth filled results compete against the accuracy of contemporary filling approaches in the field.

Based on extensive assessments, the approaches proposed in this chapter have contributed to the literature on RGB-D completion by advancing the state of the art and improving on and outperforming well-established existing techniques such as [41, 60, 66, 68, 125].

However, despite the higher accuracy of the first approach proposed in Section 3.1 and even with the improved efficiency of the second approach proposed in Section 3.2 of this chapter, it cannot be used in real-time applications as the run-time is still in the order of minutes. Consequently, in Chapter 4 we shift our focus towards a much more efficient approach, which can potentially be used in real time.

Chapter 4

Efficient Depth Completion Based on Prior Scene Segmentation

One of the most major issues commonly associated with most effective depth completion techniques, such as the ones proposed in Chapter 3, is considerable computational requirements. As a result, in this chapter, we propose a simple and efficient method for depth image completion that utilises a prior semantic segmentation labelling of the accompanying colour image [277]. The depth completion process proposed in this chapter is performed with reference to object boundaries on a pixel-wise basis in the depth image based on a grammar of holes, where pixel values are parsed to identify and fill instances of holes.

The material presented in this chapter of the thesis has been published in the following peer-reviewed publication:

- **A. Atapour-Abarghouei** and T. P. Breckon. 'DepthComp: Real-Time Depth Image Completion based on Prior Semantic Scene Segmentation.' In Proc. British Machine Vision Conference, 2017 [36].

Our process leverages advances in semantic scene segmentation [277], such that completion can be performed with reference to object boundaries within the scene. Here, focusing on the challenge of outdoor driving scenes, we utilise SegNet [277, 305], a deep convolutional architecture trained for urban scene segmentation in the context of vehicle autonomy. However, in general, any such approach that can perform accurate and efficient object or instance wise scene segmentation can suffice. In fact, as seen in Figure 4.1, the propose approach is capable of completing the depth image using different forms

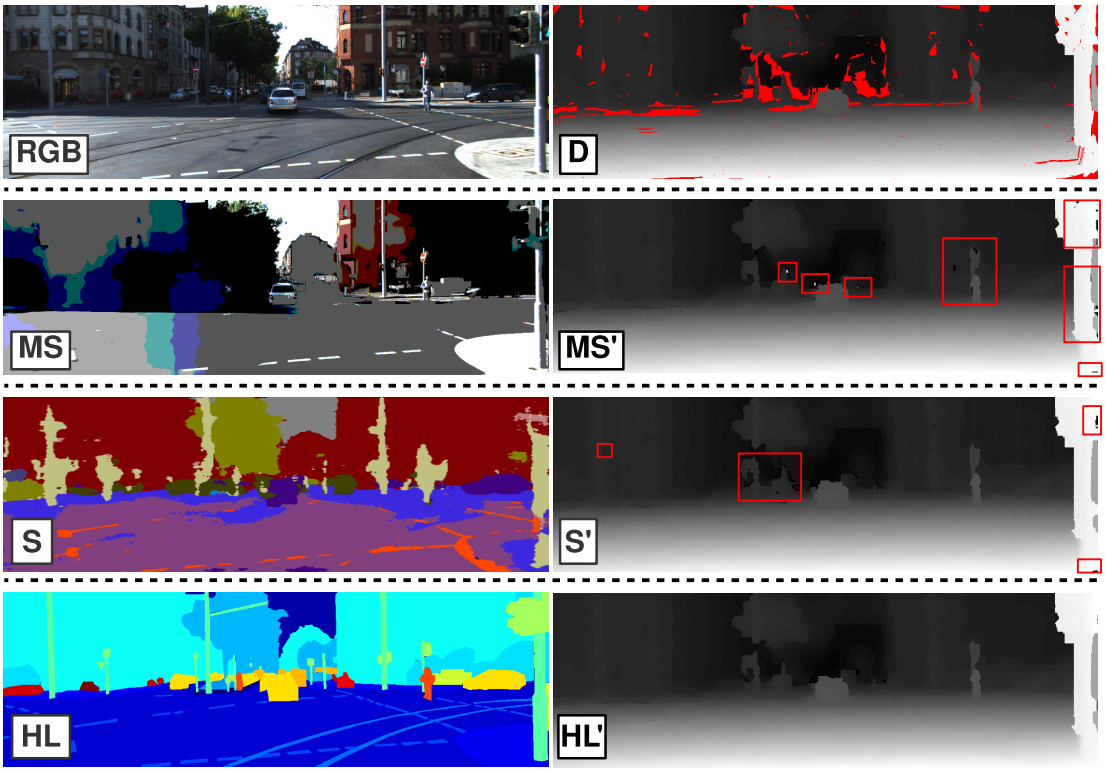

Figure 4.1: Comparison of the proposed method using different initial segmentation techniques on the KITTI dataset [301]. Original colour and disparity image (RGB, D), results with manual labels (HL), results with SegNet [277] (S) and results with mean-shift [306] (MS) are presented.

of segmented images [277, 306] with the quality of the final depth output depending on the quality of the segmentation. Our technique is a computationally inexpensive completion approach requiring a maximum of three passes over the image on a row and column-wise basis. Within this context, a hole is now defined as a sequence of missing depth values constrained to one scene object within a single row/column of the depth image. To these ends, a depth hole (i.e. missing region in the image) is now comprised of multiple such *constrained holes*, all limited to the completion of a single row/column, with respect to a single adjacent scene object at the hole boundary. In a sense, we now have a grammar of holes, with each instance parsed based on row-wise adjacent depth availability for a consistent scene object.

As the image is scanned in raster order, each hole discovered is identified as one of twelve possible completion cases with reference to both the pattern of missing depth and the consistency of surrounding segmented pixel labelling. Whilst each case may be efficiently implemented in isolation, a common

Type	%	Type	%
Case 1	11.2	Case 7	7.75
Case 2	0.32	Case 8	0.33
Case 3	57.0	Case 9	1.93
Case 4	1.99	Case 10	2.47
Case 5	3.44	Case 11	10.3
Case 6	0.22	Case 12	3.03
Filled	98.8	Unfilled	1.17

Table 4.1: Frequency of the hole occurrences using data from KITTI [301].

notion of informed re-sampling, behind the overall solution to all cases, provides underpinning plausible completion in the general sense. This avoids the simplicity of a brittle rule-based technique whilst taking advantage of a discrete set of hole occurrences at the local level to aid efficient implementation.

4.1 Semantic Segmentation for Object-Wise Depth Completion

Here, we primarily use SegNet [277] to perform the initial segmentation task. The SegNet architecture consists of corresponding encoder and decoder layers and a pixel-wise classification layer. The encoder is similar to the convolutional layers in VGG16 [307], while the decoder maps the low-resolution feature maps from the encoder as inputs for subsequent pixel-wise semantic classification.

Although SegNet [277] shows sufficient accuracy for our task, an *idealized* scene depth completion method requires absolute labelling accuracy beyond that of SegNet. As we illustrate (Figure 4.1), alternative segmentation models (object, instance or otherwise) [308–310] can similarly be used, provided they produce limited mis-segmentation artefacts.

4.2 Details of the Proposed Approach

Depth completion is performed in three image passes: primary row-wise, column-wise and secondary row-wise. For explanation purposes, we detail the outline of our approach solely in terms of image rows with the intermediate column-wise pass being purely a rotational analogue of the same. Each constrained depth hole can be identified as either one of eight non-parametrically solvable cases (subsequently outlined with a corresponding algorithmic solution) or as one of four remaining unresolvable cases. When a case does not conform to a solvable case in a given pass, it is left to subsequent passes whereby the completion of other neighbourhood pixels may allow subsequent resolution into one of these cases. In

cases where a pixel remains unresolvable after all three passes, we refer to the use of bilinear interpolation. From Table 4.1, we see the occurrence of these non-parametrically unresolvable cases is indeed very limited. Figure 4.2 provides a first glimpse into this process. As can be seen, holes conforming to the first eight cases, which will be subsequently explained in detail, have been successfully filled in the first pass and the remaining holes are left unfilled to be reclassified and dealt with in subsequent passes.

Algorithm 4.1: Depth completion based on hole cases as defined in Section 4.2.

```
l ← length of the hole.
c ← completion case identifier.
if c in {1, 2, 3, 4} then
    i ← index of leftmost pixel in the hole
end
else if c in {5, 6, 7, 8} then
    i ← index of rightmost pixel in the hole
end
assign initial v_0(i) according to case c.
assign slope according to case c.
while i is in the hole region do
    update v(i) according to case c
    if c in {1, 2, 3, 4} then
        i ← i + 1.
    end
    else if c in {5, 6, 7, 8} then
        i ← i − 1.
    end
end
```

When a hole of a specific length is identified within a row, the information available to the left and the right of the hole within the same object boundaries is surveyed and surface depth pattern is propagated into the hole region. A continuity coefficient ($slope$) is taken into account during this propagation to plausibly bridge the depth values on both sides of the hole. Although all constrained hole cases are essentially processed identically, the availability of valid depth values and appropriate sampling region govern the categorisation of such row-wise constrained hole occurrences into a number of discrete cases. Of these twelve such completion cases, many are inherently similar in their characteristics with our detailed separation on a case-wise basis only aimed at maximizing accuracy and efficiency.

***Case 1*:** where the constrained hole ends at the rightmost boundary of the object, i.e. all depth values on the right side of the current object are missing but the number of preceding depth values to the left of the

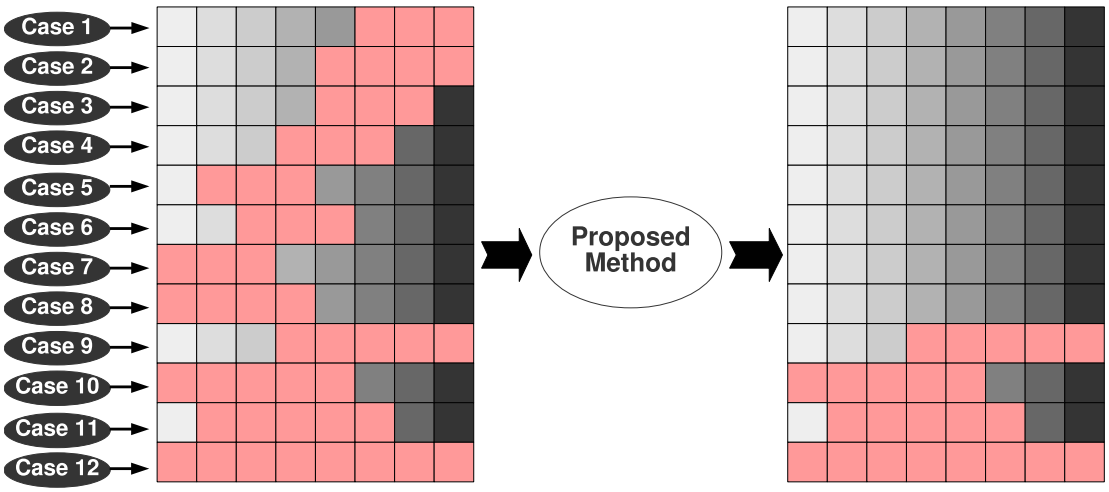

Figure 4.2: Exemplar completion cases (left) with primary row-wise completion (right).

hole exceeds the length of the hole itself.

$$v(i) = v(i-1) + [v(i-l) - v(i-l-1)] \times slope. \tag{4.1}$$

Since such holes extend to the rightmost pixel in the current object, no depth information is available to the right of the hole and as such there is no need to account for any in-filling continuity. Consequently, it suffices to identify the pattern of depth change to the left of the hole, the length of which is greater than the length of the hole itself and propagate this pattern rightward, replicating the texture and relief detail present within the object boundary. As a result, $slope = 1$, and $v(i)$ is initialised to zero with updates as per Equation 4.1 with reference to Algorithm 4.1. See the illustration of Equation 4.1 terms in Figure 4.3 (Case 1).

***Case 2*:** where the constrained hole ends at the rightmost boundary the object (as per Case 1) but here, the number of preceding depth values to the left of the hole is exactly the same as the length of the hole itself.

$$v(i) = v(i-1) + [v(i-l+1) - v(i-l)] \times slope. \tag{4.2}$$

Here, we proceed as per Case 1, but with less depth information present to the left of the hole to identify and propagate any pattern rightward. As a result, $slope = 1$, and $v(i)$ is initialised to zero with updates as per Equation 4.2 with reference to Algorithm 4.1. See the illustration of Equation 4.2 terms in Figure 4.3 (Case 2).

***Case 3*:** where the constrained hole does not reach the leftmost or rightmost boundary edges of the scene object, i.e. the hole is contained within the object itself with valid depth values to both the left and right. In this case, the pattern of depth change can be sampled from either side depending on valid depth value availability within the same scene object. Assuming sufficient depth values exist to the left of the hole (by default, even if sufficient on the right also), we proceed as follows:

$$v_0(i) = v(i-1) + v(i-l) - v(i-l-1), \tag{4.3}$$

$$slope = \frac{v(i+l) - v(i-1)}{v_0(i) - v(i-l-1)}, \tag{4.4}$$

$$v(i) = v(i-1) + [v(i-l) - v(i-l-1)] \times slope. \tag{4.5}$$

To predict the missing depth values correctly considering the pattern of texture and relief, continuity between the valid values to the left and the right side of the hole is taken into account. The continuity coefficient ($slope$) is utilised to ensure that the predicted values plausibly bridge the depth values to the left and right of the hole. The pattern of change in the valid values is propagated rightward with each value being multiplied by $slope$, calculated by dividing the difference between the values surrounding the hole into the difference between the values surrounding the sample area (Figure 4.3 (Case 3) and Algorithm 4.1). The initial value of $v_0(i)$ and $slope$ in Algorithm 4.1 are respectively calculated based on Equations 4.3 and 4.4. Within Algorithm 4.1, $v(i)$ is updated according to Equation 4.5. See the illustration of Equations 4.3, 4.4, and 4.5 terms in Figure 4.3 (Case 3).

***Case 4*:** as per Case 3, but such that the number of valid depth values to the left of the constrained hole is exactly the same as the length of the hole itself.

$$v_0(i) = v(i-1) + v(i-l+1) - v(i-l), \tag{4.6}$$

$$slope = \frac{v(i+l) - v_0(i)}{v_0(i) - v(i-l)}, \tag{4.7}$$

$$v(i) = v(i-1) + [v(i-l+1) - v(i-l)] \times slope. \tag{4.8}$$

The difference between this completion process and that of Case 3 is the same as the difference between Cases 1 and 2. The completion order and the $slope$ coefficient are applied similarly to Case 3. The initial value of $v_0(i)$ and $slope$ in Algorithm 4.1 are respectively calculated based on Equations 4.6 and 4.7. Within Algorithm 4.1, $v(i)$ is updated according to Equation 4.8. See the illustration of Equations 4.6, 4.7, and 4.8 terms in Figure 4.3 (Case 4).

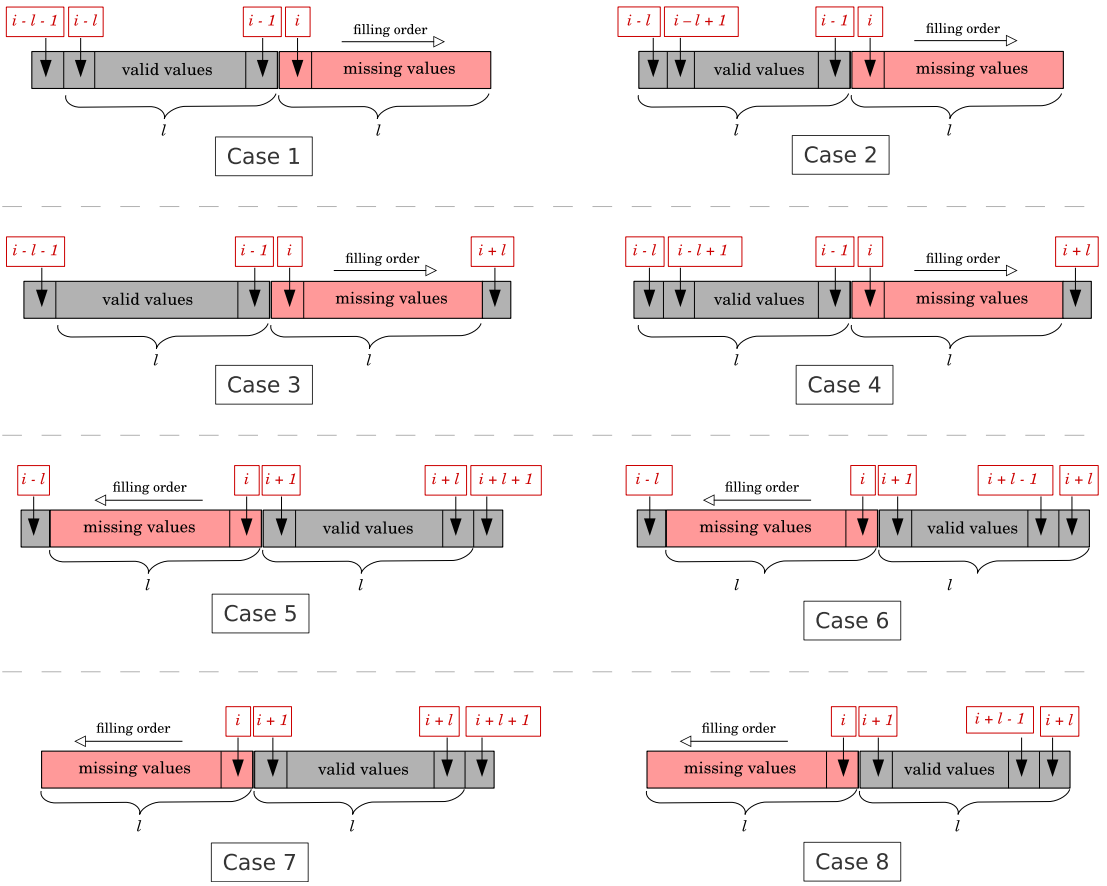

Figure 4.3: Exemplar constrained holes (row-wise), Cases 1-8.

***Case 5*:** where the constrained hole does not reach the leftmost or rightmost boundary of the object (as per Case 3) but the number of valid depth values to the left of the hole is smaller than the length of the hole itself, while sufficient valid depth values exist to the right of the hole for completion.

$$v_0(i) = v(i+1) + v(i+l) - v(i+l+1), \tag{4.9}$$

$$slope = \frac{v(i+1) - v(i-l)}{v(i+l+1) - v_0(i)}, \tag{4.10}$$

$$v(i) = v(i+1) + [v(i+l) - v(i+l+1)] \times slope. \tag{4.11}$$

Following a symmetric completion process to that of Case 3, the pattern of change in the valid depth values is propagated leftward as per Algorithm 4.1. The initial value of $v_0(i)$ and $slope$ in Algorithm

4.1 are respectively calculated based on Equations 4.9 and 4.10. Within Algorithm 4.1, $v(i)$ is updated according to Equation 4.11. See the illustration of Equations 4.9, 4.10, and 4.11 terms in Figure 4.3 (Case 5).

***Case 6*:** as per Case 5, but such that the number of valid depth values to the right of the constrained hole is exactly the same as the length of the hole itself.

$$v_0(i) = v(i+1) + v(i+l-1) - v(i+l), \tag{4.12}$$

$$slope = \frac{v_0(i) - v(i-l)}{v(i+l) - v_0(i)}, \tag{4.13}$$

$$v(i) = v(i+1) + [v(i+l-1) - v(i+l)] \times slope. \tag{4.14}$$

Following a symmetric completion process to that of Case 4, the pattern of change in the valid depth values is propagated leftward as per Algorithm 4.1. The initial value of $v_0(i)$ and $slope$ in Algorithm 4.1 are respectively calculated based on Equations 4.12 and 4.13. Within Algorithm 4.1, $v(i)$ is updated according to Equation 4.14. See the illustration of Equations 4.12, 4.13, and 4.14 terms in Figure 4.3 (Case 6).

***Case 7*:** where the constrained hole starts at the leftmost boundary edge of the scene object (symmetric to that of Case 1). Conversely, the number of valid values on the right of the hole is greater than the length of the hole itself.

$$v(i) = v(i+1) + [v(i+l) - v(i+l+1)] \times slope. \tag{4.15}$$

Following a symmetric completion process to that of Case 1, the pattern of change in the valid depth values is propagated leftward as per Algorithm 4.1. Since no continuity is required, $slope = 1$. The initial value of $v(i)$ is zero and this value is updated iteratively based on Equation 4.15. See the illustration of Equation 4.15 terms in Figure 4.3 (Case 7).

***Case 8*:** as per Case 7, but such that the number of valid depth values to the right of the constrained hole is exactly the same as the length of the hole itself.

$$v(i) = v(i+1) + [v(i+l-1) - v(i+l)] \times slope. \tag{4.16}$$

The difference between this completion process and that of Case 7 is the same as the difference between Cases 1 and 2. The depth completion order and the $slope$ coefficient are applied similarly to Case 7. Since no continuity is required, $slope = 1$. In Algorithm 4.1, the initial value of $v(i)$ is zero and this value is updated iteratively based on Equation 4.16. See the illustration of Equation 4.16 terms in Figure 4.3 (Case 8).

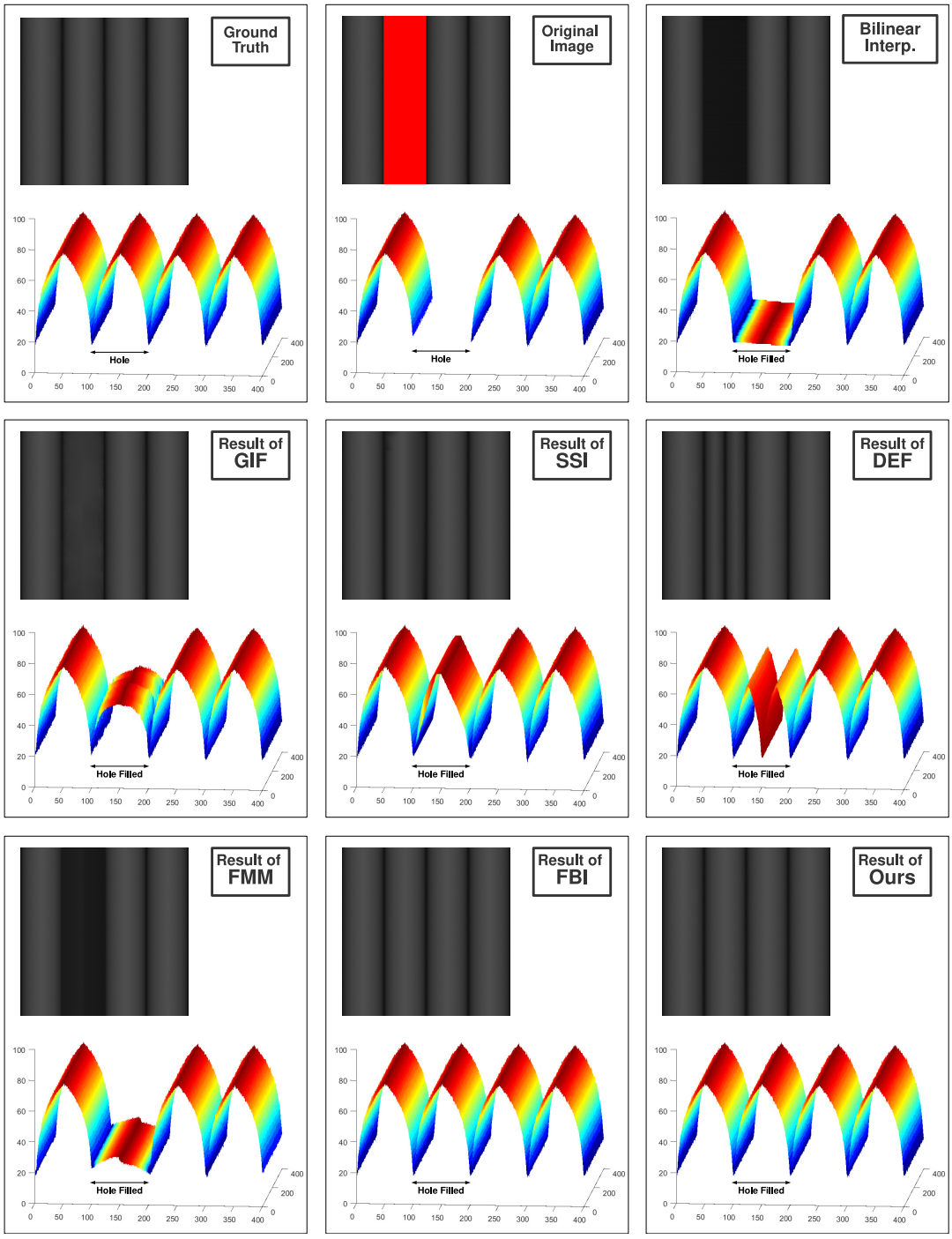

Figure 4.4: Comparing the results of guided inpainting and filtering (GIF) [41], second-order smoothing inpainting (SSI) [125], fast marching based inpainting (FMM) [68], the Fourier-based inpainting approach (FBI) proposed in Section 3.1, diffusion-based exemplar filling (DEF) [66], bilinear interpolation and the proposed approach using a synthetic test image with available ground truth depth.

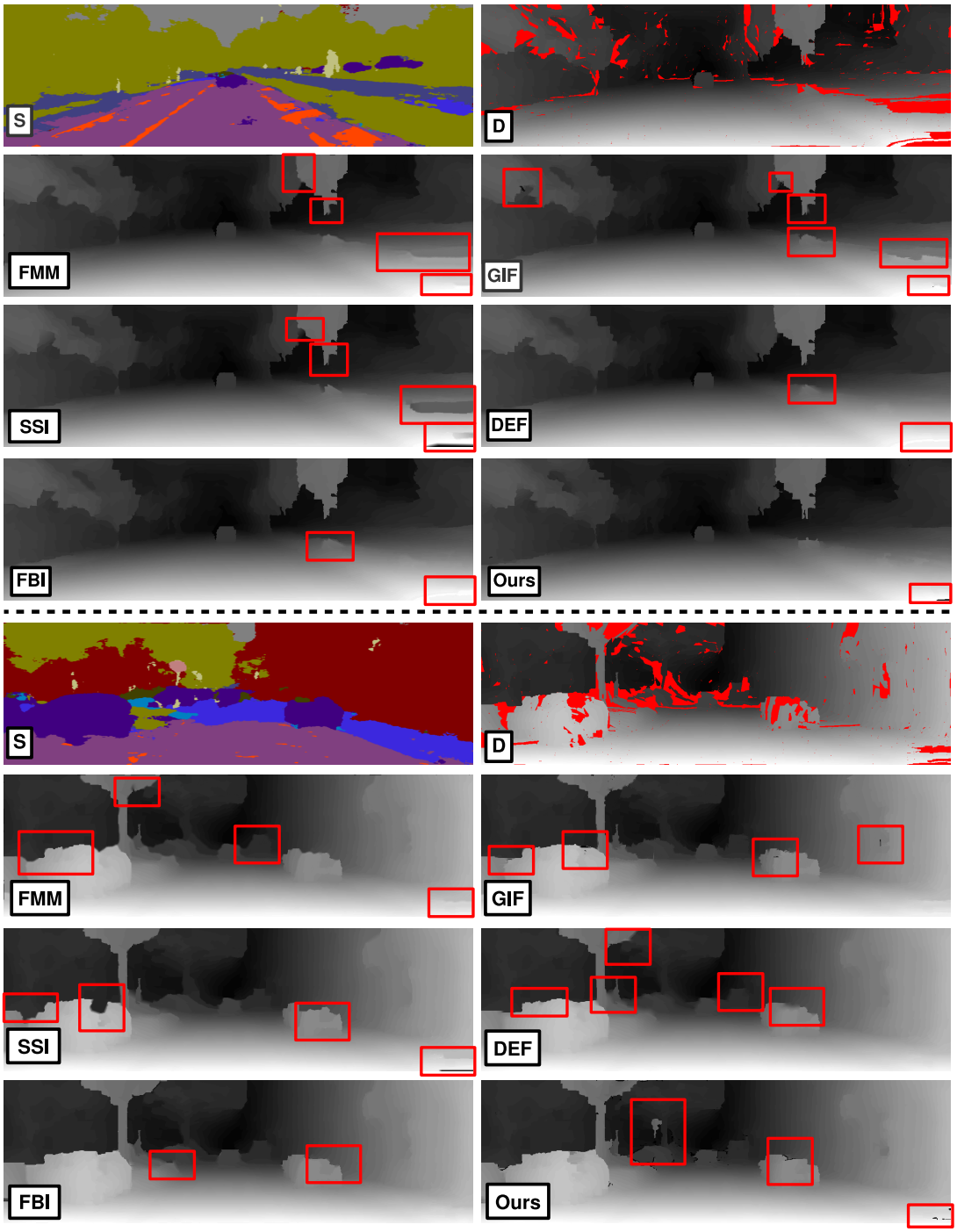

Figure 4.5: An example of the results of proposed depth completion approach applied to the KITTI dataset [301] compared to the depth inpainting approach using second-order smoothness prior (SSI) [125], guided inpainting and filtering (GIF) [41], the fast marching method based inpainting (FMM) [68], diffusion-based exemplar filling (DEF) [66], the Fourier-based inpainting method proposed in Section 3.1 (FBI). **S** denotes the segmented images [277] and **D** the original (unfilled) disparity maps.

***Case 9*:** where the constrained hole extends to the rightmost pixel within the object (similar to Cases 1 and 2), but we cannot employ a non-parametric approach (as per Cases 1 and 2) because the number of valid depth values to the left of the hole is smaller than the length of the hole itself. As a result, there is not enough information to accurately fill these holes. Instances of these cases are left unfilled if identified during the scan in progress. In subsequent scans, many of these unresolvable (Case 9) patterns are broken due to the use of alternating row-wise and column-wise scan passes (resulting in an alternative resolvable case instance). For Case 9 instances that are not resolved after all three image passes, simple (cubic) interpolation is used (in an insignificant number of cases, Table 4.1).

***Case 10*:** where the constrained hole extends to the leftmost pixel within the scene object (similar to Cases 7 and 8), but again we cannot employ a non-parametric approach (as per Cases 7 and 8) because the number of valid depth values to the right of the hole is smaller than the length of the hole itself. As a result, there is again not enough information to accurately fill these holes and we proceed as per Case 9.

***Case 11*:** where the constrained hole is located in the middle of an object but with insufficient valid depth values to the left and right side to facilitate non-parametric filling. Again, there is not enough information to accurately fill these holes and we proceed as per Case 9.

***Case 12*:** where the constrained hole spans over the entire length of the scene object, (i.e. no depth is available for an object known to be present in the scene from the semantically segmented colour image) making it the most challenging case of all. For instances of this case not resolved within the three scan passes (row-wise, column-wise, secondary row-wise), a clear ambiguity exists as there is no valid depth information available for the object at all. As Table 4.1 illustrates, this is an incredibly rare occurrence in practice and the hole is best left uncompleted rather than using invalid or implausible values (as per other work, [41, 45, 60, 66, 68, 125, 127, 131–133, 139, 148, 149, 156, 157, 170]). Table 4.1 illustrates the typical occurrence frequency of the cases (1-12) on the KITTI dataset [301] (using [304] for depth estimation). As seen in Table 4.1, less than 2% of hole occurrences cannot be completed using our three-pass approach (row-wise, column-wise, secondary row-wise) through the resolution outlined. This includes the challenging Case 12, which cannot be accurately filled due to the lack of surrounding valid depth values. Although our three pass processing of these 12 cases noticeably uses no explicit inter-row/column support regions (from adjacent row/columns) as may be ordinarily expected, this is in fact implicit in our formulation based on the use of the prior region-based scene segmentation which inherently provides semantically defined support regions.

4.3 Experimental Results

With the asymptotic runtime of $O(n)$ for n image pixels, the approach proposed in this chapter is comparable to simple interpolation methods in complexity but with accuracy exceeding that of more

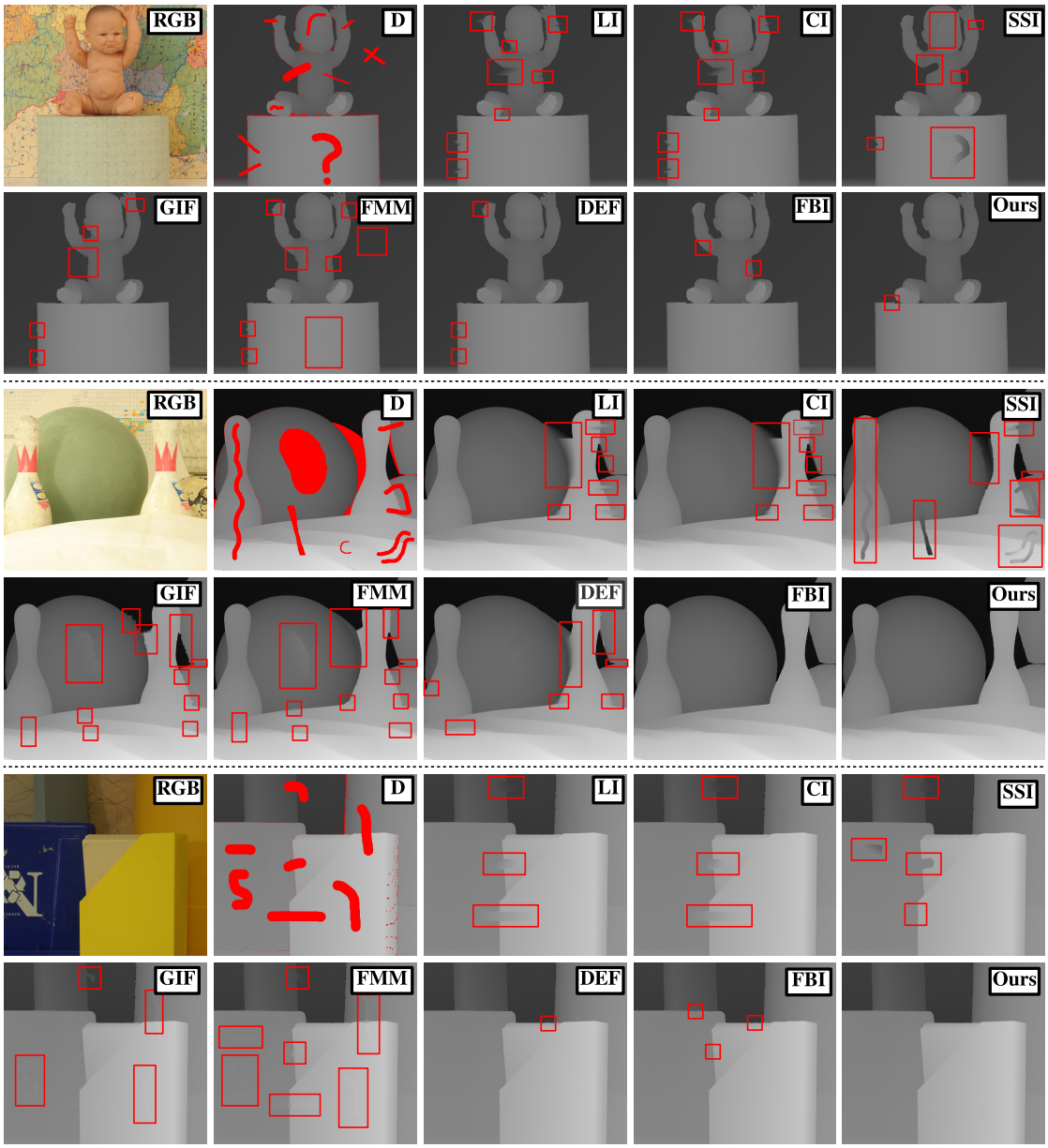

Figure 4.6: An example of the results of proposed exemplar-based depth completion approach using the Middlebury dataset [303] compared to depth inpainting approach using second-order smoothness prior (SSI) [125], guided inpainting and filtering (GIF) [41], the fast marching method based inpainting (FMM) [68], diffusion-based exemplar filling (DEF) [66], the Fourier-based inpainting method proposed in Section 3.1 (FBI) and bilinear and bicubic interpolation techniques.

Approach	RMSE	PBMP	Run-time
Bilinear	22.5868	0.2601	3.05
Bicubic	22.3810	0.2598	3.18
GIF [41]	7.3281	0.2496	$1.07e3$
SSI [125]	3.7970	0.1893	$5.92e3$
FMM [68]	18.9501	0.2663	$1.10e3$
DEF [66]	10.5448	0.1513	$>1.2e5$
FBI	0.8372	0.0863	$>3.6e6$
Ours	0.8617	0.0917	3.83

Table 4.2: Comparing the RMSE, PBMP and the run-time (ms) of the guided inpainting and filtering (GIF) approach [41], second-order smoothing inpainting (SSI) [125], fast marching based inpainting (FMM) [68], the Fourier-based inpainting approach (FBI) proposed in Section 3.1, diffusion-based exemplar filling (DEF) [66], bilinear interpolation, bicubic interpolation and the proposed approach using a synthetic test image with ground truth depth.

complex methods [41, 66, 68, 125]. The approach is first tested using a synthetically generated depth image, which can be seen in Figure 4.4. This image contains steep curves and sharp peaks to simulate exaggerated texture to evaluate the performance of the approach in the presence of surface relief within the image (under a single scene object assumption, with no need for prior segmentation). Here, Gaussian noise ($mean = 0$, $variance = 0.0001$) is added to the depth image to avoid completely smooth surfaces and a topological colour scale is used to guide methods that require additional colour image input ([41, 125]) and to aid visualization of the final result.

The superiority of the proposed approach is clearly seen in Figure 4.4. Additionally, the root-mean-square error (RMSE) and the percentage of bad pixels produced by the proposed method are far smaller than comparators, as seen in Table 4.2.

Figure 4.5 demonstrates the results of our approach compared to others when applied to examples from the KITTI dataset [301]. Depth is calculated using [304] with significant disparity speckles filtered out and SegNet [277] is used to perform the initial semantic scene understanding. The proposed method results in sharper images with no additional artefacts (Figure 4.5) and performs more efficiently than comparators (Table 4.3).

As previously discussed, the initial segmentation step can indeed be performed using any technique with the efficacy of results depending on the accuracy of this segmentation. Figure 4.1 compares the results of our approach obtained through the use of varying segmentation methods. When a manually labelled image is used (ground truth, [311]), the results are more accurate than when SegNet [277] or mean-shift [306, 309, 312] segmentation is employed.

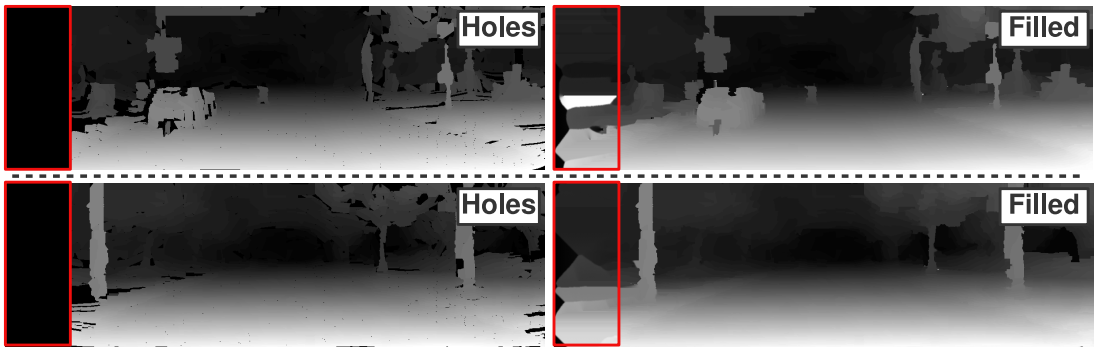

Figure 4.7: Examples demonstrating the limitations of the proposed approach. Note how unrealistic content and noise is used to fill the *large* holes artificially created in the depth images.

We also utilise the Middlebury dataset [303] to provide additional qualitative and quantitative evaluation. Figure 4.6 demonstrates that the proposed method generates more plausible results without invalid outliers, blurring, jagging or other artefacts than comparator approaches. Table 3.1 provides quantitative evaluation of the proposed approach against the same comparator set. As shown in Table 3.1, the method is faster (real time, excluding segmentation) and has a smaller root-mean-square error and fewer bad pixels [13] than comparators. Experiments were performed on a 2.30GHz CPU (Tables 4.2, 3.1 and 4.3).

4.4 Limitations

The main advantage of the technique presented here is its efficiency, as seen in Tables 4.4, 4.2 and 4.3. The approach also produces highly accurate results when applied to depth images acquired through the

Approach	Run-time	Approach	Run-time
Bilinear	7.0355	FMM [68]	83.15e1
GIF [41]	15.23e2	DEF [66]	$>$12e5
SSI [125]	20.44e3	FBI	$>$36e5 ms
Ours	11.171	Ours+[277]	632.613

Table 4.3: Comparing the average run-time (ms) of the guided inpainting and filtering (GIF) approach [41], second-order smoothing inpainting (SSI) [125], fast marching based inpainting (FMM) [68], the Fourier-based inpainting approach (FBI) proposed in Section 3.1, diffusion-based exemplar filling (DEF) [66], bilinear interpolation and the proposed approach using images from the KITTI dataset [301]. Note that our depth completion method (Ours) is real-time and even when combined with the segmentation method of [277] (Ours+[277]), it is very efficient.

	RMSE	PBMP	Run-time
Ours	0.4012	0.0021	97.38 ms

Table 4.4: Average RMSE, PBMP, & run-time using images form the Middlebury [303] dataset.

depth estimation approach of [304]. This stereo correspondence approach leads to small, albeit numerous, holes within the resulting depth image, which makes it ideal for our efficient depth completion approach. However, if large missing sections (holes) exist within a depth image, it becomes more and more probable that entire objects (as detected by the semantic segmentation step prior to depth completion) fall within the hole region boundaries. As explained in Section 4.2, when a hole covers an entire object (Case 12), our approach cannot fill the hole in a semantically meaningful way since not enough information exists to synthesise any potential content in a plausible manner, as seen in Figure 4.7. Although other conventional spatial-based depth completion techniques face a similar problem, our approach is more susceptible to this issue as it explicitly leaves such holes unfilled. As a solution to this problem, in Chapter 5, a leaning-based approach is proposed that can learn to generate content for large missing sections of a depth image, based on the information available within the corresponding colour image.

4.5 Summary

In this chapter, we attempt to address the problem of depth completion with efficiency and attention to surface (relief) detail accuracy. As a first step, our approach requires an accurate semantic segmentation over an accompanying colour image, which is commonly available from contemporary sensing arrangements, to facilitate depth completion on an object-wise basis. Subsequently, missing depth values are filled via a three-pass non-parametrically driven approach, using twelve discrete completion case occurrences. Experiments point to the efficiency of the proposed approach being comparable to simple interpolation methods while the plausibility and statistical relevance of the depth filled results compete against the accuracy of other contemporary depth filling approaches. Surface detail and relief texture is preserved within a highly efficient framework driven by recent and ongoing advances in scene labelling.

Via its capability to be deployed in real time and its high accuracy and attention to surface detail, our approach has advanced the state of the art by performing better than contemporary techniques such as [41, 66, 68, 125] and the two approaches proposed in Chapter 3 and in terms of run-time, it is only outperformed by simple interpolation techniques such as bilinear interpolation.

This approach, however, cannot successfully complete very large holes when these holes cover entire scene objects, since not enough information exists to synthesise new content. As a result, our proposed

approach is not capable of filling very large holes. Consequently, in the next chapter, we focus on a leaning-based approach capable of generating content to complete depth images with large missing sections, based on the information available within the corresponding colour image.

Chapter 5

Learning to Complete Scene Depth

Deep neural networks have recently been successfully utilised for various computer vision tasks that have previously been hitherto challenging to solve using more conventional methods, such as image style transfer [97, 98], super-resolution [101, 102, 313] and colourization [103]. As mentioned in Chapter 2, great strides have been made towards colour image completion using deep convolutional neural networks but the literature on using deep networks to solve the depth completion problem is very limited. In this chapter, we utilise a deep convolutional neural network trained on synthetic data [38] to complete depth images.

The material presented in this chapter of the thesis has been published in the following peer-reviewed publication:

- **A. Atapour-Abarghouei**, S. Akcay, G. P. de La Garanderie and T. P. Breckon, 'Generative Adversarial Framework for Depth Filling via Wasserstein Metric, Cosine Transform and Domain Transfer'. In Pattern Recognition, 2019 [34].

Since the approach we propose in this chapter is expected to synthesise large portions of depth, it has to adapt to learning image structures and semantics. In the existing body of work on learning-based colour image completion [104, 106, 107], training requires large datasets. The complete RGB image (widely available in various extremely large public datasets) is often considered as the ground truth and the input is created by adding noise or sparse corruptions [106], removing rectangular blocks [104, 106, 107], or cutting random regions from the image [104]. In the context of depth completion, however, no datasets exist that contain large quantities of naturally-sensed ground truth (hole-free) depth. Consequently,

we choose to utilise synthetic data acquired from a graphically rendered virtual environment primarily designed for a gaming application [38].

Since depth holes are neither random nor manually created, they are *predictable*, in that they occur due to specific scene features or the hardware device used for capture. For instance, featureless surfaces such as blank walls and roads, reflective objects and depth discontinuities, among others can cause depth holes. As a result of this *predictability*, the location of a hole occurrence can be learned via a separate model trained to predict where holes would be in a depth image based on the features present in the scene and the assumption of a specific capture approach.

When high-quality ground truth exists, a model can be naively trained based on a simplistic reconstruction loss (L_1 or L_2). However, due to the multi-modality of image completion, a model trained in this way tends to generate the average of the multiple possible modes in the predictions, which results in an output containing blurring effects. This is why the techniques in [104, 106, 107] and other generative models [314, 315] leverage adversarial training [105] as this assists with mode selection to generate realistic results. However, approaches using Generative Adversarial Networks (GAN) [104–107] suffer from certain flaws such as unstable training, difficulties in reaching an equilibrium and vanishing gradients due to premature discriminator optimality and other issues [316–318]. Here, we utilise an improved adversarial framework [317] that avoids such issues.

Even though an adversarial loss can help diffuse blurring effects, the goal of the adversary should be generating a more realistic image across the board and blurring artefacts still occasionally make their way to the output. This is because the generator feels safer averaging than selecting values. To ease the burden of de-blurring on the adversary, we propose the addition of a loss term based on the Discrete Cosine Transform (DCT) in addition to the conventional L_1 loss. The DCT preserves an accurate representation of the image structure in its spatial frequency content, which is why it has long been used in de-blurring [319], compression [320] and alike. We utilise the absolute deviations loss (L_1) in the frequency domain, as this error is far more obvious when the DCT is applied to a blurry averaged image. As seen in Figure 5.1, the L_1 distance between the original image and the blurry image, both in the spatial domain, is not very large but when the same images are transformed into the frequency domain using the DCT, the L_1 error is much larger and therefore a better indicator of blurring effects.

Our generator network, which is primarily designed to create the completed depth images, carries out its task in two stages: reducing the input into a compact representation of itself in the feature space (encoding) and reconstructing the image from these compact features (decoding). Up-convolutions, of any kind, are fraught with intrinsic unpredictability and can lead to bad salient edges and absence of fine texture. As a result, ensuring that the reconstruction starts from a correct and viable feature representation is paramount. We use the feature representations produced in the generator bottleneck in our loss to make

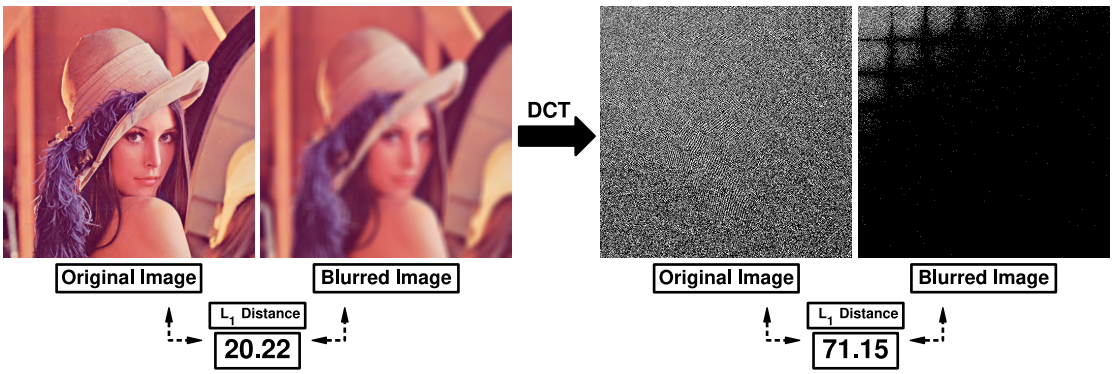

Figure 5.1: A demonstration of how the DCT makes the absolute deviations loss more susceptible to blurring. Note that the L_1 distance between the images in the frequency domain is higher and therefore a great tool to identify blurring.

sure the scene representation is correctly captured before reconstruction. While the sole use of this as a loss function is inadvisable and can lead to high-frequency artefacts, it is a helpful complement to the reconstruction and adversarial losses.

Our approach is meant to fill holes in depth images acquired via commercially and computationally inexpensive tools (a stereo camera and established stereo correspondence approaches such as [12, 321]) and not in pixel-perfect synthetic depth images only. Therefore, as part of our training procedure, it is vital to guide the model toward capturing the distribution of the natural data. With this in mind, a domain adaptation network is trained within the framework to rectify the model such that real-world images can be viable inputs during inference.

A major component of our model heavily depends on successful adversarial training to produce high fidelity depth outputs. Generative Adversarial Networks (GAN) are capable of producing semantically sound samples by creating a competition between a generator (G), which endeavours to capture the data distribution and a discriminator (D), which judges the generator output and penalises unrealistic images. Both networks are trained simultaneously to achieve an equilibrium. More formally put, this competition follows the minimax objective [105]:

$$\min_G \max_D \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r} [log(D(x))] + \mathop{\mathbb{E}}_{\tilde{x} \sim \mathbb{P}_g} [log(1 - D(\tilde{x}))], \tag{5.1}$$

where $\mathbb{P}_r$ is the data distribution, $\mathbb{P}_g$ is the model distribution defined by $\tilde{x} = G(z), z \sim p(z)$, with z being the random noise vector used as the generator input.

Training a GAN is rife with instability and potential issues [318], one of which is that the discriminator can rapidly reach optimality and easily distinguish between generator outputs and samples from the real distribution and hence, will not produce meaningful gradients for training. In [316], the Earth Mover's distance (EM) or Wasserstein-1 metric is used to measure the distance between two distributions. The EM distance, $EM(p, q)$, is the minimum cost of moving distribution elements (earth mass) to transform a distribution q to distribution p (cost = mass $\times$ transport distance) and the Wasserstein GAN [316] has an aptly named *critic* (C) instead of the conventional *discriminator* since it is no longer a classifier. Using the EM distance, the critic will not solely judge whether a sample is fake or real as a discrete binary decision but how real or how fake the generated sample is as a continuous regressive output. The critic will converge to a linear function with ever-present meaningful gradients and cannot saturate. The loss in the Wasserstein GAN is created via the Kantorovich-Rubinstein duality [316]:

$$\min_{G} \max_{C \in \mathcal{F}} \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r} [C(x)] - \mathop{\mathbb{E}}_{\tilde{x} \sim \mathbb{P}_g} [C(\tilde{x})], \tag{5.2}$$

where $\mathcal{F}$ is the set of 1-Lipschitz functions, $\mathbb{P}_r$ the true distribution, $\mathbb{P}_g$ the model distribution defined by $\tilde{x} = G(z), z \sim p(z)$ and z random noise. If C is optimal, minimising the value function with respect to G minimises $EM(\mathbb{P}_r, \mathbb{P}_g)$.

The Wasserstein GAN does not suffer from vanishing gradients and is immune to mode collapse. However, to guarantee continuity, a Lipschitz constraint must be enforced, which is achieved in [316] by clamping the weights. This creates a new clamping hyper-parameter, which needs to be carefully tuned to the distribution. A gradient norm penalty with respect to the critic input is proposed in [317] to replace clamping. Since a differentiable function is 1-Lipschitz if and only if its gradient norm is no more than 1 everywhere, [317] limits the critic gradient norm by penalising the function on the gradient norm for samples $\hat{x} \sim \mathbb{P}_{\hat{x}}$, where $\hat{x} = \epsilon x + (1 - \epsilon)\tilde{x},\ 0 < \epsilon < 1$. The new loss is therefore as follows [317]:

$$\min_{G} \max_{C} \mathop{\mathbb{E}}_{\tilde{x} \sim \mathbb{P}_g} [C(\tilde{x})] - \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r} [C(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} [(||\nabla_{\hat{x}} C(\hat{x})||_2 - 1)^2], \tag{5.3}$$

where $\mathbb{P}_g$ is the model distribution defined by $\tilde{x} = G(z), z \sim p(z)$, with z being the random noise vector, $\mathbb{P}_r$ is the true data distribution and $\mathbb{P}_{\hat{x}}$ is implicitly defined to sample uniformly along straight lines between pairs of points sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$ [317]. Here, we use the same critic for our adversarial loss.

Our model is trained on a *synthetic* dataset of RGB-D images to perform depth completion. However, due to domain bias [219], a model trained using data from a specific domain does not necessarily generalise to

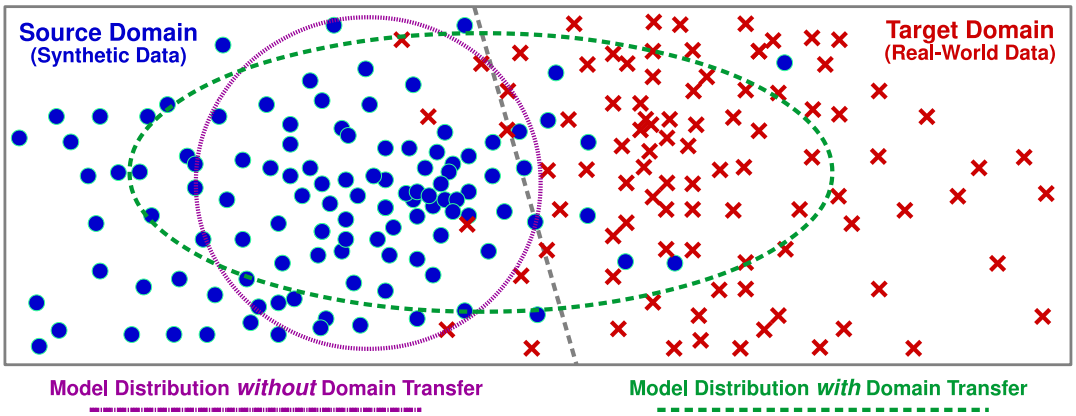

Figure 5.2: A demonstration of modelling two separate data domain distributions via domain adaptation. A model only trained on samples from the source domain (blue) can capture the target data distribution (red) using domain adaptation.

other data domains. In other words, a model trained on *synthetic* data may not perform well on *real-world* data. Therefore, our model may not succeed with naturally obtained depth images, which would make it utterly useless from a practical standpoint.

While the typical solution to this problem is to fine-tune the network on the novel data, fitting the large number of parameters in a deep network to a new dataset requires a large amount of data, which can be very time-consuming, expensive, or intractable to obtain. This is often the reason why synthetic data is used in the first place, as it is in our case. One strategy often used to solve this problem is to minimise the distance between the source and target feature distributions [225, 226]. Figure 5.2 demonstrates how domain adaptation can aid in modelling the distribution of both the source domain (represented in blue), used for training the model and the target domain (represented in red), which is the focus of the final objective. Using domain adaptation, both distributions can be captured within the model even if the model is only trained on one of them.

Some approaches have taken advantage of MMD (maximum mean discrepancy), which calculates the norm of the distance between the domains to reduce the discrepancy [228], whereas others have taken to using adversarial training which leads to a representation that minimises the domain discrepancy while able to discriminate the source labels easily [226]. Although most of the above-mentioned techniques focus on discriminative models, research concentrating on generative tasks has also utilised domain adaptation [229].

We propose a domain critic network, which uses the Wasserstein metric to measure the distance between the source (synthetic data) and the target (real-world data) and minimises this difference by comparing

the generator outputs when synthetic and real-world images are used as the input, while the generator is simultaneously trained to fill synthetic holes using synthetic ground truth. Further details of the inner-workings of the proposed approach are explained in Section 5.2.

5.1 Data Preparation

In a supervised learning approach, such as ours, ground truth labels are required during training. Since the objective here is to fill depth holes, ground truth hole-free depth is required. However, obtaining complete depth from the real world is not practically possible. Consequently, we use synthetic data acquired from a graphically rendered gaming environment focusing on driving scenarios, akin to [38].

Necessary steps were taken to avoid dataset bias. Co-registered colour and depth images are captured from a camera view set in front of a virtual car as it automatically drives. An image is captured every 60 frames as the height and field of view of the camera are randomly changed after every capture. The process is carried out in numerous weather and lighting conditions at different times of day to avoid any possible model over-fitting. A total of 130,000 images were captured with 100,000 used for training and 30,000 set aside for testing.

During training, depth images are used as ground truth but corrupted depth images (with holes) of the same scenes are required as inputs. Rather than randomly cutting out sections of the image, we opt for creating realistic holes with the characteristics of those found in real-world depth images, which occur in stereo correspondence due to the existence of featureless or shiny surfaces, unclear object separation and distant objects, among others. To produce these semantically meaningful holes, a separate model is needed to predict depth holes by means of pixel-wise classification. The objective is to produce a hole mask, which represents regions in the depth image likely to contain holes. Since within our synthetic dataset, only complete pixel-perfect depth is available, simulating corrupted depth, similar to what is naturally sensed in the real world, is important.

5.1.1 Hole Prediction

Our hole prediction model is a fully-convolutional encoder-decoder network inspired by [277, 307] with nine convolutional layers in both the encoder and the decoder. No fully-connected layers are used to maintain a smaller number of network parameters and therefore, easier and faster training and inference. Every decoder layer corresponds to an encoder layer, with the last decoder layer connecting to a soft-max classifier. Each convolutional layer is followed by batch normalisation [322] and a ReLU. Max-pooling is used in the sub-sampling to produce features that are invariant to small translational shifts in the input. In the decoder, max-unpooling [277] (which uses the recorded locations of maxima within the region of

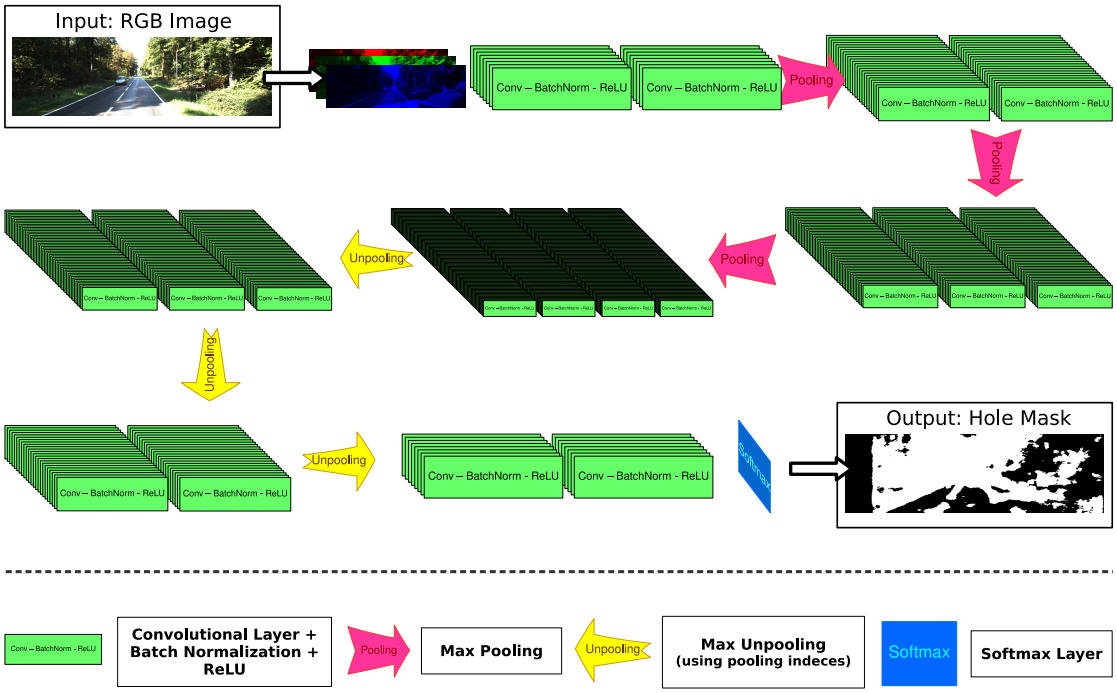

Figure 5.3: An overview of the network architecture used during the hole prediction stage.

each max-pooling operation) is applied to preserve the feature structure and boundary information. The network architecture is seen in Figure 5.3.

A number of stereo images (40,000) from the KITTI dataset [301] was used to train the network by estimating the disparity via Semi-Global Matching (SGM) [12] and generating a hole mask (M) which indicates which pixels are holes, i.e. regions for which disparity was not recovered (with a value of zero) and which are non-holes (with a value of one). Although SGM is used, this is interchangeable with any depth via disparity or active depth capture approach. The left RGB images are used as inputs with the generated masks as ground truths. Cross-entropy is used as the loss function with the network weights randomly initialised.

Our hole prediction process is self-supervised, meaning no human annotation or intervention is necessary at any point, with the ground truth calculated using a disparity estimation approach [12]. Although this makes the ground truth for hole prediction unreliable, consequently making any accurate quantitative analyses meaningless, it suits the purposes of this endeavour.

Qualitative evaluations reveal that holes are predicted where expected. From Figure 5.4, we see that in regions where camera overlap is absent or featureless surfaces, sparse shrubbery, unclear object boundaries and very distant objects are present, such pixels are correctly classified as holes. This model is

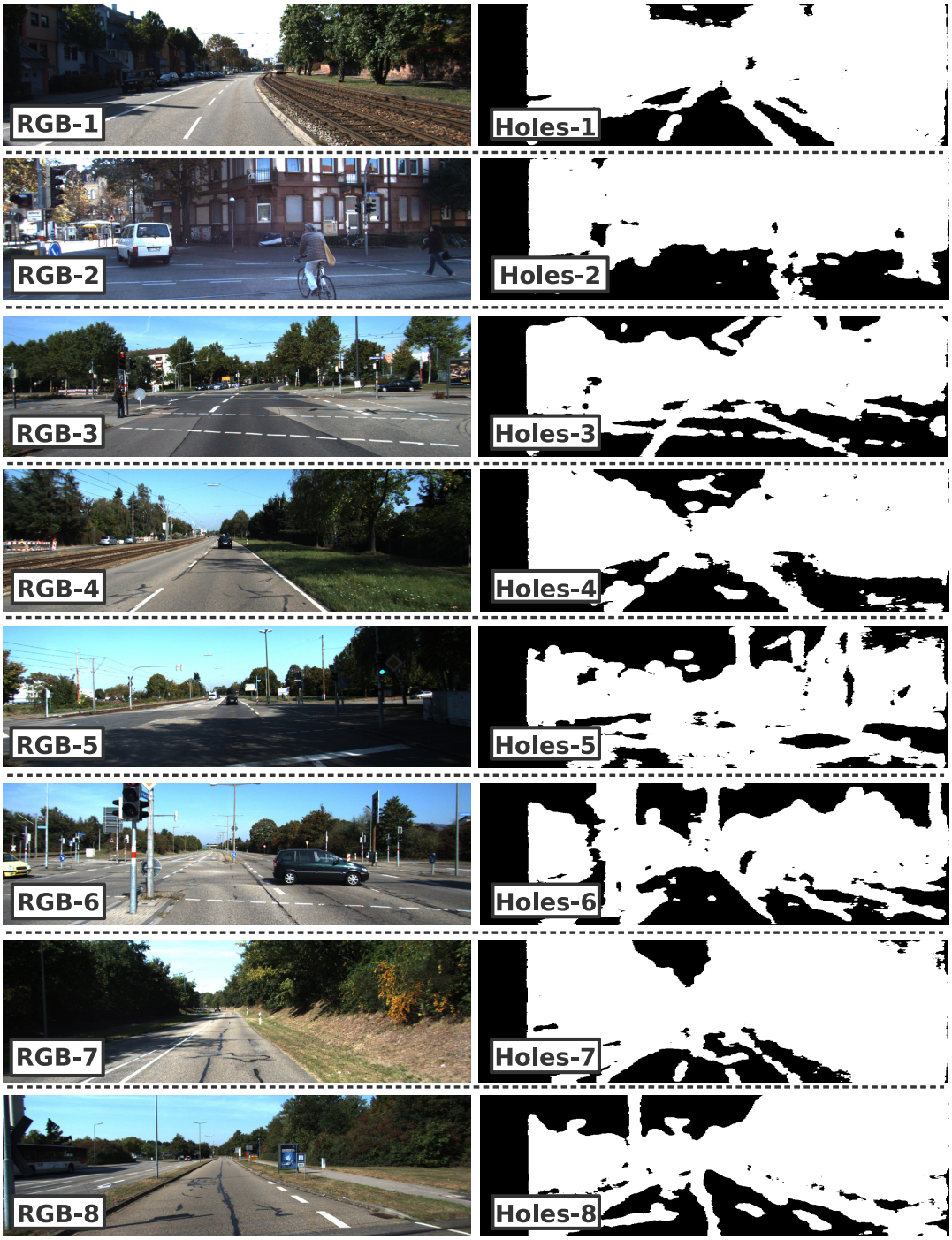

Figure 5.4: Examples of the hole prediction model applied to a test set of 5,000 images (RGB) from [301]. Note that featureless surfaces and sky are correctly identified as holes in the outputs (Holes).

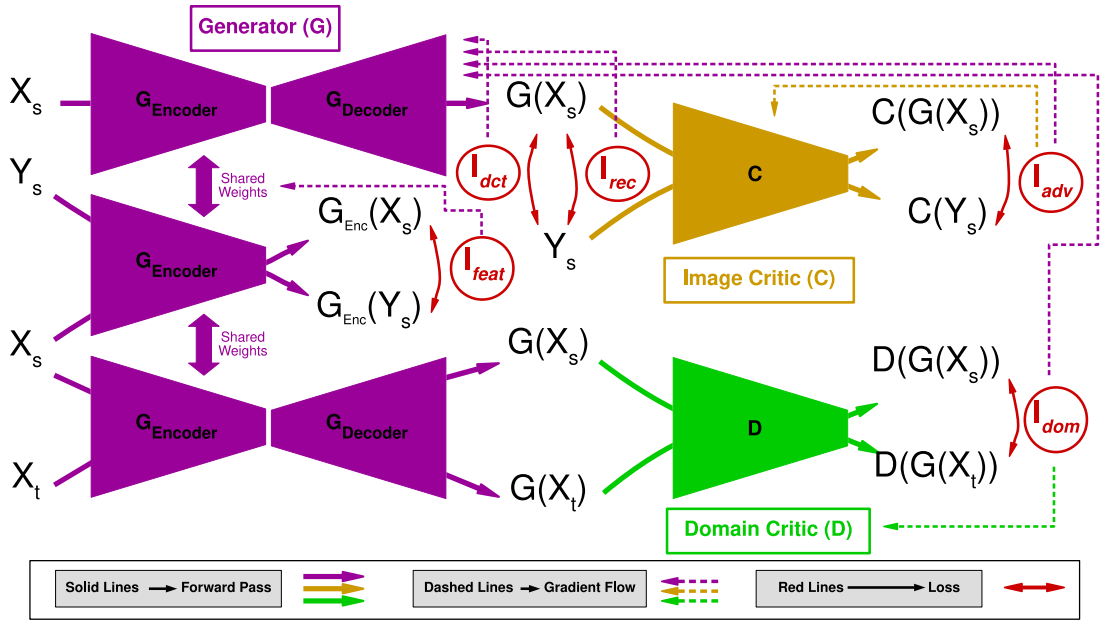

Figure 5.5: The general framework of the entire model. The pipeline contains a generator (the only network used during inference, as explained in Section 5.2.1), an image critic to ensure the high fidelity of the generated depth (Section 5.2.2) and a domain critic to enforce generalisation over real-world data (Section 5.2.2). Loss functions and gradient flows are shown for all components. G represents the generator network, C the image critic and D the domain critic network. t and s respectively refer to the target (real-world data) and source (synthetic data) domains.

subsequently used to infer where the holes would be in the hole-free ground truth synthetic RGB-D images needed for training the hole filling model.

5.2 Details of the Approach

Taking advantage of the adversarial training procedures present within the literature [316, 317], our process involves three networks: a generator (G) which follows an encoder-decoder pipeline and is tasked with generating the completed depth, an image critic (C) which judges the generator output in an adversarial fashion and a domain adaptation network (D) which provides the possibility of applying the model (trained on synthetic data) to natural images without ground truth depth.

While the image critic (C) ensures the higher fidelity of the overall depth output, the domain adaptation network (D) guarantees that the generator network (G) is encouraged to perform well on the real-world test data as well as the synthetic training data. The generator produces the complete depth output by

encapsulating the underlying distribution of the data. As a result, it would be very easy for G to overfit to the synthetic domain if synthetic training images are all it ever sees. Consequently, D is used to push the generator to capture the distribution of the target domain (real-world data - denoted by red in Figure 5.2) along with that of the source domain (synthetic training data - represented in blur in Figure 5.2).

The interactions between the networks are demonstrated in Figure 5.5. An overview of the architecture of the networks is seen in Figure 5.6. Let `CRnSa-k` denote an $n \times n$ Convolution-BatchNorm-ReLU with k filters and a stride of a. Similarly, `CLnSa-k` represents an $n \times n$ Convolution-LeakyReLU with k filters and a stride of a, `MP` a max-pooling operation with a kernel size of 2 and a stride of 2 and `MU` a max-unpooling operation with similar parameters. Consequently, the generator architecture is made up of:

`CR3S1-16, CR3S1-16, MP, CR3S1-32, CR3S1-64, MP, CR3S1-128, CR3S1-128, CR3S1-128, MP, CR3S1-256, CR3S1-256, CR3S1-256, CR3S1-256, MU, CR3S1-128, CR3S1-128, CR3S1-128, MU, CR3S1-64, CR3S1-32, MU, CR3S1-16, CR3S1-1`

The image critic and the domain critic networks consist of:

`CL4S2-64, CL4S2-128, CL3S1-128, CL4S2-256, CL7S3-512, CL(3,6)S(1,3)-1024, CL4S1-1`

In the following, details regarding the hole filling process are briefly outlined.

5.2.1 Missing Depth Prediction

Depth completion is performed by a generator with an encoder-decoder pipeline (the only network used during inference). A synthetic 4-channel RGB-D image containing holes (predicted by the model discussed in Section 5.1.1) is used as the encoder input, which creates a compact set of feature representations. This set of feature representations is then passed through the decoder, creating a single channel depth image with the missing regions filled if necessary (exceptions being very distant objects and sky, for which no valid depth should or does exist).

For the sake of consistency, the same architecture (Figure 5.6) is used for both the hole prediction network (Section 5.1.1) and the generator. Since the goal is to test the learning capabilities of the model, the weights are randomly initialised and training procedure commences from scratch. The network is fully-convolutional with nine convolutional layers, batch normalisation and max-pooling operations (Figure 5.6). A large feature map of $78 \times 24 \times 256$ is produced in the bottleneck. Many past works [104, 106, 107] advocate sub-sampling the image down to a small feature map passed through a fully-connected layer to allow for '*entire image context reasoning for each unit*' [104]. We experimented with fully-connected layers but other than a significant increase in the number of parameters, training difficulty and

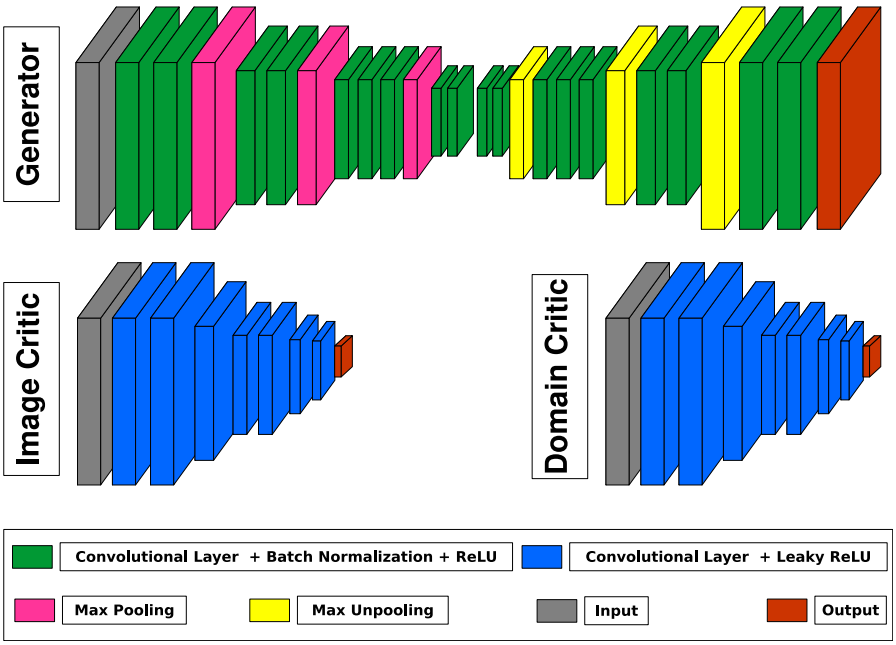

Figure 5.6: Network architectures (generator, image critic and domain critic) used in the training.

inference time, no noticeable difference in the quality of the results was observed. This means direct connections between different regions of a single feature map is not necessary for this task.

During inference, after the generator outputs the completed depth image, the regions filled by the network are blended using the approach in [323] into the hole regions within the original hole-ridden input image to create the final results. The approach in [323] follows a Laplacian pyramid framework, in which different levels of the composite pyramid obtained via independent mixing of the source at different resolutions are summed to achieve the final blended output after carefully-designed mixing procedures for different frequency bands. In is important to note that while we have opted for this approach [323], any other blending or mixing approach with visually satisfactory results would suffice.

5.2.2 Loss Function

Our resulting model performs depth completion by regressing to the ground truth depth content of the unknown regions. Using a reconstruction loss, we ensure the filled regions are contextually sound, coherent and in accordance with the known regions. The addition of an adversarial loss [317] results in plausible outputs since the adversarial framework will enforce mode selection. To ensure robust contextual feature extraction and better encoder training, the distance is measured between the feature

maps produced in the generator bottleneck when the depth channel of the input contains holes and when the ground depth is used as the input. The generator is then trained to minimise this distance. This guarantees correct and balanced training of the encoder and the decoder within the generator.

Additionally, even with perfect training, a network trained on synthetic data cannot be expected to perform equally well on naturally sensed depth images. A domain adaptation loss is consequently used to ensure that our approach can complete naturally sensed depth images. A joint loss function is thus formulated consisting of four components:- reconstruction loss (Section 5.2.2), adversarial loss (Section 5.2.2), bottleneck feature loss (Section 5.2.2) and domain adaptation loss (Section 5.2.2) - each of which are subsequently detailed.

Reconstruction Loss

To maintain structural continuity and semantic coherence in the output, a reconstruction error against the ground truth is needed. However, to achieve sharper and more crisp results and to ease the burden on the adversarial image critic to enforce realism, we utilise a two-term reconstruction loss. Given a ground truth depth y, our generator (G) takes an input x, which itself is created based on y and generates $G(x)$. In this context, our hole prediction model (Section 5.1.1) has produced a binary hole mask, M, in which 0 denotes an unknown region (hole) and 1 a known depth region. The generator input, x, is obtained as follows:

$$x = y \odot M, \tag{5.4}$$

where $\odot$ is the element-wise product operation. We use a masked L_1 distance as part of our reconstruction loss:

$$\mathcal{L}_{rec-L_1} = ||(1 - M) \odot G(x) - (1 - M) \odot y||_1 \tag{5.5}$$

Experiments with L_2 loss returned the same results. With the known issues of a reconstruction loss, blurry images are often produced, which is why the use of adversarial losses is prevalent. However, here we add another term to our loss to partly alleviate the issue of blurring. Since the Discrete Cosine Transform (DCT) can be used to encode a unique embedding of spatial image structure, avoiding the limitations of L_1 pixel space embedding (Figure 5.1), the entire (unmasked) generated output $G(x)$ and the ground truth depth y both undergo the transform and the distance is measured within the projected DCT space:

$$\mathcal{L}_{rec-dct} = ||DCT(G(x)) - DCT(y)||_1 \tag{5.6}$$

The final reconstruction loss used in this work is therefore:

$$\mathcal{L}_{rec} = \mathcal{L}_{rec-L_1} + \mathcal{L}_{rec-dct} \tag{5.7}$$

Although the addition of the $\mathcal{L}_{rec-dct}$ reduces blurring, the overall quality of the output is still subject to issues due to the generality of the L_1 distance, ensuring an adversarial component is subsequently needed.

Adversarial Loss

Unlike most generative models, our network is conditioned on the known regions of the depth and the entire RGB and is tasked with generating the full depth. Our generator approximates a function which maps samples from the noisy distribution x to the true data distribution y, $G : x \mapsto y$. No noise or drop-out is used and the image critic is not conditioned like the generator and sees the entire generator output, such that it cannot take advantage of structural discontinuities or possible differences in the overall intensities within the depth in its judgement.
Therefore, it improves the whole generator output and not just the missing regions. The objective of the image critic is hence measuring the difference (using the EM metric) between real data samples and generated ones. Given that $\tilde{y} = G(x)$, the objective function of the critic is:

$$\min_G \max_C \mathop{\mathbb{E}}_{\tilde{y}\sim\mathbb{P}_g}[C(\tilde{y})] - \mathop{\mathbb{E}}_{y\sim\mathbb{P}_r}[C(y)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(||\nabla_{\hat{x}} C(\hat{x})||_2 - 1)^2], \tag{5.8}$$

where $\mathbb{P}_g$ is the model distribution defined by $\tilde{y} = G(x)$, with x being the generator input sampled from the noisy distribution, $\mathbb{P}_r$ is the true data distribution and $\mathbb{P}_{\hat{x}}$ is implicitly defined to sample uniformly along straight lines between pairs of points sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$ [317]. The generator objective is to fool the image critic by creating increasingly more realistic outputs and getting closer to the true data distribution. The adversarial loss is thus as follows:

$$\mathcal{L}_{adv} = \max_C - \mathop{\mathbb{E}}_{\tilde{y}\sim\mathbb{P}_g}[C(\tilde{y})], \tag{5.9}$$

where once again, $\mathbb{P}_g$ is the model distribution defined by $\tilde{y} = G(x)$, with x being the generator input sampled from the noisy distribution. The generator and the image critic are trained iteratively while the critic is kept optimal at all times (in each epoch, it is trained 25 times per each generator training iteration for the first 100 generator iterations and 5 times per each generator iteration for the rest of the training process). The critic is a fully-convolutional network with no batch normalisation. An overview of its architecture is seen in Figure 5.6 (image critic).

Bottleneck Feature Loss

In a typical convolutional encoder-decoder pipeline [104, 277], the convolutional layers in the encoder and the decoder learn independently. This can be advantageous as it provides a wide learning domain for the network. However, convergence to optimality can be slow and difficult.

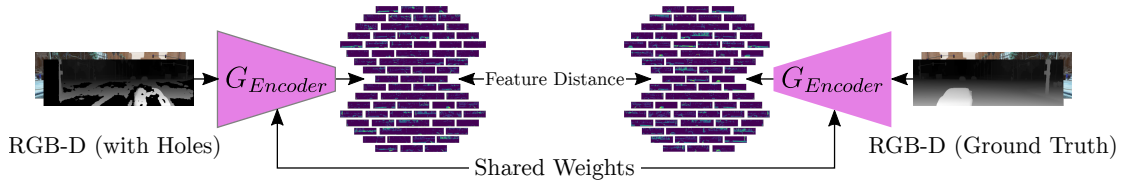

Figure 5.7: A demonstration of how the bottleneck feature distance is calculated. Depth with holes (left) and ground truth depth (right) are used as inputs to the generator encoder. minimising the absolute difference between the feature maps extracted from the bottleneck is part of the objective.

Since the generator needs to predict any missing depth based on the RGB view and known depth regions, we can improve the generator training by making sure the encoder is creating the right feature representation of the entire scene and the decoder is, in turn, starting from the best set of feature maps to produce the output.

Using the ground truth depth as the input and comparing the generated bottleneck features with the features produced from the regular input (depth with holes), we can guarantee the encoder is rightly trained to capture the full information available in the scene based on context and inferred geometry rather than local low-level scene features. As Figure 5.7 demonstrates, the ground truth depth is used as the input for the generator (right side of the figure) and the depth image with holes is also used as the input (left side of the figure). The distance between such features extracted from the generator bottleneck is then used as a component of the loss. Subsequently, our loss includes the distance between the generated bottleneck features from the ground truth and the noisy input:

$$\mathcal{L}_{feat} = ||G_{encoder}(x) - G_{encoder}(y)||_1 \tag{5.10}$$

In all previous loss terms, x as the input to the generator was a 4-channel RGB-D image with the depth channel containing holes and y a single-channel hole-free depth image. In Equation 5.10, however, y is also a 4-channel RGB-D image but the depth channel is the ground truth depth (hole-free). For the sake of consistency, the same notation is used in Equation 5.10.

Domain Adaptation Loss

All the training data used here are synthetic images, yet for the model to be practically viable, it has to perform on real-world images. Since no naturally-obtained ground truth is available for training, the generator is also trained to recognize natural data in an adversarial fashion (similar to Section 5.2.2).

Let all synthetic inputs (source domain) be denoted by x_s with synthetic ground truth y_s. All naturally-obtained data (target domain) are denoted by x_t. Note that there is *no* y_t since our target domain

(naturally-sensed images) has no ground truth (hole-free) depth. A domain critic network (D) is used to measure the difference (in EM distance) between the generator output when the input is sampled from the source domain (x_s) and when the input is from the target domain x_t. The gradients will be used to train the generator and the generator is subsequently forced to model the distribution of both the source and the target domains. Given that $\tilde{y_s} = G(x_s)$ and $\tilde{y_t} = G(x_t)$, the objective function of the domain critic is:

$$\min_{G} \max_{D} \mathop{\mathbb{E}}_{\tilde{y_t} \sim \mathbb{P}_t} [D(\tilde{y_t})] - \mathop{\mathbb{E}}_{y_s \sim \mathbb{P}_s} [D(\tilde{y_s})] + \lambda \mathop{\mathbb{E}}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} [(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2], \tag{5.11}$$

where $\mathbb{P}_t$ is the target model distribution defined by $\tilde{y_t} = G(x_t)$, with x_t being the generator input sampled from the natural data distribution, $\mathbb{P}_s$ is the source data distribution defined by $\tilde{y_s} = G(x_s)$, with x_s being the generator input sampled from the synthetic data distribution and $\mathbb{P}_{\hat{x}}$ is implicitly defined to sample uniformly along straight lines between pairs of points from $\mathbb{P}_t$ and $\mathbb{P}_s$ [317].

The objective of the generator is to fool the domain critic by approximating both domain distributions. The domain adaptation loss is as follows:

$$\mathcal{L}_{dom} = \max_{D} - \mathop{\mathbb{E}}_{\tilde{y_t} \sim \mathbb{P}_t} [D(\tilde{y_t})], \tag{5.12}$$

where $\mathbb{P}_t$ is the natural domain distribution defined by $\tilde{y_t} = G(x_t)$, with x_t being the generator input from natural data. The generator and the domain critic are trained iteratively while the domain critic is always kept optimal, much like the critic in Section 5.2.2. The domain critic architecture is the same as the image critic, as seen in Figure 5.6. Weight sharing between the image critic and the domain critic was attempted but we could not achieve convergence with that setup.

Synthetic ground truth y_s does not come into play in domain adaptation training and the model is trained on the source domain to approximate the data distribution (from which y_s is sampled). The domain adaptation loss thus forces the generator to comprehend both the natural and synthetic distributions. Additionally, over-training the model using domain adaptation leads to artefacts in the outputs. Thus, this term is only used in a quarter of the total number of epochs (Section 5.3.1). It is important to note that this loss component was originally used in the training objective of the hole prediction network (Section 5.1.1) as well but with no evidence for any significant improvement in the output.

Weighting Coefficients	Mean L_1 Error	Mean L_2 Error	PSNR (dB)	SSIM $(-1, 1)$
$\lambda_{rec} = 0.01, \lambda_{adv} = 0.01, \lambda_{feat} = 10.0, \lambda_{dom} = 100$	2.01	0.12	28.68	0885
$\lambda_{rec} = 0.01, \lambda_{adv} = 100, \lambda_{feat} = 100, \lambda_{dom} = 10.0$	2.09	0.13	28.63	0.882
$\lambda_{rec} = 0.01, \lambda_{adv} = 100, \lambda_{feat} = 100, \lambda_{dom} = 100$	2.13	0.15	28.51	0.879
$\lambda_{rec} = 100, \lambda_{adv} = 0.01, \lambda_{feat} = 1.00, \lambda_{dom} = 0.01$	1.83	0.09	20.82	0.919
$\lambda_{rec} = 100, \lambda_{adv} = 0.01, \lambda_{feat} = 10.0, \lambda_{dom} = 0.01$	1.82	0.09	31.85	0.921
$\lambda_{rec} = 100, \lambda_{adv} = 0.01, \lambda_{feat} = 0.01, \lambda_{dom} = 0.01$	**1.79**	**0.08**	**31.89**	**0.928**

Table 5.1: Numerical comparison of the coefficients weighting the loss components in the approach in Chapter 5. While disparity error values are lower for more realistic images, with Peak Signal-to-Noise Ratio (PNSR) and Structural Similarity index (SSIM), higher values are better. The three worst (top) and the three best (bottom) performing combinations of weights are included here for better analysis.

Joint Loss Function

Based on Equation 5.7 (Section 5.2.2), Equation 5.9 (Section 5.2.2), Equation 5.10 (Section 5.2.2) and Equation 5.12 (Section 5.2.2), our overall joint loss function is finally defined as:

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{feat}\mathcal{L}_{feat} + \lambda_{dom}\mathcal{L}_{dom} \tag{5.13}$$

The choice of the weights λ_{rec}, λ_{adv}, λ_{feat} and λ_{dom} is empirical.

5.3 Experimental Results

A total of 30,000 synthetic images were used as part of the test set. Moreover, a set of 5,000 locally-captured images consisting of RGB and registered depth containing holes were used for training as part of domain critic training and subsequently used to test the model on real-world natural images.

5.3.1 Implementation Details

All network implementation and training is done in *PyTorch* [324] and *Caffe* [325]. The Adam optimisation method [326] is used for this problem (momentum $\beta_1 = 0.5$, $\beta_2 = 0.999$ and initial learning rate $\alpha = 0.0001$) and the coefficients in the loss function are empirically chosen to be $\lambda_{rec} = 100, \lambda_{adv} = 0.01, \lambda_{feat} = 0.01, \lambda_{dom} = 0.01$ based on a preliminary grid search with coefficients changing an order of magnitude between 0.01 and 100. Since the images are normalized to values within the range [0,1] before they are passed into the model, the reconstruction loss, which is essentially the Euclidean distance between the output and the ground truth (both with values in the range [0,1]) would be very small.

Figure 5.8: Example of the results when different components of the loss function are added to the joint loss function. Note that the results of the full proposed approach (with all four components) are substantially superior. **A:** RGB image; **B:** depth image with holes; **C:** ground truth depth; **D:** approach using conventional GAN, L_1, and no RGB image; **E:** approach with L_1 loss ; **F:** approach with L_1 and dct loss; **G:** approach with L_1, dct and adv loss; **H:** full proposed approach.

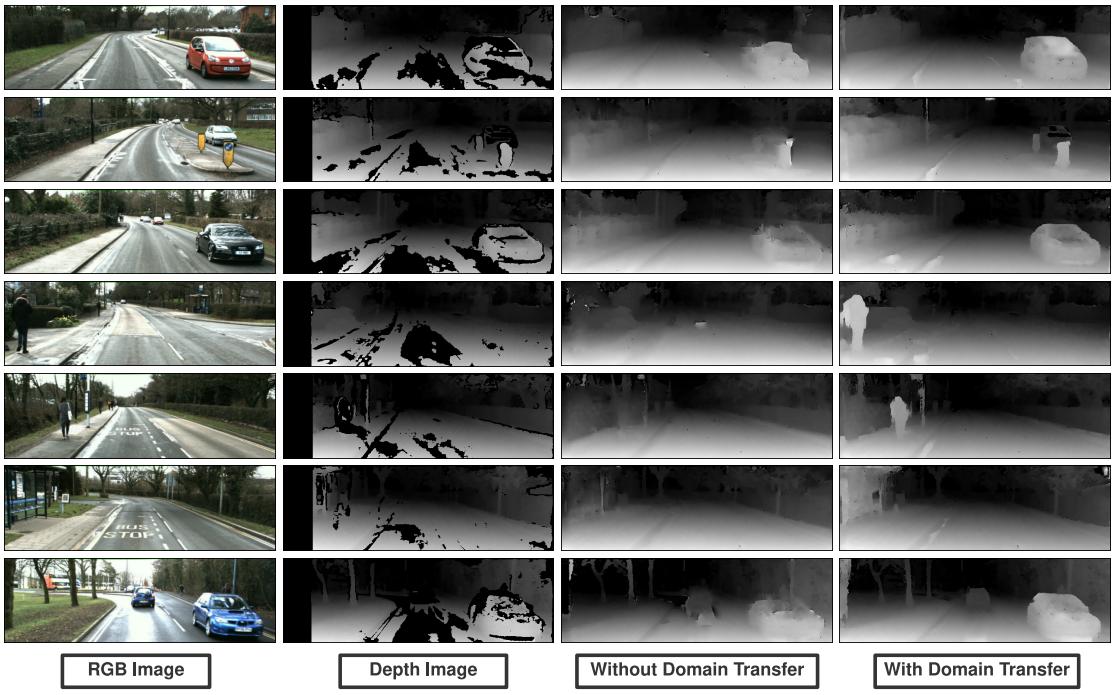

Figure 5.9: Results of our approach on real data with and without domain adaptation.

It is consequently expected that the best results could be achieved if all the loss components are scaled to have roughly similar contributions. It is important to note, however, that the approach is not very sensitive to the coefficients weighting the loss components, as the overall model can eventually learn its way around any imbalances between the loss components. This is demonstrated in Table 5.1, where the results of the best-performing and the worst-performing combinations of the coefficients are included. As seen in Table 5.1, despite the selected weights clearly outperforming all others, the overall difference between the best and the worst results is small, making the approach indifferent to an imbalance between the loss components, which would be expected from most learning-based methodologies. The networks used in our model are trained for 20 epochs over the entire dataset with a batch-size of 7 images. The domain adaptation loss, $\mathcal{L}_{dom}$, is used only every other epoch and only in the last 10 epochs to avoid introducing undesirable effects in the outputs.

5.3.2 Ablation Studies

A crucial part of this work was interpreting the necessity of the components of our loss function. The model was trained from random initialisation each time after adding a single loss component.

As seen in Figure 5.8, when a simple reconstruction loss (L_1) is solely used, large holes are ubiquitously filled with averaged blurry content (blue boxes in Figure 5.8). The addition of the DCT helps in alleviating the issue but blurring and unwanted artefacts still exist. The adversarial loss clears the image to a great extent but the use of full joint loss function (except of course the domain adaptation portion, which is only relevant to natural images) creates a sharp and realistic image, with minimal differences with the ground truth. The significant similarities seen between the final results and the ground truth is in part because the model is conditioned on the RGB view, as seen in Figure 5.8 (**D**) where the RGB is not used in the training.

Not only are the images realistic to the human eye, the quantitative results in Table 5.3 demonstrate that our results are clearly superior to the prior works of [125], [66], [68] and [41]. As seen in Table 5.3, we use four metrics to compare the results against the ground truth (mean absolute difference, mean squared difference, peak signal-to-noise ratio and the structural similarity index). Overall, Table 5.3 shows a significant reduction in prediction errors of the proposed approach against ground truth with negligible standard deviation indicating consistent performance over the randomly selected test set of 30,000 synthetic images.

5.3.3 Evaluation using Non-Synthetic Natural Data

The last component of our loss fits the model to naturally-sensed images as well as the synthetic data. The effectiveness of this loss term is demonstrated in Figure 5.9. Without data domain adaptation, the network is incapable of producing valid and meaningful results. The adversarial nature of our domain adaptation can result in the generator attempting to produce pixel-perfect depth images when a real-world image is given as its input and therefore removes entire objects or synthesises ones that should not be in the scene. This is primarily due to variations between the contents of the scenes in the images from the synthetic and real-world domains. Since the depth estimation portion of the approach is only trained using synthetic data, when scene components that do not exist in synthetic images appear in real-world images, an overly-trained domain adaptation pipeline may encourage the generator to synthesise the depth information for the non-existant objects or vice versa.

Method	FBI [37]	EBI [66]	SSI [125]	GIF [41]	FMM [68]	Our Approach
Run-Time (ms)	$>36 \times 10^5$	$>12 \times 10^5$	33.4×10^3	14.32×10^2	82.8×10^1	7.47×10^0

Table 5.2: Comparing the run-time of our approach with conventional depth completion. Note that only requiring a single forward pass, our approach is highly efficient using modern hardware.

Method	Mean L_1 Error	Mean L_2 Error	PSNR (dB)	SSIM $(-1, 1)$
SSI [125]	5.66 ± 1.033	2.96 ± 0.512	16.04 ± 4.819	0.772 ± 0.220
FMM [68]	2.85 ± 0.491	0.89 ± 0.198	20.66 ± 2.030	0.780 ± 0.082
EBI [66]	2.39 ± 0.629	0.92 ± 0.091	20.65 ± 3.122	0.787 ± 0.139
GIF [41]	2.77 ± 0.518	0.86 ± 0.068	20.78 ± 1.910	0.764 ± 0.125
FBI [37]	2.36 ± 0.602	0.91 ± 0.105	20.67 ± 2.891	0.788 ± 0.106
L_1 Loss Only	2.96 ± 0.489	0.28 ± 0.038	25.99 ± 2.890	0.819 ± 0.112
$L_1 + dct$ Loss	2.47 ± 0.422	0.19 ± 0.047	27.98 ± 2.019	0.872 ± 0.132
$L_1 + dct + adv$ Loss	2.33 ± 0.405	0.17 ± 0.050	28.50 ± 1.105	0.882 ± 0.096
CE [104] ($L_2 + adv$ Loss)	2.18 ± 0.391	0.18 ± 0.034	28.21 ± 1.359	0.877 ± 0.108
Full Proposed Approach	**1.79** ± 0.401	**0.08** ± 0.011	**31.89** ± 2.012	**0.928** ± 0.110

Table 5.3: Numerical comparison of our approach, our ablated method and other techniques, such as the Fourier-based inpainting (FBI) method proposed in Section 3.1, smoothing second order inpainting [125] (SSI), exemplar-based inpainting [66], fast marching method [68] (FFM), guided inpainting and filtering technique [41] (GIF). Disparity error values are lower for more realistic images. For Peak Signal-to-Noise Ratio (PNSR) and Structural Similarity index (SSIM), higher values are better.

To remedy this issue, the domain adaptation loss is only used in a quarter (1 in 4) of all training epochs. This balanced training procedure will push the generator to fill the real-world depth images that are of significantly lower quality than the synthetic ones in a realistic and consistent manner, while at the same time, scene integrity is not compromised by the domain adaptation component of the network. As Figure 5.9 demonstrates, naturally-sensed depth images are filled in a more plausible manner with domain adaptation as part of the training (Figure 5.9 - fourth column) than with no domain adaptation (Figure 5.9 - third column).

5.3.4 Comparison to Contemporary Approaches

The approach is also evaluated against classical hole filling techniques. We used both synthetic and natural images to test the performance and since ground truth depth is available for the synthetic data, numerical analysis is possible in the evaluation. The Fourier-based inpainting approach (FBI) proposed in Section 3.1, the smoothing second order inpainting [125] (SSI), diffusion-based exemplar filling (DEF) [66], fast marching method [68] (FMM) and a guided inpainting and filtering technique [41] (GIF) are chosen for their accuracy and their capability of handling relatively large holes.

As indicated in Table 5.3, our approach outperforms the comparators by a large margin, even if the loss is stripped down to a simple reconstruction loss.

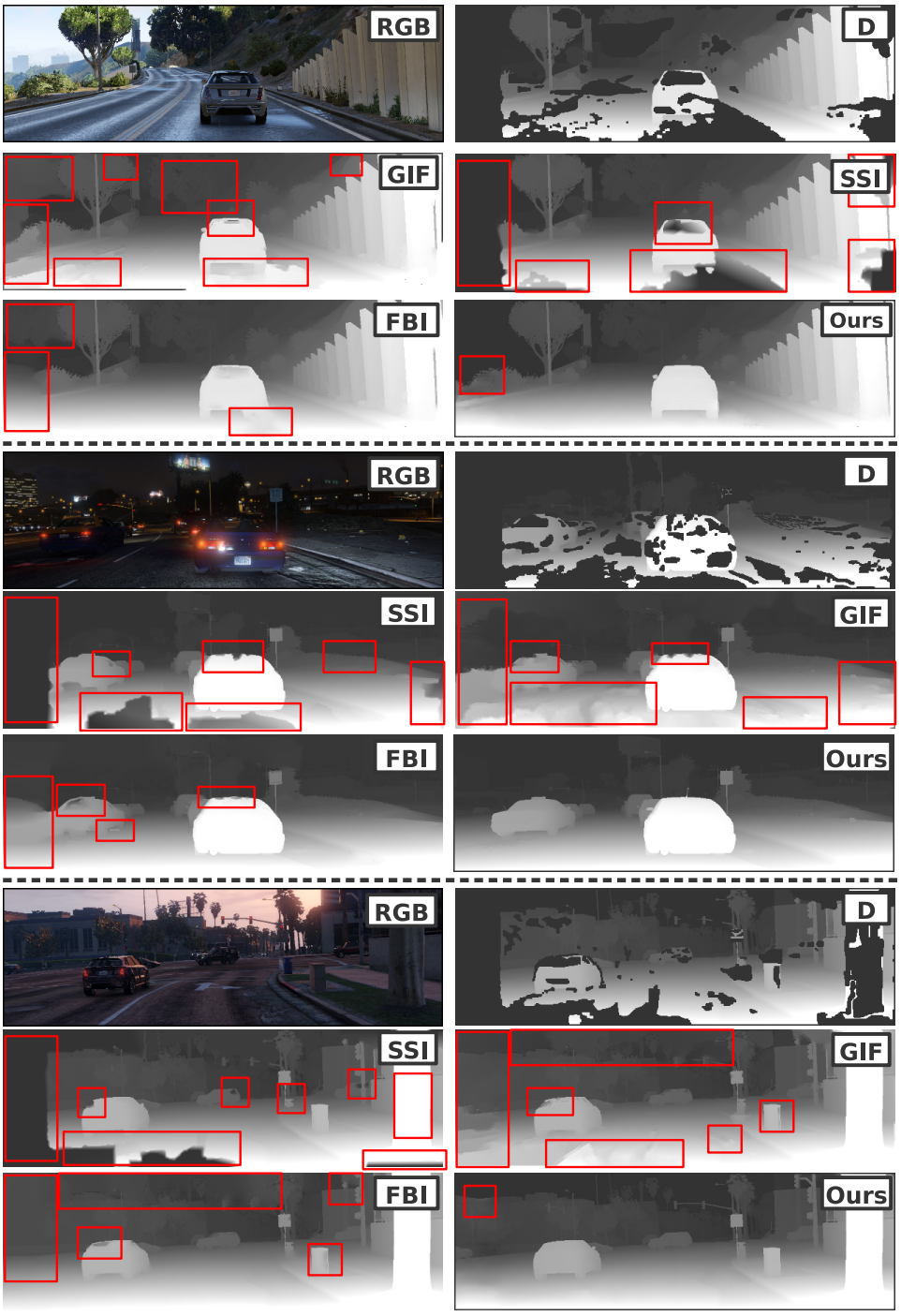

Figure 5.10: Comparing the results of our approach with the depth inpainting approach using second-order smoothness prior (SSI) [125], guided inpainting and filtering (GIF) [41] and the Fourier-based inpainting method proposed in Section 3.1 (FBI) using synthetic test data. Anomalies and undesirable artefacts are marked in red.

Since the synthetic images are of extremely high quality (pixel-perfect dense depth information with granular texture and accurate object boundaries), they should be prime candidates for traditional hole filling methods. However, since learning the semantics, structures and the context of a scene plays a vital role in predicting its contents, our approach produces more realistic results with almost no anomalies, blurring or any undesirable artefacts, as seen in Figure 5.10. Based on Figure 5.11, similar conclusions can be drawn when it comes to natural real-world images, where the depth is of significantly lower quality compared to synthetic data. The capabilities of our approach over real-world data are owed to the domain adaptation component of our loss function (Section 5.3.3).

Regarding efficiency, the runtime of our approach heavily depends on the hardware. All training and inference were done using an NVIDIA GeForce GTX 1080 Ti GPU and our mean inference time (requiring a single forward pass) is 7.47 milliseconds based on processing a 192×640 image (4 channel, RGB-D). Table 5.2 provides a comparative analysis of our approach and the comparators.

5.3.5 Feature Learning

Our model is shown to be capable of learning scene context and content in its attempt to produce a complete hole-free depth image. Since our technique does not utilise off-the-shelf classic network architectures, quantifying the feature strength within the network weights used as a pre-training stage for tasks such as classification and detection would not be possible. However, we could evaluate our features in a task somewhat similar to depth completion, such as monocular depth estimation.

The overall similarities between the two problems make this feature evaluation possible despite the differences between the two problems, e.g. the different low and high level cues that need to be learned by the network. We re-purpose our model to estimate scene depth based on a single RGB view by initialising the network with the pre-trained weights from the depth completion model (excluding the depth channel of the first convolutional layer). Fine-tuning is only performed over a single epoch of the dataset without any layer freezing. The results are compared with state-of-the-art approaches [47, 216]. Qualitative results based on synthetic images used as inputs are seen in Figure 5.12.

No domain adaptation to our real-world set was performed during this experiment but the models are evaluated using our real-world test images, nonetheless. As seen in Figure 5.13, even though our network has never seen a real-world image and data domain has not been transferred, we can see that our approach produces sharper and more crisp depth information despite the anomalies that persist due to domain bias issues. Quantitative analysis using synthetic ground truth images is presented in Table 5.4. We can see that our approach has succeeded in a task it is not primarily designed for due to its strength in scene feature learning.

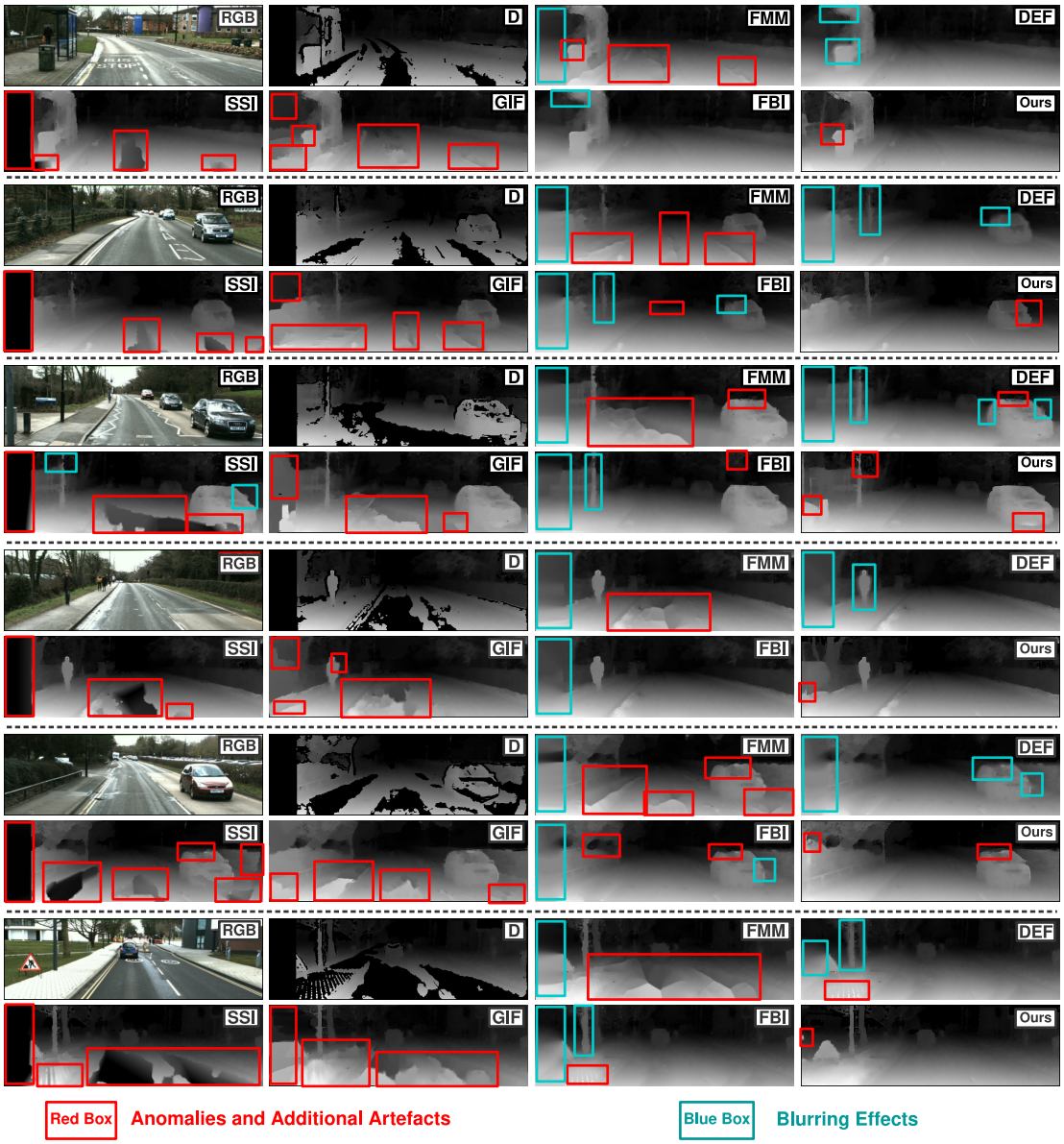

Figure 5.11: Comparing the results of our approach with the depth inpainting approach using second-order smoothness prior (SSI) [125], guided inpainting and filtering (GIF) [41], the fast marching method based inpainting (FMM) [68], diffusion-based exemplar filling (DEF) [66], the Fourier-based inpainting method proposed in Section 3.1 (FBI) using natural real-world data.

Figure 5.12: The results of our approach re-purposed to estimate depth from an RGB image compared against the state-of-the-art approaches of depth and ego-motion from video (DEV) [216] and estimation based on left/right consistency (LRC) [47] using synthetic data.

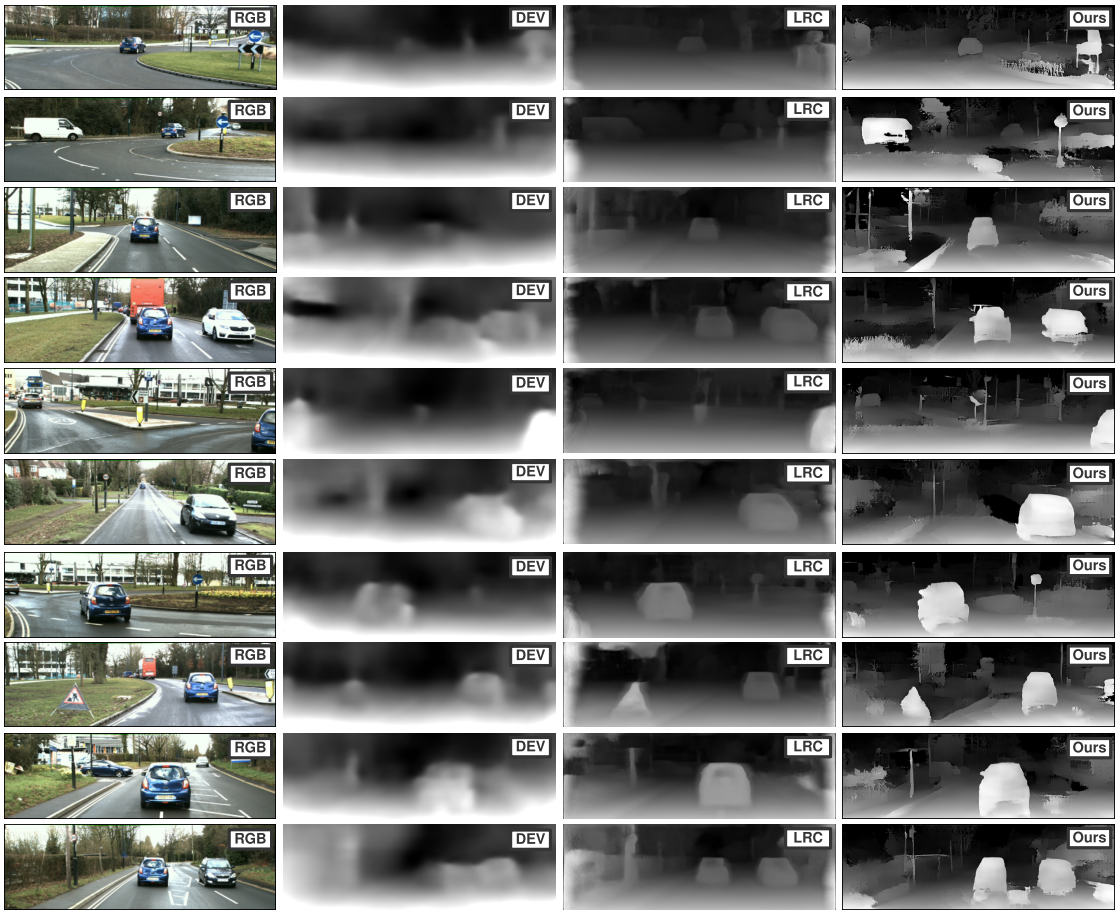

Figure 5.13: The results of our approach re-purposed to estimate depth from an RGB image compared against the state-of-the-art approaches of depth and ego-motion from video (DEV) [216] and estimation based on left/right consistency (LRC) [47] using natural real-world data.

Method	Mean L_1 Error	Mean L_2 Error	PSNR (dB)	SSIM $(-1, 1)$
Result of [216]	28.61	13.68	10.15	0.374
Result of [47]	14.46	3.93	14.22	0.565
Our Result	4.97	0.88	22.35	0.778

Table 5.4: Our pre-trained model tasked with monocular depth estimation compared to [47, 216]. More realistic images have lower error values but with PNSR and SSIM, higher values are better.

5.4 Limitations

This approach is capable of completing depth images in a numerically accurate and visually plausible manner regardless of the number or the size of the holes that exist within the depth image. The approach learns to generate depth content within holes not only based on the content that is already available within the existing portions of the depth image but also based on the context inferred from corresponding colour image. Despite its promising results, the approach suffers from certain challenges. The main issue stems from the fact that the approach is trained on synthetic images but is intended to function in real-world contexts. While the domain adaptation component of the approach explained in Section 5.2.2 positively influences the functionality of the approach, samples from the target domain need to be used during the training process.

Additionally, if a depth completion approach such as ours is to be deployed in real-world scenarios, temporal consistency should be one of the intended outcomes. Our approach is completely based on the spatial information available within the scene and solely operates on a single frame level. Temporal consistency is one of the main avenues any future depth estimation or completion approaches need to focus on.

5.5 Summary

In this chapter, we have attempted to approach depth completion from a learning perspective using an adversarially trained encoder-decoder architecture. If sufficient information is learned from the scene content and context, potential content for large holes within a depth image can be synthesised based on the known regions of the image and the full RGB view. The ground truth depth used to train our learning-based approach is obtained from a virtual environment primarily designed for gaming and a separate network is trained to estimate where holes would be if the data had been obtained via stereo correspondence. The objective of the overall approach is to minimise a loss consisting of four loss components: reconstruction, adversarial, bottleneck feature and domain transfer loss, which results in filling depth holes, not only in synthetic depth images but also in real-world data with no ground truth.

Despite the dependence of the approach on synthetic images for training, qualitative and quantitative experiments point to the capability of the approach to outperform competing contemporary depth completion methods [41, 66, 68, 125] and our best-performing approach presented previously within this thesis in terms of accuracy, the Fourier-based RGB-D image completion technique proposed in Section 3.1. While the domain adaptation component of our approach positively influences the functionality of the approach, samples from the target domain need to be used during the training process. Moreover, for

any effective completion approach to be deployed in the real world, temporal consistency should be one of its most important features. The approach proposed in this chapter is completely based on the spatial information available within the scene and does not take any form of temporal information into account.

In the next chapter, we explore the possibility of estimating full depth of the entire scene from a single RGB image without the need for depth completion or any other post-processing approach. Additionally, we investigate the use of image style transfer as a domain adaptation technique and the incorporation of temporal consistency to enhance the accuracy of the depth output.

Chapter 6

Monocular Depth Estimation

A solution to many of the challenges often associated with conventional depth sensing techniques such as stereo correspondence [13, 296], structure from motion [49, 50], depth from shading and light diffusion [51, 52, 327] is monocular depth estimation. An ideal monocular depth estimation approach is capable of estimating depth from a single RGB image in a very efficient and inexpensive way without the need for careful calibration of additional hardware. Moreover, the estimated depth is often without any invalid or missing regions (holes), therefore eliminating the need for any post-processing or refinement post capture. Consequently, it has the potential to circumvent the need for the depth completion approaches presented in this thesis thus far.

Despite the many benefits that monocular depth estimation can offer, however, certain inherent challenges exist within the problem domain that can restrict the use of such depth estimation approaches in specific applications. For instance, most monocular depth estimation approaches within the existing literature are highly adept at estimating relative depth within the scene but find obtaining absolute depth values describing the exact distance of a given object to the camera significantly more difficult. Another issue, not necessarily unconnected from the former, relates to the scale of the objects in a scene. While any learning-based approach has the potential to learn about the semantics and thus the expected size of the objects, discriminating between small objects close to the camera and larger objects farther from the camera is not as trivial a task for a monocular depth estimation approach as it is for a stereo correspondence technique. In spite of all such challenges, the advantages of monocular depth estimation have transformed it into an active subject of research.

Over the past few years, a number of supervised learning approaches have emerged that take advantage of off-line training on ground truth depth data to make monocular depth prediction possible [46, 189, 190,

Figure 6.1: An example of the results of our monocular depth estimation (KITTI [329]). $\mathbf{RGB_O}$ denotes the original real-world RGB images, $\mathbf{RGB_S}$ represents the domain of real-world images stylised into the synthetic domain and **D** is the estimated depth.

193, 194, 328]. However, ground truth depth is extremely difficult and expensive to acquire in the real world and when it is obtained it is often sparse and flawed, constraining the practical use of many of these approaches.

Certain monocular depth estimation approaches, sometimes referred to as *unsupervised*, do not require direct ground truth depth but instead utilise a secondary supervisory signal during training which indirectly results in producing the desired depth [47, 191, 192, 196]. Training data for these approaches is abundant and easily obtainable but they suffer from undesirable artefacts, such as blurring and incoherent content, due to the nature of their secondary supervision. In this chapter, we attempt to produce high-quality pixel-perfect depth images from a single RGB input image using two novel learning based models trained on synthetic images. The approaches are evaluated using the metrics conventionally used within the literature and outlined in Section 2.3.3.

The material presented in this chapter of the thesis has been published in the following peer-reviewed publications:

- **A. Atapour-Abarghouei** and T. P. Breckon. 'Real-Time Monocular Depth Estimation using Synthetic Data with Domain Adaptation via Image Style Transfer.' In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2018 [33].

- **A. Atapour-Abarghouei** and T. P. Breckon, 'Veritatem Dies Aperit - Temporally Consistent Depth Prediction Enabled by a Multi-Task Geometric and Semantic Scene Understanding Approach.' In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2019 [32].

6.1 Monocular Depth Estimation using Synthetic Data and Image Style Transfer

While using unsupervised monocular depth estimation techniques such as [47, 191, 216] can be an effective solution to the problem of the scarcity of ground truth data (which hinders supervised approaches), an often overlooked fact is that the same technology that facilitates training large-scale deep neural networks can also assist in acquiring synthetic data for these neural networks [38, 330]. Nearly photo-realistic graphically rendered environments primarily used for video gaming can be used to capture homogeneous synthetic depth maps which are then utilised in training a depth estimating model, just as they have been used for depth completion (Chapter 5).

While the use of synthetic data is not novel [330–333], domain adaptation has always been the greatest challenge in this area. Here, we explore the possibility of training a depth estimation model on synthetic data using the new findings regarding the connection between style transfer and domain adaptation [230] (Figure 6.1). Our approach consists of two stages, the operations of which are carried out by two separate models, trained at the same time. The first stage includes training a depth estimation model over synthetic data [38] (Section 6.1.1). However, as the eventual goal involves real-world images, we attempt to reduce the domain discrepancy between the synthetic data distribution and the real-world data distribution using a model trained to transfer the style of synthetic images to real-world images in the second stage (Section 6.1.2).

Here, we perform the overall process within two separate stages. A monocular depth estimation approach is trained using synthetic data and an entirely separate style transfer pipeline will stylise any real-world image to be used as the input to the depth estimation model to appear more similar to the synthetic data to reduce domain shift. This disparity between the two stages is primarily there to emphasise the connection between style transfer and domain adaptation. In any real use case or deployment scenario, one can stylise synthetic images to appear as if they have been captured using real-world sensors and with the aid of the corresponding synthetic depth images, train a single model capable of performing monocular depth estimation on real-world images. It is important to note that functionally both approaches will lead to the same result.

6.1.1 Stage 1 - Monocular Depth Estimation Model

We treat monocular depth estimation as an image-to-image mapping problem, with the RGB image used as the input to our mapping function, which produces depth as its output. With the advent of convolutional neural networks, image-to-image translation and prediction problems have become significantly more

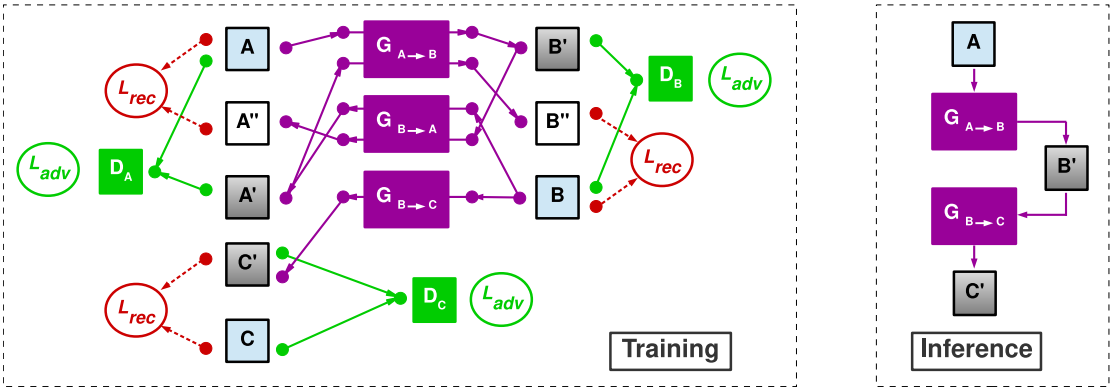

Figure 6.2: An overview of the pipeline of our approach using [315]. Domain **A** (real-world RGB) is transformed into **B** (synthetic RGB) and then to **C** (pixel-perfect depth). **A**, **B**, **C** denote ground truth, **A'**, **B'**, **C'** generated images and **A''**, **B''** cyclically regenerated images.

tractable. A naive solution would be utilising a network that minimises a reconstruction loss (Euclidean distance) between the pixel values of the network output and the ground truth. However, due to the inherent multi-modality of the monocular depth estimation problem (several plausible depth maps can correspond with a single RGB view), any model trained to predict depth based on a sole reconstruction loss (L_1 or L_2) tends to generate values that are the average of all the possible modes in the predictions. This results in blurry outputs.

For this reason, many prediction-based approaches [104, 315, 334–337] and other generative models [314, 338] leverage adversarial training [105] since the use of an adversarial loss helps the model select a single mode from the distribution and generate more realistic results without blurring. While most generative models generate images from a latent noise vector as the input to the generator, our model is conditioned on an input image (RGB).

More formally, our generative model learns a mapping from the input image x (RGB view) to the output image y (scene depth) $G : x \rightarrow y$. The generator (G) attempts to produce fake samples $G(x) = \tilde{y}$ that cannot be distinguished from real ground truth samples y by the discriminator (D) that is adversarially trained to detect the fake samples produced by the generator.

Loss Function

Our objective is achieved using a loss function consisting of two components: a reconstruction loss, which incentivises the generator to produce images that are structurally and contextually as close as possible to the ground truth. We utilise the L_1 loss:

$$\mathcal{L}_{rec} = ||G(x) - y||_1 \tag{6.1}$$

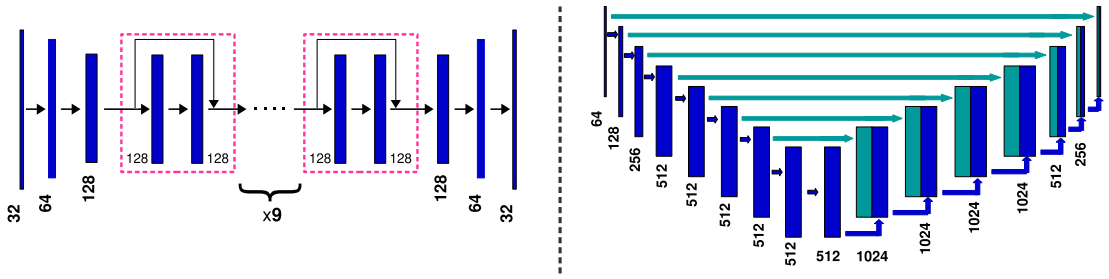

Figure 6.3: An overview of the architecture of the networks used in the approach in Section 6.1. The architecture of the depth generator network is seen on the *right* and the architecture of the style transfer models is seen on the *left*.

However, with the sole use of a reconstruction loss, the generator optimises towards averaging all possible values (blurring) rather than selecting one (sharpness). To remedy this, an adversarial loss is introduced:

$$\mathcal{L}_{adv} = \min_{G} \max_{D} \mathop{\mathbb{E}}_{x,y\sim\mathbb{P}_d(x,y)} [logD(x,y)] + \mathop{\mathbb{E}}_{x\sim\mathbb{P}_d(x)} [log(1 - D(x, G(x)))] \tag{6.2}$$

where $\mathbb{P}_d$ is our data distribution defined by $\tilde{y} = G(x)$, with x being the generator input and y the ground truth. Subsequently, the joint loss function is as follows:

$$\mathcal{L} = \lambda\mathcal{L}_{rec} + (1 - \lambda)\mathcal{L}_{adv} \tag{6.3}$$

with λ selected empirically. This forces optimisation towards explicit value selection and content preservation.

Implementation Details

Since synthetic data is needed to train the model, colour and disparity images are captured from a camera view set in front of a virtual car as it automatically drives around the virtual environment and images are captured every 60 frames with randomly varying height, field of view, weather and lighting conditions at different times of day to avoid over-fitting. 80,000 images were captured with 70,000 used for training and 10,000 set aside for testing. Our model trained using this synthetic data outputs a disparity image which is converted to depth using focal length and scaled to the depth range of the KITTI image frame [329].

An important aspect of the monocular depth estimation problem is that overall structures within the RGB image (input) and the depth image (output) are aligned as they provide different types of information

Method	Training Data	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
		Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Data Set Mean [301]	[301]	0.403	0.530	8.709	0.403	0.593	0.776	0.878
Eigen et al. Coarse [46]	[301]	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen et al. Fine [46]	[301]	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [190]	[301]	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Zhou et al. [216]	[301]	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou et al. [216]	[301]+[339]	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Garg et al. [191]	[301]	0.152	1.226	5.849	0.246	0.784	0.921	0.967
Godard et al. [47]	[301]	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Godard et al. [47]	[301]+[339]	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Zhan et al. [48]	[301]	0.144	1.391	5.869	0.241	0.803	0.928	0.969
Our Approach	S^*	**0.110**	**0.929**	**4.726**	**0.194**	**0.923**	**0.967**	**0.984**

Table 6.1: Comparing the results of monocular depth estimation techniques over the KITTI dataset using the data split in [193]. S^* denotes the synthetic data captured from a graphically environment.

for the exact same scene. As a result, much information (e.g. structure, geometry, object boundaries and alike) is shared between the input and output. We attempt to ensure that this data shared between the input and the output, especially the high-frequency information which is easily lost during the down-sampling process within the network, is preserved.

Consequently, we utilise skip connections in the generator rather than using a classic encoder-decoder pipeline with no skip connections [98, 104, 277, 340, 341]. By taking advantage of these skip connections, the generator has the opportunity to directly pass geometric information between corresponding layers in the encoder and the decoder without having to go through every single layer in between. These short-cuts created by the skip connections within the network will lead to a higher-quality output, as the high-frequency information in the input can be directly passed through to the output and no useful information is lost as the features are down-sized in the network bottleneck.

While Many other generative approaches incorporate a random noise vector z or drop-outs into the generator training to prevent deterministic mapping and induce stochasticity [104, 336, 341, 342], despite our experiments with both random noise as part of the generator input and drop-outs, no significant difference in the output distribution could be achieved.

Following the success of U-net [275] which contains an efficient light-weight architecture, our generator consists of a similar pipeline, with the exception that skip connections exist between every pair of corresponding layers in the encoder and decoder. For our discriminator, we deploy the basic architecture used in [343]. Both generator and discriminator use the convolution-BatchNorm-ReLu module [322] with the discriminator using leaky ReLUs ($slope = 0.2$).

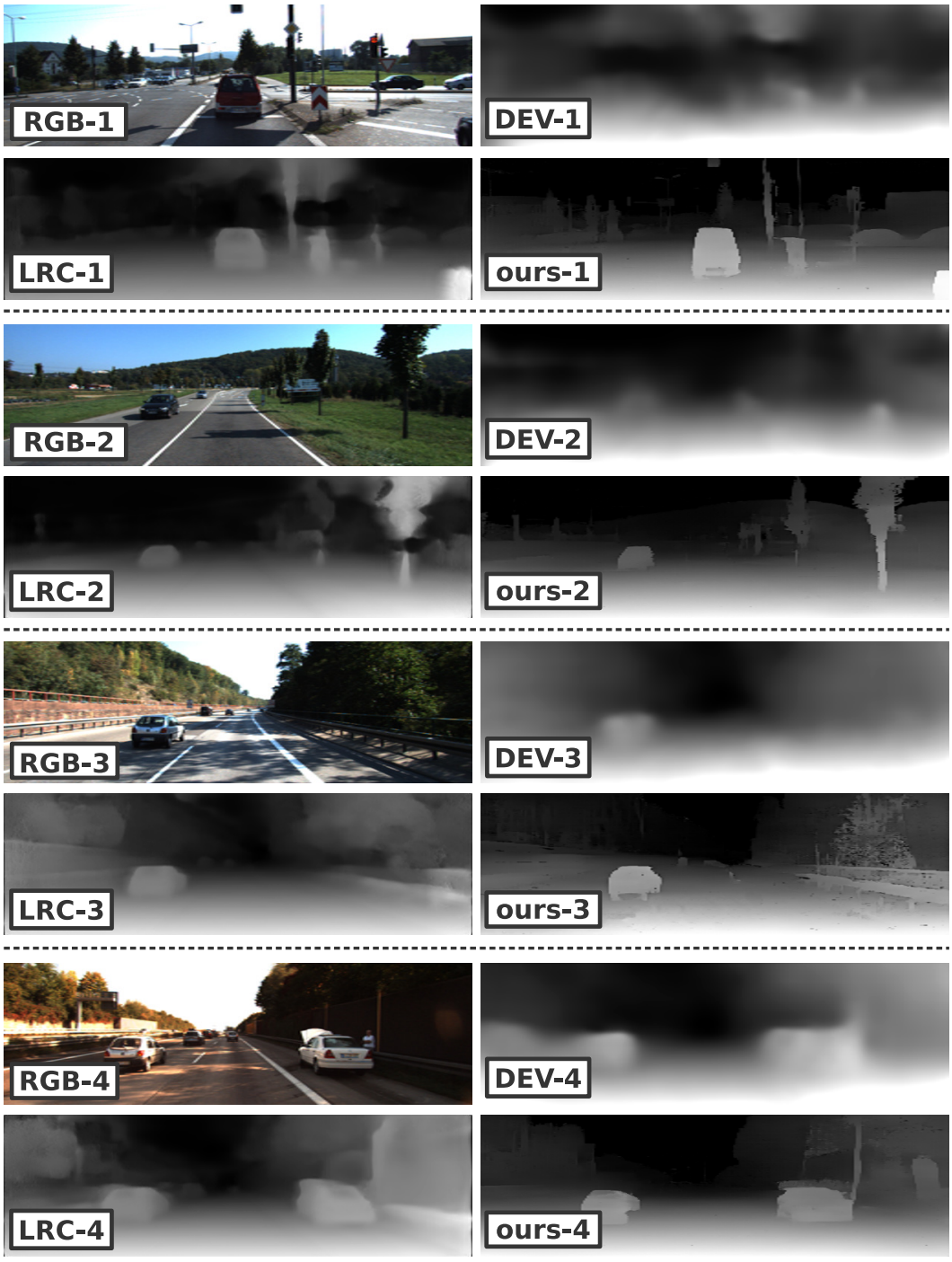

Figure 6.4: Qualitative comparison of our results against the state-of-the-art methods of depth and ego-motion from video (DEV) [216] and estimation based on left/right consistency (LRC) [47] over the KITTI split. Note that despite the sharp and crisp results, anomalies (such as the noise in the third result) can occur due to environmental conditions (in this case the saturated shrubbery).

All implementation and training is done in *PyTorch* [324], with the Adam [326] providing experimentally superior optimisation (momentum $\beta_1 = 0.5$, $\beta_2 = 0.999$, initial learning rate $\alpha = 0.0002$). The coefficient in the joint loss function was empirically chosen to be $\lambda = 0.99$.

6.1.2 Stage 2 - Domain Adaptation via Style Transfer

Assuming the monocular depth estimation procedure presented in Section 6.1.1 performs well, since the model is trained on synthetic images, the idea of estimating depth from RGB images captured in the real-world is still far fetched as the synthetic and real-world images are from widely different domains.

Our goal is thus to learn a mapping function $\mathcal{D} : X \rightarrow Y$ from the source domain X (real-world images) to the target domain Y (synthetic images) in a way that the distributions $\mathcal{D}(X)$ and Y are identical. When images from X are mapped into Y, their depth can be inferred using our monocular depth estimator (Section 6.1.1) that is specifically trained on images from Y.

While the notion of transforming images from one domain to the other is not new [229, 315, 344–346], we utilise image style transfer using generative adversarial networks, as proposed in [315], to reduce the discrepancy between our source domain (real-world data) and our target domain (synthetic data on which our depth estimator in Section 6.1.1 functions). This approach uses adversarial training [105] and cycle-consistency [47, 347–349] to translate between two sets of unaligned images from different domains.

Formally put, the objective is to map images between the two domains X, Y with distributions $x \sim \mathbb{P}_d(x)$ and $y \sim \mathbb{P}_d(y)$. The mapping functions are approximated using two separate generators, G_{XtoY} and G_{YtoX} and two discriminators D_X (discriminating between $x \in X$ and $G_{YtoX}(y)$) and D_Y (discriminating between $y \in Y$ and $G_{XtoY}(x)$). The loss contains two components: an adversarial loss [105] and a cycle consistency loss [315]. The general pipeline of the approach (along with the depth estimation model described in Section 6.1.1) is seen in Figure 6.2, with three generators $G_{A_{to}B}$, $G_{B_{to}A}$ and $G_{B_{to}C}$ and three discriminators D_A, D_B and D_C.

As seen in Figure 6.2, the generator networks $G_{A_{to}B}$ and $G_{B_{to}A}$ are responsible for transferring the style of synthetic images to natural real-world images and vice versa. The training of these generators is accomplished using a reconstruction loss and an adversarial loss, using the discriminator networks D_A and D_B. For further details of the style transfer approach, refer to [315]. The generator network of $G_{B_{to}C}$ is directly responsible for estimating the scene depth output, the fidelity of which is ensured by the discriminator network of D_C.

Similar architectures are used for the networks for the sake of consistency. In the depth estimation model, an architecture inspired by that of U-net [275] is used for the generator, $G_{B_{to}C}$. Let `Ck` represent a Convolution-BatchNorm-ReLU layer with k filters. The architecture thus consists of:

`C64-C128-C256-C512-C512-C512-C512-C512-C512-C1024-C1024-C1024-C1024`

`-C512-C256-C128`

Skip connections directly connect every single pair of corresponding layers in the encoder and the decoder. An overview of the architecture is depicted in Figure 6.3 (right).

As for the generator networks of $G_{A_{to}B}$ and $G_{B_{to}A}$, the architecture is inspired by that of [98]. Let `Ck` denote a Convolution-InstanceNorm- ReLU layer with k filters and stride 1, `dk` a Convolution-InstanceNorm-ReLU layer with k filters and stride 2, and `uk` a fractionally-strided-Convolution-InstanceNorm-ReLU layer with k filters, and stride $1/2$. `Rk` represents a residual block that contains two convolutional layers. The full architecture, therefore, consists of:

`C32-d64-d128-R128-R128-R128-R128-R128-R128-R128-R128-R128-C128-u64-u32`

The overall architecture of the generator networks in charge of style transfer, $G_{A_{to}B}$ and $G_{B_{to}A}$, can be seen in Figure 6.3 (left).

Loss Function

Since there are two generators to constrain the content of the images, there are two mapping functions, each with its own loss but with similar formulations. The use of an adversarial loss guarantees the style of one domain is transferred to the other. The loss for G_{XtoY} with D_Y is as follows:

$$\mathcal{L}_{adv}(X \to Y) = \min_{G_{XtoY}} \max_{D_Y} \mathop{\mathbb{E}}_{y \sim \mathbb{P}_d(y)} [logD_Y(y)] + \mathop{\mathbb{E}}_{x \sim \mathbb{P}_d(x)} [log(1 - D_Y(G_{XtoY}(x)))] \tag{6.4}$$

where $\mathbb{P}_d$ is the data distribution, X the source domain with samples x and Y the target domain with samples y. Similarly, for G_{YtoX} and D_X, the adversarial loss is:

$$\mathcal{L}_{adv}(Y \to X) = \min_{G_{YtoX}} \max_{D_X} \mathop{\mathbb{E}}_{x \sim \mathbb{P}_d(x)} [logD_X(x)] + \mathop{\mathbb{E}}_{y \sim \mathbb{P}_d(y)} [log(1 - D_X(G_{YtoX}(y)))] \tag{6.5}$$

The original work in [315] replaces the log likelihood by a least square loss to improve training stability [350]. We experimented with that setup but noticed no significant improvement in training stability or the quality of the results. Therefore the original adversarial loss is used.

In order to constrain the adversarial loss of the generators to encourage the model to produce desirable contextually coherent images rather than random images with the target domain, a cycle-consistency loss is added that prompts the model to become capable of bringing an image x that is translated into the target

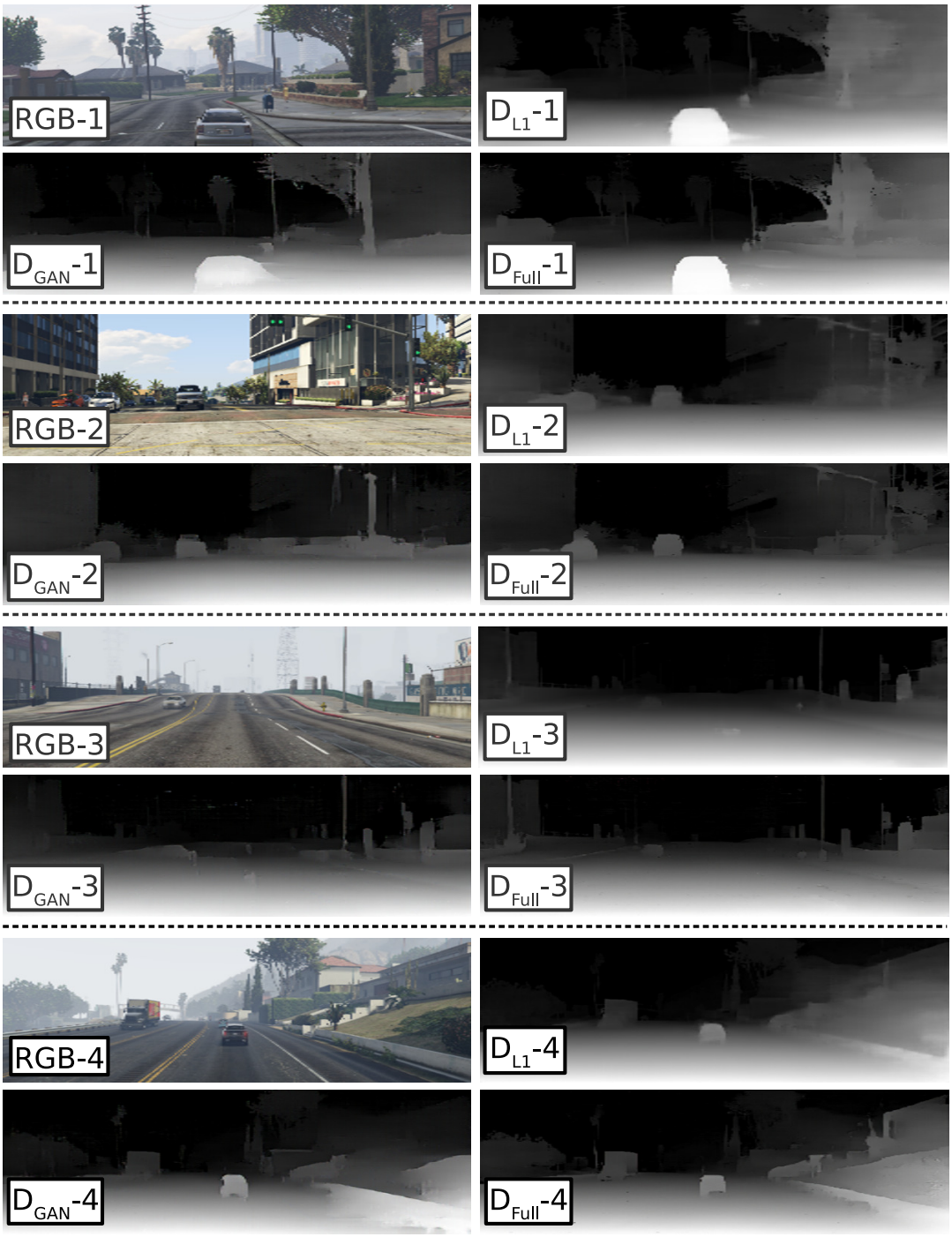

Figure 6.5: Comparison of the results with different components of the loss in the depth estimation model (Section 6.1.1). **RGB** denotes the synthetic colour images used as inputs to the model and **D** represents the generated depth images.

Method	Training Data	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
		Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Ours w/o domain adaptation	K+S*	0.498	6.533	9.382	0.609	0.712	0.823	0.883
Ours w/ the approach of [98]	K+S*	0.154	1.338	6.470	0.296	0.874	0.962	0.981
Ours w/ the approach of [315]	K+S*	**0.101**	**1.048**	**5.308**	**0.184**	**0.903**	**0.988**	**0.992**

Table 6.2: Ablation study over the KITTI dataset using the KITTI split. our approach is trained using, KITTI (K) and synthetic data (S*). The approach with domain adaptation using cycleGAN [315] provides the best results.

domain Y using G_{XtoY} back into the source domain X using G_{YtoX}. Essentially, after a full cycle, we should have: $G_{YtoX}(G_{XtoY}(x)) = x$ and vice versa. As a result, the cycle-consistency loss is:

$$\mathcal{L}_{cyc} = ||G_{YtoX}(G_{XtoY}(x)) - x||_1 + ||G_{XtoY}(G_{YtoX}(y)) - y||_1 \tag{6.6}$$

Subsequently, the joint loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{adv}(X \rightarrow Y) + \mathcal{L}_{adv}(Y \rightarrow X) + \lambda \mathcal{L}_{cyc} \tag{6.7}$$

with λ selected empirically.

Implementation Details

The style generator architectures are based on the work in [98] with two convolutional layers followed by nine residual blocks [351] and two up-convolutions that bring the image back to its original input size.

As for the discriminators, the same architecture is used as was in Section 6.1.1. Additionally, the discriminators are updated based on the last 50 generator outputs and not just the last generated image [315, 346].

All implementation and training is done in *PyTorch* [324] and Adam [326] is used to perform the optimisation for the task (momentum $\beta_1 = 0.5$, $\beta_2 = 0.999$ and initial learning rate $\alpha = 0.0001$). The coefficient in the joint loss function in Equation 6.7 is empirically chosen to be $\lambda = 10$. Since the images are normalized to values within the range [0,1] before they are passed into the model, the cycle consistency loss ($\mathcal{L}_{cyc}$), which is essentially the sum of the Euclidean distances between the cyclically-stylised outputs and the ground truth original inputs (both with values in the range [0,1]) would be very small. This is the primary reason why the weighting coefficient λ in Equation 6.7 emphasises the cycle consistency to balance the values within the overall loss function. It is important to note that since style transfer is primarily a visual problem, the empirical analyses that led to the choice of the value of λ were all visual as well, with the overall quality being the main determining factor.

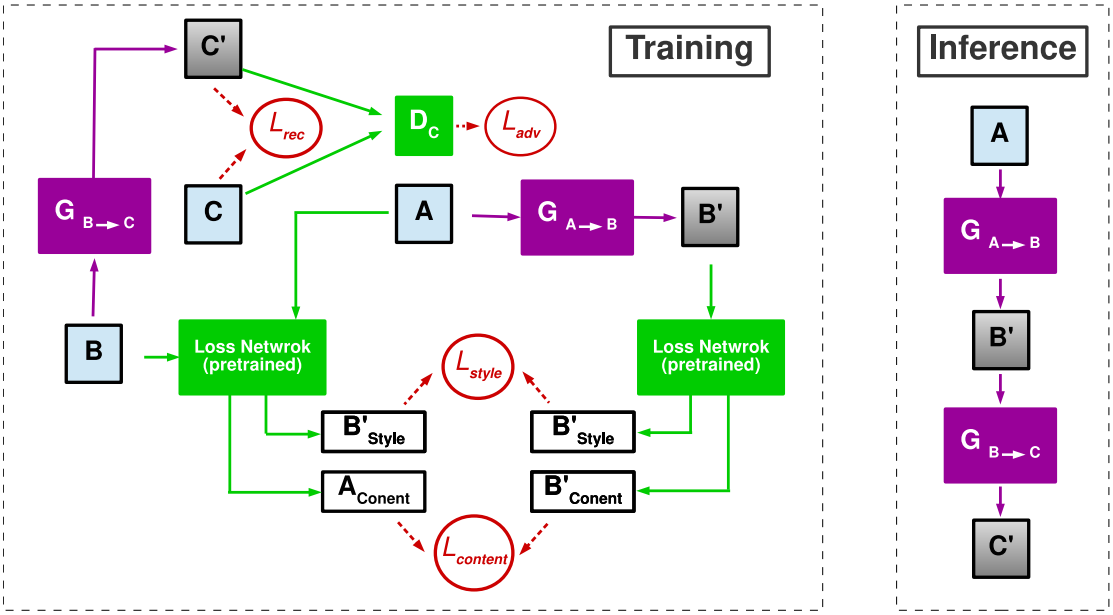

Figure 6.6: Our approach using the style transfer approach of [98]. Images from domain **A** (real-world) are transformed into **B** (synthetic) and then to **C** (pixel-perfect depth maps). **A, B, C** represent ground truth images and **A', B', C'** denote generated images.

6.1.3 Experimental Results

In this section, we evaluate our approach using ablation studies and both qualitative and quantitative comparisons with state-of-the-art monocular depth estimation methods via publicly available datasets. We use KITTI [329] for our comparisons and Make3D [205] in addition to data captured locally to test how our approach generalises over unseen data domains. Using a GeForce GTX 1080 Ti GPU, the entire two passes take an average of 22.7 milliseconds ($\sim$ 44 fps).

Comparisons with Other Approaches

To facilitate better numerical comparisons against existing approaches within the literature, we test our model using the 697 images from the data split suggested in [193]. As seen in Table 6.1, our approach performs better than the current state-of-the-art [47, 190, 193, 216] with lower error and higher accuracy. Measurement metrics are based on [193]. Some of the comparators [47, 216] use a combination of different datasets for training and fine-tuning to boost performance, while we only use the synthetic data and KITTI [329]. Additionally, following the conventions of the literature [47, 190, 193, 216], the error

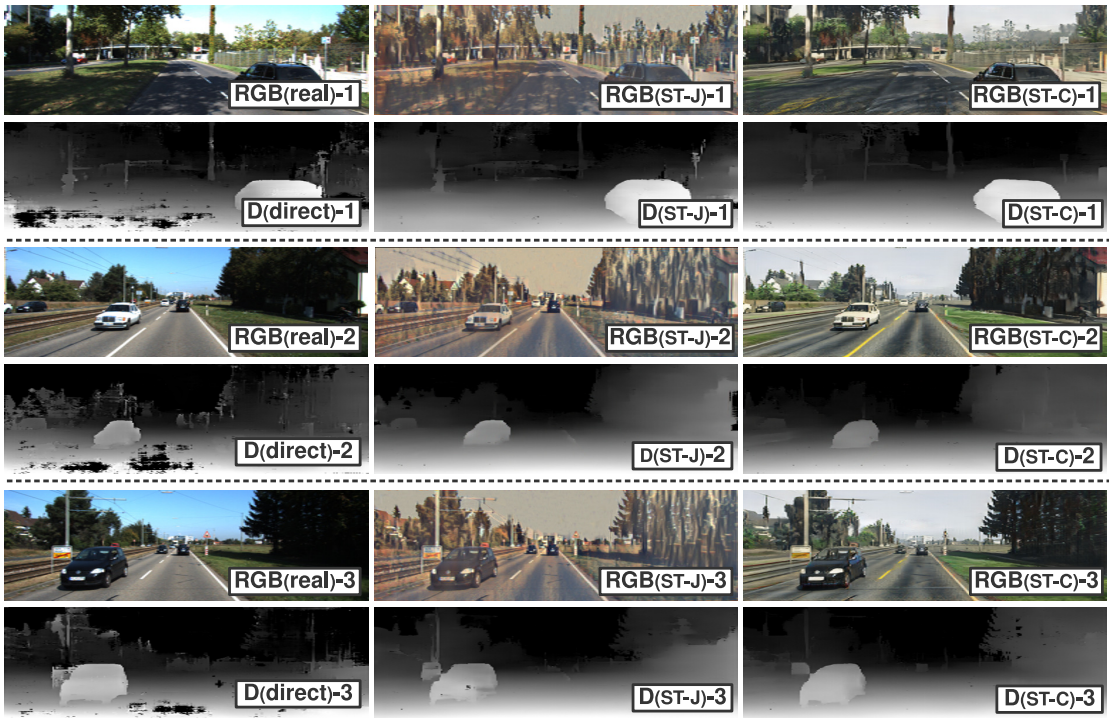

Figure 6.7: Results demonstrating the importance of style transfer. Left column shows results with no domain adaptation (**direct**). Middle column contains results when the style transfer used as domain transfer is based on the work of Johnson et al. [98] (**ST-J**) and the right column indicates results when the style transfer used as domain transfer is based on CycleGAN [315] (**ST-C**).

measurements are all performed in depth space, while our approach produces disparities and as a result, small precision issues are expected.

We also used the data split of 200 images in KITTI [329] to provide better *qualitative* evaluation, since the ground truth disparity images within this split are of considerably higher quality than laser data and provide CAD models as replacements for moving cars. As is clearly shown in Figure 6.4, compared to the state-of-the-art approaches [47, 216] trained on similar data domains, our approach generates sharper and more crisp outputs in which object boundaries and thin structures are well preserved.

Ablation Studies

A crucial part of this study is interpreting the necessity of the components of the approach. Our monocular depth estimation model (Section 6.1.1) utilises a combination of reconstruction and adversarial losses (Equation 6.7). We trained our model using the reconstruction loss only and the adversarial loss only to

Method	Error Metrics (lower, better)			
	Abs. Rel.	Sq. Rel.	RMSE	RMSE log
Train Set Mean	0.814	12.992	11.411	0.285
Karsch et al. [201]	0.398	4.723	7.801	0.138
Liu et al. [207]	0.441	6.102	9.346	0.153
Godard et al. [47]	0.505	10.172	10.936	0.179
Zhou et al. [216]	0.356	4.948	9.737	0.443
Laina et al. [211]	0.189	1.711	5.285	0.079
Our Approach	0.423	9.343	9.002	0.122

Table 6.3: Comparative results on Make3D [205], on which the approach is not trained. [201, 207, 211] are specifically trained on Make3D. Following [47, 216], errors are only calculated where depth is less than 70 meters in a central image crop.

test their importance. Figure 6.5 demonstrates the effects of removing parts of the training objective. The model based only on the reconstruction loss (L_1) produces contextually sound but blurry results, while the adversarial loss generates sharp outputs that contain artefacts. The full approach creates more accurate results without unwanted effects. Further evidence of the efficacy of a combination of a reconstruction and adversarial loss is found in [336].

Another important aspect of our ablation study entails evaluating the importance of domain transfer (Section 6.1.2) within our framework. Due to the differences in the domains of the synthetic and natural data, our depth estimator directly applied to real-world data does not produce qualitatively or quantitatively desirable results, which makes the domain adaptation step necessary.

While our approach requires an adversarial discriminator [315] to carry out the style transfer needed for our domain adaptation, [230] has suggested that more conventional style transfer, which involves matching the Gram matrices [231] of feature maps, is theoretically equivalent to minimising the Maximum Mean Discrepancy with the second order polynomial kernel and leads to domain adaptation.

As evidence for the notion that a discriminator can perform the task even better, we also experiment with the style transfer approach of [98], which improves on the original work [97] by training a generator that can transfer a specific style (that of our synthetic domain in our work) onto a set of images of a specific domain (real-world images). A loss network (pre-trained VGG [307]) is used to extract the image style and content (as in [97]). This network calculates the loss values for content (based on the L_2 difference between feature representations extracted from the loss network) and style (from the squared Frobenius norm of the distance between the *Gram matrices* of the input and main style images) that are used to train the generator. An overview of the entire pipeline using [98] (along with the depth estimation model - Section 6.1.1) can be seen in Figure 6.6.

Figure 6.8: Qualitative results of our approach on urban driving scenes captured locally without further training. **RGB$_O$** denotes the original real-world RGB images, **RGB$_S$** represents the real-world images stylised into the synthetic domain and **D** is the estimated depth.

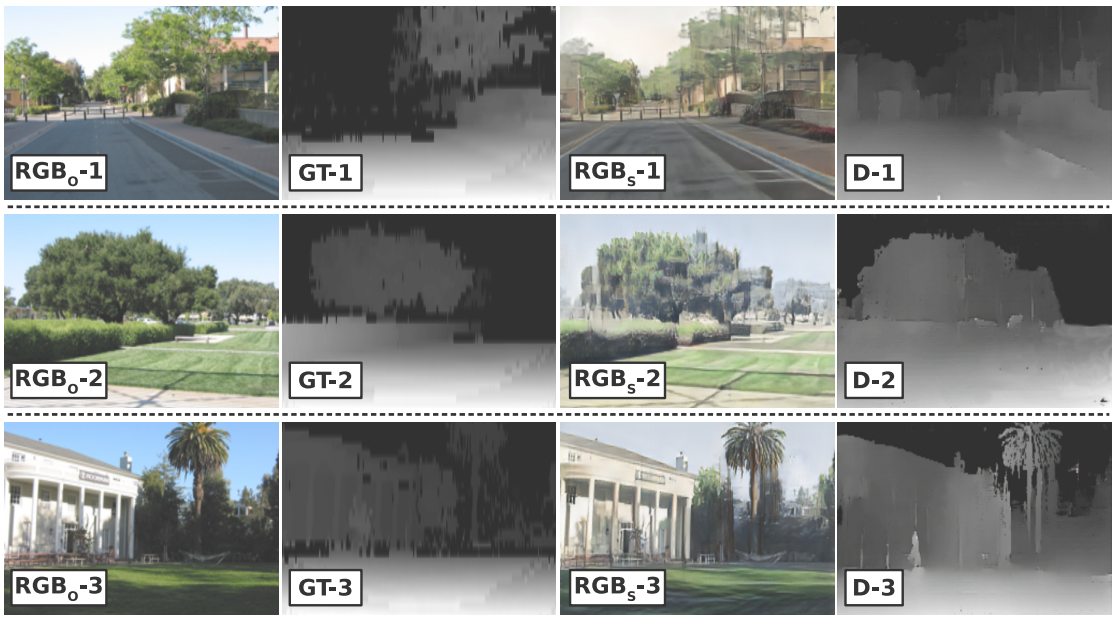

Figure 6.9: Results on the Make3D test set [205]. Note the quality of our outputs despite the vast differences between this dataset and the images used in our training. $\mathbf{RGB_O}$ denotes the original real-world RGB images, $\mathbf{RGB_S}$ represents the real-world images stylised into the synthetic domain, **GT** is the ground-truth depth and **D** is the estimated depth.

Whilst [315] transfers the style between two *sets* of unaligned images from different domains, [98] requires *one* specific image to be used as the style image. We have access to tens of thousands of images representing the same style. This is remedied by collecting a number of synthetic images that contain a variety of objects, textures and colours that represent their domain and pooling their features to create a single image that holds our desired style.

The data split of 200 images in KITTI [329] was used to evaluate our approach regarding the effects of domain adaptation via style transfer. We experimented with both [315] and [98], in addition to feeding real-world images to our depth estimator without domain adaptation. As seen in the results presented in Table 6.2, not using style transfer ends in considerable amount of anomalies in the output while translating images into synthetic space using [315] before depth estimation generates significantly better outputs. Figure 6.7 provides qualitative results leading to the same conclusion.

Model Generalisation

Images used in our training procedure come from the synthetic environment [38] and the KITTI dataset [329] but we evaluate our approach on additional data to test the model generalisation capabilities. Using

data captured locally in an urban environment (Durham, UK), we generated visually convincing depth without any training on our data which are sharp, coherent and plausible, as seen in Figure 6.8.

Furthermore, we tested our model on the Make3D dataset [205], which contains paired RGB and depth images from a different domain and compared our results against supervised methods trained on said dataset and state-of-the-art monocular depth estimation methods. Even though our approach does not numerically beat comparators that are trained on Make3D [201, 207, 211], as seen in Table 6.3, our results are promising despite no training over this data and outputs are highly plausible qualitatively, even compared to the ground truth. Some results are seen in Figure 6.9.

6.1.4 Limitations

Even though the proposed approach is capable of generating high quality depth by taking advantage of domain adaptation through image style transfer, the very idea of style transfer brings forth certain shortcomings. The biggest issue is that the approach is incapable of adapting to sudden lighting changes and saturation during style transfer. When the two domains significantly vary in intensity differences between lit areas and shadows (as is the case with our approach), shadows can be recognised as elevated surfaces or foreground objects post style transfer. Figure 6.10 contains some examples of how these issues arise.

Moreover, despite the fact that depth holes are generally considered undesirable [41, 133], certain areas within the scene depth should remain without depth values (e.g. very distant objects and sky). However, a supervised monocular depth estimation approach such as ours (even with style transfer) is incapable of distinguishing the sky from other extremely saturated objects within the scene, which can lead to small holes within the scene. This issue can be tackled in any future work by adding a weighted loss component that can penalise the generator when holes are misplaced based on the approximate location of the sky and other distant background objects.

6.1.5 Summary

In Section 6.1 of this chapter, we have primarily focused on investigating the use of efficient learning-based models capable of estimating complete scene depth from a single RGB image. In this vein, we have proposed a monocular depth estimation approach that is capable of producing high quality pixel-prefect depth images based on a monocular RGB images.

Since the issue of lack of ground truth dense hole-free depth images has long plagued the area of monocular depth estimation, or approach is based on training an effective model supervised using synthetic data captured from a graphically rendered virtual environment [38]. However, such a model only trained on

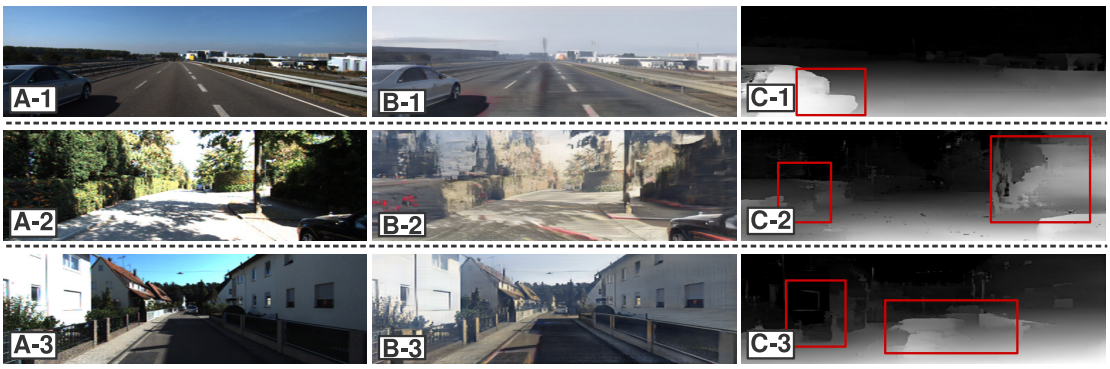

Figure 6.10: Examples of failures, due to light saturation and shadows. Issues are marked with red boxes.

synthetic imagery cannot be expected to perform well on images naturally captured from the real world as the data distributions to which real-world images belong significantly differs from the synthetic training images.

As a result, we rely on an adversarially-trained style transfer approach [315] to adapt the real-world data to fit into the distribution approximated within the depth estimation model. Consequently, the model is capable of producing high quality depth outputs (matching the quality of synthetic images) for previously unseen RGB images captured from real-world scenarios.

Using extensive experimentation on publicly available data [301], we demonstrate that our monocular depth estimation approach outperforms contemporary state-of-the-art techniques focusing on images from urban driving scenarios [46–48, 190, 191, 216]. Additionally, to illustrate the generalisation capabilities of the our technique, it is tested using the Make3D dataset [205] from a completely different domain. Our approach remains closely competitive with techniques that are specifically trained on the dataset never before seen by our approach [201, 207, 211].

In Section 6.2, we attempt to extend the notion of monocular depth estimation trained on synthetic data to be compatible with temporal data in a multi-task pipeline capable of performing semantic segmentation as well depth estimation. By providing the model with a better geometric and semantic understanding of the scene, the model is able to learn more crucial details about the context and the content of the scene and its objects and can perform well on real-world data even without domain adaptation.

6.2 Temporally Consistent Depth via a Multi-Task Approach

In this section, we propose a model capable of semantically understanding a scene by jointly predicting depth and pixel-wise semantic classes (Figure 6.11). The network performs semantic segmentation

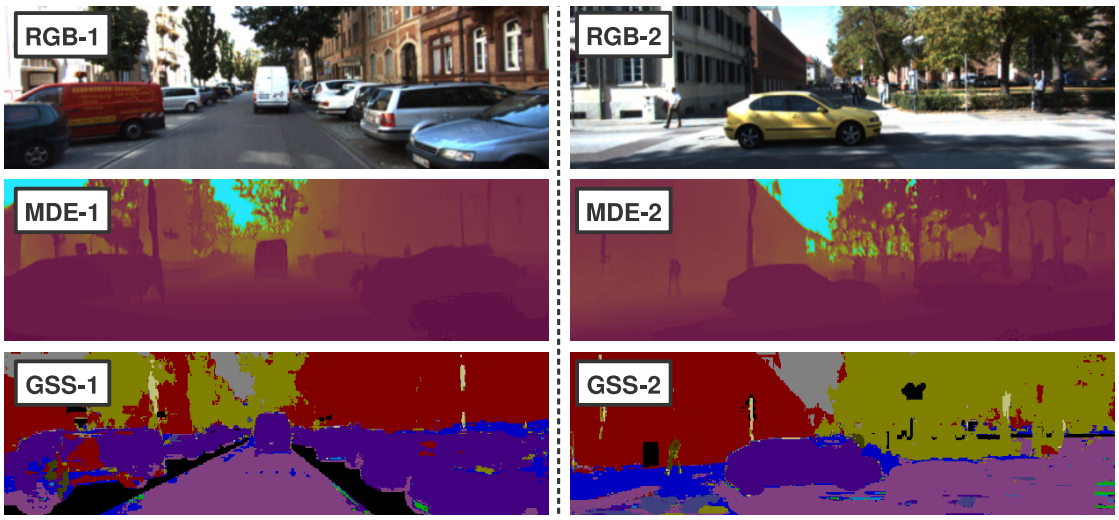

Figure 6.11: Example of the results of the proposed approach. **RGB:** input colour image; **MDE:** Monocular Depth Estimation; **GSS:** Generated Semantic Segmentation.

(Section 6.2.3) in conjunction with monocular depth estimation (i.e. predicting scene depth based on a single RGB image) or depth completion (i.e. completing missing regions of existing depth sensed through other imperfect means, Section 6.2.2). Our approach performs these tasks within a single model capable of two separate scene understanding objectives requiring low-level feature extraction and high-level inference, which leads to better and deeper representation learning within the model [352]. This is empirically demonstrated via the notably improved results obtained for each individual task when performed simultaneously in this manner.

Within the current literature, many techniques focus on individual frames to spatially accomplish their objectives, ignoring temporal consistency in video sequences, one of the most valuable sources of information widely available within real-world applications. Here, we propose a feedback network that at each time step takes the output generated at the previous time step as a recurrent input. Furthermore, using a pre-trained optical flow estimation model, we ensure the temporal information is explicitly considered by the overall model during training.

In recent years, skip connections have been demonstrated to be very effective when the input and output of a CNN share roughly similar high-level spatial features [275, 353–355]. We make use of a complex network of skip connections throughout the architecture to guarantee that no high-level spatial features are lost during training when the features are down-sampled.

Our approach is designed to perform two tasks using a single joint model: depth estimation/completion (Section 6.2.2) and semantic segmentation (Section 6.2.3). This has been made possible using a synthetic

dataset [39] in which both ground truth depth and pixel-wise segmentation labels are available for video sequences of urban driving scenarios.

6.2.1 Overall Architecture

Our single network takes three different inputs producing two separate outputs for two tasks - *depth prediction and semantic segmentation*. Moreover, temporal information is explicit in our formulation, as one of the inputs at every time step is an output from the previous time step via recurrence. The network comprises three different components: the input streams (Figure 6.12 - left), in which the inputs are encoded, the middle stream (Figure 6.12 - middle), which fuses the features and begins the decoding process and finally the output streams (Figure 6.12 - right), in which the results are generated.

As seen in Figure 6.13, two of the inputs are RGB or RGB-D images (depending on whether monocular depth estimation to create depth, or depth completion to fill holes within an existing depth image, is the focus) from the current and previous time steps. The two input streams that decode these share their weights. The third input is the depth generated at the previous time step. The middle section of the network fuses and decodes the input features and finally the output streams produce the results (scene depth and segmentation). Every layer of the network contains two convolutions, batch normalisation [322] and PReLU [356].

Following recent successes of approaches using skip connections [275, 353–355], we utilise a series of skip connections within our architecture. Figure 6.12 demonstrate an overview of our overall architecture, in which the flow of these skip connections within the model can be clearly seen. Our inputs and outputs, despite containing different types of information (RGB, depth and pixel-wise class labels), relate to consecutive frames from the same scene and therefore, share high-frequency information such as certain object boundaries, structures, geometry and others, so skip connections can be of significant value in improving results. By combining two separate objectives (predicting depth and pixel-wise class labels) within our network, in which the input streams and middle streams are fully trained on both tasks, the results are better than when two separate networks are individually trained to perform the same tasks.

Even though the entire network is trained as one entity, in our discussions, the parts of the network responsible for predicting depth will be referred to as G_1 and the portions involved in semantic segmentation G_2. These two modules are essentially the same except for their output streams.

6.2.2 Depth Estimation / Completion

Similar to the solution presented in Section 6.1, we consider depth prediction as a supervised image-to-image translation problem, wherein an input RGB image (for depth estimation) or RGB-D image (with

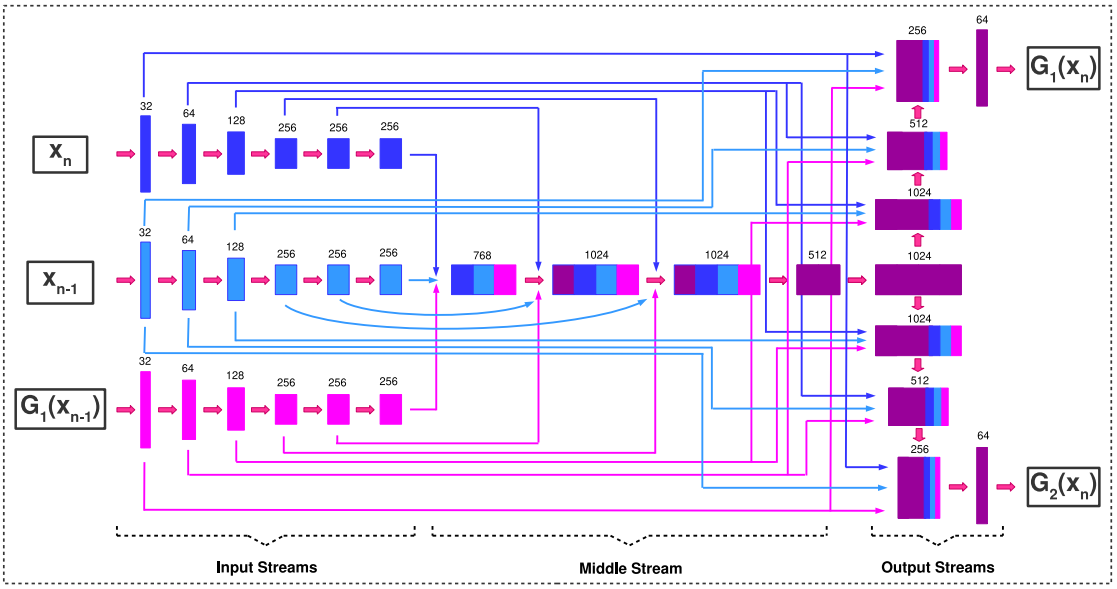

Figure 6.12: An overview of the detailed outline of the generator architecture.

the depth channel containing holes for depth completion) is translated to a complete depth image. More formally, a generative model (G_1) approximates a mapping function that takes as its input an image x (RGB or RGB-D with holes) and outputs an image y (complete depth image) $G_1 : x \rightarrow y$.

The initial solution would be to minimise the Euclidean distance between the pixel values of the output ($G_1(x)$) and the ground truth depth (y). This simple reconstruction mechanism forces the model to generate images that are structurally and contextually close to the ground truth. For monocular depth estimation, this reconstruction loss is:

$$\mathcal{L}_{rec} = ||G_1(x) - y||_1, \tag{6.8}$$

Method	Depth Error (lower, better)				Depth Accuracy (higher, better)			Segmentation (higher, better)	
	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$	Accuracy	IoU
Two Models	0.245	1.513	6.323	0.274	0.803	0.856	0.882	0.604	0.672
One Model	0.208	1.402	6.026	0.269	0.836	0.901	0.926	0.748	0.764

Table 6.4: Comparison of depth prediction and segmentation tasks performed in one single network and two separate networks.

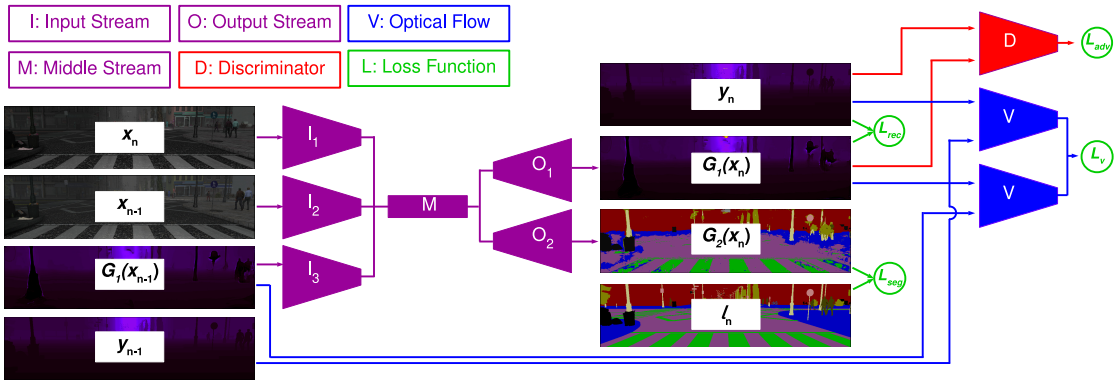

Figure 6.13: Overall training procedure of the model.

where x is the input image, $G_1(x)$ is the output and y the ground truth. For depth completion, however, the input x is a four-channel RGB-D image with the depth containing holes that would occur during depth sensing. Since we use synthetic data [39], we only have access to hole-free pixel-perfect ground truth depth. While one could naïvely cut out random sections of the depth image to simulate holes, we opt for creating realistic and semantically meaningful holes with characteristics of those found in real-world images. As a result, similar to the approach in Section 5.1.1, a separate model is thus created and tasked with predicting where holes would be by means of pixel-wise segmentation. The training process is of this *hole prediction* model is the same as that in Section 5.1.1.

When our main model is being trained to perform depth completion, the hole mask generated by the *hole prediction* network is employed to create the depth channel of the input RGB-D image. Subsequently, the reconstruction loss is:

$$\mathcal{L}_{rec} = ||(1 - M) \odot G_1(x) - (1 - M) \odot y||_1, \tag{6.9}$$

where $\odot$ is the element-wise product operation and x the input RGB-D image in which the depth channel is $y \odot M$. Experiments with an $L2$ loss returned similar results.

However, the sole use of a reconstruction loss would lead to blurry outputs since monocular depth estimation and depth completion are multi-modal problems, i.e. several plausible depth outputs can correctly correspond to a region of an RGB image. This multi-modality results in the generative model (G_1) averaging all possible modes rather than selecting one, leading to blurring effects in the output.

To prevent this, adversarial training [105] has become prevalent within the literature [104, 106, 314, 336] since it forces the model to select a mode from the distribution resulting in better quality outputs. In this vein, our depth generation model (G_1) takes x as its input and produces fake samples $G_1(x) = \tilde{y}$ while a

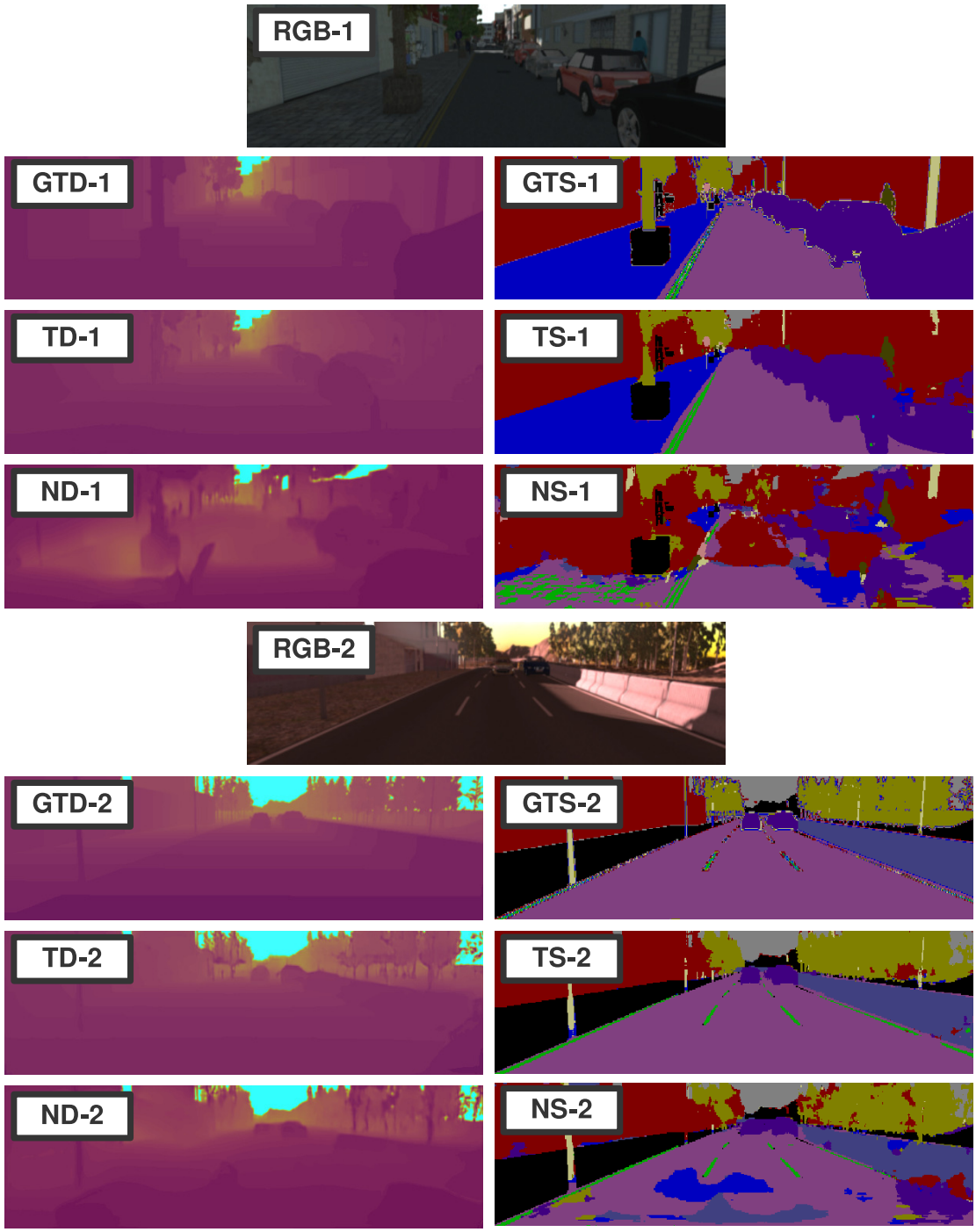

Figure 6.14: Comparing the results of the approach on the synthetic test set when the model is trained with and without temporal consistency. **RGB:** input colour image; **GTD:** Ground Truth Depth; **GTS:** Ground Truth Segmentation; **TS:** Temporal Segmentation; **TD:** Temporal Depth; **NS:** Non-Temporal Segmentation; **ND:** Non-Temporal Depth.

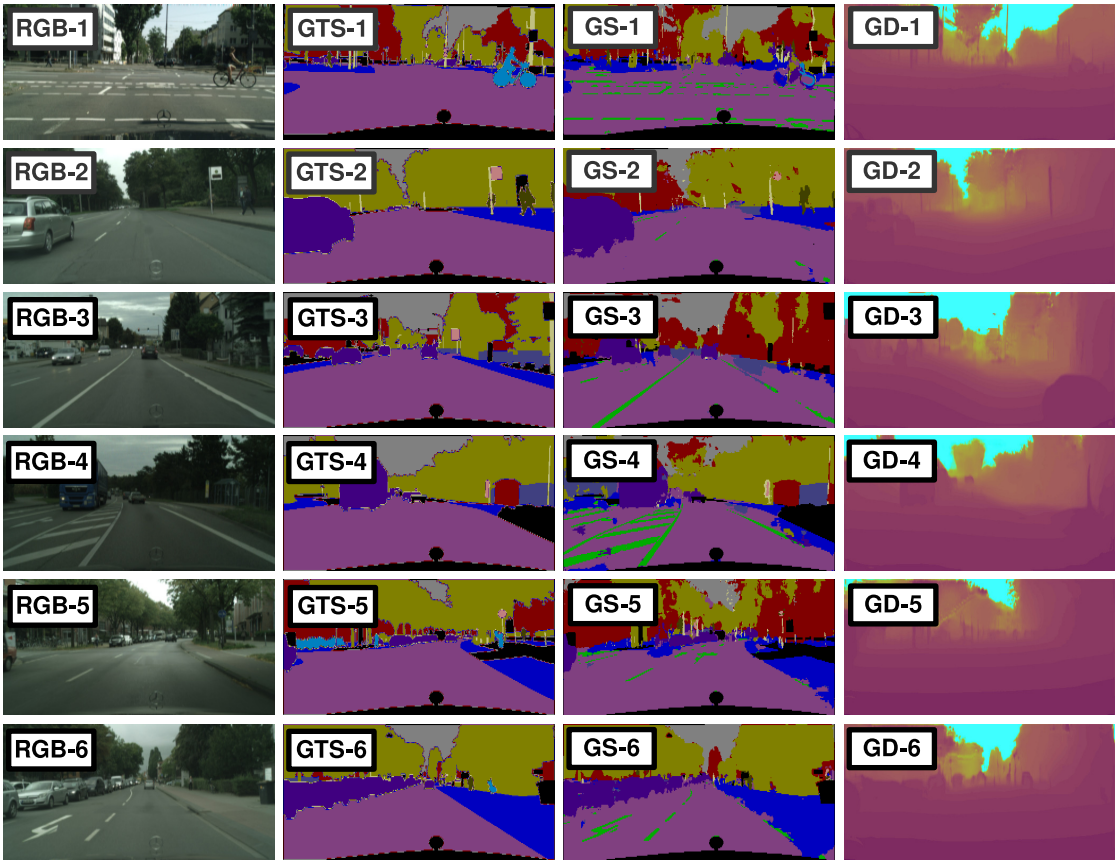

Figure 6.15: Results of the proposed approach on the on the Cityscapes dataset [339]. **RGB:** input colour image; **GTS:** Ground Truth Segmentation; **GS:** Generated Segmentation; **GD:** Generated Depth.

Method	Depth Error (lower, better)				Depth Accuracy (higher, better)			Segmentation (higher, better)	
	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$	Accuracy	IoU
T/R	0.991	1.964	7.393	0.402	0.598	0.684	0.698	0.156	0.335
T/R/A	0.851	1.798	6.826	0.368	0.692	0.750	0.778	0.341	0.435
T/R/A/SC	0.655	1.616	6.473	0.278	0.753	0.812	0.838	0.669	0.738
T/R/A/SC/S	0.412	1.573	6.256	0.258	0.793	0.875	0.887	0.693	0.741
N/R/A/SC/S	0.534	1.602	6.469	0.275	0.758	0.820	0.856	0.614	0.681
T/R/A/SC/S/OF	**0.208**	**1.402**	**6.026**	**0.269**	**0.836**	**0.901**	**0.926**	**0.748**	**0.764**

Table 6.5: Numerical results with different components of loss. **T:** Temporal training; **T:** Non-Temporal training; **R:** Reconstruction loss; **A:** Adversarial loss; **SC:** Skip Connections; **S:** Smoothing loss; **OF:** Optical Flow.

discriminator (D) is adversarially trained to distinguish fake samples $\tilde{y}$ from ground truth samples y. The adversarial loss is thus as follows:

$$\mathcal{L}_{adv} = \min_{G_1} \max_{D} \mathop{\mathbb{E}}_{x,y\sim\mathbb{P}_d(x,y)} [logD(x,y)] + \mathop{\mathbb{E}}_{x\sim\mathbb{P}_d(x)} [log(1 - D(x, G_1(x)))], \tag{6.10}$$

where $\mathbb{P}_d$ is the data distribution defined by $\tilde{y} = G_1(x)$, with x being the generator input and y the ground truth.

Additionally, a smoothing term [47, 357] is utilised to encourage the model to generate more locally-smooth depth outputs. Output depth gradients ($\partial G_1(x)$) are penalised using $L1$ regularisation and an edge-aware weighting term based on input image gradients (∂x) is used since image gradients are stronger where depth discontinuities are most likely found. The smoothing loss is therefore as follows:

$$\mathcal{L}_s = |\partial G_1(x)| e^{||\partial x||}, \tag{6.11}$$

where x is the input and $G_1(x)$ the depth output. The gradients are summed over vertical and horizontal axes.

Another important consideration is ensuring the depth outputs are temporally consistent. While the model is capable of implicitly learning temporal continuity when the output at each time step is recurrently used as the input at the next time step, we incorporate a light-weight pre-trained optical flow network [271], which utilises a coarse-to-fine spatial pyramid to learn residual flow at each scale, into our pipeline to explicitly enforce consistency in the presence of camera or scene motion.

At each time step n, the flow between the ground truth depth frames n and $n-1$ is estimated using our pre-trained optical flow network [271] as well as the flow between generated outputs from the same frames.

The gradients from the optical flow network (F) are used to train the generator (G_1) to capture motion information and temporal continuity by minimising the End Point Error (EPE) between the produced flows. Hence, the last component of our loss function is:

$$\mathcal{L}_{V_n} = ||F(G_1(x_n), G_1(x_{n-1})) - F(y_n, y_{n-1})||_2, \tag{6.12}$$

where x and y are input and ground truth depth images respectively and n the time step. While we utilise ground truth depth as inputs to the optical flow network, colour images can also be equally viable inputs. However, since our training data contains noisy environmental elements (e.g. lighting variations, rain, etc.), using the sharp and clean depth images leads to more desirable results.

Within the final decoder used exclusively for depth prediction, outputs are produced at four scales, following [47]. Each scale output is twice the spatial resolution of its previous scale. The overall depth loss is therefore the sum of losses calculated at every scale c:

$$\mathcal{L}_{depth} = \sum_{c=1}^{4}(\lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_s\mathcal{L}_s + \lambda_V\mathcal{L}_{V_n}). \tag{6.13}$$

The weighting coefficients (λ) are empirically selected (Section 6.2.4). These loss components, used to optimise depth fidelity, are used alongside the semantic segmentation loss, explained in Section 6.2.3.

6.2.3 Semantic Segmentation

As semantic segmentation is not the main focus of our approach but only used to enforce deeper and better representation learning within our model, we opt for a simple and efficient fully-supervised training procedure for our segmentation (G_2). The RGB or RGB-D image is used as the input and the network outputs class labels. Pixel-wise softmax with cross-entropy is used as the loss function, with the loss summed over all the pixels within a batch:

$$P_k(x) = \frac{e^{a_k(x)}}{\sum_{k'=1}^{K} e^{a_{k'}(x)}}, \tag{6.14}$$

$$\mathcal{L}_{seg} = -log(P_l(G_2(x))), \tag{6.15}$$

where $G_2(x)$ denotes the network output for the segmentation task, $a_k(x)$ is the feature activation for

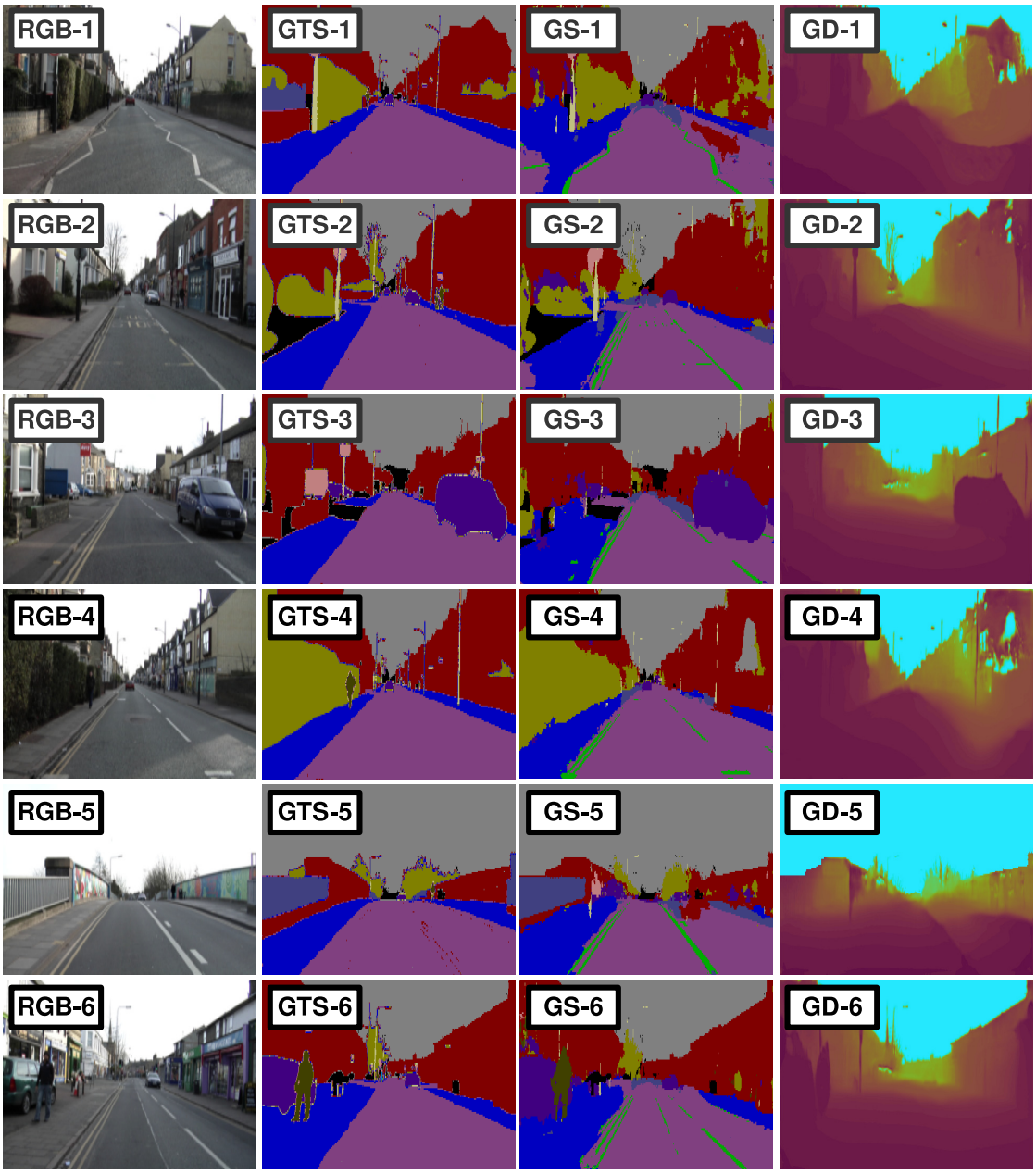

Figure 6.16: Results of the proposed approach on the CamVid dataset [358]. **RGB:** input colour image; **GTS:** Ground Truth Segmentation; **GS:** Generated Segmentation; **GD:** Generated Depth.

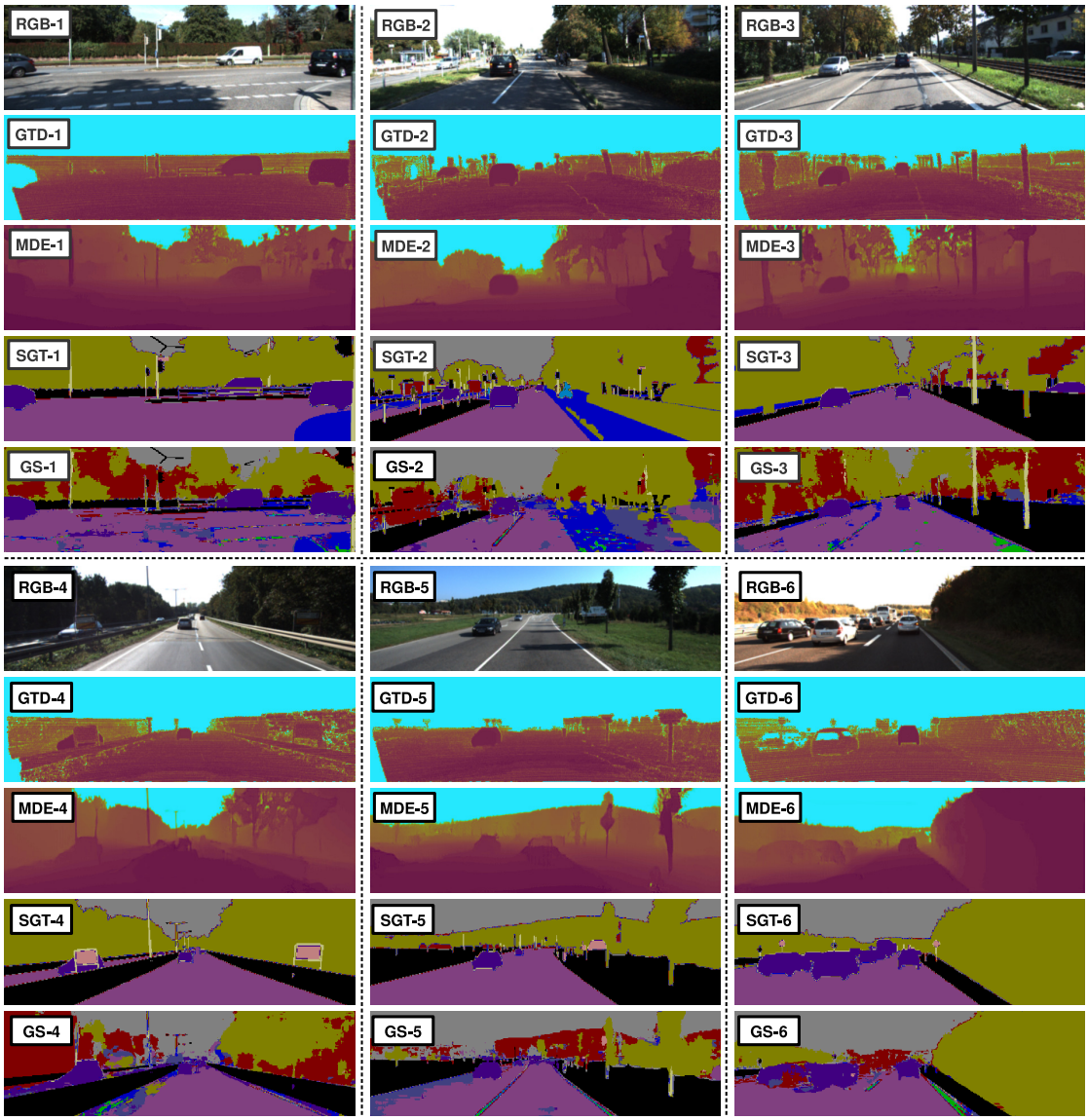

Figure 6.17: Results of our approach applied to KITTI [359, 360]. **RGB:** input colour image; **GTD:** Ground Truth Depth; **MDE:** Monocular Depth Estimation; **GTS:** Ground Truth Segmentation; **GS:** Generated Segmentation.

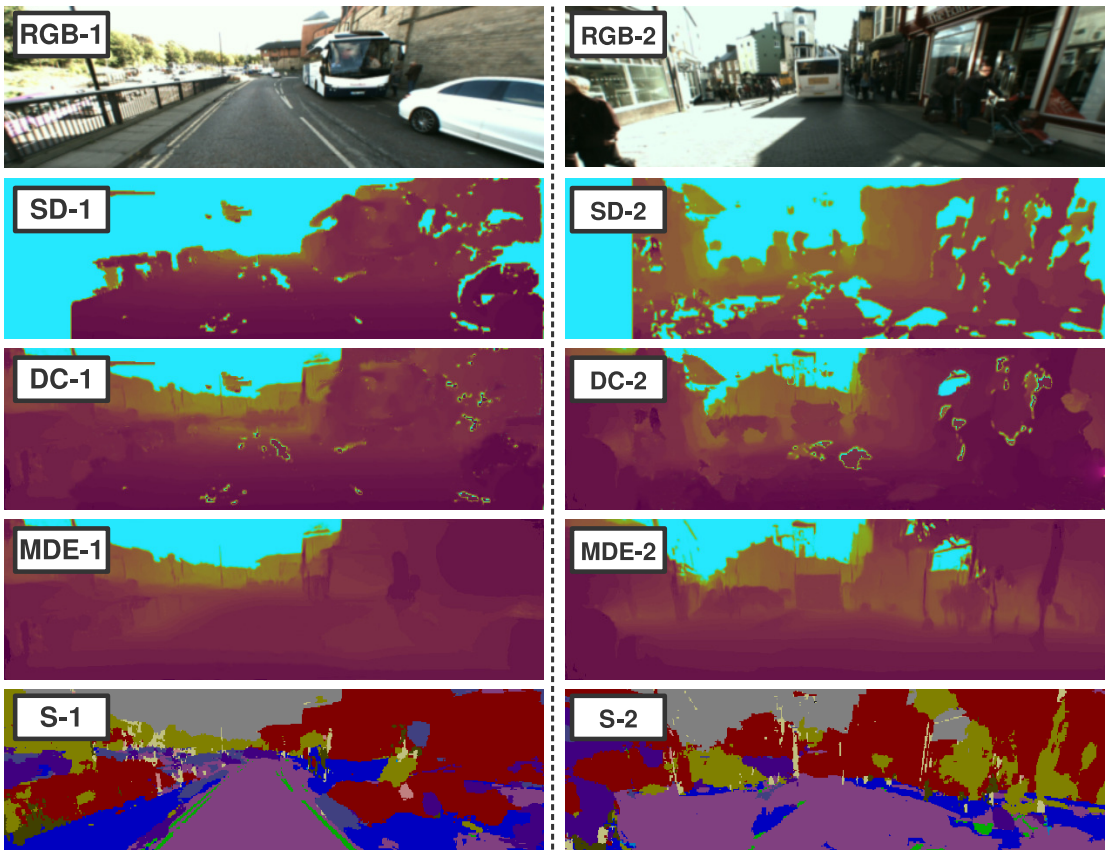

Figure 6.18: Our results on locally captured data. **SD:** Depth via Stereo Correspondence; **DC:** Depth Completion; **MDE:** Monocular Depth Estimation; **S:** Semantic Segmentation.

channel k, K is the number of classes, $P_k(x)$ is the approximated maximum function and l is the ground truth label for image pixels. The loss is summed for all pixels within the images.

Finally, since the entire network is trained as one unit, the joint loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{depth} + \lambda_{rec}\mathcal{L}_{seg}. \tag{6.16}$$

with coefficients selected empirically (Section 6.2.4).

Method	IoU	Method	IoU
CRF-RNN [362]	62.5	DeepLab [274]	63.1
Pixel-level Encoding [363]	64.3	FCN-8s [273]	65.3
DPN [364]	66.8	Our Approach	**67.0**

Table 6.6: Segmentation results of the approach on the Cityscapes [339] test set.

6.2.4 Implementation Details

Synthetic data [39] consisting of RGB, depth and class labels are used for training. The discriminator follows the architecture of [343] and the optical flow network [271] is pre-trained on the KITTI dataset [360]. Experiments with the Sintel dataset [361] returned similar, albeit slightly inferior, results.

The discriminator uses convolution-BatchNorm-leaky ReLU ($slope = 0.2$) modules. The dataset [39] contains numerous sequences some spanning thousands of frames. However, a feedback network taking in high-resolution images (512×128) back-propagating over thousands of time steps is intractable to train. Empirically, we found training over sequences of 10 frames offers a reasonable trade-off between accuracy and training efficiency. Mini-batches are loaded in as tensors containing two sequences of 10 frames each, resulting in roughly $10,000$ batches overall.

All implementation is done in *PyTorch* [324], with Adam [326] providing the best optimisation ($\beta_1 = 0.5$, $\beta_2 = 0.999$, $\alpha = 0.0002$). The weighting coefficients in the loss function are empirically chosen to be $\lambda_{rec} = 1000, \lambda_{adv} = 100, \lambda_s = 10, \lambda_V = 1, \lambda_{seg} = 10$.

6.2.5 Experimental Results

We assess our approach using ablation studies and both qualitative and quantitative comparisons with state-of-the-art methods applied to publicly available datasets [301, 339, 358–360]. We also utilise our own synthetic test set and data captured locally to further evaluate the approach.

Ablation Studies

A crucial part of our work is demonstrating that every component of the approach is integral to the overall performance. We train our model to perform two tasks based on the assumption that the network is forced to learn more about the scene if different objectives are to be accomplished. We demonstrate this by training one model performing both tasks and two separate models focusing on each and conducting tests on randomly selected synthetic sequences [39]. As seen in Table 6.4, both tasks (monocular depth

Figure 6.19: Comparison of various completion methods applied to the synthetic test set. **RGB:** input colour image; **GTD:** Ground Truth Depth; **DH:** Depth Holes; **FBI:** Fourier-based inpainting presented in Section 3.1; **GLC:** Global and Local Completion [109]; **ICA:** Inpainting with Contextual Attention [110]; **GIF:** Guided Inpainting and Filtering [41].

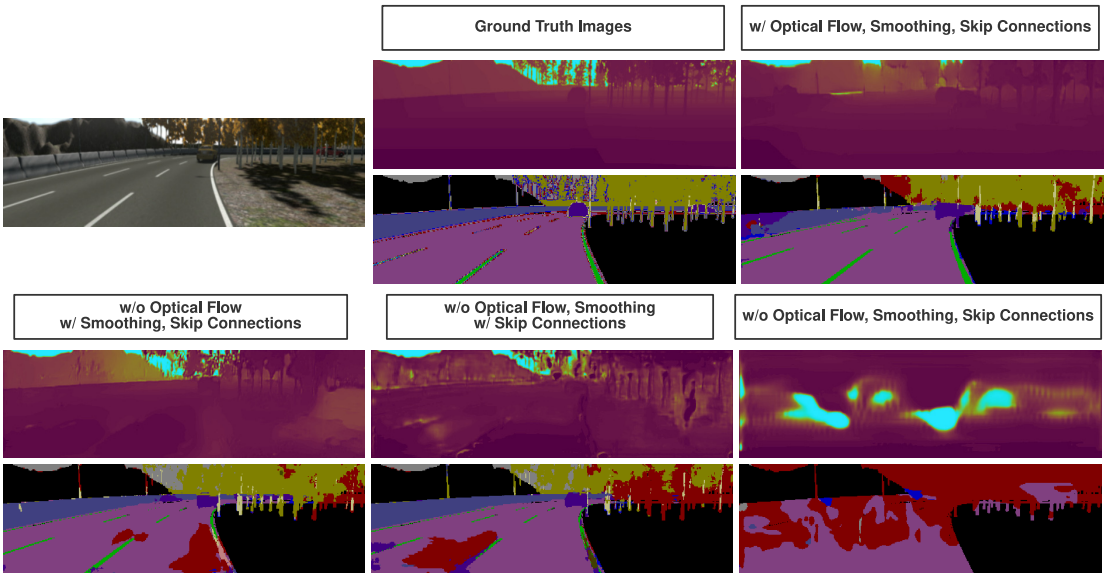

Figure 6.20: Comparing the performance of the approach with differing components of the loss function removed.

estimation and semantic segmentation) perform better when the model is trained on both. Moreover, since the segmentation pipeline does not receive any explicit temporal supervision (from the temporal flow network) and its temporal continuity is only enforced by the input and middle streams trained by the depth pipeline, when the two pipelines are disentangled, the segmentation results become far worse than the depth results.

Figure 6.14 depicts the quality of the outputs when the model is a feedback network trained temporally compared to our model when the output depth from the previous time step is not used as the input during training. We can clearly see that both depth and segmentation results are of higher fidelity when temporal information is used during training.

Additionally, our depth prediction pipeline uses several loss functions. We employ the same test sequences to evaluate our model trained as different components are removed. Table 6.5 demonstrates the network temporally trained with all the loss components (T/R/A/SC/S/OF) outperforms models trained without specific ones. Qualitatively, we can see in Figure 6.20 that the results are far better when the network is fully trained with all the components. Specifically, the set of skip connections used in the network make a significant difference in the quality of the outputs.

Semantic Segmentation

Segmentation is not the focus of this work and is mainly used to boost the performance of depth prediction. However, we extensively evaluate our segmentation pipeline which outperforms several well-known comparators. We utilise Cityscapes [339] and CamVid [358] test sets for our performance evaluation despite the fact that our model is solely trained on synthetic data and *without any domain adaptation* should not be expected to perform well on naturally sensed real-world data. The effective performance of our segmentation points to the generalisation capabilities of our model. When tested on CamVid [358], our approach produces better results compared to well-established techniques such as [277, 305, 365, 366] despite the lower quality of the input images as seen in Table 6.7. As for Cityscapes [339], the test set does not contain video sequences but our temporal model still outperforms approaches such as [273, 274, 362–364], as demonstrated in Table 6.6.

Examples of the segmentation results over both Cityscapes [339] and CamVid [358] datasets are respectively seen in Figures 6.15 and 6.16. As can be seen, despite the approach not being primarily designed to perform semantic segmentation, the results are very promising and with minimal inaccuracies and artefacts. Additionally, we also used the KITTI semantic segmentation data [359] in our tests and as shown in Figure 6.17, our approach produces high fidelity semantic class labels despite including *no domain adaptation*.

Depth Completion

Evaluation for depth completion ideally requires dense ground truth scene depth. However, no such dataset exists for urban driving scenarios, which is why we utilise randomly selected previously unseen synthetic data with available dense depth images to assess the results. Our model generates full scene depth and the predicted depth values for the missing regions of the depth image are subsequently blended in with the known regions of the image using [323]. Figure 6.19 shows a comparison of our results against other contemporary approaches [37, 41, 109, 110]. As seen from the enlarged sections, our approach produces minimal artefacts (blurring, streaking, etc.) compared to the other techniques. To evaluate

Method	IoU	Method	IoU
SegNet-Basic [277]	46.4	DeconvNet [365]	48.9
SegNet [277]	50.2	Bayesian SegNet-Basic [305]	55.8
Reseg [366]	58.8	Our Approach	**59.1**

Table 6.7: Segmentation results of the approach on the CamVid [358] test set.

Method	PSNR	SSIM	Method	PSNR	SSIM
Holes	33.73	0.372	GTS [109]	31.47	0.672
ICA [110]	31.01	0.488	GIF [41]	44.57	0.972
FDF [37]	46.13	0.986	Ours	**47.45**	**0.991**

Table 6.8: Structural integrity analysis post depth completion.

Method		Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
		Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Train Set Mean	[301]	0.403	0.530	8.709	0.403	0.593	0.776	0.878
Eigen et al.	[193]	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al.	[190]	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Zhou et al.	[216]	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Godard et al.	[47]	0.148	**1.344**	5.927	0.247	0.803	0.922	0.964
Zhan et al.	[48]	**0.144**	1.391	**5.869**	0.241	0.803	0.928	**0.969**
Our Approach		0.193	1.438	5.887	**0.234**	**0.836**	**0.930**	0.958

Table 6.9: Numerical comparison of monocular depth estimation over the KITTI [301] data split in [193]. All comparators are trained and tested on the same dataset (KITTI [301]) while our approach is trained on [39] and tested using [301].

the structural integrity of the results post completion, we also numerically assess the performance of our approach and the comparators. As seen in Table 6.8, our approach quantitatively outperforms the comparators as well.

While blending [323] might work well for colour images with a connected missing region, significant quantities of small and large holes in depth images can lead to undesirable artefacts such as stitch mark or burning effects post blending. Examples of artefacts can be seen in Figure 6.18, which demonstrates the results of the approach applied to locally captured data (Durham, UK). This is further discussed in Section 6.2.6.

Monocular Depth Estimation

As the main focus of our model, our monocular depth estimation model is evaluated against contemporary state-of-the-art approaches [47, 48, 190, 193, 216]. Following the conventions of the literature, we use the data split suggested in [193] as the test set. These images are selected from random sequences and do not follow a temporally sequential pattern, while our full approach requires video sequences as its input. As a result, we apply our approach to all the sequences from which the images are chosen but the evaluation itself is only performed on the 697 test images.

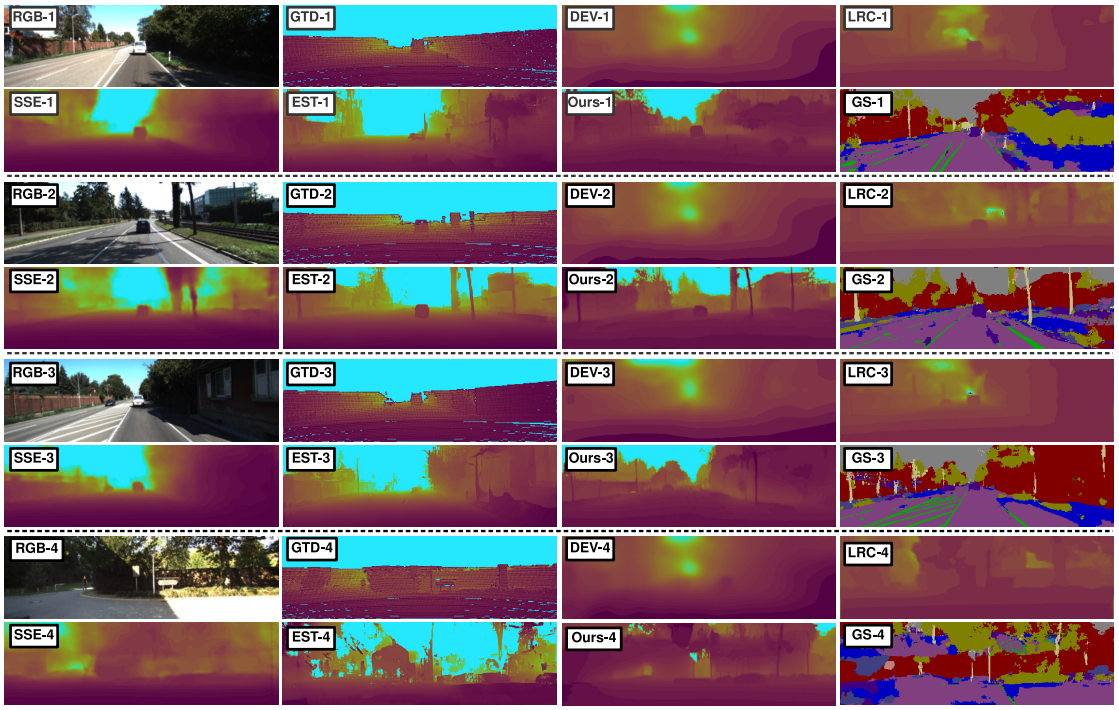

Figure 6.21: Comparing the results of the approach against [47, 216, 217]. Images have been adjusted for better visualisation. **RGB:** input colour image; **GTD:** Ground Truth Depth; **DEV:** Depth and Ego-motion from Video [216]; **LRC:** Left-Right Consistency [47]; **SSE:** Semi-supervised Estimation [217]; **EST:** the Estimation via Style Transfer approach presented in Section 6.1; **GS:** Generated Segmentation.

For numerical assessment, the generated depth is corrected for the differences in focal length between the training [39] and testing data [301]. As seen in Table 6.9, our approach outperforms [190, 193, 216] across all metrics and stays competitive with [47, 48]. It is important to note that all of these comparators are trained on the *same* dataset as the one used for testing [301] while our approach is trained on synthetic data [39] *without domain adaptation* and has not seen a single image from [301]. Additionally, none of the other comparators is capable of producing temporally consistent outputs as all of them operate on a frame level.

We also assess our model using the data split of KITTI [360] and qualitatively evaluate the results, since the ground truth images in [360] are of higher quality than the laser data and provide CAD models as replacements for the cars in the scene. As shown in Figure 6.17, our method produces sharp and crisp depth outputs with segmentation results in which object boundaries and thin structures are well preserved.

6.2.6 Limitations

Even though our approach can generate temporally consistent depth and segmentation by utilising a feedback network, this can lead to error propagation (i.e. when an erroneous output is generated at one time step, the invalid values will continually propagate to future frames). This can be resolved by exploring the use of 3D convolutions or regularisation terms aimed at penalising propagated invalid outputs. Moreover, as mentioned in Section 6.2.5, blending the depth output into the known regions of the depth [323] produces undesirable artefacts in the results. This can be rectified by incorporating the blending operation into the training procedure.

In other words, the blending itself will take place before the supervisory signal is back-propagated through the network during training, which would force the network to learn these artefacts, removing any need for post-processing. As for our segmentation component, no explicit temporal consistency enforcement or class balancing is performed, which has lead to frame-to-frame flickering and lower accuracy with unbalanced classes (e.g. pedestrians, cyclists). By improving segmentation, the entire model can potentially benefit from a significant performance boost.

6.2.7 Summary

The approach presented in Section 6.2 is based on a multi-task model capable of performing depth prediction and semantic segmentation in a temporally-consistent manner using a feedback network that takes as its recurrent input the output generated at the previous time step.

In addition to the use of the recurrent network, temporal consistency is further enforced using a pre-trained frozen optical flow network [271]. This network receives generated and ground truth depth images from consecutive frames and its gradients are used to train the primary network as we attempt to minimise the distance between the flow vectors it generates.

In this approach, depth prediction is considered within the areas of depth completion and monocular depth estimation and therefore separate models are trained to achieve both objectives. Our evaluations demonstrate that our proposed approach can generate more plausible and accurate results when the tasks of depth prediction and semantic segmentation are performed at the same time. The use of skip connections is also demonstrated to be highly effective in improving the results for both depth prediction and semantic scene segmentation.

Our experiments demonstrate the efficacy of our proposed approach and its capability to outperform prior work across the domains of monocular depth estimation [33, 47, 48, 190, 193, 216], depth completion [37, 41, 109, 110] and semantic segmentation [273, 274, 277, 305, 362–366].

Chapter 7

Conclusions

Accurate three-dimensional scene understanding is now a crucial component of many commercial and prototypical vision-based and robotic systems such as autonomous driving [367], augmented reality [368], environment modelling [369] and alike. Consequently, as a vital ingredient of such applications, acquiring complete scene depth information has become the basis for the emergence of an active area of research in recent years.

Despite the expansive and fast-growing literature within the field, most existing 3D scene understanding approaches often deal with challenges arising from incomplete, corrupt, noisy and partially invalid scene depth. In this thesis, we have explored the use of various techniques to complete and refine depth images captured via common consumer scene sensing devices and the possibility of estimating complete depth from a single RGB view of the scene without the need for any refinement.

Succinctly put, the focus of this thesis has been to provide answers to the following research questions:

- Can we provide the entire scene depth captured through imperfect and flawed means by completing the missing regions of the acquired depth in a post-processing stage?
- Can we obtain the entire scene depth without any need for post processing from a single RGB image by learning about content, context and other cues present within the scene?

In the earlier parts of this thesis, we have primarily focused on answering the first research question; the possibility of plausibly and accurately completing existing scene depth with missing regions by proposing four different depth completion techniques (Chapters 3, 4 and 5). Inspired by various RGB inpainting methods [42, 53, 60, 67, 104, 106, 110], our proposed approaches are purpose-built to tackle the problem

of depth completion, taking into account different application requirements such as accuracy, efficiency, underlying scene geometry, surface detail, texture and alike.

In later portions of the thesis, we have focused on answering the second research question posed above, as the natural progression of this field of research would be and has been to circumvent the need for any depth completion as a post-processing stage and to produce complete and accurate depth information in the first place. With that in mind, and encouraged by the recent advances in learning-based depth estimation methods [47, 190, 191, 216], we have presented two monocular depth estimation techniques which are capable of producing complete scene depth from a single RGB image (Chapter 6).

As a result of the research carried out for this thesis, various contributions have been made to the literature on the areas of depth completion and estimation. In the following, we will briefly revisit the key contributions made within this thesis.

7.1 Contributions

The research conducted as part of the preparation of this thesis began with a completion approach based on an exemplar-based framework aimed at object removal and depth completion within the context of RGB-D images. The proposed approach (Chapter 3 - Section 3.1) focuses on filling holes within RGB-D imagery, resulting from either object removal [129] or limitations in 3D sensor capabilities [44, 151, 155], uniquely taking advantage of the Fourier space to facilitate independent completion of both high frequency depth detail (relief) and low frequency surface continuation (shape). As its first step, the approach employs the Discrete Fourier Transform to decompose a depth image into separate high and low frequency components via Butterworth filtering. An improved texture synthesis method is then proposed to fill the high frequency depth detail and a structural inpainting approach is used to complete the underlying structures in the image. The two are then recombined in Fourier space to create a depth image, where the holes are plausibly filled and the surface relief and edges preserved.

Through detailed evaluations, we have demonstrated that the approach proposed in Section 3.1 is capable of object removal and hole filling in RGB-D images in a plausible and accurate manner, outperforming comparators [41, 60, 66, 68, 125] used to achieve the same objectives. However, the primary challenge the approach faces stems from its computational requirements and slow run-time. This prompted us to shift our attention towards a more balanced approach focusing on the compromise between accuracy and efficiency.

As a result, we arrive at the second approach proposed in Chapter 3, which addresses the problem of depth completion in an exemplar-based framework with a focus on a more balanced trade-off between efficiency and attention to surface (relief) detail accuracy. While exemplar-based methods are mostly

used for colour images, their ability to preserve texture in the target region makes them suitable for depth filling when texture is of importance. In this approach, the priority term that determines the order of patch sampling has been modified to allow for a better propagation of strong linear structures and texture into the target region. Moreover, by constraining the query space, the method performs more efficiently than other exemplar-based approaches [60, 66].

Our experiments demonstrate that not only is the efficiency of the proposed approach superior compared to many of its predecessors [60, 66], the plausibility and accuracy of the depth outputs rival those of many contemporary completion approaches within the field [41, 60, 66, 68, 125]. However, despite the improved efficiency of the approach proposed in Section 3.2, neither of the approaches in Chapter 3 can be used in real-time applications due to their slow run-time.

Consequently, in Chapter 4, the problem of depth completion is addressed by proposing a significantly more efficient approach which can potentially be used in real time. The real-time depth completion technique is designed with a focus on efficiency and attention to surface (relief) detail accuracy, with reference to a prior object-wise scene labelling. This first step of the approach requires an accurate semantic segmentation [277] over an accompanying colour image, which is commonly available from contemporary sensing arrangements, to facilitate depth completion on an object-wise basis. Missing depth values are subsequently filled via a three-pass non-parametrically driven approach, using a grammar of twelve discrete completion case occurrences.

Experiments demonstrate that while the efficiency of the proposed method is comparable to simple interpolation methods, the approach can outperform contemporary techniques such as [41, 66, 68, 125] and the two approaches proposed in Chapter 3. Despite its strengths, however, this approach is not capable of completing very large depth holes. A possible solution to this challenge can be found in a learning-based depth completion approach capable of filling larger holes by synthesising new content for the missing sections of the depth image.

Subsequently, we approach the problem of depth completion from a learning perspective in Chapter 5 by employing an adversarially trained encoder/decoder architecture. It is expected that if enough is learned about the contents and semantics of a scene, large missing regions of a depth image can be inferred given its known regions and the full RGB view. The training is fully self-supervised, without the need for any annotation or human intervention. The ground truth depth used for training is acquired from a graphical environment developed for gaming [38] and a separate model is trained to infer where holes would be if the data were obtained via stereo correspondence. The model objective is to minimise a loss consisting of four loss components: reconstruction, adversarial, bottleneck feature and domain transfer loss, which results in filling depth holes, not only in synthetic depth images but also in real-world data with no ground truth.

Even though the approach utilises synthetic images for training and requires a complicated mixture of parameters with their own weighting coefficients, qualitative and quantitative evaluations demonstrate how it can outperform competing contemporary depth filling techniques [41, 66, 68, 125]. Moreover, we hypothesise that due to its complicated learning procedure, the model contains highly robust feature learning capabilities. To test this, with minimal transfer learning, we apply the network used within this depth completion approach to perform monocular depth estimation, a task it is not primarily designed or trained to perform. The resulting monocular depth estimation model originally trained to perform depth completion can remarkably outperform some of contemporary monocular depth estimation techniques [47, 216], mainly due to its accurate synthetic training data, complex training procedure and domain adaptation capabilities.

As a result of this success in monocular depth estimation, our focus within the overall three-dimensional scene understanding paradigm is slightly shifted towards circumnavigating the need for depth completion by investigating the possibility of using efficient learning-based techniques capable of estimating complete hole-free depth from a single RGB image. In this vein, two novel monocular depth estimation approaches are presented in Chapter 6.

Using synthetic data captured from the same graphically rendered urban environment as the one used in Chapter 5, the first approach presented in Chapter 6 focuses on an effective depth estimation model trained in a supervised manner. However, any such model trained on synthetic data cannot perform well on real-world data as the domain distributions to which these two sets of data belong are widely different. Relying on new theoretical studies connecting style transfer and domain adaptation [230], the proposed method relies on a GAN-based style transfer approach [315] to adapt the real-world data to fit into the distribution approximated by the generator in the depth estimation model. Despite certain isolated issues, experimental results prove the superiority of this monocular depth estimation approach both in terms of visual quality and numerical accuracy compared to contemporary state-of-the-art comparators [46–48, 190, 191, 216].

In order to extend the capabilities of the model with our primary focus still on depth estimation, we present the second approach in Chapter 6, which contains a multi-task model capable of performing both depth prediction and semantic segmentation in a temporally-consistent manner using a feedback network that takes as its recurrent input the output generated at the previous time step. Using a series of dense skip connections, we ensure that no high-frequency spatial information is lost during feature down-sampling within the training process.

Using extensive experimentation, we demonstrate that our model achieves much better results when it performs depth prediction and segmentation at the same time compared to two separate networks performing the same tasks. The use of skip connections is also illustrated to be significantly effective

in improving the results for both depth prediction and segmentation tasks. Additionally, since this multi-task objective can provide the model with a better geometric and semantic understanding of the scene, the model is able to learn more crucial details about the context and the content of the scene and can subsequently achieve impressive results when applied to natural real-world data even without domain adaptation. Experimental evaluation superior performance of this compared to prior work across the domains of monocular depth estimation [33, 47, 48, 190, 193, 216], depth completion [37, 41, 109, 110] and semantic segmentation [273, 274, 277, 305, 362–366].

Overall, the aim of this thesis has been to demonstrate the possibility of achieving the entire scene depth captured through imperfect and flawed means by completing the missing regions of the acquired depth in a post-processing stage and obtaining the entire scene depth without any need for post processing from a single RGB image by learning about content, context and other cues present within the scene. By presenting four depth completion techniques in Chapters 3, 4 and 5 and two monocular depth estimation techniques in 6, we believe we have sufficiently investigated the possibilities of recovering complete three-dimensional scene information by means of producing complete depth images through completion and estimation techniques, thus answering the primary research questions posed in Chapter 1.

7.2 Future Work

Although we have demonstrated the capabilities of our novel depth completion and monocular depth estimation approaches in recovering complete and accurate scene depth, there remains areas in which this work, not unlike any other work, can be improved. Even though we have clearly discussed the limitations of every proposed approach in their corresponding chapters, in the following, we will focus on more general aspects where our work can be built upon and the directions which future research can take.

7.2.1 Computational Efficiency

One of the greatest issues facing depth prediction approaches is their computational requirements. Since any depth processing method, be it stereo matching [13], depth completion [31], monocular depth estimation [47, 48, 216] or any other similar approach [49, 51], needs to perform in real-time to provide immediate accurate data for other downstream applications, efficiency is of significant importance.

There are currently many commercial and industrial systems that heavily rely on complete and accurate scene depth for an acceptable level of performance within their pipeline. Some of these include of various applications within robotics to achieve better navigation, localisation and mapping, advances in facial recognition for security and surveillance purposes, gesture and behaviour recognition for improved entertainment systems, among others. With real-time capabilities, an efficient depth estimation or completion

technique can be used to supply such applications with accurate depth information, thus improving the overall quality of the system, making computational efficiency a high-priority and impactful direction for future research.

In Chapter 4, we have proposed an approach focusing on efficiency within the context of depth completion. While this approach is highly efficient and comparable with a simple bilinear interpolation technique, it suffers from its own flaws (Section 4.4) and cannot be directly deployed in real-world applications. Additionally, novel learning-based depth prediction approaches, such as those presented in Chapter 6 can process images in real time or near real time, relying on modern advances in graphical hardware. However, deploying such expensive and fragile hardware in consumer systems is problematic and non-economical.

Consequently, creating simpler yet effective approaches with a smaller number of parameters, specific implementations with efficiency as the key objective, optimising depth prediction on a hardware level, and alike can all be great steps towards future research on scene depth recovery.

7.2.2 Merging Existing and Estimated Depth

Many vision-based systems currently deployed in real-world applications make use of flawed, imperfect and sometimes very expensive depth sensing technologies such as stereo correspondence and LiDAR. Most existing work in the current academic literature is focused on proposing more economical and efficient alternative approaches to obtaining accurate scene depth.

Depth completion techniques similar to those proposed in Chapters 3, 4 and 5 can be a great solution to providing complete and dense depth information of the entire scene after partial depth has been captured via other means [13, 49]. However, the necessity of applying such completion methods as a post-processing operation is itself an obstacle to achieving a light-weight real-time system.

Consequently, monocular depth estimation approaches (Chapter 6) can be a great stride towards providing a real-time alternative to currently-used depth capture technologies [13, 19, 21, 26]. However, despite their ability to provide complete hole-free scene depth, the resulting information output by such monocular depth estimation techniques usually suffers from its own particular challenges. While the depth information estimated for every single object, instance or component present within the scene is often highly accurate relative to other scene objects, the overall absolute depth range produced by many of these approaches is unreliable and in need of adjustment post estimation [47, 216, 217].

As part of a future work, the approaches presented in this thesis, especially the monocular depth estimation techniques proposed in Chapter 6, can be extended to include depth information captured from

current technologies such as stereo vision systems [13]. Being readily available through inexpensive consumer devices, stereo depth can be used to aid in improving the overall absolute accuracy of the estimated scene depth.

7.2.3 Training Dataset

A significant portion of the work done for this thesis has involved the use of novel state-of-the-art machine learning techniques [97, 105, 315, 317, 370] in order to successfully generate complete and accurate scene depth by learning the context and contents of a scene.

Not unlike most other forms of accurately-collated labelled data, obtaining ground truth dense depth images to train a completion or estimation model is very difficult, highly expensive and time-consuming. Since any supervised training procedure requires very large quantities of training data, one of the biggest challenges any learning-based depth prediction model faces is the lack of data. Although it is possible to acquire depth outputs in a learning-based pipeline without using ground truth dense depth information [47, 191, 192, 216], such approaches can suffer from issues depending on the objective function they utilise during training (e.g. blurring, unwanted artefacts, random content in the presence of occlusion and alike).

In Chapters 5 and 6, we have opted for utilising synthetic images to fill the gap created by the scarcity of naturally-sensed ground truth depth images. The data is captured from different virtual environments designed for research or commercial purposes. However, this leads to issue of domain shift, which is a significant problem as in practice, any depth prediction approach needs to function well in the presence of real-world images.

By introducing enough variation into the training dataset, i.e. using several different virtual environments, randomising texture and incorporating sufficient quantities of real-world data, a more robust model is created that can better encounter the issue of domain variation. In this vein, an interesting direction for future research can be a method of randomising content captured from a simulator or virtual environment [38, 330] in a semantically meaningful manner guided by a secondary attention model [371] trained to focus the content randomisation process towards regions of the image that lead to a larger amounts of domain shift.

Bibliography

1. Tippetts, B., Lee, D. J., Lillywhite, K. & Archibald, J. Review of Stereo Vision Algorithms and their Suitability for Resource-Limited Systems. *Real-Time Image Processing* **11,** 5–25 (2016) (*pp.* 2, 8).
2. Zhang, Z. Microsoft Kinect Sensor and its Effect. *IEEE Multimedia* **19,** 4–10 (2012) (*pp.* 2–3, 8).
3. Cong, P., Xiong, Z., Zhang, Y., Zhao, S. & Wu, F. Accurate Dynamic 3D Sensing with Fourier-Assisted Phase Shifting. *Selected Topics in Signal Processing* **9,** 396–408 (2015) (*pp.* 2, 8).
4. Yang, Q., Tan, K.-H., Culbertson, B. & Apostolopoulos, J. *Fusion of Active and Passive Sensors for Fast 3D Capture* in *Int. Workshop on Multimedia Signal Processing* (2010), 69–74 (*pp.* 2, 8).
5. Gudmundsson, S. A., Aanaes, H. & Larsen, R. Fusion of Stereo Vision and Time-of-Flight Imaging for Improved 3D Estimation. *Intelligent Systems Technologies and Applications* **5,** 425–433 (2008) (*pp.* 2, 8).
6. Cruz, L., Lucio, D. & Velho, L. *Kinect and RGB-D Images: Challenges and Applications* in *Conf. Graphics, Patterns and Images Tutorials* (2012), 36–49 (*pp.* 2, 8).
7. Hughes, J. F., Van Dam, A., Foley, J. D., McGuire, M., Feiner, S. K., Sklar, D. F. & Akeley, K. *Computer Graphics: Principles and Practice* (Pearson Education, 2014) (*p.* 2).
8. Fishman, E. K., Ney, D. R., Heath, D. G., Corl, F. M., Horton, K. M. & Johnson, P. T. Volume Rendering versus Maximum Intensity Projection in CT Angiography: What Works Best, When, and Why. *Radiographics* **26,** 905–922 (2006) (*p.* 2).
9. Breckon, T. P. & Fisher, R. B. Amodal Volume Completion: 3D Visual Completion. *Computer Vision and Image Understanding* **99,** 499–526 (2005) (*pp.* 2, 10).
10. Rom, M. & Brakhage, K.-H. *Volume Mesh Generation for Numerical Flow Simulations Using Catmull-Clark and Surface Approximation Methods* (Inst. für Geometrie und Praktische Mathematik, 2011) (*p.* 2).

11. Rusinkiewicz, S. & Levoy, M. *QSplat: A Multiresolution Point Rendering System for Large Meshes* in *Conf. Computer graphics and interactive techniques* (2000), 343–352 (*p.* 2).
12. Hirschmuller, H. Stereo Processing by Semi-Global Matching and Mutual Information. *IEEE Trans. Pattern Analysis and Machine Intelligence* **30,** 328–341 (2008) (*pp.* 3, 89, 93).
13. Scharstein, D. & Szeliski, R. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Int. J. Computer Vision* **47,** 7–42 (2002) (*pp.* 3, 8, 84, 114, 154–156).
14. Seitz, S. M., Curless, B., Diebel, J., Scharstein, D. & Szeliski, R. *A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms* in *IEEE Conf. Computer Vision and Pattern Recognition* **1** (2006), 519–528 (*p.* 3).
15. Brown, L. G. A Survey of Image Registration Techniques. *Computing Surveys* **24,** 325–376 (1992) (*p.* 3).
16. Pages, J., Salvi, J., Garcia, R. & Matabosch, C. *Overview of Coded Light Projection Techniques for Automatic 3D Profiling* in *Int. Conf. Robotics and Automation* **1** (2003), 133–138 (*p.* 3).
17. Han, J., Shao, L., Xu, D. & Shotton, J. Enhanced Computer Vision with Microsoft Kinect Sensor: A Review. *IEEE Trans. Cybernetics* **43,** 1318–1334 (2013) (*p.* 3).
18. Khoshelham, K. *Accuracy Analysis of Kinect Depth Data* in *ISPRS Workshop on Laser Scanning* **38** (2011), W12 (*p.* 3).
19. Kolb, A., Barth, E., Koch, R. & Larsen, R. *Time-of-Flight Cameras in Computer Graphics* in *Computer Graphics Forum* **29** (2010), 141–159 (*pp.* 3, 155).
20. Sell, J. & O'Connor, P. The xbox One System on a Chip and Kinect Sensor. *IEEE Micro* **34,** 44–53 (2014) (*p.* 3).
21. Gokturk, S. B., Yalcin, H. & Bamji, C. *A Time-of-Flight Depth Sensor-System Description, Issues and Solutions* in *Workshop in IEEE Conf. Computer Vision and Pattern Recognition* (2004), 35–35 (*pp.* 3, 155).
22. El-laithy, R. A., Huang, J. & Yeh, M. *Study on the Use of Microsoft Kinect for Robotics Applications* in *Position Location and Navigation Symposium* (2012), 1280–1288 (*p.* 3).
23. Berger, K., Ruhl, K., Schroeder, Y., Bruemmer, C., Scholz, A. & Magnor, M. A. *Markerless Motion Capture using Multiple Color-Depth Sensors* in *Vision Modeling and Visualization* (2011), 317–324 (*p.* 3).
24. Butler, A., Izadi, S., Hilliges, O., Molyneaux, D., Hodges, S. & Kim, D. *Shake'n'Sense: Reducing Interference for Overlapping Structured Light Depth Cameras* in *Conf. Human Factors in Computing Systems* (2012), 1933–1936 (*p.* 3).

25. Sabov, A. & Krüger, J. *Identification and Correction of Flying Pixels in Range Camera Data* in *Conf. Computer Graphics* (2008), 135–142 (*p.* 3).

26. Sarbolandi, H., Lefloch, D. & Kolb, A. Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect. *Computer Vision and Image Understanding* **139,** 1–20 (2015) (*pp.* 3, 155).

27. Hansard, M., Lee, S., Choi, O. & Horaud, R. P. *Time-of-Flight Cameras: Principles, Methods and Applications* (Springer Science & Business Media, 2012) (*p.* 3).

28. Ihrke, I., Kutulakos, K. N., Lensch, H., Magnor, M. & Heidrich, W. *Transparent and Specular Object Reconstruction* in *Computer Graphics Forum* **29** (2010), 2400–2426 (*p.* 3).

29. Ringbeck, T., Möller, T. & Hagebeuker, B. Multidimensional Measurement by using 3D PMD Sensors. *Advances in Radio Science* **5,** 135 (2007) (*p.* 3).

30. Lindner, M., Schiller, I., Kolb, A. & Koch, R. Time-of-Flight Sensor Calibration for Accurate Range Sensing. *Computer Vision and Image Understanding* **114,** 1318–1328 (2010) (*p.* 3).

31. Atapour-Abarghouei, A. & Breckon, T. A Comparative Review of Plausible Hole Filling Strategies in the Context of Scene Depth Image Completion. *Computers and Graphics* **72,** 39–58 (2018) (*pp.* 5, 9, 154).

32. Atapour-Abarghouei, A. & Breckon, T. *Veritatem Dies Aperit - Temporally Consistent Depth Prediction Enabled by a Multi-Task Geometric and Semantic Scene Understanding Approach* in *IEEE Conf. Computer Vision and Pattern Recognition* (IEEE, 2019) (*pp.* 5, 115).

33. Atapour-Abarghouei, A. & Breckon, T. *Real-Time Monocular Depth Estimation using Synthetic Data with Domain Adaptation via Image Style Transfer* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 2800–2810 (*pp.* 5, 8, 115, 149, 154).

34. Atapour-Abarghouei, A., Akcay, S., Payen de La Garanderie, G. & Breckon, T. Generative Adversarial Framework for Depth Filling via Wasserstein Metric, Cosine Transform and Domain Transfer. *Pattern Recognition* **91,** 232–244 (2019) (*pp.* 5, 87).

35. Atapour-Abarghouei, A. & Breckon, T. *Extended Patch Prioritization for Depth Filling Within Constrained Exemplar-Based RGB-D Image Completion* in *Int. Conf. Image Analysis and Recognition* (2018), 306–314 (*pp.* 5, 53).

36. Atapour-Abarghouei, A. & Breckon, T. *DepthComp: Real-Time Depth Image Completion Based on Prior Semantic Scene Segmentation* in *British Machine Vision Conference* (BMVA, 2017), 1–13 (*pp.* 6, 24, 39–40, 71).

37. Atapour-Abarghouei, A., Payen de La Garanderie, G. & Breckon, T. P. *Back to Butterworth - a Fourier Basis for 3D Surface Relief Hole Filling within RGB-D Imagery* in *Int. Conf. Pattern Recognition* (2016), 2813–2818 (*pp.* 6, 39–40, 53, 105–106, 146–147, 149, 154).

38. Miralles, R. *An Open-Source Development Environment for Self-Driving Vehicles* in *Universitat Oberta de Catalunya* (2017), 1–31 (*pp.* 6, 87–88, 92, 116, 129–130, 152, 156).

39. Ros, G., Sellart, L., Materzynska, J., Vazquez, D. & Lopez, A. *The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 3234–3243 (*pp.* 6, 133, 135, 143, 147–148).

40. Lai, P., Tian, D. & Lopez, P. *Depth Map Processing with Iterative Joint Multilateral Filtering* in *Picture Coding Symposium* (2010), 9–12 (*pp.* 8, 24, 28–29, 39).

41. Liu, J., Gong, X. & Liu, J. *Guided Inpainting and Filtering for Kinect Depth Maps* in *Int. Conf. Pattern Recognition* (2012), 2055–2058 (*pp.* 8, 17, 20, 22, 24, 26, 31, 33–35, 38–40, 63, 67–70, 79–85, 105–107, 109, 112, 130, 144, 146–147, 149, 151–154).

42. Po, L.-M., Zhang, S., Xu, X. & Zhu, Y. *A New Multi-Directional Extrapolation Hole-Filling Method for Depth-Image-Based Rendering* in *Int. Conf. Image Processing* (2011), 2589–2592 (*pp.* 8, 24, 28–29, 39, 150).

43. Garro, V., Mutto, C. D., Zanuttigh, P. & Cortelazzo, G. M. *A Novel Interpolation Scheme for Range Data with Side Information* in *Conf. Visual Media Production* (2009), 52–60 (*pp.* 8, 26, 30, 39).

44. Matyunin, S., Vatolin, D., Berdnikov, Y. & Smirnov, M. *Temporal Filtering for Depth Maps Generated by Kinect Depth Camera* in *3DTV Conference* (2011), 1–4 (*pp.* 8, 24, 26–27, 35, 39, 151).

45. Camplani, M. & Salgado, L. *Efficient Spatiotemporal Hole Filling Strategy for Kinect Depth Maps* in *IS&T/SPIE Electronic Imaging* (2012), 82900E–82900E (*pp.* 8, 22, 24, 29, 34–37, 39–40, 81).

46. Eigen, D. & Fergus, R. *Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture* in *Int. Conf. Computer Vision* (2015), 2650–2658 (*pp.* 8, 40, 43, 114, 119, 131, 153).

47. Godard, C., Aodha, O. M. & Brostow, G. J. *Unsupervised Monocular Depth Estimation with Left-Right Consistency* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 6602–6611 (*pp.* 8, 40–41, 44–45, 108, 110–111, 115–116, 119–121, 125–127, 131, 138–139, 147–149, 151, 153–156).

48. Zhan, H., Garg, R., Weerasekera, C. S., Li, K., Agarwal, H. & Reid, I. *Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 340–349 (*pp.* 8, 44, 119, 131, 147–149, 153–154).

49. Ding, L. & Sharma, G. *Fusing Structure from Motion and Lidar for Dense Accurate Depth Map Estimation* in *Int. Conf. Acoustics, Speech and Signal Processing* (2017), 1283–1287 (*pp.* 8, 114, 154–155).

50. Cavestany, P., Rodriguez, A., Martinez-Barbera, H. & Breckon, T. *Improved 3D Sparse Maps for High-performance Structure from Motion with Low-cost Omnidirectional Robots* in *Int. Conf. Image Processing* (2015), 4927–4931 (*pp.* 8, 114).

51. Tao, M. W., Srinivasan, P. P., Malik, J., Rusinkiewicz, S. & Ramamoorthi, R. *Depth from Shading, Defocus, and Correspondence using Light-Field Angular Coherence* in *IEEE Conf. Computer Vision and Pattern Recognition* (2015), 1940–1948 (*pp.* 8, 114, 154).

52. Abrams, A., Hawley, C. & Pless, R. Heliometric Stereo: Shape from Sun Position. *Euro. Conf. Computer Vision,* 357–370 (2012) (*pp.* 8, 114).

53. Efros, A. A. & Leung, T. K. *Texture Synthesis by Non-Parametric Sampling* in *Int. Conf. Computer Vision* **2** (1999), 1033–1038 (*pp.* 9–10, 13, 53–55, 57, 150).

54. Heeger, D. J. & Bergen, J. R. *Pyramid-Based Texture Analysis/Synthesis* in *Conf. Computer Graphics and Interactive Techniques* (1995), 229–238 (*pp.* 9–10, 53).

55. Simoncelli, E. P. & Portilla, J. *Texture Characterization via Joint Statistics of Wavelet Coefficient Magnitudes* in *Int. Conf. Image Processing* **1** (1998), 62–66 (*pp.* 9, 53).

56. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G. & Verdera, J. Filling-in by Joint Interpolation of Vector Fields and Gray Levels. *IEEE Trans. Image Processing* **10,** 1200–1211 (2001) (*p.* 9).

57. Bertalmio, M., Bertozzi, A. L. & Sapiro, G. *Navier-stokes, Fluid Dynamics, and Image and Video Inpainting* in *IEEE Conf. Computer Vision and Pattern Recognition* **1** (2001), I–355 (*pp.* 9, 53).

58. Bertalmio, M., Sapiro, G., Caselles, V. & Ballester, C. *Image Inpainting* in *Int. Conf. Computer Graphics and Interactive Techniques* (2000), 417–424 (*pp.* 9–11, 15, 23, 28, 39, 53).

59. Rane, S. D., Sapiro, G. & Bertalmio, M. Structure and Texture Filling-in of Missing Image Blocks in Wireless Transmission and Compression Applications. *IEEE Trans. Image Processing* **12,** 296–303 (2003) (*p.* 9).

60. Criminisi, A., Pérez, P. & Toyama, K. Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Trans. Image Processing* **13,** 1200–1212 (2004) (*pp.* 9–12, 19–21, 29, 31, 36–37, 59, 61–65, 67–70, 81, 150–152).

61. Popat, K. & Picard, R. W. *Novel Cluster-Based Probability Model for Texture Synthesis, Classification, and Compression* in *Visual Communications* (1993), 756–768 (*p.* 10).

62. Liang, L., Liu, C., Xu, Y.-Q., Guo, B. & Shum, H.-Y. Real-Time Texture Synthesis by Patch-Based Sampling. *ACM Trans. Graphics* **20,** 127–150 (2001) (*pp.* 10, 59).

63. Efros, A. A. & Freeman, W. T. *Image Quilting for Texture Synthesis and Transfer* in *Conf. Computer Graphics and Interactive Techniques* (2001), 341–346 (*pp.* 10, 59).

64. Nealen, A. & Alexa, M. *Hybrid Texture Synthesis* (Techn. Univ., Fachbereich Informatik, Fachgebiet Graphisch-Interaktive Systeme, 2003) (*p.* 10).

65. Bertalmio, M., Vese, L., Sapiro, G. & Osher, S. Simultaneous Structure and Texture Image Inpainting. *IEEE Trans. Image Processing* **12,** 882–889 (2003) (*pp.* 10–11).

66. Arias, P., Facciolo, G., Caselles, V. & Sapiro, G. A Variational Framework for Exemplar-Based Image Inpainting. *Int. J. Computer Vision* **93,** 319–347 (2011) (*pp.* 10, 12–13, 54–55, 57, 59–61, 63, 66–70, 79–85, 105–106, 109, 112, 151–153).

67. Jia, J. & Tang, C.-K. *Image Repairing: Robust Image Synthesis by Adaptive n-D Tensor Voting* in *IEEE Conf. Computer Vision and Pattern Recognition* **1** (2003), I–643 (*pp.* 10–11, 150).

68. Telea, A. An Image Inpainting Technique Based on the Fast Marching Method. *Graphics Tools* **9,** 23–34 (2004) (*pp.* 11, 17, 19–20, 29, 31, 33–34, 39–40, 61, 63, 67–70, 79–85, 105–106, 109, 112, 151–153).

69. Chan, T. & Shen, J. *Mathematical Models for Local Deterministic Inpaintings* technical report (Technical Report CAM TR 00-11, UCLA, 2000) (*p.* 11).

70. Chan, T. F. & Shen, J. Non-texture Inpainting by Curvature-Driven Diffusions. *Visual Communication and Image Representation* **12,** 436–449 (2001) (*pp.* 11, 15, 54).

71. Richard, M. M. O. B. B. & Chang, M. Y.-S. *Fast Digital Image Inpainting* in *Int. Conf. Visualization, Imaging and Image Processing* (2001), 106–107 (*pp.* 11, 31).

72. Harrison, P. A Non-Hierarchical Procedure for Resynthesis of Complex Textures (2001) (*p.* 11).

73. Daribo, I. & Saito, H. A Novel Inpainting-Based Layered Depth Video for 3DTV. *IEEE Trans. Broadcasting* **57,** 533–541 (2011) (*pp.* 11, 19, 21, 24, 31, 40).

74. Hervieu, A., Papadakis, N., Bugeau, A., Gargallo, P. & Caselles, V. *Stereoscopic Image Inpainting: Distinct Depth Maps and Images Inpainting* in *Int. Conf. Pattern Recognition* (2010), 4101–4104 (*pp.* 11, 19, 21, 24, 31, 40, 61).

75. Mansfield, A., Prasad, M., Rother, C., Sharp, T., Kohli, P. & Van Gool, L. J. *Transforming Image Completion* in *British Machine Vision Conference* (2011), 1–11 (*p.* 12).

76. Darabi, S., Shechtman, E., Barnes, C., Goldman, D. B. & Sen, P. Image Melding: Combining Inconsistent Images Using Patch-Based Synthesis. *ACM Trans. Graphics* **31,** 82–1 (2012) (*pp.* 12, 20).

77. Huang, J.-B., Kang, S. B., Ahuja, N. & Kopf, J. Image Completion Using Planar Structure Guidance. *ACM Trans. Graphics* **33,** 129 (2014) (*pp.* 12–13).

78. Kumar, V., Mukherjee, J. & Mandal, S. K. D. Image Inpainting through Metric Labeling via Guided Patch Mixing. *IEEE Trans. Image Processing* **25,** 5212–5226 (2016) (*p.* 12).

79. Kwatra, V., Essa, I., Bobick, A. & Kwatra, N. *Texture Optimization for Example-Based Synthesis* in *ACM Trans. Graphics* **24** (2005), 795–802 (*pp.* 12–13, 17).

80. Wexler, Y., Shechtman, E. & Irani, M. Space-Time Completion of Video. *IEEE Trans. Pattern Analysis and Machine Intelligence* **29,** 463–476 (2007) (*pp.* 12–13, 17, 20, 61).

81. Barnes, C., Shechtman, E., Finkelstein, A. & Goldman, D. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Trans. Graphics* **28,** 24 (2009) (*pp.* 12–13, 17).

82. Komodakis, N. & Tziritas, G. Image Completion Using Efficient Belief Propagation via Priority Scheduling and Dynamic Pruning. *IEEE Trans. Image Processing* **16,** 2649–2661 (2007) (*pp.* 12–13).

83. Pritch, Y., Kav-Venaki, E. & Peleg, S. *Shift-Map Image Editing* in *Int. Conf. Computer Vision* **9** (2009), 151–158 (*pp.* 12–13).

84. Sun, J., Yuan, L., Jia, J. & Shum, H.-Y. *Image Completion with Structure Propagation* in *ACM Trans. Graphics* **24** (2005), 861–868 (*p.* 12).

85. Liu, Y. & Caselles, V. Exemplar-Based Image Inpainting using Multiscale Graph Cuts. *IEEE trans. Image Processing* **22,** 1699–1711 (2013) (*pp.* 12, 20).

86. Bugeau, A., Bertalmio, M., Caselles, V. & Sapiro, G. A Comprehensive Framework for Image Inpainting. *IEEE Trans. Image Processing* **19,** 2634–2645 (2010) (*pp.* 12–13, 15, 31).

87. Mittal, A., Moorthy, A. K. & Bovik, A. C. No-Reference Image Quality Assessment in the Spatial Domain. *IEEE Trans. Image Processing* **21,** 4695–4708 (2012) (*p.* 13).

88. Fang, C.-W. & Lien, J.-J. J. Rapid Image Completion System Using Multi-Resolution Patch-Based Directional and Non-Directional Approaches. *IEEE Trans. Image Processing* **18,** 2769–2779 (2009) (*pp.* 13, 60).

89. Demanet, L., Song, B. & Chan, T. Image Inpainting by Correspondence Maps: a Deterministic Approach. *Computational and Applied Mathematics* **1100,** 217–50 (2003) (*p.* 13).

90. Aujol, J.-F., Ladjal, S. & Masnou, S. Exemplar-Based Inpainting from a Variational Point of View. *Mathematical Analysis* **42,** 1246–1285 (2010) (*p.* 13).

91. Ashikhmin, M. *Synthesizing Natural Textures* in *Symp. Interactive 3D graphics* (2001), 217–226 (*p.* 13).

92. Wexler, Y., Shechtman, E. & Irani, M. *Space-Time Video Completion* in *IEEE Conf. Computer Vision and Pattern Recognition* **1** (2004), I–120 (*p.* 13).

93. Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Computer Vision* **60,** 91–110 (2004) (*p.* 13).

94. Hays, J. & Efros, A. A. Scene Completion Using Millions of Photographs. *ACM Trans. Graphics* **26,** 4 (2007) (*pp.* 13–14).

95. Whyte, O., Sivic, J. & Zisserman, A. *Get Out of My Picture! Internet-Based Inpainting* in *British Machine Vision Conference* (2009), 1–11 (*pp.* 13–14).

96. Philbin, J., Chum, O., Isard, M., Sivic, J. & Zisserman, A. *Object Retrieval with Large Vocabularies and Fast Spatial Matching* in *IEEE Conf. Computer Vision and Pattern Recognition* (2007), 1–8 (*p.* 13).

97. Gatys, L. A., Ecker, A. S. & Bethge, M. *Image Style Transfer using Convolutional Neural Networks* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 2414–2423 (*pp.* 14, 47–48, 87, 127, 156).

98. Johnson, J., Alahi, A. & Fei-Fei, L. *Perceptual Losses for Real-Time Style Transfer and Super-Resolution* in *Euro. Conf. Computer Vision* (2016), 694–711 (*pp.* 14, 47–48, 87, 119, 122, 124–127, 129).

99. Ghiasi, G., Lee, H., Kudlur, M., Dumoulin, V. & Shlens, J. *Exploring the Structure of a Real-Time, Arbitrary Neural Artistic Stylization Network* in *British Machine Vision Conference* (2017), 1–12 (*pp.* 14, 48).

100. Jackson, P. T., Atapour-Abarghouei, A., Bonner, S., Breckon, T. & Obara, B. Style Augmentation: Data Augmentation via Style Randomization. *arXiv preprint arXiv:1809.05375,* 1–13 (2018) (*p.* 14).

101. Wang, L., Huang, Z., Gong, Y. & Pan, C. Ensemble based Deep Networks for Image Super-Resolution. *Pattern Recognition* **68,** 191–198 (2017) (*pp.* 14, 87).

102. Nguyen, K., Fookes, C., Sridharan, S., Tistarelli, M. & Nixon, M. Super-Resolution for Biometrics: A Comprehensive Survey. *Pattern Recognition* **78,** 23–42 (2018) (*pp.* 14, 87).

103. Zhang, R., Isola, P. & Efros, A. A. *Colorful Image Colorization* in *Euro. Conf. Computer Vision* (2016), 649–666 (*pp.* 14, 87).

104. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. & Efros, A. A. *Context Encoders: Feature Learning by Inpainting* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 2536–2544 (*pp.* 14, 22, 87–88, 96, 99, 106, 117, 119, 135, 150).

105. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. *Generative Adversarial Nets* in *Advances in Neural Information Processing Systems* (2014), 2672–2680 (*pp.* 14, 88–89, 117, 121, 135, 156).

106. Yeh, R. A., Chen, C., Lim, T. Y., G., S. A., Hasegawa-Johnson, M. & Do, M. N. *Semantic Image Inpainting with Deep Generative Models* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 6882–6890 (*pp.* 14, 22, 87–88, 96, 135, 150).

107. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O. & Li, H. *High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 4076–4084 (*pp.* 14, 22, 87–88, 96).

108. Li, C. & Wand, M. *Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 2479–2486 (*p.* 14).

109. Iizuka, S., Simo-Serra, E. & Ishikawa, H. Globally and Locally Consistent Image Completion. *ACM Trans. Graphics* **36,** 107 (2017) (*pp.* 14–15, 22, 144, 146–147, 149, 154).

110. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X. & Huang, T. S. *Generative Image Inpainting with Contextual Attention* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 1–15 (*pp.* 14–15, 22, 144, 146–147, 149–150, 154).

111. Wei, L.-Y., Lefebvre, S., Kwatra, V. & Turk, G. *State of the Art in Example-Based Texture Synthesis* in *Eurographics State of the Art Report* (2009), 93–117 (*p.* 15).

112. Guillemot, C. & Le Meur, O. Image Inpainting: Overview and Recent Advances. *Signal Processing Magazine* **31,** 127–144 (2014) (*p.* 15).

113. Fidaner, I. B. A Survey on Variational Image Inpainting, Texture synthesis and Image Completion. *Bogazici University* (2008) (*p.* 15).

114. Zhang, H.-y. & Peng, Q.-c. A Survey on Digital Image Inpainting. *Image and Graphics* **12,** 1–10 (2007) (*p.* 15).

115. Janarthanan, V. & Jananii, G. A Detailed Survey on Various Image Inpainting Techniques. *Advances in Image Processing* **2,** 1 (2012) (*p.* 15).

116. Perona, P. & Malik, J. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Trans. Pattern Analysis and Machine Intelligence* **12,** 629–639 (1990) (*pp.* 15–16).

117. Ballester, C., Caselles, V., Verdera, J., Bertalmio, M. & Sapiro, G. *A Variational Model for Filling-in Gray Level and Color Images* in *Int. Conf. Computer Vision* **1** (2001), 10–16 (*pp.* 15, 54).

118. Hu, G., Huang, S., Zhao, L., Alempijevic, A. & Dissanayake, G. *A Robust RGB-D SLAM Algorithm* in *Int. Conf. Intelligent Robots and Systems* (2012), 1714–1719 (*pp.* 16, 40).

119. Mur-Artal, R. & Tardos, J. D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robotics* **33,** 1255–1262 (2017) (*pp.* 16, 40).

120. Vijayanagar, K. R., Loghman, M. & Kim, J. Real-Time Refinement of Kinect Depth Maps Using Multi-Resolution Anisotropic Diffusion. *Mobile Networks and Applications* **19,** 414–425 (2014) (*pp.* 16, 20, 24, 32, 39).

121. Miao, D., Fu, J., Lu, Y., Li, S. & Chen, C. W. *Texture-Assisted Kinect Depth Inpainting* in *Int. Symp. Circuits and Systems* (2012), 604–607 (*pp.* 16, 24, 32).

122. Liu, S., Wang, Y., Wang, J., Wang, H., Zhang, J. & Pan, C. *Kinect Depth Restoration via Energy Minimization with TV 21 Regularization* in *Int. Conf. Image Processing* (2013), 724–724 (*pp.* 17, 24, 33–34, 39–40).

123. Chen, C., Cai, J., Zheng, J., Cham, T. J. & Shi, G. Kinect Depth Recovery Using a Color-Guided, Region-Adaptive, and Depth-Selective Framework. *ACM Trans. Intelligent Systems and Technology* **6,** 12 (2015) (*pp.* 17, 24, 32–33).

124. Barbero, A. & Sra, S. *Fast Newton-Type Methods for Total Variation Regularization* in *Int. Conf. Machine Learning* (2011), 313–320 (*pp.* 17, 34).

125. Herrera, D., Kannala, J., Heikkila, J. *et al. Depth Map Inpainting under a Second-Order Smoothness Prior* in *Scandinavian Conf. Image Analysis* (2013), 555–566 (*pp.* 17, 40, 63, 67–70, 79–85, 105–107, 109, 112, 151–153).

126. Xu, X., Po, L.-M., Cheung, C.-H., Feng, L., Ng, K.-H. & Cheung, K.-W. *Depth-Aided Exemplar-Based Hole Filling for DIBR View Synthesis* in *Int. Symp. Circuits and Systems* (2013), 2840–2843 (*pp.* 18, 20, 24, 32, 39–40).

127. Zhang, L., Shen, P., Zhang, S., Song, J. & Zhu, G. *Depth Enhancement with Improved Exemplar-Based Inpainting and Joint Trilateral Guided Filtering* in *Int. Conf. Image Processing* (2016), 4102–4106 (*pp.* 20, 65, 81).

128. Baek, S.-H., Choi, I. & Kim, M. H. *Multiview Image Completion with Space Structure Propagation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 488–496 (*pp.* 20, 39).

129. Wang, L., Jin, H., Yang, R. & Gong, M. *Stereoscopic Inpainting: Joint Color and Depth Completion from Stereo Images* in *IEEE Conf. Computer Vision and Pattern Recognition* (2008), 1–8 (*pp.* 20, 39, 151).

130. McMillan Jr, L. *An Image-Based Approach to Three-Dimensional Computer Graphics* PhD thesis (Citeseer, 1997) (*p.* 21).

131. Lu, S., Ren, X. & Liu, F. *Depth Enhancement via Low-Rank Matrix Completion* in *IEEE Conf. Computer Vision and Pattern Recognition* (2014), 3390–3397 (*pp.* 21–22, 39–40, 81).

132. Richardt, C., Stoll, C., Dodgson, N. A., Seidel, H.-P. & Theobalt, C. *Coherent Spatiotemporal Filtering, Upsampling and Rendering of RGBZ Videos* in *Computer Graphics Forum* **31** (2012), 247–256 (*pp.* 22, 24, 37, 39–40, 81).

133. Qi, F., Han, J., Wang, P., Shi, G. & Li, F. Structure Guided Fusion for Depth Map Inpainting. *Pattern Recognition Letters* **34,** 70–76 (2013) (*pp.* 22, 24, 31, 39–40, 81, 130).

134. Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. Image Denoising by Sparse 3D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Processing* **16,** 2080–2095 (2007) (*p.* 22).

135. Xue, H., Zhang, S. & Cai, D. Depth Image Inpainting: Improving Low Rank Matrix Completion with Low Gradient Regularization. *IEEE Trans. Image Processing* **26,** 4311–4320 (2017) (*p.* 22).

136. Zhang, Y. & Funkhouser, T. *Deep Depth Completion of a Single RGB-D Image* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 175–185 (*p.* 23).

137. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A. & Zhang, Y. *Matterport3D: Learning from RGB-D data in indoor environments* in *Int Conf. 3D Vision* (2017) (*p.* 23).

138. Yang, N.-E., Kim, Y.-G. & Park, R.-H. *Depth Hole Filling Using the Depth Distribution of Neighboring Regions of Depth Holes in the Kinect Sensor* in *Int. Conf. Signal Processing, Communication and Computing* (2012), 658–661 (*pp.* 24, 26–27, 39–40, 66).

139. Chen, L., Lin, H. & Li, S. *Depth Image Enhancement for Kinect Using Region Growing and Bilateral Filter* in *Int. Conf. Pattern Recognition* (2012), 3070–3073 (*pp.* 24, 26–27, 39–40, 81).

140. Min, D., Lu, J. & Do, M. N. Depth Video Enhancement Based on Weighted Mode Filtering. *IEEE Trans. Image Processing* **21,** 1176–1190 (2012) (*pp.* 24, 27, 39–40, 66).

141. Chen, W.-Y., Chang, Y.-L., Lin, S.-F., Ding, L.-F. & Chen, L.-G. *Efficient Depth Image Based Rendering with Edge Dependent Depth Filter and Interpolation* in *Int. Conf. Multimedia and Expo* (2005), 1314–1317 (*pp.* 24, 27, 39–40).

142. Daribo, I., Tillier, C. & Pesquet-Popescu, B. *Distance Dependent Depth Filtering in 3D Warping for 3DTV* in *Workshop on Multimedia Signal Processing* (2007), 312–315 (*pp.* 24, 27, 39–40).

143. Lee, S.-B. & Ho, Y.-S. *Discontinuity-Adaptive Depth Map Filtering for 3D View Generation* in *Int. Conf. Immersive Telecommunications* (2009), 8 (*pp.* 24, 28, 39–40).

144. Xu, X., Po, L.-M., Ng, K.-H., Feng, L., Cheung, K.-W., Cheung, C.-H. & Ting, C.-W. Depth Map Misalignment Correction and Dilation for DIBR View Synthesis. *Signal Processing: Image Communication* **28,** 1023–1045 (2013) (*pp.* 24, 28–29, 39).

145. Jung, S.-W. Enhancement of Image and Depth Map Using Adaptive Joint Trilateral Filter. *IEEE Trans. Circuits and Systems for Video Technology* **23,** 258–269 (2013) (*pp.* 24, 29, 38, 40).

146. Matsuo, T., Fukushima, N. & Ishibashi, Y. *Weighted Joint Bilateral Filter with Slope Depth Compensation Filter for Depth Map Refinement* in *Int. Conf. Computer Vision Theory and Applications* (2013), 300–309 (*pp.* 24, 27, 30, 40).

147. Chen, C., Cai, J., Zheng, J., Cham, T.-J. & Shi, G. *A Color-Guided, Region-Adaptive and Depth-Selective Unified Framework for Kinect Depth Recovery* in *Int. Workshop on Multimedia Signal Processing* (2013), 007–012 (*pp.* 24, 33).

148. Yang, J., Ye, X., Li, K., Hou, C. & Wang, Y. Color-Guided Depth Recovery from RGB-D Data Using an Adaptive Autoregressive Model. *IEEE Trans. Image Processing* **23,** 3443–3458 (2014) (*pp.* 24, 26, 34–35, 38–39, 81).

149. Wang, Z., Hu, J., Wang, S. & Lu, T. Trilateral Constrained Sparse Representation for Kinect Depth Hole Filling. *Pattern Recognition Letters* **65,** 95–102 (2015) (*pp.* 24, 34–35, 81).

150. Hu, J., Hu, R., Wang, Z., Gong, Y. & Duan, M. *Color Image Guided Locality Regularized Representation for Kinect Depth Holes Filling* in *Visual Communications and Image Processing* (2013), 1–6 (*pp.* 24, 35, 39).

151. Berdnikov, Y. & Vatolin, D. *Real-Time Depth Map Occlusion Filling and Scene Background Restoration for Projected-Pattern Based Depth Cameras* in *Graphic Conf. IETP* (2011) (*pp.* 24, 35–36, 39, 151).

152. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A. *et al. KinectFusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera* in *ACM Symp. User Interface Software and Technology* (2011), 559–568 (*pp.* 24, 35, 39).

153. Fu, D., Zhao, Y. & Yu, L. *Temporal Consistency Enhancement on Depth Sequences* in *Picture Coding Symposium* (2010), 342–345 (*pp.* 24, 36).

154. Sheng, L., Ngan, K. N. & Li, S. *Temporal Depth Video Enhancement Based on Intrinsic Static Structure* in *Int. Conf. Image Processing* (2014), 2893–2897 (*pp.* 24, 36).

155. Wang, J., An, P., Zuo, Y., You, Z. & Zhang, Z. *High Accuracy Hole Filling for Kinect Depth Maps* in *SPIE/COS Photonics Asia* (2014), 92732L–92732L (*pp.* 24, 36–37, 151).

156. Camplani, M. & Salgado, L. *Adaptive Spatiotemporal Filter for Low-Cost Camera Depth Maps* in *Int. Conf. Emerging Signal Processing Applications* (2012), 33–36 (*pp.* 24, 37, 81).

157. Kim, S.-Y., Cho, J.-H., Koschan, A. & Abidi, M. A. *Spatial and Temporal Enhancement of Depth Images Captured by a Time-of-Flight Depth Sensor* in *Int. Conf. Pattern Recognition* (2010), 2358–2361 (*pp.* 24, 37–40, 81).

158. Xu, K., Zhou, J. & Wang, Z. *A Method of Hole-Filling for the Depth Map Generated by Kinect with Moving Objects Detection* in *Int. Symp. Broadband Multimedia Systems and Broadcasting* (2012), 1–5 (*pp.* 24, 37).

159. Lai, K., Bo, L., Ren, X. & Fox, D. *A Large-Scale Hierarchical Multi-View RGB-D Object Dataset* in *Int. Conf. Robotics and Automation* (2011), 1817–1824 (*p.* 26).

160. Zhang, L., Tam, W. J. & Wang, D. *Stereoscopic Image Generation Based on Depth Images* in *Int. Conf. Image Processing* **5** (2004), 2993–2996 (*p.* 26).

161. Tomasi, C. & Manduchi, R. *Bilateral Filtering for Gray and Color Images* in *Int. Conf. Computer Vision* (1998), 839–846 (*p.* 26).

162. Buades, A., Coll, B. & Morel, J.-M. *A Non-Local Algorithm for Image Denoising* in *Int. Conf. Computer Vision and Pattern Recognition* **2** (2005), 60–65 (*p.* 26).

163. Gangwal, O. P. & Djapic, B. *Real-Time Implementation of Depth Map Post-Processing for 3D-TV in Dedicated Hardware* in *Int. Conf. Consumer Electronics* (2010), 173–174 (*p.* 26).

164. Kopf, J., Cohen, M. F., Lischinski, D. & Uyttendaele, M. Joint Bilateral Upsampling. *ACM Trans. Graphics* **26,** 96 (2007) (*pp.* 26, 34, 37).

165. Kim, Y., Ham, B., Oh, C. & Sohn, K. Structure Selective Depth Superresolution for RGB-D Cameras. *IEEE Trans. Image Processing* **25,** 5227–5238 (2016) (*p.* 26).

166. Riegler, G., Ferstl, D., Rüther, M. & Horst, B. *A Deep Primal-Dual Network for Guided Depth Super-Resolution* in *British Machine Vision Conference* (2016), 1–14 (*p.* 26).

167. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H. & Toyama, K. *Digital Photography with Flash and no-Flash Image Pairs* in *ACM Trans. Graphics* **23** (2004), 664–672 (*p.* 26).

168. Liu, S., Lai, P., Tian, D., Gomila, C. & Chen, C. W. *Joint Trilateral Filtering for Depth Map Compression* in *Visual Communications and Image Processing* (2010), 77440F–77440F (*p.* 26).

169. He, K., Sun, J. & Tang, X. in *Euro. Conf. Computer Vision* 1–14 (Springer, 2010) (*pp.* 26, 31).

170. Nguyen, Q. H., Do, M. N. & Patel, S. J. *Depth Image-Based Rendering from Multiple Cameras with 3D Propagation Algorithm* in *Int. Conf. Immersive Telecommunications* (2009), 6 (*pp.* 27–29, 39, 81).

171. Mueller, M., Zilly, F. & Kauff, P. *Adaptive Cross-Trilateral Depth Map Filtering* in *3DTV Conference* (2010), 1–4 (*pp.* 27, 39).

172. Nguyen, H. T. & Do, M. N. *Image-Based Rendering with Depth Information Using the Propagation Algorithm* in *Int. Conf. Acoustics, Speech, and Signal Processing* (2005), 589–592 (*p.* 28).

173. Shen, J. & Cheung, S.-C. *Layer Depth De-noising and Completion for Structured-Light RGB-D Cameras* in *IEEE Conf. Computer Vision and Pattern Recognition* (2013), 1187–1194 (*pp.* 29, 39–40).

174. Vázquez, C., Tam, W. J. & Speranza, F. *Stereoscopic Imaging: Filling Disoccluded Areas in Depth Image-Based Rendering* in *Optics East* (2006), 63920D–63920D (*p.* 29).

175. Meyer, F. *Color Image Segmentation* in *Int. Conf. Image Processing and its Applications* (1992), 303–306 (*p.* 29).

176. Crabb, R., Tracey, C., Puranik, A. & Davis, J. *Real-Time Foreground Segmentation via Range and Color Imaging* in *Workshop in IEEE Conf. Computer Vision and Pattern Recognition* (2008), 1–5 (*p.* 30).

177. Ma, Y., Worrall, S. & Kondoz, A. M. *Automatic Video Object Segmentation using Depth Information and an Active Contour Model* in *Workshop on Multimedia Signal Processing* (2008), 910–914 (*p.* 30).

178. Felzenszwalb, P. F. & Huttenlocher, D. P. Efficient Graph-Based Image Segmentation. *Int. J. Computer Vision* **59,** 167–181 (2004) (*p.* 30).

179. Levin, A., Lischinski, D. & Weiss, Y. A Closed-Form Solution to Natural Image Matting. *IEEE Trans. Pattern Analysis and Machine Intelligence* **30,** 228–242 (2008) (*p.* 34).

180. Tošić, I., Olshausen, B. A. & Culpepper, B. J. Learning Sparse Representations of Depth. *Selected Topics in Signal Processing* **5,** 941–952 (2011) (*p.* 35).

181. Tosic, I. & Drewes, S. Learning Joint Intensity-Depth Sparse Representations. *IEEE Trans. Image Processing* **23,** 2122–2132 (2014) (*p.* 35).

182. Harsha, G. N., Majumdar, A. & Ward, R. Disparity Map Computation for Stereo Images using Compressive Sampling. *IASTED Signal and Image Processing,* 804–809 (2013) (*p.* 35).

183. Gortler, S. J., Grzeszczuk, R., Szeliski, R. & Cohen, M. F. *The lumigraph* in *Conf. Computer Graphics and Interactive Techniques* (1996), 43–54 (*p.* 37).

184. Islam, A. T., Scheel, C., Pajarola, R. & Staadt, O. Robust Enhancement of Depth Images from Depth Sensors. *Computers & Graphics* **68,** 53–65 (2017) (*pp.* 38–39).

185. Rousseeuw, P. J. Least Median of Squares Regression. *American Statistical Association* **79,** 871–880 (1984) (*p.* 38).

186. Gupta, S., Girshick, R., Arbelaez, P. & Malik, J. *Learning Rich Features from RGB-D Images for Object Detection and Segmentation* in *Euro. Conf. Computer Vision* (2014), 345–360 (*p.* 40).

187. Bo, L., Ren, X. & Fox, D. *Unsupervised Feature Learning for RGB-D Based Object Recognition* in *Experimental Robotics* (2013), 387–402 (*p.* 40).

188. Spinello, L. & Arras, K. O. *People Detection in RGB-D Data* in *Int. Conf. Intelligent Robots and Systems* (2011), 3838–3843 (*p.* 40).

189. Ladicky, L., Shi, J. & Pollefeys, M. *Pulling Things out of Perspective* in *IEEE Conf. Computer Vision and Pattern Recognition* (2014), 89–96 (*pp.* 40–41, 46, 114).

190. Liu, F., Shen, C., Lin, G. & Reid, I. Learning Depth from Single Monocular Images using Deep Convolutional Neural Fields. *IEEE Trans. Pattern Analysis and Machine Intelligence* **38,** 2024–2039 (2016) (*pp.* 40, 114, 119, 125, 131, 147–149, 151, 153–154).

191. Garg, R., Carneiro, G. & Reid, I. *Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue* in *Euro. Conf. Computer Vision* (2016), 740–756 (*pp.* 40–41, 44, 115–116, 119, 131, 151, 153, 156).

192. Xie, J., Girshick, R. & Farhadi, A. *Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks* in *Euro. Conf. Computer Vision* (2016), 842–857 (*pp.* 40–41, 44, 115, 156).

193. Eigen, D., Puhrsch, C. & Fergus, R. *Depth Map Prediction from a Single Image using a Multi-Scale Deep Network* in *Advances in Neural Information Processing Systems* (2014), 2366–2374 (*pp.* 40, 43, 46, 114, 119, 125, 147–149, 154).

194. Zhuo, W., Salzmann, M., He, X. & Liu, M. *Indoor Scene Structure Analysis for Single Image Depth Estimation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2015), 614–622 (*pp.* 40, 42, 115).

195. Zheng, C., Cham, T.-J. & Cai, J. *T2Net: Synthetic-to-Realistic Translation for Solving Single-Image Depth Estimation Tasks* in *Euro. Conf. Computer Vision* (2018), 767–783 (*pp.* 41, 44).

196. Chen, W., Fu, Z., Yang, D. & Deng, J. *Single-Image Depth Perception in the Wild* in *Advances in Neural Information Processing Systems* (2016), 730–738 (*pp.* 41, 115).

197. Huang, X., Wang, L., Huang, J., Li, D. & Zhang, M. *A Depth Extraction Method Based on Motion and Geometry for 2D to 3D Conversion* in *Intelligent Information Technology Application* **3** (2009), 294–298 (*p.* 41).

198. Zhang, G., Jia, J., Hua, W. & Bao, H. Robust Bilayer Segmentation and Motion/Depth Estimation with a Handheld Camera. *IEEE Trans. Pattern Analysis and Machine Intelligence* **33,** 603–617 (2011) (*p.* 41).

199. Hoiem, D., Efros, A. A. & Hebert, M. *Automatic Photo Pop-up* in *ACM trans. Graphics* **24** (2005), 577–584 (*p.* 41).

200. Hoiem, D., Efros, A. A. & Hebert, M. *Geometric Context from a Single Image* in *It. Conf. Computer Vision* **1** (2005), 654–661 (*p.* 41).

201. Karsch, K., Liu, C. & Kang, S. B. Depth Transfer: Depth Extraction from Video using Non-Parametric Sampling. *IEEE Trans. Pattern Analysis and Machine Intelligence* **36,** 2144–2158 (2014) (*pp.* 41, 127, 130–131).

202. Liu, B., Gould, S. & Koller, D. *Single Image Depth Estimation from Predicted Semantic Labels* in *IEEE Conf. Computer Vision and Pattern Recognition* (2010), 1253–1260 (*p.* 41).

203. Delage, E., Lee, H. & Ng, A. Y. *A Dynamic Bayesian Network Model for Autonomous 3D Reconstruction from a Single Indoor Image* in *IEEE Conf. Computer Vision and Pattern Recognition* **2** (2006), 2418–2428 (*p.* 42).

204. Saxena, A., Chung, S. H. & Ng, A. Y. *Learning Depth from Single Monocular Images* in *Advances in Neural Information Processing Systems* (2006), 1161–1168 (*p.* 42).

205. Saxena, A., Sun, M. & Ng, A. Y. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Trans. Pattern Analysis and Machine Intelligence* **31,** 824–840 (2009) (*pp.* 42, 46, 125, 127, 129–131).

206. Heitz, G., Gould, S., Saxena, A. & Koller, D. *Cascaded Classification Models: Combining Models for Holistic Scene Understanding* in *Advances in Neural Information Processing Systems* (2009), 641–648 (*pp.* 42, 46).

207. Liu, M., Salzmann, M. & He, X. *Discrete-Continuous Depth Estimation from a Single Image* in *IEEE Conf. Computer Vision and Pattern Recognition* (2014), 716–723 (*pp.* 42, 127, 130–131).

208. Ihler, A. & McAllester, D. *Particle Belief Propagation* in *Artificial Intelligence and Statistics* (2009), 256–263 (*p.* 42).

209. Peng, J., Hazan, T., McAllester, D. & Urtasun, R. *Convex Max-Product Algorithms for Continuous MRFs with Applications to Protein Folding* in *Int. Conf. Machine Learning* (2011), 729–736 (*p.* 42).

210. Wu, Y., Ying, S. & Zheng, L. Size-to-Depth: A New Perspective for Single Image Depth Estimation. *arXiv preprint arXiv:1801.04461* (2018) (*p.* 42).

211. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F. & Navab, N. *Deeper Depth Prediction with Fully Convolutional Residual Networks* in *Int. Conf. 3D Vision* (2016), 239–248 (*pp.* 43, 127, 130–131).

212. Liu, F., Shen, C. & Lin, G. *Deep Convolutional Neural Fields for Depth Estimation from a Single Image* in *IEEE Conf. Computer Vision and Pattern Recognition* (2015), 5162–5170 (*p.* 43).

213. Cao, Y., Wu, Z. & Shen, C. Estimating Depth from Monocular Images as Classification using Deep Fully Convolutional Residual Networks. *IEEE Trans. Circuits and Systems for Video Technology* **28,** 3174–3182 (2017) (*p.* 43).

214. Li, B., Dai, Y. & He, M. Monocular Depth Estimation with Hierarchical Fusion of Dilated CNNs and Soft-Weighted-Sum Inference. *Pattern Recognition* (2018) (*p.* 43).

215. Hu, J., Ozay, M., Zhang, Y. & Okatani, T. Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps with Accurate Object Boundaries. *arXiv preprint arXiv:1803.08673* (2018) (*p.* 43).

216. Zhou, T., Brown, M., Snavely, N. & Lowe, D. G. *Unsupervised Learning of Depth and Ego-Motion from Video* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 6612–6619 (*pp.* 45, 108, 110–111, 116, 119–120, 125–127, 131, 147–149, 151, 153–156).

217. Kuznietsov, Y., Stuckler, J. & Leibe, B. *Semi-Supervised Deep Learning for Monocular Depth Map Prediction* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 6647–6655 (*pp.* 45, 148, 155).

218. Jaderberg, M., Simonyan, K., Zisserman, A. *et al. Spatial Transformer Networks* in *Advances in Neural Information Processing Systems* (2015), 2017–2025 (*p.* 44).

219. Quiñonero Candela, J., Sugiyama, M., Schwaighofer, A. & Lawrence, N. Covariate Shift by Kernel Mean Matching. *Dataset Shift in Machine Learning,* 131–160 (2009) (*pp.* 46, 90).

220. Long, M., Cao, Y., Wang, J. & Jordan, M. *Learning Transferable Features with Deep Adaptation Networks* in *Int. Conf. Machine Learning* (2015), 97–105 (*p.* 47).

221. Ghifary, M., Kleijn, W. B., Zhang, M., Balduzzi, D. & Li, W. *Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation* in *Euro. Conf. Computer Vision* (2016), 597–613 (*p.* 47).

222. Ganin, Y. & Lempitsky, V. *Unsupervised Domain Adaptation by Backpropagation* in *Int. Conf. Machine Learning* (2015), 1180–1189 (*p.* 47).

223. Tzeng, E., Hoffman, J., Darrell, T. & Saenko, K. *Simultaneous Deep Transfer across Domains and Tasks* in *Int. Conf. Computer Vision* (2015), 4068–4076 (*p.* 47).

224. Donahue, J., Krähenbühl, P. & Darrell, T. *Adversarial Feature Learning* in *Int. Conf. Learning Representations* (2016), 1–11 (*p.* 47).

225. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. & Lempitsky, V. Domain-Adversarial Training of Neural Networks. *J. Machine Learning Research* **17,** 1–35 (2016) (*pp.* 47, 91).

226. Tzeng, E., Hoffman, J., Saenko, K. & Darrell, T. *Adversarial Discriminative Domain Adaptation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 2962–2971 (*pp.* 47, 91).

227. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K. & Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv preprint arXiv:1412.3474* (2014) (*p.* 47).

228. Sun, B. & Saenko, K. *Deep Coral: Correlation Alignment for Deep Domain Adaptation* in *Workshop in Int. Conf. Computer Vision* (2016), 443–450 (*pp.* 47, 91).

229. Liu, M.-Y. & Tuzel, O. *Coupled Generative Adversarial Networks* in *Advances in Neural Information Processing Systems* (2016), 469–477 (*pp.* 47, 91, 121).

230. Li, Y., Wang, N., Liu, J. & Hou, X. *Demystifying Neural Style Transfer* in *Int. Joint Conf. Artificial Intelligence* (2017), 2230–2236 (*pp.* 47–48, 116, 127, 153).

231. Schwerdtfeger, H. *Introduction to Linear Algebra and the Theory of Matrices* (P. Noordhoff Groningen, 1950) (*pp.* 47, 127).

232. Ulyanov, D., Lebedev, V., Vedaldi, A. & Lempitsky, V. S. *Texture Networks: Feed-Forward Synthesis of Textures and Stylized Images* in *Int. Conf. Machine Learning* (2016), 1349–1357 (*pp.* 47–48).

233. Chen, T. Q. & Schmidt, M. *Fast Patch-based Style Transfer of Arbitrary Style* in *Workshop in Constructive Machine Learning* (2016), 1–5 (*pp.* 47–48).

234. Schwerdtfeger, H. *Introduction to Linear Algebra and the Theory of Matrices* (P. Noordhoff, 1950) (*p.* 47).

235. Dumoulin, V., Shlens, J. & Kudlur, M. *A Learned Representation for Artistic Style* in *Int. Conf. Learning Representations* (2017), 1–11 (*p.* 48).

236. Ulyanov, D., Vedaldi, A. & Lempitsky, V. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv:1607.08022* (2016) (*p.* 48).

237. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. *Rethinking the Inception Architecture for Computer Vision* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 2818–2826 (*p.* 48).

238. Huang, X. & Belongie, S. *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization* in *Int. Conf. Computer Vision* (2017), 1510–1519 (*p.* 48).

239. Yanai, K. *Unseen Style Transfer Based on a Conditional Fast Style Transfer Network* in *Learning Representations Workshops* (2017), 1–4 (*p.* 48).

240. Jing, Y., Yang, Y., Feng, Z., Ye, J. & Song, M. Neural Style Transfer: A Review. *arXiv preprint arXiv:1705.04058* (2017) (*p.* 48).

241. Guo, G. & Lai, A. A Survey on Still Image Based Human Action Recognition. *Pattern Recognition* **47,** 3343–3361 (2014) (*p.* 49).

242. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. & Fei-Fei, L. *Large-Scale Video Classification with Convolutional Neural Networks* in *IEEE Conf. Computer Vision and Pattern Recognition* (2014), 1725–1732 (*p.* 49).

243. Simonyan, K. & Zisserman, A. *Two-Stream Convolutional Networks for Action Recognition in Videos* in *Advances in Neural Information Processing Systems* (2014), 568–576 (*p.* 49).

244. Wang, Y., Zhou, L. & Qiao, Y. *Temporal Hallucinating for Action Recognition With Few Still Images* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 5314–5322 (*p.* 49).

245. Carreira, J. & Zisserman, A. *Quo vadis, Action Recognition? a New Model and the Kinetics Dataset* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 4724–4733 (*p.* 49).

246. Tran, D., Bourdev, L., Fergus, R., Torresani, L. & Paluri, M. *Learning Spatiotemporal Features with 3D Convolutional Networks* in *Int. Conf. Computer Vision* (2015), 4489–4497 (*p.* 49).

247. Ji, S., Xu, W., Yang, M. & Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* **35,** 221–231 (2013) (*p.* 49).

248. Jain, A., Tompson, J., LeCun, Y. & Bregler, C. *Modeep: A Deep Learning Framework using Motion Features for Human Pose Estimation* in *Asian Conf. Computer Vision* (2014), 302–315 (*p.* 49).

249. Pfister, T., Simonyan, K., Charles, J. & Zisserman, A. *Deep Convolutional Neural Networks for Efficient Pose Estimation in Gesture Videos* in *Asian Conf. Computer Vision* (2014), 538–552 (*p.* 49).

250. Pfister, T., Charles, J. & Zisserman, A. *Flowing ConvNets for Human Pose Estimation in Videos* in *Int. Conf. Computer Vision* (2015), 1913–1921 (*p.* 49).

251. Charles, J., Pfister, T., Magee, D., Hogg, D. & Zisserman, A. *Upper Body Pose Estimation with Temporal Sequential Forests* in *British Machine Vision Conference* (2014), 1–12 (*p.* 49).

252. Oh, J., Guo, X., Lee, H., Lewis, R. & Singh, S. *Action-Conditional Video Prediction using Deep Networks in Atari Games* in *Advances in Neural Information Processing Systems* (2015), 2863–2871 (*p.* 49).

253. Yoo, Y., Yun, K., Yun, S., Hong, J., Jeong, H. & Young Choi, J. *Visual Path Prediction in Complex Scenes with Crowded Moving Objects* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 2668–2677 (*p.* 49).

254. Denton, E. L. *et al. Unsupervised Learning of Disentangled Representations from Video* in *Advances in Neural Information Processing Systems* (2017), 4414–4423 (*p.* 49).

255. Villegas, R., Yang, J., Hong, S., Lin, X. & Lee, H. Decomposing Motion and Content for Natural Video Sequence Prediction, 1–11 (2017) (*p.* 49).

256. Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K. & Woo, W.-c. *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting* in *Advances in Neural Information Processing Systems* (2015), 802–810 (*p.* 49).

257. Marwah, T., Mittal, G. & Balasubramanian, V. *Attentive Semantic Video Generation using Captions* in *Int. Conf. Computer Vision* (2017), 1435–1443 (*p.* 49).

258. Stollenga, M. F., Byeon, W., Liwicki, M. & Schmidhuber, J. *Parallel Multi-Dimensional LSTM, with Application to Fast Biomedical Volumetric Image Segmentation* in *Advances in Neural Information Processing Systems* (2015), 2998–3006 (*p.* 49).

259. Ondruska, P. & Posner, I. Deep Tracking: Seeing beyond Seeing using Recurrent Neural Networks. *Conf. Artificial Intelligence,* 3361–3367 (2016) (*p.* 49).

260. Xu, J., Ni, B., Li, Z., Cheng, S. & Yang, X. *Structure Preserving Video Prediction* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 1460–1469 (*p.* 49).

261. Zhu, X., Wang, Y., Dai, J., Yuan, L. & Wei, Y. *Flow-Guided Feature Aggregation for Video Object Detection* in *Int. Conf. Computer Vision* (2017), 408–417 (*p.* 49).

262. Saligrama, V. & Chen, Z. *Video Anomaly Detection based on Local Statistical Aggregates* in *IEEE. Conf. Computer Vision and Pattern Recognition* (2012), 2112–2119 (*p.* 49).

263. Huang, D.-A., Ramanathan, V., Mahajan, D., Torresani, L., Paluri, M., Fei-Fei, L. & Niebles, J. C. *What Makes a Video a Video: Analyzing Temporal Information in Video Understanding Models and Datasets* in *IEEE. Conf. Computer Vision and Pattern Recognition* (2018), 7366–7375 (*p.* 49).

264. Richard, A., Kuehne, H., Iqbal, A. & Gall, J. *NeuralNetwork-Viterbi: A Framework for Weakly Supervised Video Learning* in *IEEE. Conf. Computer Vision and Pattern Recognition* (2018), 1–10 (*p.* 49).

265. Horn, B. K. & Schunck, B. G. Determining Optical Flow. *Artificial Intelligence* **17,** 185–203 (1981) (*p.* 49).

266. Weickert, J., Bruhn, A., Brox, T. & Papenberg, N. in *Mathematical Models for Registration and Applications to Medical Imaging* 103–136 (2006) (*p.* 49).

267. Brox, T. & Malik, J. Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence* **33,** 500–513 (2011) (*p.* 49).

268. Bailer, C., Taetz, B. & Stricker, D. *Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation* in *Int. Conf. Computer Vision* (2015), 4015–4023 (*p.* 49).

269. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D. & Brox, T. *Flownet: Learning Optical Flow with Convolutional Networks* in *Int. Conf. Computer Vision* (2015), 2758–2766 (*pp.* 49–50).

270. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A. & Brox, T. *Flownet 2.0: Evolution of Optical Flow Estimation with Deep Networks* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 1647–1655 (*p.* 49).

271. Ranjan, A. & Black, M. J. *Optical Flow Estimation using a Spatial Pyramid Network* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 2720–2729 (*pp.* 49, 138, 143, 149).

272. Mostajabi, M., Yadollahpour, P. & Shakhnarovich, G. *Feedforward Semantic Segmentation with Zoom-out Features* in *IEEE Conf. Computer Vision and Pattern Recognition* (2015), 3376–3385 (*p.* 49).

273. Long, J., Shelhamer, E. & Darrell, T. *Fully Convolutional Networks for Semantic Segmentation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2015), 3431–3440 (*pp.* 49–50, 143, 146, 149, 154).

274. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Analysis and Machine Intelligence* **40,** 834–848 (2018) (*pp.* 49, 143, 146, 149, 154).

275. Ronneberger, O., Fischer, P. & Brox, T. *U-net: Convolutional Networks for Biomedical Image Segmentation* in *Int. Conf. Medical Image Computing and Computer-Assisted Intervention* (2015), 234–241 (*pp.* 49–50, 119, 121, 132–133).

276. Wang, X., You, S., Li, X. & Ma, H. *Weakly-Supervised Semantic Segmentation by Iteratively Mining Common Object Features* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 1354–1362 (*pp.* 49–50).

277. Badrinarayanan, V., Kendall, A. & Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* **39,** 2481–2495 (2017) (*pp.* 50, 71–73, 80, 83–84, 92, 99, 119, 146, 149, 152, 154).

278. Lin, G., Milan, A., Shen, C. & Reid, I. D. *RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation* in *IEEE Conf. Computer Vision and Pattern Recognition* **1** (2017), 5168–5177 (*p.* 50).

279. Zhao, H., Shi, J., Qi, X., Wang, X. & Jia, J. *Pyramid Scene Parsing Network* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 2881–2890 (*p.* 50).

280. Chen, L.-C., Yang, Y., Wang, J., Xu, W. & Yuille, A. L. *Attention to Scale: Scale-Aware Semantic Image Segmentation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 3640–3649 (*p.* 50).

281. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G. & Sang, N. *Learning a Discriminative Feature Network for Semantic Segmentation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 1–10 (*p.* 50).

282. Zhang, Y., Qiu, Z., Yao, T., Liu, D. & Mei, T. *Fully Convolutional Adaptation Networks for Semantic Segmentation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 6810–6818 (*p.* 50).

283. Sankaranarayanan, S., Balaji, Y., Jain, A., Lim, S. N. & Chellappa, R. *Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 1–10 (*p.* 50).

284. Lin, D., Dai, J., Jia, J., He, K. & Sun, J. *Scribblesup: Scribble-Supervised Convolutional Networks for Semantic Segmentation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 3159–3167 (*p.* 50).

285. Qi, X., Liu, Z., Shi, J., Zhao, H. & Jia, J. *Augmented Feedback in Semantic Segmentation under Image Level Supervision* in *Euro. Conf. Computer Vision* (2016), 90–105 (*p.* 50).

286. Fayyaz, M., Saffar, M. H., Sabokrou, M., Fathy, M., Klette, R. & Huang, F. *STFCN: Spatio-Temporal FCN for Semantic Video Segmentation* in *Workshop in Asian Conf. Computer Vision* (2016), 493–509 (*p.* 50).

287. Nilsson, D. & Sminchisescu, C. *Semantic Video Segmentation by Gated Recurrent Flow Propagation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 1–11 (*p.* 50).

288. Gadde, R., Jampani, V. & Gehler, P. V. *Semantic Video CNNs through Representation Warping* in *Int. Conf. Computer Vision* (2017), 4463–4472 (*p.* 50).

289. Shelhamer, E., Rakelly, K., Hoffman, J. & Darrell, T. *Clockwork ConvNets for Video Semantic Segmentation* in *Euro. Conf. Computer Vision* (2016), 852–868 (*p.* 50).

290. Zhu, X., Xiong, Y., Dai, J., Yuan, L. & Wei, Y. *Deep Feature Flow for Video Recognition* in *IEEE. Conf. Computer Vision and Pattern Recognition* (2017), 4141–4150 (*p.* 50).

291. Xu, Y.-S., Fu, T.-J., Yang, H.-K. & Lee, C.-Y. *Dynamic Video Segmentation Network* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 6556–6565 (*p.* 50).

292. Li, Y., Shi, J. & Lin, D. *Low-Latency Video Semantic Segmentation* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 5997–6005 (*p.* 50).

293. Masnou, S. & Morel, J.-M. *Level Lines Based Disocclusion* in *Int. Conf. Image Processing* (1998), 259–263 (*p.* 54).

294. Chum, O., Mikulik, A., Perdoch, M. & Matas, J. *Total Recall II: Query Expansion Revisited* in *IEEE Conf. Computer Vision and Pattern Recognition* (2011), 889–896 (*pp.* 54, 58).

295. Arandjelovic, R. & Zisserman, A. *Three Things Everyone Should Know to Improve Object Retrieval* in *IEEE Conf. Computer Vision and Pattern Recognition* (2012), 2911–2918 (*p.* 54).

296. Hamilton, O. & Breckon, T. *Generalized Dynamic Object Removal for Dense Stereo Vision Based Scene Mapping using Synthesised Optical Flow* in *Int. Conf. Image Processing* (IEEE, 2016), 3439–3443 (*pp.* 54, 66, 114).

297. Mroz, F. & Breckon, T. An Empirical Comparison of Real-time Dense Stereo Approaches for Use in the Automotive Environment. *Image and Video Processing* **2012,** 1–19 (2012) (*pp.* 54, 66).

298. Butterworth, C. Filter Approximation Theory. *Engineer* **7,** 536–541 (1930) (*pp.* 54, 56).

299. Solomon, C. & Breckon, T. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab* ISBN: 0470844736 (Wiley-Blackwell, 2010) (*p.* 56).

300. Gilboa, G. & Osher, S. Nonlocal Linear Image Regularization and Supervised Segmentation. *Multiscale Modeling & Simulation* **6,** 595–630 (2007) (*p.* 59).

301. Geiger, A., Lenz, P., Stiller, C. & Urtasun, R. Vision Meets Robotics: The KITTI Dataset. *Robotics Research,* 1231–1237 (2013) (*pp.* 63, 65–66, 72–73, 80–81, 83–84, 93–94, 119, 131, 143, 147–148).

302. Kumar, V., Mukhopadhyay, J. & Mandal, S. K. D. *Modified exemplar-based image inpainting via primal-dual optimization* in *Int. Conf. Pattern Recognition and Machine Intelligence* (2015), 116–125 (*p.* 65).

303. Hirschmuller, H. & Scharstein, D. *Evaluation of Cost Functions for Stereo Matching* in *IEEE Conf. Computer Vision and Pattern Recognition* (2007), 1–8 (*pp.* 65, 67–68, 82, 84–85).

304. Yamaguchi, K., McAllester, D. & Urtasun, R. *Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation* in *Euro. Conf. Computer Vision* (2014), 756–771 (*pp.* 67, 81, 83, 85).

305. Kendall, A., Badrinarayanan, V. & Cipolla, R. *Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding* in *British Machine Vision Conference* (2017), 1–12 (*pp.* 71, 146, 149, 154).

306. Fukunaga, K. & Hostetler, L. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Trans. Information Theory* **21,** 32–40 (1975) (*pp.* 72, 83).

307. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014) (*pp.* 73, 92, 127).

308. Sengupta, S., Greveson, E., Shahrokni, A. & Torr, P. H. *Urban 3D Semantic Modelling using Stereo Vision* in *Int. Conf. Robotics and Automation* (2013), 580–585 (*p.* 73).

309. Comaniciu, D. & Meer, P. Mean shift: A Robust Approach toward Feature Space Analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence* **24,** 603–619 (2002) (*pp.* 73, 83).

310. Beucher, S. & Meyer, F. The Morphological Approach to Segmentation: the Watershed Transformation. *Optical Engineering - New York* **34,** 433–433 (1992) (*p.* 73).

311. Xu, P., Davoine, F., Bordes, J.-B., Zhao, H. & Denœux, T. Multimodal Information Fusion for Urban Scene Understanding. *Machine Vision and Applications* **27,** 331–349 (2016) (*p.* 83).

312. Cheng, Y. Mean Shift, Mode Seeking, and Clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence* **17,** 790–799 (1995) (*p.* 83).

313. Kumar, N., Verma, R. & Sethi, A. Convolutional Neural Networks for Wavelet Domain Super-Resolution. *Pattern Recognition Letters* **90,** 65–71 (2017) (*p.* 87).

314. Dosovitskiy, A. & Brox, T. *Generating Images with Perceptual Similarity Metrics Based on Deep Networks* in *Advances in Neural Information Processing Systems* (2016), 658–666 (*pp.* 88, 117, 135).

315. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *Int. Conf. Computer Vision,* 2242–2251 (2017) (*pp.* 88, 117, 121–122, 124, 126–127, 129, 131, 153, 156).

316. Arjovsky, M., Chintala, S. & Bottou, L. *Wasserstein Generative Adversarial Networks* in *Int. Conf. Machine Learning* (2017), 214–223 (*pp.* 88, 90, 95).

317. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. C. *Improved Training of Wasserstein GANs* in *Advances in Neural Information Processing Systems* (2017), 5769–5779 (*pp.* 88, 90, 95, 97, 99, 101, 156).

318. Arjovsky, M. & Bottou, L. *Towards Principles for Training Generative Adversarial Networks* in *Int. Conf. Learning Representations* (2017), 1–17 (*pp.* 88, 90).

319. Chan, R. H., Chan, T. F. & Wong, C.-K. Cosine Transform Based Preconditioners for Total Variation Deblurring. *IEEE Trans. Image Processing* **8,** 1472–1478 (1999) (*p.* 88).

320. Raid, A., Khedr, W., El-Dosuky, M. & Ahmed, W. JPEG Image Compression using Discrete Cosine Transform - A Survey. *Computer Science & Engineering Survey* **5,** 1–9 (2014) (*p.* 88).

321. Chang, J. Y., Lee, K. M. & Lee, S. U. Stereo Matching using Iterative Reliable Disparity Map Expansion in the Color-Spatial-Disparity Space. *Pattern Recognition* **40,** 3705–3713 (2007) (*p.* 89).

322. Ioffe, S. & Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* in *Int. Conf. Machine Learning* (2015), 448–456 (*pp.* 92, 119, 133).

323. Pérez, P., Gangnet, M. & Blake, A. *Poisson Image Editing* in *ACM Trans. Graphics* **22** (ACM, 2003), 313–318 (*pp.* 97, 146–147, 149).

324. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. & Lerer, A. *Automatic Differentiation in PyTorch* in *Advances in Neural Information Processing Systems* (2017), 1–4 (*pp.* 102, 121, 124, 143).

325. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. & Darrell, T. *Caffe: Convolutional Architecture for Fast Feature Embedding* in *Int. Conf. Multimedia* (2014), 675–678 (*p.* 102).

326. Kingma, D. & Ba, J. *Adam: A Method for Stochastic Optimization* in *Int. Conf. Learning Representations* (2014), 1–15 (*pp.* 102, 121, 124, 143).

327. Woodham, R. J. Photometric Method for Determining Surface Orientation from Multiple Images. *Optical Engineering* **19,** 191139 (1980) (*p.* 114).

328. Li, B., Dai, Y., Chen, H. & He, M. Single Image Depth Estimation by Dilated Deep Residual Convolutional Neural Network and Soft-Weight-Sum Inference. *arXiv preprint arXiv:1705.00534* (2017) (*p.* 115).

329. Menze, M. & Geiger, A. *Object Scene Flow for Autonomous Vehicles* in *IEEE Conf. Computer Vision and Pattern Recognition* (2015), 3061–3070 (*pp.* 115, 118, 125–126, 129).

330. Shah, S., Dey, D., Lovett, C. & Kapoor, A. *AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles* in *Field and Service Robotics* (2017), 621–635 (*pp.* 116, 156).

331. Le, T. A., Baydin, A. G., Zinkov, R. & Wood, F. *Using Synthetic Data to Train Neural Networks is Model-Based Reasoning* in *Int. Joint Conf. Neural Networks* (2017), 3514–3521 (*p.* 116).

332. Rajpura, P. S., Hegde, R. S. & Bojinov, H. Object Detection using Deep CNNs Trained on Synthetic Images. *arXiv preprint arXiv:1706.06782* (2017) (*p.* 116).

333. Gaidon, A., Wang, Q., Cabon, Y. & Vig, E. *Virtual Worlds as Proxy for Multi-Object Tracking Analysis* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 4340–4349 (*p.* 116).

334. Yeh, R., Chen, C., Lim, T. Y., Hasegawa-Johnson, M. & Do, M. N. *Semantic Image Inpainting with Deep Generative Models* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 6882–6890 (*p.* 117).

335. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O. & Li, H. *High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 4076–4084 (*p.* 117).

336. Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. *Image-to-image translation with conditional adversarial networks* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 5967–5976 (*pp.* 117, 119, 127, 135).

337. Li, C., Zhao, X., Zhang, Z. & Du, S. Generative adversarial dehaze mapping nets. *Pattern Recognition Letters* (2017) (*p.* 117).

338. Walker, J., Marino, K., Gupta, A. & Hebert, M. *The pose knows: Video forecasting by generating pose futures* in *Int. Conf. Computer Vision* (2017), 3352–3361 (*p.* 117).

339. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. & Schiele, B. *The Cityscapes Dataset for Semantic Urban Scene Understanding* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 3213–3223 (*pp.* 119, 137, 143, 146).

340. Hinton, G. E. & Salakhutdinov, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science* **313,** 504–507 (2006) (*p.* 119).

341. Wang, X. & Gupta, A. *Generative Image Modeling using Style and Structure Adversarial Networks* in *Euro. Conf. Computer Vision* (2016), 318–335 (*p.* 119).

342. Mathieu, M., Couprie, C. & LeCun, Y. *Deep Multi-Scale Video Prediction beyond Mean Square Error* in *Int. Conf. Learning Representations* (2016), 1–14 (*p.* 119).

343. Radford, A., Metz, L. & Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434* (2015) (*pp.* 119, 143).

344. Rosales, R., Achan, K. & Frey, B. J. *Unsupervised Image Translation* in *Int. Conf. Computer Vision* (2003), 472–478 (*p.* 121).

345. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D. & Krishnan, D. *Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks* in *IEEE Conf. Computer Vision and Pattern Recognition* **1** (2017), 7 (*p.* 121).

346. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W. & Webb, R. *Learning from Simulated and Unsupervised Images through Adversarial Training* in *IEEE Conf. Computer Vision and Pattern Recognition* (2017), 2242–2251 (*pp.* 121, 124).

347. Wang, F., Huang, Q. & Guibas, L. J. *Image Co-Segmentation via Consistent Functional Maps* in *Int. Conf. Computer Vision* (2013), 849–856 (*p.* 121).

348. Zhou, T., Krahenbuhl, P., Aubry, M., Huang, Q. & Efros, A. A. *Learning Dense Correspondence via 3D-Guided Cycle Consistency* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 117–126 (*p.* 121).

349. Yi, Z., Zhang, H., Gong, P. T. *et al. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation* in *Int. Conf. Computer Vision* (2017), 2868–2876 (*p.* 121).

350. Mao, X., Li, Q., Xie, H., Lau, R. Y. & Wang, Z. Multi-Class Generative Adversarial Networks with the L2 Loss Function. *arXiv preprint arXiv:1611.04076* (2016) (*p.* 122).

351. He, K., Zhang, X., Ren, S. & Sun, J. *Deep Residual Learning for Image Recognition* in *IEEE Conf. Computer Vision and Pattern Recognition* (2016), 770–778 (*p.* 124).

352. Kendall, A., Gal, Y. & Cipolla, R. *Multi-Task Learning using Uncertainty to Weigh Losses for Scene Geometry and Semantics* in *IEEE Conf. Computer Vision and Pattern Recognition* (2018), 1–10 (*p.* 132).

353. Orhan, A. E. & Pitkow, X. *Skip Connections Eliminate Singularities* in *Int. Conf. Learning Representations* (2018), 1–11 (*pp.* 132–133).

354. Tong, T., Li, G., Liu, X. & Gao, Q. *Image Super-Resolution using Dense Skip Connections* in *Int. Conf. Computer Vision* (2017), 4809–4817 (*pp.* 132–133).

355. Yamanaka, J., Kuwashima, S. & Kurita, T. *Fast and Accurate Image Super Resolution by Deep CNN with Skip Connection and Network in Network* in *Neural Information Processing* (2017), 217–225 (*pp.* 132–133).

356. He, K., Zhang, X., Ren, S. & Sun, J. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification* in *Int. Conf. Computer Vision* (2015), 1026–1034 (*p.* 133).

357. Heise, P., Klose, S., Jensen, B. & Knoll, A. *Pm-huber: Patchmatch with Huber Regularization for Stereo Matching* in *Int. Conf. Computer Vision* (2013), 2360–2367 (*p.* 138).

358. Brostow, G. J., Fauqueur, J. & Cipolla, R. Semantic Object Classes in Video: A High-Definition Ground Truth Database. *Pattern Recognition Letters* **30,** 88–97 (2009) (*pp.* 140, 143, 146).

359. Alhaija, H., Mustikovela, S., Mescheder, L., Geiger, A. & Rother, C. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes. *Int. J. Computer Vision* **126,** 961–972 (2018) (*pp.* 141, 143, 146).

360. Menze, M., Heipke, C. & Geiger, A. Object Scene Flow. *Photogrammetry and Remote Sensing,* 60–76 (2018) (*pp.* 141, 143, 148).

361. Butler, D. J., Wulff, J., Stanley, G. B. & Black, M. J. *A Naturalistic Open Source Movie for Optical Flow Evaluation* in *Euro. Conf. Computer Vision* (2012), 611–625 (*p.* 143).

362. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C. & Torr, P. H. *Conditional Random Fields as Recurrent Neural Networks* in *Int. Conf. Computer Vision* (2015), 1529–1537 (*pp.* 143, 146, 149, 154).

363. Uhrig, J., Cordts, M., Franke, U. & Brox, T. *Pixel-level encoding and depth layering for instance-level semantic labeling* in *German Conf. Pattern Recognition* (2016), 14–25 (*pp.* 143, 146, 149, 154).

364. Liu, Z., Li, X., Luo, P., Loy, C.-C. & Tang, X. *Semantic Image Segmentation via Deep Parsing Network* in *Int. Conf. Computer Vision* (2015), 1377–1385 (*pp.* 143, 146, 149, 154).

365. Noh, H., Hong, S. & Han, B. *Learning Deconvolution Network for Semantic Segmentation* in *Int. Conf. Computer Vision* (2015), 1520–1528 (*pp.* 146, 149, 154).

366. Visin, F., Ciccone, M., Romero, A., Kastner, K., Cho, K., Bengio, Y., Matteucci, M. & Courville, A. *Reseg: A Recurrent Neural Network-Based Model for Semantic Segmentation* in *Workshop in IEEE Conf. Computer Vision and Pattern Recognition* (2016), 41–48 (*pp.* 146, 149, 154).

367. Geiger, A., Lenz, P. & Urtasun, R. *Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite* in *IEEE Conf. Computer Vision and Pattern Recognition* (2012), 3354–3361 (*p.* 150).

368. Benko, H., Jota, R. & Wilson, A. *MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop* in *SIGCHI Conf. Human Factors in Computing Systems* (2012), 199–208 (*p.* 150).

369. Castellani, U., Livatino, S. & Fisher, R. B. *Improving Environment Modelling by Edge Occlusion Surface Completion* in *Int. Symp. 3D Data Processing Visualization and Transmission* (2002), 672–675 (*p.* 150).

370. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. *et al.* Gradient-based Learning Applied to Document Recognition. *Proc. IEEE* **86,** 2278–2324 (1998) (*p.* 156).

371. Mejjati, Y. A., Richardt, C., Tompkin, J., Cosker, D. & Kim, K. I. *Unsupervised Attention-Guided Image-to-Image Translation* in *Advances in Neural Information Processing Systems* (2018), 3693–3703 (*p.* 156).