

Durham E-Theses

Microstructure of absorber layers in CdTe/Cds solar cells

Michael Andrew Cousins

How to cite:

Cousins, Michael Andrew (2001) Microstructure of absorber layers in CdTe/Cds solar cells. Doctoral thesis, Durham University.

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a <https://etheses.durham.ac.uk/id/eprint/4266/> is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full Durham E-Theses policy](#) for further details.

Microstructure of Absorber Layers in CdTe/CdS Solar Cells

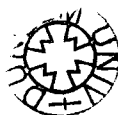
The copyright of this thesis rests with the author. No quotation from it should be published in any form, including Electronic and the Internet, without the author's prior written consent. All information derived from this thesis must be acknowledged appropriately.

By

Michael Andrew Cousins, BSc, MSc, DIC

*A thesis presented in candidature for the degree
of Doctor of Philosophy in the University of
Durham*

*Department of Physics
April 2001*



- 8 MAR 2002

ABSTRACT

This work concerns the microstructure of CSS-grown CdTe layers used for CdTe/CdS solar cells. Particular attention is given to how the development of microstructure on annealing with CdCl₂ may correlate with increases in efficiency.

By annealing pressed pellets of bulk CdTe powder, it is shown that microstructural change does occur on heating the material, enhanced by the inclusion of CdCl₂ flux. However, the temperature required to cause significant effects is demonstrated to be higher than that at which heavy oxidation takes place. The dynamics of this oxidation are also examined.

To investigate microstructural evolution in thin-films of CdTe, bi-layers of CdTe and CdS are examined by bevelling, thus revealing the microstructure to within ~1 µm of the interface. This allows optical microscopy and subsequent image analysis of grain structure. The work shows that the grain-size, which is well described by the Rayleigh distribution, varies linearly throughout the layer, but is *invariant* under CdCl₂ treatment. Electrical measurements on these bi-layers, however, showed increased efficiency, as is widely reported. This demonstrates that the efficiency of these devices is not dictated by the bulk microstructure.

Further, the region within 1 µm of the interface, of similar bi-layers to above, is examined by plan-view TEM. This reveals five-fold grain-growth on CdCl₂ treatment. Moreover, these grains show a considerably smaller grain size than expected from extrapolating the linear trend in the bulk. These observations are explained in terms of the pinning of the CdTe grain size to the underlying CdS, and the small grain size this causes. A simple model was proposed for a link between the grain-growth to the efficiency improvement. The study also examines the behaviour of defects within grains upon CdCl₂ treatment provided the first direct evidence of recovery on CdCl₂ treatment in this system.

Finally, a computer model is presented to describe the evolution of microstructure during growth. This is shown to be capable of reproducing the observed variation in grain size, but its strict physical accuracy is questioned.

DECLARATION

I declare that all the work presented in this thesis was carried out by the candidate. I also declare that none of this work has previously been submitted for any degree and that it is not being submitted for any other degree.


.....

Dr. K. Durose

Supervisor


.....

M. A. Cousins

Candidate

The copyright of this thesis rests with the author. No quotation from it should be published without their prior written consent and information derived from it should be acknowledged.

ACKNOWLEDGEMENTS

I would like to thank everyone who has helped me during the course of my PhD. Particular thanks are due to my supervisor Dr. Ken Durose: this work would not have been possible without his encouragement, enthusiasm, and appreciation of a good double-entendre.

I am also grateful to Dr. Andy Brinkman for his input, and for the invaluable technical assistance of Norman Thompson, David Pattinson and John Scott.

Acknowledgement must also be given to members of the II-VI group past and present, who have not only given valuable input, but have also made the group a pleasure to work in: Nick Aitken, Saleh Al-Amri, Josep Carles-Alabert, Arnab Basu, Nick Boyall, Ben Cantwell, Paul Edwards, Tooraj Hashemi, Matt Hogan, John Mullins, Mehmet Parlak, Mark Potter, Harpreet Sanghera, Rainer Schmidt, Thomas Schmidt, David Smyth-Boyle and Andrew Yates.

Finally, the utmost thanks must go to Siobhan, who kept me at least close to sane throughout my studies.

This work was funded by the Engineering and Physical Sciences Research Council.

ABBREVIATIONS

AFM– Atomic Force Microscopy

CBD– Chemical Bath Deposition

CSS– Close Space Sublimation

CVD– Chemical Vapour Deposition

EBIC– Electron-Beam Induced Current

ED– Electro-deposition

HRTEM– High Resolution Transmission Electron Microscopy

PVD– Physical Vapour Deposition

RBS– Rutherford Back-Scattering

REBIC– Remote Electron-Beam Induced Current

RFS– Radio Frequency Sputtering

SEM– Secondary Electron Microscopy

TEM– Transmission Electron Microscopy

XRD– X-Ray Diffraction

CONTENTS

1 Introduction	1
2 Fundamentals of Photovoltaic Energy Conversion	5
2.1 Introduction	5
2.2 The Photovoltaic Effect	6
2.3 Solar Cell Characterisation and Concepts	7
2.3.1 CURRENT-VOLTAGE CHARACTERISTICS	7
2.3.2 SOLAR RADIATION STANDARDS	11
2.3.3 QUANTUM EFFICIENCY	12
2.4 Solar Cell Configurations	12
2.4.1 THE SCHOTTKY JUNCTION	12
2.4.2 THE HOMOJUNCTION	14
2.4.3 THE HETEROJUNCTION	15
2.5 Solar Cell Materials	16
2.5.1 SINGLE CRYSTAL MATERIAL	16
2.5.2 MULTI-CRYSTALLINE MATERIAL	17
2.5.3 AMORPHOUS MATERIAL	17
2.5.4 POLYCRYSTALLINE MATERIAL	17
2.6 References	19
3 Aspects of Polycrystallinity	20
3.1 Introduction	20
3.2 Structural Changes during Growth and Processing	21
3.2.1 RECOVERY	21
3.2.2 RECRYSTALLISATION	22
3.3 Grain Growth	23
3.3.1 STABLE POLYCRYSTALLINE STRUCTURES	24
3.3.2 GRAIN-GROWTH DYNAMICS	24
3.4 References	29

4 CdTe/CdS Thin-Film Solar Cells	30
4.1 Introduction	30
4.2 Cell Structure, Composition and Construction	31
4.2.1 THE LAYERS AND THEIR FUNCTIONS	32
4.2.2 DEPOSITION TECHNIQUES	34
4.3 Microstructure of the Untreated CdTe Layer	37
4.3.1 PHYSICAL VAPOUR DEPOSITION	37
4.3.2 CLOSE SPACE SUBLIMATION	38
4.3.3 ELECTRODEPOSITION	40
4.3.4 RADIO FREQUENCY SPUTTERING	40
4.3.5 CHEMICAL VAPOUR DEPOSITION	40
4.3.6 SCREEN PRINTING AND SPRAY PYROLYSIS	41
4.4 Post-deposition treatment of CdTe/CdS Structures	41
4.5 Microstructural Effects of the CdCl ₂ Treatment	43
4.5.1 RECOVERY	44
4.5.2 RECRYSTALLISATION	44
4.5.3 GRAIN GROWTH	46
4.6 References	48
5 Experimental Techniques and Image Analysis	53
5.1 Introduction	53
5.2 Material Growth	53
5.3 Solar Cell Characterisation	54
5.3.1 CURRENT-VOLTAGE MEASUREMENTS	54
5.3.2 SPECTRAL RESPONSE	56
5.4 Bevelling	58
5.5 Defect Revealing Etches	61
5.6 Image Analysis	64
5.6.1 SPATIAL FILTERING	64
5.6.2 FLOOD-FILLING	66
5.6.3 SKELETONISATION	68
5.7 TEM	70
5.8 References	73

6 Sintering Effects in CdTe Pressed Pellets	74
6.1 Introduction	74
6.2 Sintering of Pressed Pellets	75
6.2.1 SINTERING KINETICS AND MECHANISM	75
6.2.2 EXPERIMENTAL DETAILS	77
6.2.3 SINTERING RESULTS & DISCUSSION	78
6.3 Oxidation Study of CdTe	82
6.3.1 INTRODUCTION	82
6.3.2 THE DEAL-GROVE MODEL FOR OXIDATION	83
6.3.3 OXIDATION RESULTS AND DISCUSSION	84
6.4 Conclusions	87
6.5 References	89
7 Grain Size and Performance of CdTe/CdS Solar Cells	90
7.1 Introduction	90
7.2 Sample Preparation and Measurement	91
7.2.1 CdCl_2 TREATMENT	92
7.2.2 DEVICE FABRICATION	92
7.2.3 POLISHING	93
7.2.4 BEVELLING	93
7.2.5 DEFECT SELECTIVE ETCHING	94
7.2.6 PROFILOMETRY AND MICROSCOPY	94
7.3 Data Extraction and Interpretation Methods	95
7.3.1 DATA EXTRACTION FROM THE IMAGES	95
7.3.2 QUANTITATIVE INTERPRETATION OF GRAIN SIZE AND LOCATION	98
7.4 Results	101
7.4.1 SPECTRAL RESPONSE AND J-V	101
7.4.2 GRAIN SIZE DISTRIBUTIONS	103
7.4.3 THE AVERAGE GRAIN SIZE AS A FUNCTION OF DEPTH	107
7.4.4 SPATIAL ANALYSIS	110
7.5 Discussion	111
7.6 Conclusions	115
7.7 References	119
8 TEM of the Near-Interface Region	121
8.1 Introduction	121

8.2 Experimental Details	122
8.3 Results	123
8.3.1 OVERVIEW	123
8.3.2 GRAIN SIZE	125
8.3.3 FURTHER RESULTS	127
8.4 Discussion	130
8.5 Conclusions	138
8.6 References	140
9 Computer Simulation of Absorber-Layer Growth	142
9.1 Introduction	142
9.2 Precepts of Model Behaviour	143
9.3 Results	146
9.3.1 OVERVIEW	146
9.3.2 QUANTIFICATION OF STRUCTURE	147
9.4 Discussion	151
9.4.1 COLUMNAR GROWTH	152
9.4.2 GRAIN SIZE VARIATION	153
9.4.3 THE PHYSICALITY OF THE MODEL	157
9.5 Conclusions	157
9.6 References	159
10 Conclusions	160
Appendix A— C++ Code: Image Analysis	167
A.1 Introduction	167
A.2 Image Analysis	168
A.2.1 SPATIAL FILTERING	168
A.2.2 SKELETONISATION	168
A.2.3 FLOOD FILLING	169
A.3 Driver Routine	170
A.4 Miscellaneous Routines	172
A.4.1 IMAGE MANIPULATION ROUTINES	172

A.4.2 GENERAL AND FILE I/O ROUTINES	173
A.5 Utility Routines and Header Files	175
A.5.1 BITMAP UTILITY ROUTINE	175
A.5.2 BITMAP HEADER	177
A.5.3 ARRAY HEADER	177
A.5.4 DATA TYPES HEADER	178
Appendix B— C++ Code: Levenberg-Marquardt Non-Linear Fitting	179
B.1 Introduction	179
B.2 The Levenberg-Marquardt Method	180
B.3 The Code	181
B.3.1 THE DRIVER ROUTINE	181
B.3.2 THE FITTING CLASS	182
B.3.3 THE FITTING CLASS HEADER	183
B.3.4 THE FUNCTIONS	183
B.3.5 THE MATRIX SOLVER	183
B.3.6 FILE I/O HEADER	184
Appendix C— C++ Code: Deposition Model	185
C.1 Introduction	185
C.2 The Driver Routine	185
C.3 The Main Model Class	187
C.4 The Model Header	189
C.5 The 3D Array Class	189
Appendix D— List of Publications	191

Introduction

In the recent past, it has become increasingly clear that fossil fuels provide neither a long-term nor an environmentally benign means of energy production. Increasing fears concerning global warming amongst the populace— and its acceptance as fact in scientific circles— has lead to an increased interest in clean, renewable energy sources to satisfy the world's growing demands.

One energy source that is both renewable and non-polluting is solar energy conversion, by means of photovoltaic solar cells. These produce electricity directly from sunlight by means of carrier generation at a semiconductor junction, and in addition to having reasonably high efficiencies, are maintenance free, and produce no by-products, other than heat, during operation. Although fears have been expressed concerning the environmental impact of some materials used in their construction, it is clear that photovoltaic solar cells have considerable advantages over fossil fuel alternatives.



Silicon homojunction solar cells have been available since the 1950's, and are now very widely used for small scale and space-based applications. However, the commercial production of such cells has proved to be prohibitively expensive for large-area applications, such as would be required for high-wattage terrestrial installations. Thus, attention has turned to cheaper photovoltaic solar-cell technologies. One of the most promising candidates for large-scale devices uses a heterojunction between thin films of polycrystalline CdTe and CdS. CdTe has long been known as a promising candidate for solar cell manufacture as it is an excellent absorber, and its band-gap compares well to the maximum in the solar spectrum. The wide band-gap of CdS and acceptable lattice mismatch with CdTe makes it an excellent material to complement CdTe in forming a p - n junction. In addition, the polycrystalline nature of the layers allows very economical fabrication, and the minimal thickness of the layers minimises materials costs. Commercial production of these devices is soon to be commenced: in mass-produced form, module efficiencies approach a modest 10%. However, comparing the cost of these devices ($\sim$ €1/W) with single crystal silicon modules ($\sim$ €20/W and $\sim$ 20% efficiency), the economy of the CdTe/CdS technology is clear.

As is usual, laboratory values of current maximum efficiency are slightly higher than the module figure quoted above. For small-area CdTe/CdS devices, current maximum efficiencies are about 16%. However, theory predicts a maximum device efficiency of $\sim$ 30%: the reasons for the discrepancy in practice and theory form an active area of research by a considerable number of different groups. However, the understanding of these devices in terms of traditional models (derived from considerations of single crystal device behaviour) is confounded by the polycrystalline nature of CdTe/CdS cells. Moreover, a post-deposition annealing stage of the device in

the presence of cadmium chloride (CdCl_2) has been shown to improve device performance, often by an order of magnitude over the as-grown state. Although polycrystalline material is generally viewed as a poor choice for electronic applications, the CdCl_2 anneal and grain-boundary structure interact to produce an acceptable device: the exact mechanisms for this remain unclear.

The purpose of this work is to study the microstructure of the CdTe absorber layers in CdTe/CdS solar cells and how the CdCl_2 treatment modifies this, with a view to better understanding the mechanisms behind the increase in efficiency on annealing. Initially, this work will discuss the principles of photovoltaic energy conversion, and the diverse structures and materials used to achieve this (chapter 2). Following on from this, in chapter 3, the way in which polycrystalline microstructures are able to evolve during annealing processes will be examined in terms of standard metallurgical phenomena. To conclude the review part of this work, chapter 4 will turn to previously published work on CdTe/CdS polycrystalline thin-film solar cells to outline the state of current thinking on manufacture and processing.

The experimental results of this work are divided into four sections, preceded by a chapter describing the experimental techniques used in them (chapter 5). The results chapters are divided as follows:

- a) Firstly, a study of how CdCl_2 and heat affect the microstructure of compressed CdTe powder is described (chapter 6). In addition, this chapter also studies how the annealing process affects the onset and nature of oxidation in CdTe.
- b) Following this, in chapter 7, the action of annealing and CdCl_2 on CdTe is further studied, but in CdTe/CdS bi-layers suitable for devices. This

study reveals the microstructure throughout the bulk of the CdTe layer.

In addition to these measurements, the material studied was also subject to electrical measurements.

- c) Chapter 8 examines the microstructure of the CdTe/CdS interface region using TEM, work which is especially significant, as the metallurgical interface is the region of most intense carrier generation
- d) Finally in Chapter 9 the results of microstructural investigation are further examined by means of a computer model. The aim of this is to reproduce microstructural phenomena found in the CdTe layers.

After concluding in chapter 10, the computer code required for this work is described and reproduced in the appendices. As mentioned above, however, this work commences by describing the general principles of photovoltaic energy conversion.

Chapter 2

Fundamentals of Photovoltaic Energy Conversion

2.1 INTRODUCTION

The discovery of the photovoltaic effect is attributed to Becquerel in 1839: he noted that on illuminating an electrolytic solution, current flowed between the electrodes immersed therein [1]. It was, however, a further 38 years until Adams detected this effect in the solid state [2] in a selenium device. The first photovoltaic device with sufficient efficiency to develop a useful amount of power (and thus the first true solar cell) was reported in 1956 by Chapin *et al.* [3]. It was constructed using a single-crystal silicon homojunction, and this remains the most common solar cell construction to date for small-area applications. The first feasible CdTe/CdS cells were produced in the early 1970's (for example, see Bonnet and Rabenhorst [4]),

and so it can be seen that the devices that concern us here are relative newcomers to the field of photovoltaics.

The following two chapters discuss some features of the behaviour of polycrystalline material (chapter 3), and attributes specific to polycrystalline thin-film CdTe/CdS solar cells in chapter 4. Firstly, though, this chapter will describe some of the fundamental concepts behind photovoltaics and photovoltaic cell construction.

2.2 THE PHOTOVOLTAIC EFFECT

The photovoltaic effect can be conveniently separated into two processes: firstly, carriers are generated by the action of light, and secondly, these carriers are separated by a potential, and collected to produce a photocurrent. These two effects will be studied in turn.

If a photon of energy $h\nu$ interacts with an ideal semiconductor with band-gap E_g , there are three possible outcomes:

- a) Firstly, if $E_g \leq h\nu < 2E_g$, an electron-hole pair will be created by the promotion of an electron across the energy gap. If $h\nu > E_g$, then the electron will usually thermalise to the band-edge by the emission of phonons.
- b) Secondly, If $h\nu < E_g$, nothing will occur, as there are no inter-band states in this ideal material.
- c) Finally, if $h\nu \geq 2E_g$, an electron promoted across the band-gap has sufficient energy to create one or more further electron-hole pairs in a process known as 'impact ionisation'. This can continue until the electron's energy drops below $2E_g$ after which it will then behave as in a) above.

Any carrier pairs generated by such photonic interactions will quickly recombine unless an electric field is applied to separate them. In a real device, of course, extrinsic transitions will both allow the absorption of photons with $h\nu < E_g$, and encourage the recombination of electrons and holes. This is especially true at material interfaces, where surface states produce deep donor and acceptor levels.

As mentioned above, carrier collection is achieved by separating the electrons and holes by means of an electric field. In some devices, this is supplied externally by biasing the device (such as in solid state x-ray detectors). In the case of solar cells, though, the electric field associated with a junction between two differently doped materials is used to sweep the carriers apart. The characteristics associated with a typical junction of this type, the p - n junction, are described below.

2.3 SOLAR CELL CHARACTERISATION AND CONCEPTS

2.3.1 CURRENT-VOLTAGE CHARACTERISTICS

Current density-voltage (J - V) measurements are the most important method of characterising a solar cell. The basis of the method is to measure the current produced by a solar cell at different biases either with illumination (so-called 'light' J - V) or without ('dark' J - V). All the major parameters pertaining to device performance can be extracted from such measurements. The origin of these parameters is outlined below.

The current density without illumination, J_d , flowing through an ideal p - n junction solar cell with a voltage V across it is given by Shockley's diode equation [5]:

$$J_d = J_0 \left[\exp\left(\frac{qV}{AkT}\right) - 1 \right], \quad (2.1)$$

where J_0 is a constant dependent on materials properties known as the reverse saturation current, and A is the diode ideality factor, and is dependent upon the type of transport involved. Under illumination, a term J_{phot} is added to account for the extra current generated by the photon flux, giving the light current, J_l :

$$J_l = J_0 \left[\exp\left(\frac{qV}{AkT}\right) - 1 \right] - J_{phot}. \quad (2.2)$$

Curves for equations 2.1 and 2.2 are shown in figure 2.1. It is also common to see these curves inverted around the V axis, as in the results of section 7.4.1.

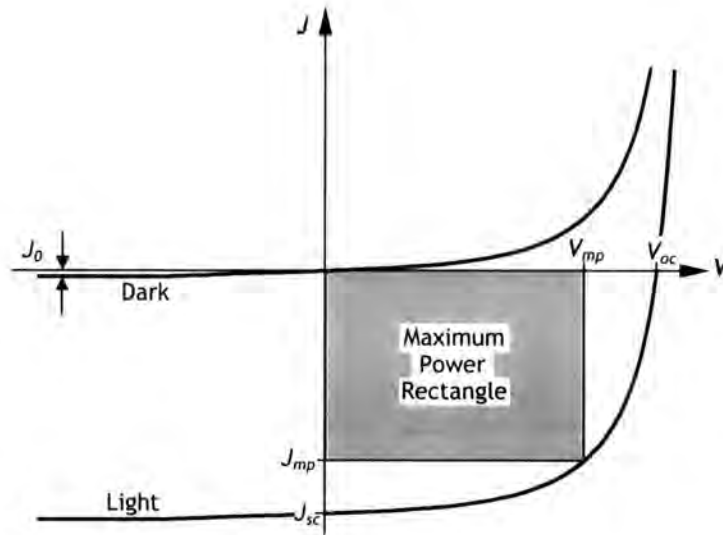


Figure 2.1: Typical J - V curves for a solar cell, with illumination and without: the parameters marked are described in the text. The typical size of J_0 has been increased for clarity.

Whilst parameters of the light J - V curve could be expressed in terms of those in equation 2.2, it is more usual to use a number of other descriptors. There are four of these, indicated in the figure:

- a) V_{oc} , the open circuit voltage, is the potential at which the dark current is balanced by the light induced current;
- b) J_{sc} , the short-circuit current, is the current flowing at zero bias voltage (in fact, equal to J_{phot}), and;
- c) J_{mp} and V_{mp} , the current and voltage defined by the maximum power rectangle. This represents the largest rectangle that can be fitted inside the curve as it passes through the fourth quadrant, and thus the highest power available from the device.

These parameters can be used to calculate two further important parameters. The fill factor, FF , is given by,

$$FF = \frac{J_{mp} V_{mp}}{J_{sc} V_{oc}}, \quad (2.3)$$

and describes the ‘squareness’ of the curve in the fourth quadrant: the squarer the curve, the larger the values of J_{mp} and V_{mp} for a given J_{sc} and V_{oc} . Most important of all is the efficiency,

$$\eta = \frac{J_{mp} V_{mp}}{P_{in}}, \quad (2.4)$$

the ratio of the maximum power available to the incident photon power, P_{in} .

Whilst the Shockley equation is valid for an ideal p - n junction, a considerable number of factors can contribute to deviations from this behaviour, primarily resistances. The series resistance, R_{se} , arises simply from the resistivity of the layers of the device, and acts to limit the current available from the device. Shunt resistance, R_{sh} can often be attributed to short-circuit paths through the device layers, such as pinholes, but excess

recombination will also contribute to this. In contrast to the series resistance, this will limit V_{oc} . The simplest equivalent circuit to model these effects [6] is shown on the left-hand side of figure 2.2. Examination of this circuit yields a modified equation for the J - V relationship:

$$J = J_0 \left\{ e^{\frac{q(V - R_{se}J)}{AkT}} - 1 \right\} + \frac{V - R_{se}J}{R_{sh}} - J_L. \quad (2.5)$$

Whilst this equivalent circuit and equation may be suitable for solar cells with ohmic contacts, some solar cells to which it is difficult to form a back contact (a category which includes CdTe/CdS cells) show so-called ‘roll-over’. This levelling off of the J - V characteristic at forward bias has been attributed to rectifying behaviour at the back contact, and can be modelled by the addition of another diode in series [7], shown on the right hand side of figure 2.2.

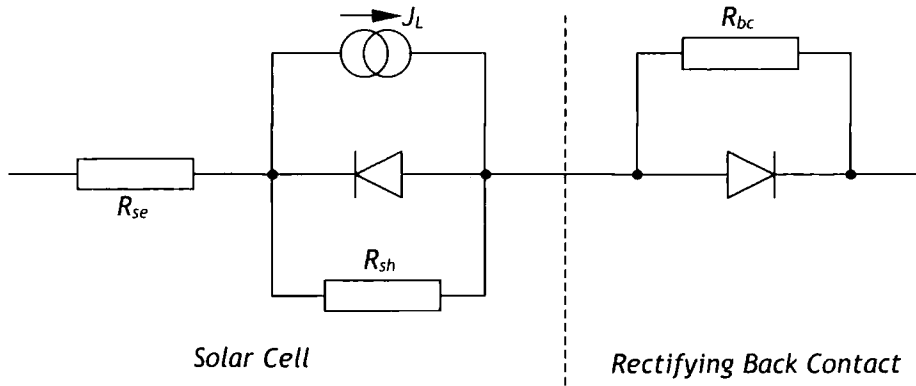


Figure 2.2: On the left of the figure is shown a simple equivalent circuit for a solar cell, valid for devices with ohmic back contacts. The circuit on the right can be added to model the ‘roll-over’ behaviour associated with a rectifying back contact of resistance R_{bc} .

2.3.2 SOLAR RADIATION STANDARDS

All of the parameters described in the previous section are affected by the nature of the incident light. In order to standardise the characterisation procedure, an ‘air mass’ figure can be defined. In this standard, the thickness of the atmosphere is defined as unity, and the air mass (AM) number reflects how many effective atmospheres light has to pass through to reach the earth's surface, averaged over one year. Thus, the spectral irradiance arriving at the outer atmosphere would be AM 0, and at the equator, AM 1. A comparison of some such spectra is given by Kazmerski [6]. With simple geometry, it can be shown that the latitude, θ , is related to the air mass number by,

$$AM \approx \sec \theta. \quad (2.6)$$

A commonly used standard is AM 1.5, reflecting the spectrum at temperate latitudes. This is shown in figure 2.3.

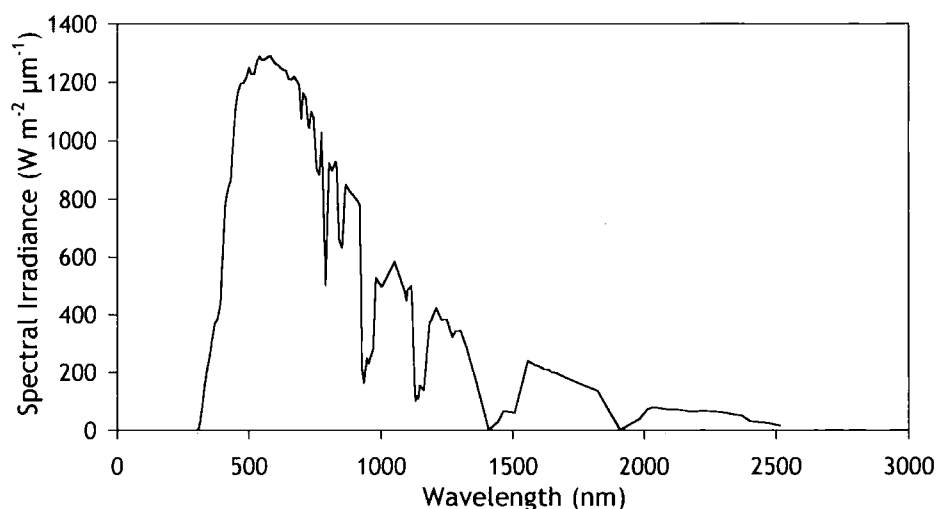


Figure 2.3: The AM 1.5 Spectrum (data from Fahrenbruch and Bube [8]).

Also of importance in standardising test results for solar cells is the light power incident on the cell. In common with most standards laboratories, a power of 100 mW cm^{-2} was used in this work, roughly equal to the solar power density at the earth's surface at temperate latitudes.

2.3.3 QUANTUM EFFICIENCY

In contrast to the power efficiency described above, the quantum efficiency (QE) describes the number of electrons produced per incident photon. Strictly speaking, two different quantum efficiencies can be defined: the internal quantum efficiency, which is adjusted for reflection losses at the front surface; and external quantum efficiency, the uncorrected figure. It should also be noted that quantum efficiencies greater than unity are possible through impact ionisation, discussed in section 2.2.

As the number of electrons produced by a photon is strongly dependent on its energy, QE is commonly measured as a function of wavelength, a so-called spectral response measurement. The response of a device to a particular photon energy can provide significant information concerning structure and composition of a device. Spectral response, and the apparatus for taking such measurements, is discussed further in section 5.3.2.

2.4 SOLAR CELL CONFIGURATIONS

A number of different junction structures can be used to produce the necessary field for charge separation. Whilst all have esoteric adaptations, the main structures from which they are derived are described here. In all cases, however, the most important parameter is V_{bi} , the built in potential which gives rise to the electric field. In each case below, a particular combination of p - and n -type materials has been arbitrarily chosen: in all cases, band diagrams can be constructed for alternative materials choices.

2.4.1 THE SCHOTTKY JUNCTION

The Schottky junction (also known as the metal-semiconductor junction) solar cell is perhaps the simplest of the device configurations. A

layer of metal, thin enough to allow most light to pass through, is deposited on a semiconductor substrate. This can be performed by a low temperature deposition method, such as evaporation, giving economy and simplicity of construction. The band bending caused by the contact between these differing materials produces the necessary electric field to separate the charge carriers, as shown in the band diagram of a Schottky junction shown in figure 2.4.

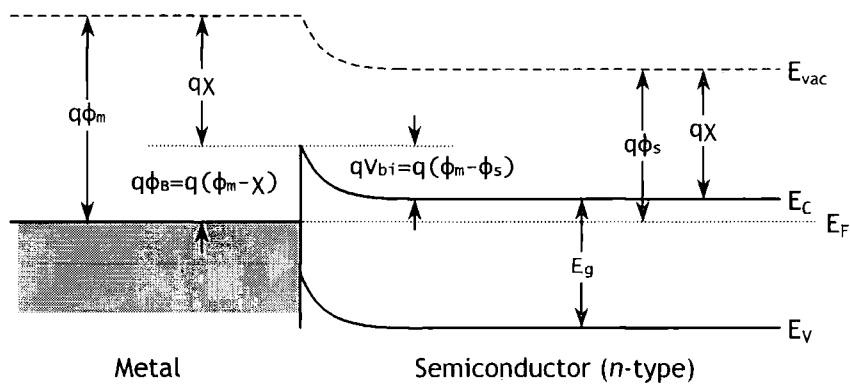


Figure 2.4: Band diagram for a Schottky junction. The junction is formed between an n -type semiconductor— of band-gap E_g , work function χ , and electron affinity ϕ_s — and a metal of electron affinity ϕ_m . The Schottky barrier height formed is ϕ_B .

Although the presence of a depletion region at the semiconductor surface gives rise to high currents from this type of device, in applications where a higher voltage is needed a metal-insulator-semiconductor (MIS) structure can be used. This involves the insertion of a thin layer of oxide at the interface. The mechanisms by which the oxide improves performance are diverse, and are reviewed by Pulfrey [9]. Efficiencies of up to 18% (at AM 1) have been achieved for these structures [5].

2.4.2 THE HOMOJUNCTION

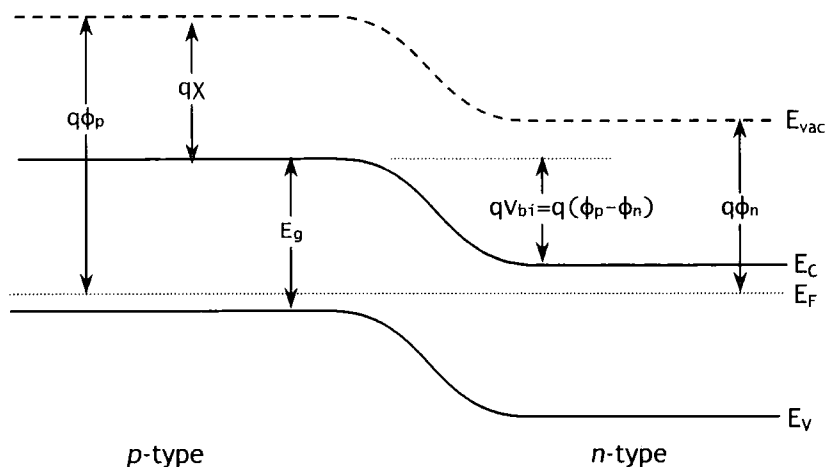


Figure 2.5: Band diagram of a homojunction between p - and n -type regions of electron affinity ϕ_p and ϕ_n respectively. This gives rise to a built-in potential V_{bi} . The energy gap, E_g , and electron affinity, χ , are also shown.

Homojunctions differ from the other two types of cell described here inasmuch as both sides of the junction are formed from the same substance. In order to produce a carrier concentration differential, species are introduced into one side of the cell to change the doping, often by diffusion or ion implantation.

In its favour, this system has no possibilities of discontinuities in its band structure to act as barriers to the flow of electrons, as shown in figure 2.5, and lacks interface states that could encourage recombination. Disadvantageously, though, whilst the material must be a good absorber, this also forces the p - n junction to be near the surface. This can cause problems with surface recombination and thus current loss. These problems notwithstanding, homojunctions form the vast majority of commercially available cells.

2.4.3 THE HETEROJUNCTION

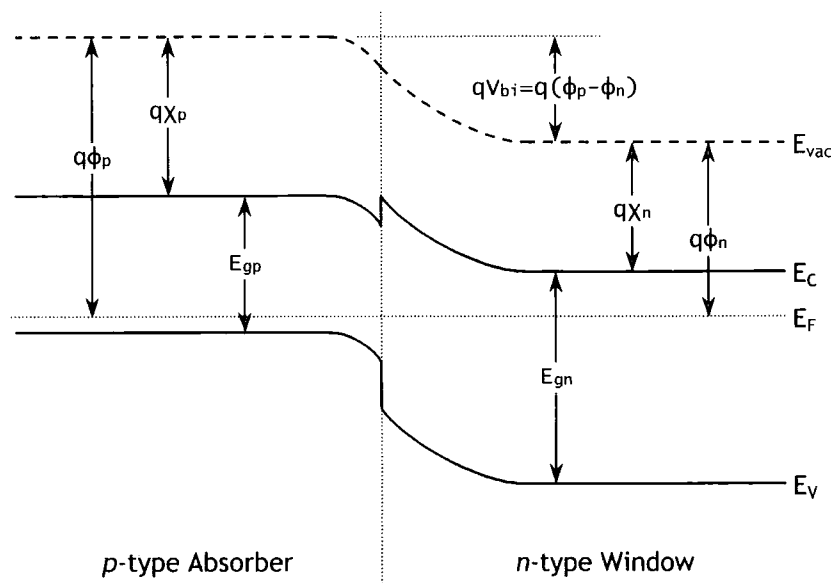


Figure 2.6: The Anderson model for the band structure of a heterojunction. The energy gap, electron affinity, and work function are marked with appropriate subscripts for *p*- and *n*-type material.

Placing two different materials in intimate contact forms a heterojunction. The primary purpose of such an arrangement is to avoid the problems associated with surface recombination discussed above in the context of homojunctions. This is achieved by making one of the materials of a very large band-gap, thus allowing most of the light to pass through it (a 'window' layer). The second layer is formed from a good solar absorber, which produces the carriers.

The band diagram for such a structure is complicated, however a first approximation is the Anderson model (described in Fahrenbruch and Aranovich [10]) shown in figure 2.6. A 'spike' in the conduction band can clearly be seen, and in systems where this arises, it will act as a barrier to the flow of carriers.

In this work, we are concerned with a heterojunction device. There are a number of other considerations with this type of device, such as the materials properties and how these affect the device characteristics. These issues will be addressed for the specific case of polycrystalline CdTe/CdS solar cells in chapter 4.

2.5 SOLAR CELL MATERIALS

In this section a brief review of current materials used in the production of solar cells is given. All efficiencies quoted are for cells, rather than modules, and are taken from Green *et al.* [11], except where stated.

2.5.1 SINGLE CRYSTAL MATERIAL

As mentioned in the introduction to this chapter, single crystal solar cells have the longest history of commercial exploitation and are the most widely used for small area applications. They have a relatively high efficiency (~24%) but the production of the silicon wafers, from which the cells are manufactured, is expensive with no real prospect of cheapening significantly. In addition to this, the size of an individual cell is limited to the maximum wafer size available.

Another material used for making single crystal solar cells is GaAs, with a current efficiency of 25.1%. Although more costly than Si, the good lattice matching of III-V materials has allowed the production of multi-junction cells with efficiencies in excess of 30%.

Both Si and GaAs will continue, no doubt, to be used for small-scale and space applications, but are ultimately unsuitable for large-scale terrestrial applications, due to their prohibitive cost.

2.5.2 MULTI-CRYSTALLINE MATERIAL

To reduce the cost of silicon solar cell production, multi-crystalline cells are increasingly popular. These devices are similar to the single-crystal counterpart, but are produced from slices cut from very large cast ingots with grain sizes of the order of millimetres. Whilst the reduction in materials quality causes a corresponding decrease in price, it is also deleterious to the performance; efficiencies are no higher than 20%, and the cells use a large amount of material compared to thin-film cells. These cells are, though, popular for medium scale installations, such as domestic use.

2.5.3 AMORPHOUS MATERIAL

Amorphous silicon solar cells now account for a large proportion of solar module production. In the hydrogenated form, it behaves as a semiconductor of band-gap 1.7 eV, with higher optical absorption than single crystal silicon. Thin film devices manufactured using this material have achieved efficiencies of up to 12.7%.

2.5.4 POLYCRYSTALLINE MATERIAL

This type of solar cell commonly uses cheap and simple deposition techniques on glass substrates, and use minimal thicknesses for their active layers (typically one to ten microns, but in some cases hundreds of microns). The efficiencies are also reasonable making them strong contenders for large-scale terrestrial applications.

The heterojunction devices that are of concern in this work, based on a CdTe absorber, have a record efficiency of 16.4% [12], and will be reviewed in detail in chapter 4. The main rival to this technology is that based on I-III-VI chalcopyrites. One member of this family of cells, using $\text{Cu}(\text{In}, \text{Ga})\text{Se}_2$, has a current highest efficiency of 18.8% [13]. In addition to being of a slightly

higher efficiency than their cadmium-based rivals, they have the added advantage of lower toxicity.

A further polycrystalline variant exists, based on silicon, albeit commonly referred to as 'microcrystalline' material. These have a similar efficiency to the above CdTe cells of ~16%.

2.6 REFERENCES

- [1] E. Becquerel *On Electric Effects under the Influence of Solar Radiation*, Comptes Rendus Hebdomadaires des Seances de l'Academie des Sciences **9** (1839) 561-67 (In French)
- [2] W. G. Adams and R. E. Day *The Action of Light on Selenium*, Proceedings of the Royal Society of London A **25** (1877) 113-17
- [3] D. M. Chapin, C. S. Fuller and G. L. Pearson *A New Silicon p-n Junction Photocell for Converting Solar Radiation into Electrical Power*, Journal of Applied Physics **25** (1954) 676-77
- [4] D. Bonnet and H. Rabenhorst *New Results on the Development of a Thin-Film p-CdTe n-CdS Heterojunction Solar Cell*, Proceedings of the 9th IEEE Photovoltaic Specialists Conference (1972) 129-32
- [5] S. M. Sze "Semiconductor Devices, Physics and Technology" (Wiley, New York, 1985)
- [6] L. L. Kazmerski *Introduction to Photovoltaics: Physics, Materials, and Technology* in "Solar Materials Science", Ed. L. E. Murr (Academic Press, New York, 1980)
- [7] G. Stollwerck and J. R. Sites *Analysis of CdTe Back-Contact Barriers*, Proceedings of the 13th European Photovoltaic Solar Energy Conference (1995) 2020-22
- [8] A. L. Fahrenbruch and R. H. Bube "Fundamentals of Solar Cells" (Academic Press, New York, 1983)
- [9] D. L. Pulfrey *MIS Solar Cells: A Review*, IEEE Transactions on Electron Devices **25** (1978) 1308-17
- [10] A. L. Fahrenbruch and J. Aranovich *Heterojunction Phenomena and Interfacial Defects in Photovoltaic Converters* in "Solar Energy Conversion - Solid-State Physics Aspects", Ed. B. O. Seraphin (Springer-Verlag, Berlin, 1979)
- [11] M. A. Green, K. Emery, K. Bucher and S. Igari *Solar Cell Efficiency Tables (Version 12)*, Progress in Photovoltaics **6** (1998) 265-70
- [12] National Renewable Energy Laboratory *Press Release No. 1501*, 24th April 2001
- [13] M. A. Contreras, B. Egaas, K. Ramanathan, J. Hiltner, A. Swartzlander, F. Hassoon and R. Noufi *Approaching 20% Efficiency in Cu(In,Ga)Se₂ Polycrystalline Thin-Film Solar Cells*, Progress in Photovoltaics **7** (1999) 311-16

Chapter 3

Aspects of Polycrystallinity

3.1 INTRODUCTION

Polycrystalline materials are considered a promising candidate for large-scale terrestrial photovoltaic applications; their economy of manufacture, and conservative use of materials, makes them more suitable than their single and multi-crystalline counterparts as discussed in the previous chapter. However, there are a number of issues specific to polycrystalline materials, which make their study more complicated than traditional cells. Such issues include how the grains and grain structure evolve during film growth, and how grain-boundary phenomena affect the performance of a device.

The purpose of this chapter is to elucidate these phenomena specific to polycrystalline materials, with cursory reference to CdTe where relevant: a detailed review of these effects in CdTe is presented in chapter 4.

3.2 STRUCTURAL CHANGES DURING GROWTH AND PROCESSING

There are three metallurgical phenomena commonly associated with the processing of polycrystalline materials. Although descriptions of these are usually found in discourses pertaining to metals, the behaviour is equally applicable to any crystalline solid [1]. Although described as consecutive processes, the three effects— recovery, recrystallisation, and grain-growth— can, and often do, overlap. The first two of these effects are discussed in this section. The topic of grain-growth, which is most important in this work, will be discussed in section 3.3.

3.2.1 RECOVERY

In both the growth of materials and, more commonly, their processing, large numbers of dislocations can be introduced into the material. A common example of this is the cold-working of metals, in which the original unstrained grains have progressively more defects introduced into them, making the material increasingly hard to work. Material with such a high density of defects is far from equilibrium, and in the case of electronic materials, far from device quality due to the recombination centres provided by the dislocations. In order to remove these defects, raising the temperature of the material (i.e. annealing) can supply enough energy to the defects to allow them to migrate. The process of recovery can then occur, as described below.

The recovery mechanism can be divided into a number of steps, shown as a two-dimensional analogue in figure 3.1. It should be noted, however that the stages will occur, in all probability, with some degree of simultaneity in a real system.

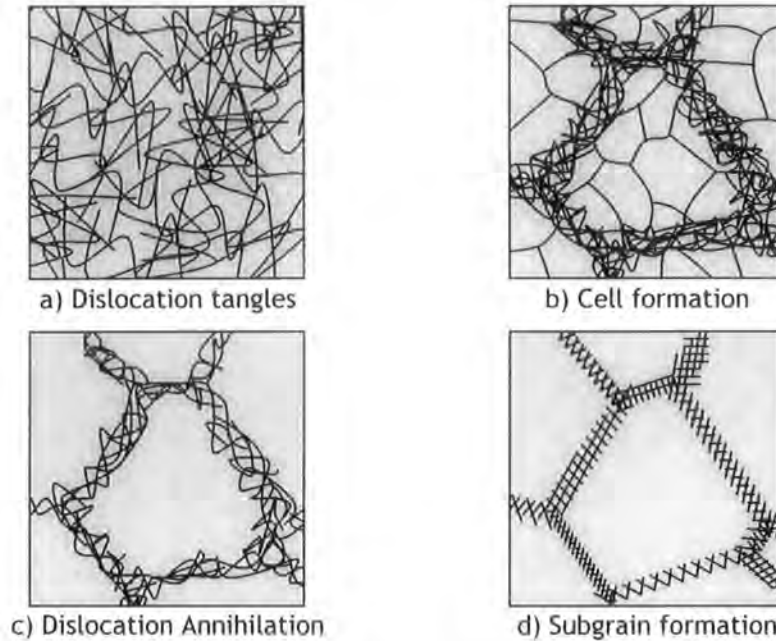


Figure 3.1: Conceptual progression of the recovery process within a single grain. The stages a) to d) are discussed in the text. (After Humphreys & Hatherley [1]).

Figure 3.1a shows a material with a large number of dislocations. On annealing, these defects will begin to migrate by climb, glide, or cross-slip [2], forming cells and defect complexes due to the attraction between defects (figure 3.1b). Subsequent annihilation of defects (the mechanisms for which are described by Li [3]) causes the creation of a cell structure (figure 3.1c) and, as the process progresses, formation of sub-grain boundaries (figure 3.1d), also called ‘polygonisation walls’. Following the formation of these structured boundaries, the sub-grains can grow in a fashion analogous to the grain-growth described in section 3.3 (‘sub-grain coarsening’).

3.2.2 RECRYSTALLISATION

The final structure described above is often not in a state of thermodynamic equilibrium. A further process that can occur to reduce the stored energy is recrystallisation. In this process, new grains nucleate at the

interstices of the old structure. As these grow, the old structure is consumed, removing any dislocations or dislocation complexes, such as sub-grain boundaries.

The energy needed to drive recrystallisation is conventionally considered to be produced by strain energy, either from planar defects such as grain boundaries, or epitaxial mismatch. If, for example, the as-grown grain size is very small, the strain energy per unit volume arising from this structure may be sufficient to cause recrystallisation on annealing. Not only this, but a small grain size also encourages recrystallisation by providing a higher density of nucleation sites (i.e. grain boundaries).

It is worth noting at this point that recovery and recrystallisation are in fact competing processes. As more of the defects are re-arranged by recovery, there will be a reduced driving force for recrystallisation. Conversely, any recrystallised structure will have no need of recovery. Thus, it is usual to find the two processes occurring concurrently, rather than consecutively, as portrayed here.

3.3 GRAIN GROWTH

In a recrystallised structure (and often before full recrystallisation has occurred), the structure of the polycrystalline material can still be far from equilibrium, as the final grain size after recrystallisation is dictated primarily by the density of nucleation sites for this process. This encourages expansion of the grains to minimise the grain-boundary energy to volume ratio, and thus the stored energy in the material. Factors affecting this process for a single-phase material are discussed below.

3.3.1 STABLE POLYCRYSTALLINE STRUCTURES

The process of grain-growth can be viewed as occurring by migration of grain boundaries, where the pressure driving this migration is a function of the radius of curvature of the boundary. Thus, we can see that a structure with purely straight boundaries would experience no grain-growth. It can also be shown, however, that there are strict conditions for a vertex where boundaries meet to be stable: in two dimensions, the vertex must consist of three arms with an angle of 120° between them; in three dimensions, four rays are needed, set at the tetrahedral angle ($109^\circ 28' 16''$). In order to satisfy both the need for straight boundaries, and the angular constraints, few tessellating structures are permitted. In two dimensions, an array of equal-sized hexagons (a 'honeycomb' structure) will fulfil these requirements. However, in three dimensions there only exist approximations:

- a) Truncated octahedra are an example of a body with plane faces, but the angles at the vertices are incorrect, and;
- b) Kelvin tetrakaidecahedra [4] have the correct angles, but require curved faces.

Thus, even in a three dimensional, ideal and directionally isotropic system, there is a driving force for grain growth, and grain growth must occur.

In real systems, however, these nearly ideal structures are rarely reached. Various contributing factors will contribute to the premature cessation of grain growth, or indeed the absence of grain growth altogether. These issues are discussed below in the context of deviations from ideal grain-growth kinetics.

3.3.2 GRAIN-GROWTH DYNAMICS

There are two distinct modes of grain-growth: normal grain-growth, in which the shape of the grain size distribution is time-invariant; and, abnormal

grain-growth in which local textural differences cause the grains in one region to grow faster than in another. These two modes of growth are described by the distribution curves in figure 3.2. As shown in the figure, abnormal grain growth continues only until the larger, abnormal grains have consumed the smaller ones: normal grain-growth then predominates. Abnormal grain growth is highly dependent on initial material conditions, making the behaviour difficult to analyse. Since no abnormal grain-growth was seen in this work, the reader is referred to Gladman [5] for further information.

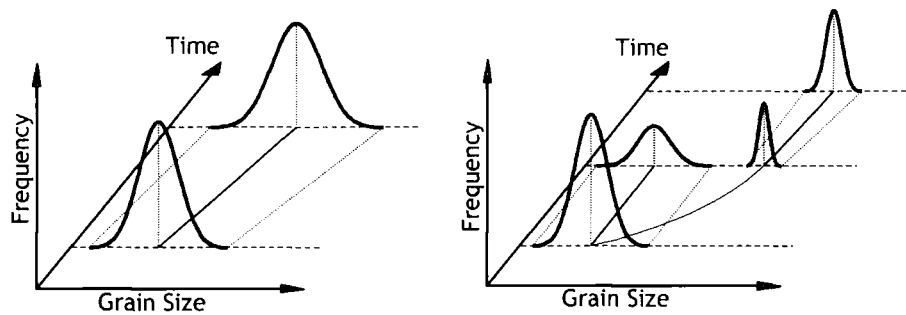


Figure 3.2: Grain size development with time showing normal grain-growth (left) and abnormal grain-growth (right). In the case of abnormal growth, the new population can be seen emerging from the old, and then commencing normal growth (after Detert [6]).

The dynamics of normal grain-growth have been extensively studied. In measuring rates of growth, reference is still made to the *grain-growth exponent*, a product of the seminal work by Burke and Turnbull [7]. In their work, they considered an isolated section of grain boundary, and the forces acting upon it. Defining this force as arising from a boundary pressure, P , for a grain-boundary with principal radii of curvature r_1 and r_2 as,

$$P = \gamma \left(\frac{1}{r_1} + \frac{1}{r_2} \right), \quad (3.1)$$

for a grain boundary energy γ , and relating this to the boundary velocity, v , using a boundary mobility μ ,

$$v = \mu P, \quad (3.2)$$

they then made a number of assumptions:

- a) γ is a constant for all grains, grain sizes and times;
- b) For a spherical grain (i.e. $r_1 = r_2 = r$) the curvature is proportional to the grain radius, R , hence $r = c_1 R$;
- c) μ is independent of R , giving $dR/dt = c_2 P$, and most importantly;
- d) The only forces acting on the boundary arise from the curvature, and thus pressure.

This gives:

$$\frac{dR}{dt} = \frac{2c_2\gamma}{c_1R}. \quad (3.3)$$

Solution of this equation and substitution of the mean value $\bar{R}$ for R , gives,

$$\bar{R}^2 - \bar{R}_0^2 = c_3 t, \quad (3.4)$$

where c_3 is a constant, R has been substituted by the general value $\bar{R}$ and $\bar{R}_0$ is derived from the constant of integration. Thus, simply,

$$\sqrt{\bar{R}^2 - \bar{R}_0^2} = Kt^{1/n}, \quad (3.5)$$

where n is the grain-growth exponent*, here equal to 2, and K a constant. This value for n gives rise to the common epithet of *parabolic* grain-growth kinetics.

A large number of studies have compared measured grain growth kinetics to the case where $n = 2$: the comparison is rarely good, however, with n ranging

* In some work, the grain-growth exponent is alternatively defined as the reciprocal of n in equation 3.5. As n is always greater than unity, though, confusion should not arise.

from about 1 to 5. A review of studies for various metals is given by Detert [8]. In the case of CdTe in CdTe/CdS solar cells, Qi *et al.* [9] also found poor agreement with parabolic behaviour, described in section 4.5.3. The deviation from the expected value of n can be attributed to a number of factors, reviewed by Atkinson [4]. The following, however, are those of most interest here:

- a) Non-isotropy of the grain structure: many thin films of material exhibit columnar grain growth, rather than the classic 'interlocked polygon' structure. CdTe is one of these materials, being well known to exhibit columnar growth. (Incidentally, this is thought to aid conduction perpendicular to the film direction).
- b) Solute pinning: the presence of a solute in the material (such as the CdCl₂ used for the treatment of CdTe) can prevent the grain boundaries from migrating, although if the solute is also a flux it can be instrumental in aiding recrystallisation and grain-growth.
- c) Substrate pinning: if grown on a dissimilar but polycrystalline substrate (such as the CdS upon which CdTe is grown), the grain boundaries in the substrate can act to pin those boundaries trying to migrate.

In addition to the above physical reasons for experiment not fitting the Burke and Turnbull model, one could also question the assumptions of the model itself. A considerable amount of effort has been expended on refining models to describe grain-growth dynamics, often concentrating on mean-field theories to avoid the unphysical assumption of the Burke and Turnbull model of a single, isolated boundary. Some of the more successful of these are reviewed by Louat *et al.* [10], used later in this work to give predicted distributions of grain sizes (section 7.4.2).

Overall, it is clear that the polycrystalline nature of CdTe films has a potentially great effect on the growth and processing of CdTe/CdS solar cells. However, observing and interpreting these polycrystalline phenomena in terms of the mechanisms described in this chapter is not trivial, and has been the subject of considerable study. The following chapter will review this body of work, in addition to a general overview of the current thinking in the field of thin-film polycrystalline CdTe/CdS solar cells.

3.4 REFERENCES

- [1] F. J. Humphreys and M. Hatherley "Recrystallization and Related Annealing Phenomena" (Pergamon, Oxford, 1995)
- [2] D. Hull and D. J. Bacon "Introduction to Dislocations- 3rd Edition" (Pergamon, Oxford, 1984)
- [3] J. C. M. Li *Recovery Processes in Metals*, Proceedings of the American Society for Metals Conference (1966) 45-98
- [4] H. V. Atkinson *Theories of Normal Grain-Growth in Pure Single-Phase Systems*, Acta Metallurgica **36** (1988) 469-91
- [5] T. Gladman *Abnormal Grain Growth During the Heat Treatment of Steel* in "Grain Growth in Polycrystalline Materials", Eds. G. Abbruzzese and P. Brozzo (Trans Tech Publications, Zurich, 1992)
- [6] K. Detert in "Recrystallisation of Metallic Materials", Ed. F. Haessner (Springer-Verlag, Berlin, 1978)
- [7] J. E. Burke and D. Turnbull *Recrystallization and Grain Growth*, Progress in Metallurgical Physics **3** (1952) 220-92
- [8] M. P. Anderson, D. J. Srolovitz, G. S. Grest and P. S. Sahni *Computer Simulation of Grain-Growth 1- Kinetics*, Acta Metallica **32** (1984) 783-91
- [9] B. Qi, D. W. Kim, D. L. Williamson and J. U. Trefny *Effects of Postdeposition Heat-Treatment on Morphology and Microstructure of CdTe Grown by Electrodeposition*, Journal of the Electrochemical Society **143** (1996) 517-23
- [10] N. P. Louat, M. S. Duesbery and K. Sadananda *On the Role of Random Walk in Grain Growth* in "Grain Growth in Polycrystalline Materials", Eds. G. Abbruzzese and P. Brozzo (Trans Tech Publications, Zurich, 1992)

Chapter 4

CdTe/CdS Thin-Film Solar Cells

4.1 INTRODUCTION

The suitability of CdTe to act as an absorber for solar energy conversion has long been recognised; as far back as 1956, Loferski [1] showed that its band-gap of 1.45 eV was close to the theoretical ideal. Some years afterwards, the production of CdTe homojunction solar cells was motivated by the ability to produce both *p*- and *n*-type CdTe. However, efficiencies in *p-n* junction cells were found to be limited to ~6% by the very high surface recombination velocity of CdTe [2]. Research into CdTe homojunction did continue for some years, however, culminating in 1982 with a 10.7% efficient *p⁺-n-n⁺* cell [3].

In order to overcome the problem of surface recombination, attention turned to heterojunction cells. Initial efforts concentrated on using a *p*-type window layer, as it was appreciated that forming an ohmic contact to *p*-CdTe was difficult. Some initial success was attained with a *p*-CdTe/*n*-Cu_{2-x}Te

system [4], but long-term stability proved to be problematic. The feasibility of metal/*n*-CdTe Schottky cells [5] was also investigated, but efficiencies were limited. Both cases, though, showed that thin-film technology based on CdTe was viable.

As candidates for forming *p*-type window layers were exhausted, attention turned to the *n*-window/*p*-CdTe structure, which interests us here. CdS was always considered a likely candidate, as in forming a *p-n* junction with CdTe there is no 'spike' in the conduction band [6] (see section 2.4.3). CdS also has a large band-gap of 2.42 eV, and high optical transparency. Adirovich *et al.* produced the first cell based upon this technology [7], albeit with 1% efficiency. However, this first cell used a 'superstrate' configuration (see following section), which remains the construction geometry of choice.

In the 28 years since the production of the first 'efficient' *p*-CdTe/*n*-CdS solar cells (for example Bonnet and Rabenhorst [8]), the efficiency of these cells has increased from 5.4 to 16.0% [9]. With a theoretical maximum efficiency of over 29% [10], though, it is quite possible that considerable improvements can be made to practical device performance.

4.2 CELL STRUCTURE, COMPOSITION AND CONSTRUCTION

Since their inception, CdTe/CdS solar cells have been considered an important topic for research due to their relatively high efficiency, their economy of construction, and tolerance to manufacturing conditions. In this section, the functions of each of the layers in the 'superstrate' design of thin film heterojunction solar cells, shown in figure 4.1, is described. In section 4.2.2, the methods needed to form the CdTe absorber layers are

reviewed as these have a direct influence of the properties of material prepared.

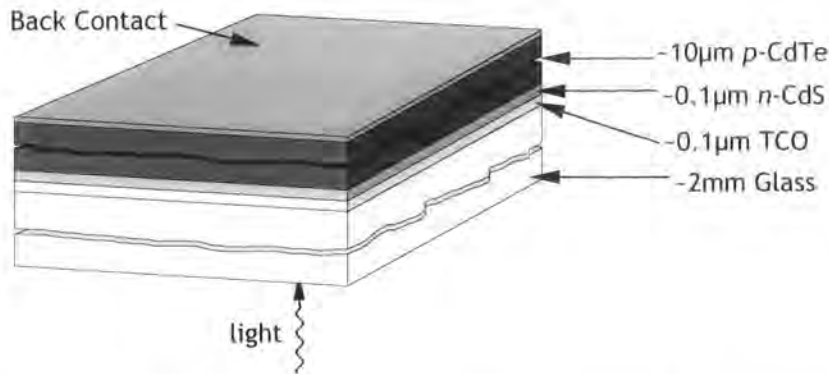


Figure 4.1: Schematic cross-section of the superstrate configuration of a CdTe/CdS solar cell.

4.2.1 THE LAYERS AND THEIR FUNCTIONS

The Substrate: This is made from a ~2mm thick layer of glass, as it is transparent, durable and cheap. Although simple soda-glass can be used, more often a low-alkali glass is used to minimise CdS acceptor species that could diffuse from the glass (for example group I ions). Corning #7059 borosilicate glass is a popular choice. Some research groups also apply anti-reflective coatings, e.g. MgF, to the glass.

The Front Contact Material: The main criteria for this layer are transparency and high conductivity, and SnO₂ is used in this capacity in many applications. Its conductivity can be enhanced further by doping with fluorine. Indium tin oxide (In₂O₃-SnO₂ or ITO) is also popular, having superior conductivity. However, indium diffusing into the active layers has been suggested to be deleterious to device performance. Thus, it is common to find the ITO covered with a layer of SnO₂ as a diffusion barrier. The thickness of the layer is chosen to provide a balance between minimising

optical losses, whilst maximising the conductivity. A value of $10 \Omega/\square$ is usually quoted as sufficient for a high efficiency cell, requiring a layer of thickness ~ 100 nm.

The Window Layer: The n -CdS window layer is usually ~ 100 nm thick, being as thin as possible to allow maximum optical transmission. This minimum thickness is dictated by the amount of the layer consumed by intermixing of the CdTe and CdS layers on treatment with CdCl_2 (see section 4.4) although the complete conversion of the CdS layer to $\text{CdS}_{1-x}\text{Te}_x$ has been shown to be deleterious to device performance [11]. The layer can be deposited by all of the methods detailed in the following section, and also by chemical bath deposition. However, the thick films produced by spray pyrolysis and screen-printing are unsuitable for high efficiency cells. For material deposited at low temperature, investigations have been made of post-growth annealing of the CdS with CdCl_2 or air before the CdTe layer is deposited. This serves to increase grain size, and reduce defect density [12].

The Absorber Layer: As mentioned previously, CdTe has a band-gap very well matched to the absorption of the solar spectrum. However, it also suffers very high surface recombination velocities, meaning that it is suitable only for use in a heterojunction cell. It is an impressive absorber in the visible range, $1 \mu\text{m}$ absorbing 99% of incident radiation. It is, though, common to make the layers considerably thicker than this as homogeneity can be a problem for thinner layers. This being the layer of interest in this work, different methods of deposition will be discussed in the following section. The microstructures which result will also be examined.

The Back Contact: Due to the very large work function of p -CdTe, there is no metal with a work function large enough to form an ohmic contact to it. Instead, making a p^+ layer at the CdTe surface produces a pseudo-

ohmic contact. This produces a narrower depletion region upon contacting, through which the carriers can more easily tunnel. There are two distinct methods used to accomplish this:

- a) A layer of conductive *p*-type semiconductor with a lower work function than CdTe can be deposited on the CdTe. ZnTe [13], HgTe [14], and Sb₂Te₃ [15] have been used with some success.
- b) A *p*-type layer can be produced on the CdTe surface by either chemical etching or doping [16].

Commonly used etchants for the latter process are nitric/phosphoric acid [17], bromine/methanol [18], or acidified dichromate [19]. Doping can be achieved by coating the CdTe with a graphite paste containing the required dopant. Subsequent annealing allows the dopant to diffuse into the CdTe. After the etching or doping procedure, a variety of metals including Au, Ni/Au, or Cu/Au can be used to supply the electrical contact.

4.2.2 DEPOSITION TECHNIQUES

This section describes the most commonly used techniques for depositing thin films of CdTe.

Close Space Sublimation: CSS is a widely used technique for depositing both CdTe and CdS. The deposition is carried out in an inert atmosphere, often of helium, at a pressure of ~10 millibar. The inclusion of a small amount of oxygen is reported to enhance the efficiency of the final cell [20]. A source, consisting of the material to be deposited, is maintained at a high temperature (e.g. ~650°C) a few millimetres from the substrate, which is at a lower temperature (e.g. ~550°C). The source material then dissociates, and then recombines at the substrate. A major advantage of this method is the rapid deposition rate (~1 µm/minute), and large grain size, which is widely

thought to be beneficial to solar cell efficiency due to decreased grain boundary density. Typical film thicknesses deposited by this method are $\sim 10\ \mu\text{m}$. All CdTe layers studied in this work are deposited using this method. Close Space Vapour Transport (CSV T) is usually synonymous with CSS although some authors [21] distinguish the two methods by the presence or absence of a gaseous ambient in the growth chamber (CSV T being the high vacuum variant). As the terms CSV T and CSS are often used interchangeably, readers are advised to check the deposition conditions used by each author.

Physical Vapour Deposition: PVD, or simply 'vacuum evaporation', is the process of heating a solid in vacuum, thus creating a vapour. This vapour then condenses onto a substrate, usually located some centimetres from the source. Substrate temperatures of between room temperature and $\sim 400^\circ\text{C}$, are reported, but this is considerably cooler than the source. Deposition rates attainable by this method are of the order of $0.3\ \mu\text{m}/\text{minute}$. Absorber layers laid down by PVD are usually $\sim 5\ \mu\text{m}$ thick with a grain size which is small compared to that of CSS material.

Electrochemical Bath Deposition: This technique, also referred to as simply 'Electrodeposition' (ED), was first demonstrated for use in CdTe/CdS solar cells by Panicker *et al.* [22]. The aqueous electrolyte used contains Cd^{2+} and HTeO_2^+ , and the net reaction can be described by the following two steps [23]:



The process gives a deposition rate of only $\sim 0.02\ \mu\text{m}/\text{minute}$, however the films are of high quality with very few pin-holes, allowing very thin ($\sim 1.5\ \mu\text{m}$)

layers to be used. Resulting grain sizes can be of the order of tens of nanometres.

Screen-Printing: This is a reasonably popular technique, as it requires no in-vacuo processing. Propylene glycol slurries of semiconductor powder are forced through a fine mesh onto the substrate, which is then dried slowly to prevent cracking. The resulting film is then sintered at 600-700°C to densify the layer. This method has the advantages that CdCl₂ can be introduced at the screen printing stage, thus removing the need for a post-deposition CdCl₂ treatment. (For example, see Clemminck *et al.* [24].)

Chemical Vapour Deposition: CVD is one of the least common deposition techniques for solar cell applications. Organic compounds of Cd and Te, such as dimethylcadmium and diisopropyltellurium are flowed over a hot (~350°C) substrate, depositing a layer of CdTe. Growth rates of ~1.5 µm/minute are typical. For an example of this type of growth see Brinkman *et al.* [25].

Spray Pyrolysis: This method of deposition is similar to CVD; organic compounds of Cd and Te are sprayed onto a heated substrate (~450°C), where the precursors decompose to leave Cd and Te [26].

Radio-Frequency Sputtering: RFS involves the sputtering of a CdTe or CdS target with an RF-plasma. The sputtered material is then incident on a substrate that may be heated, although typically at a temperature much lower than PVD or CSS: Moutinho *et al.* report a growth temperature of 150°C [27].

4.3 MICROSTRUCTURE OF THE UNTREATED CdTe LAYER

The diversity of techniques used to deposit the layers of CdTe in CdTe/CdS thin-film solar cells leads to a range of microstructural phenomena in as-deposited layers. However, most effects are consistently present, irrespective of layer deposition technique, but with their prevalence dependent on the cell's thermal history. In this section the microstructural character of CdTe layers is reviewed according to deposition technique. A summary is also presented in table 4.I, at the end of this section.

4.3.1 PHYSICAL VAPOUR DEPOSITION

Perhaps the most striking microstructural feature of vapour deposited CdTe thin-films is the columnar grain structure, as revealed by cross-section TEM [23]: the grains also contain a high density of planar defects [28], these being reported as stacking faults or twins. Certainly both types are prevalent in CdTe of all types [29].

The relationship between the morphology and microstructure of the CdTe layer and the underlying CdS is not simple. The CdTe is nucleated on the CdS grains (i.e. grain boundaries propagate across the interface), and HRTEM reveals them to be epitaxially related to the substrate grains [30]. As the grain structure is predominantly columnar, the CdS grain structure has a strong influence on the surface structure of the CdTe films. During growth, some columns widen at the expense of others leading to the observation that thick PVD CdTe films have a slightly larger grain size than thinner ones at the film surface [28]. There is, however, some evidence for a 'limiting' size for this lateral expansion. Levi *et al.* [31], for example, report the same surface grain

size for CdTe grown on both CdS and ITO, even though the relative grain size of CdS is generally larger than that of ITO.

In recent work, however, Romeo *et al.* [32] report that CdS layers with similar surface feature size, but deposited by different methods, can cause an increase (of up to ten times) in the grain size of the overlying CdTe film. In addition, x-ray diffraction data shows that the different films have considerably different lattice parameters, and the CdTe surface morphology is different. These discrepancies are explained in terms of strain. Compressive strain in as-deposited PVD films has been reported before [28], caused by either the difference in the lattice parameters of CdTe and CdS, differences in thermal expansion coefficients, or a combination of both.

Films of CdTe deposited by PVD often display {111} preferred orientation in the as-deposited state [33, 34]. This ordering is shown to decrease with increasing substrate temperature [32, 35], possibly due to the higher surface mobility of arriving species. Preferred orientation can be quantified using the method of Harris [36], which, in essence, compares the heights of XRD peaks from a sample of material to those of a powder sample of the same material. This yields a single variable to describe the degree of preferentiality.

4.3.2 CLOSE SPACE SUBLIMATION

CSS may be thought of as a relatively high temperature deposition method with typical substrate temperature being $\sim 500^{\circ}\text{C}$. Growth (and processing) at this temperature can lead to large grain sizes which increase further with increasing substrate temperature [37], and film thickness [38]. Al-Jassim *et al.* [37] have also studied the near-interface region by cross-

sectional TEM, and found a heavily twinned structure with the density of the planar defects decreasing with increasing temperature.

The grains in CSS CdTe are often 1-5 μm in size, and examination by SEM generally reveals a faceted morphology [30]. It could be argued that these films have a columnar grain structure (as described for PVD films in section 4.3.1), but as the grain size is often of the order of the film thickness, it is not necessarily a meaningful concept.

The degree of preferred orientation present in CSS-grown CdTe films is dependent upon both the substrate type and temperature. For glass substrates, Zelaya *et al.* [39] report that at deposition temperatures between 450 and 600°C a strong $\{111\}$ preferred orientation prevailed. For temperatures between 600 and 650°C, this changed to $\{220\}$. In the more technologically relevant case of CdTe layers on CdS grown at a substrate temperature of 500°C, the orientation is random [33]. Only by reducing the growth temperature to 335°C could $\{111\}$ preferred orientation be induced [38].

The effect on the microstructure of CdTe layers of including a partial pressure of O_2 in the CSS chamber during growth has been studied by Rose *et al.*. In that work, it was found that increasing the partial pressure of oxygen had the effect of increasing the density of nucleation sites on CdS window layers. This was beneficial, as it reduced the number of pinholes in the films. However, it also caused a decrease in grain size, accompanied by an apparent decrease in crystal quality as indicated by a less faceted appearance on each grain [40].

4.3.3 ELECTRODEPOSITION

This method of CdTe film deposition, in common with PVD growth, usually shows very strong {111} orientation [41], with the crystallite size being reduced by increased deposition potential [42]. Grain size is generally reported as $\sim 0.1 \mu\text{m}$ or lower. XRD shows that the CdTe films are also in a state of compression [43].

The most detailed microstructural survey of electrodeposited CdTe so far was recently published by Johnson [44]. For the as-deposited state, a columnar structure similar to that of PVD grown material is reported, with grain diameter being $\sim 0.1 \mu\text{m}$ and each grain containing a high density of stacking faults.

4.3.4 RADIO FREQUENCY SPUTTERING

This produces small-grained material, comparable to that produced by electrodeposition. However, AFM studies have shown that these $\sim 0.1 \mu\text{m}$ grains are comprised of sub-grains of diameter $\sim 50 \text{ nm}$ [45]. The films are also considerably more porous than other techniques discussed thus far [33].

4.3.5 CHEMICAL VAPOUR DEPOSITION

There is very limited information on the as-grown properties of CVD grown films. Chu *et al.* [46], however, report a closely packed columnar structure of small-grained material with {111} orientation. The material was deposited with a substrate temperature of $350\text{--}400^\circ\text{C}$. The authors do not state a grain size, although it is most likely small ($\sim 0.1 \mu\text{m}$).

4.3.6 SCREEN PRINTING AND SPRAY PYROLYSIS

Due to the practice of including CdCl_2 in the initial CdTe deposition, the resulting in-situ chlorine anneal precludes any study of an as-grown state. The reader is referred to section 4.5 for a description of the films.

Growth Method	Grain Size (μm)	Notes
PVD	~0.1	{111} orientation, columnar
CSS	1-10	Columnar
ED	~0.1	{111} orientation
RFS	~0.1	Contain sub-grains
CVD	—	{111} orientation, columnar

Table 4.1: Microstructure of as-grown CdTe/CdS thin films. See text for details. It should be noted that grain size is heavily dependent on film thickness.

4.4 POST-DEPOSITION TREATMENT OF CdTe/CdS STRUCTURES

In their as-grown state, CdTe/CdS solar cells have a typical efficiency of 1-2%. However, CdCl_2 is known to be responsible for a large increase in efficiency. A chloride treatment has now become a ubiquitous stage in the construction of finished CdTe/CdS cells. The chlorine can be introduced to the device in a number of ways:

- The CdCl_2 can form part of the CdTe deposition step, and thus post-growth annealing effects chloride treatment of the CdTe directly.
- The CdCl_2 can be deposited onto the CdTe at a post-deposition stage (usually by evaporation or dipping in a methanolic solution); this is followed by air annealing, typically at 400°C for 30 minutes.
- The cell can be annealed as above in the presence of CdCl_2 vapour, Cl_2 gas, or even gaseous HCl .

As with the layer deposition, all of these techniques appear to give an equivalent improvement in the cell's performance, often of about an order of magnitude (for example, see Edwards *et al.* [47]). Oxygen, whether introduced as part of the deposition process, or from the annealing atmosphere, is also thought to play an important part in this process, with slightly higher efficiencies achieved with oxygen present than in its absence [20].

The increase in efficiency from the CdCl_2 treatment is usually attributed to a number of effects, which may act singly or in combination. These are discussed below.

Type Conversion: In their as-deposited state, CdTe/CdS cells are inefficient in their action as solar cells, and it is widely reported that the CdCl_2 treatment enhances performance by introducing *p*-type dopant centres into the CdTe. This is generally a 'type conversion' as as-grown CdTe is usually *n*-type or intrinsic and thus resistive. This is consistent with the tendency of CdTe to self-compensate, making it difficult to dope *p*-type. Duffy *et al.* [48] have demonstrated that the chloride treatment does indeed effect the formation of *p*-CdTe. Although it has not been identified explicitly, the acceptor state which causes this type conversion is thought to be the $(\text{V}_{\text{Cd}} + \text{Cl}_{\text{Te}})^+$ complex [49]. The role of oxygen has also been speculated upon, possibly with it contributing to doping by acting as a sink for Cd atoms. The oxygen itself is also thought to modify the doping density by forming a shallow acceptor [50].

Passivation: EBIC measurements [51] have shown that the grain boundaries become a less effective site for carrier recombination after the CdCl_2 anneal. This is thought to be a result of upward band bending in the CdTe near to the grain boundaries after treatment, which leads to the repulsion of carriers [52]. The doping mechanism by which this grain

boundary passivation occurs is unclear. Cahen and Noufi, though, suggest that oxygen may play a role [53].

Interdiffusion: A further effect of the CdCl_2 treatment is to cause interdiffusion of the CdTe and CdS layers [54]. The amount to which this occurs is affected by a number of factors, including the CdS grain size [55], and the thickness of the CdS [24]. This intermixing is reported to have a number of effects:

- a) Data for CdTe–CdS composition from various authors is synthesised by Jensen *et al.* [56], showing considerable bowing of the band-gap with composition. Thus, some compositions of $\text{CdS}_{1-x}\text{Te}_x$ have a lower band-gap than CdTe, and will as such cause an increase in the optical absorption in the interface region.
- b) The in-plane stresses caused by lattice mismatch are relieved, giving improved V_{oc} [57].
- c) If the CdS layer is entirely consumed, light can be absorbed within the window layer, degrading performance [58].

This area of research is still most active.

Grain Growth: As grain growth and other microstructural phenomena form a very important part of this work, they will be discussed in detail in the following section.

4.5 MICROSTRUCTURAL EFFECTS OF THE CdCl_2 TREATMENT

The work in this thesis is primarily concerned with microstructural phenomena in CdTe/CdS solar cells. Thus, this section will review the effects of the post-growth treatment on CdTe layers in terms of the classical

metallurgical concepts of recovery, recrystallisation, and grain growth described in chapter 3.

4.5.1 RECOVERY

By its very nature, recovery is a difficult process to observe in real structures: to map the progress of defects in a particular grain during annealing is not practical. For example, monitoring recovery of layers by TEM, before and after treatment, is complicated by the fact that the density of defects in each as-grown grain is a variable. Moutinho *et al.* [59] do claim, though, that the strain relief detected at the interface by XRD is due to ‘a process of recovery’. However, that work did not attempt to investigate directly the processes of recovery at the level of dislocations. Dislocation re-arrangement has been seen in *bulk* CdTe, however, at temperatures of $\sim 500^{\circ}\text{C}$ [60], the usual growth temperature of CSS CdTe. Experimental evidence of recovery is presented in chapter 8 of this thesis.

4.5.2 RECRYSTALLISATION

In comparison with the above case, evidence for recrystallisation in CdTe on treatment with CdCl_2 is widespread, specifically for small-grained material. Those films that grow in a $\{111\}$ orientation (the majority) invariably become random on annealing in the presence of chlorine (this occurs, for example, in PVD material [28], ED material [41] and RFS material [33]). This is widely thought to be due to the formation of a new grain structure by recrystallisation, and accompanying consumption of the old microstructure. The driving forces for this type of process are the overall compressive strain in the film and strain in individual grains. The latter arises from dislocations and grain boundaries within the grains.

Direct evidence of the propensity for as-grown ED CdTe layers to recrystallise was reported by Johnson [44]. In that work such layers were seen to be of poor crystallinity and more unstable in the electron beam. Air annealing, however, produced material that was stable under the same conditions.

In real structures, direct evidence for recrystallisation and its mechanism was reported by Moutinho *et al.* [59]. In that work two different techniques were used to study the recrystallisation of PVD CdTe layers, namely XRD and AFM. In the XRD study, the Bragg-Brentano (θ - 2θ) geometry was used to investigate reflections from (h , k , l) planes with particular values of d_{hkl} perpendicular to the film, for both CdCl₂ treated and untreated material. The untreated material showed only the {111}, {222} and {333} peaks, indicating that the film had preferred orientation. On plotting the d_{hkl} values for these peaks on a Nelson-Riley plot [61], the lattice parameter perpendicular to the film was found to be larger than that for a CdTe powder standard. This indicates in-plane compressive strain. The same analysis was also performed on a CdCl₂ treated CdTe film. This showed a random film orientation (i.e. all sphalerite peaks were present), and a lattice parameter comparable to that for CdTe powder, indicative of a fully relaxed film. From these observations, it can be inferred that on CdCl₂ treatment the stressed, untreated film was restructured, presumably by recrystallisation, to form this new, relaxed microstructure.

Further to the XRD, Moutinho *et al.* also studied the surface of the CdTe film at different stages of CdCl₂ treatment by AFM. This was expedited by reducing the treatment temperature slightly to slow the process of recrystallisation. AFM images of the CdTe surface treated at 350°C revealed the presence of new recrystallised grains growing at the interstices of the old

ones. Presumably, this process would continue until the old microstructure was completely consumed, giving a restructured film.

4.5.3 GRAIN GROWTH

A consistent feature of the CdCl_2 treatment of small-grained CdTe films is grain growth, presumably post-recrystallisation. Larger columnar grains grow in the plane of the film at the expense of smaller ones, thus increasing the average size. This process is driven by the minimisation of grain boundary area, and corresponding reduction in grain boundary energy. Qi *et al.* [43] have quantified this grain growth in terms of the Burke and Turnbull model [62], (described in section 3.3.2) in which the evolution of mean grain size with time, t , follows a power law:

$$\sqrt{\bar{R}^2 - \bar{R}_0^2} = Kt^{1/n}. \quad (4.3)$$

The grain growth exponent, n , is ideally 2 for normal grain growth. In the work of Qi *et al.*, the grain growth exponent was found to be more than 2 in all cases for ED CdTe. A number of reasons were given for this:

- a) Preferred orientation, which has been shown to reduce growth rates;
- b) Columnar grains, as normal grain growth cannot be expected in a columnar structure, and,
- c) Residual stress, as normal grain growth is driven by grain boundary energy

The last case, of course, is dependent upon whether the CdTe has fully recrystallised before grain growth starts. Two other factors that may also reduce the growth rate and thus the grain growth exponent are the existence of a limiting grain size, and solute drag at the boundaries (i.e. the boundaries are pinned [63]). The former is almost certainly the case, as even prolonged CdCl_2 treatment does not lead grain sizes greater than $\sim 10 \mu\text{m}$. How much

solute drag growing grains experience will depend on the CdCl_2 concentration at the grain boundaries, and its solubility in CdTe.

At the time this work was begun, there were contradictory reports of grain-growth in CdTe upon CdCl_2 treatment (for example, grain growth had been seen by some authors [28, 43, 64], but not by others [30, 31]). Moreover, there is some evidence that effects may be non-uniform throughout a particular film: while no growth is reported at CSS film surfaces, Loginov *et al.* [65] reported preliminary observations of grain growth at the CdTe/CdS interface on CdCl_2 treatment of CSS CdTe. In the current work, the grain size distribution throughout such layers was fully investigated (chapters 7 and 8).

Recently, Moutinho *et al.* [33] correlated grain growth effects to grain size with small-grained material displaying grain-growth irrespective of the deposition technique. Large-grained material showed little or no grain growth. This is consistent with surface area to volume ratio of grains being the primary driving force for grain growth. Further discussion is deferred until section 7.4 in which the results of grain size analysis in this work are presented.

4.6 REFERENCES

- [1] J. J. Loferski *Theoretical Considerations Governing the Choice of the Optimum Semiconductor for Photovoltaic Solar Energy Conversion*, Journal of Applied Physics **27** (1956) 777-84
- [2] K. Zanio "Cadmium Telluride" (Academic Press, New York, 1978)
- [3] M. Barbe, F. Bailly, D. Lincot and G. Cohen-Solal *13% Conversion Efficiency in Cadmium Telluride Shallow Homojunction Solar Cell. Effects of Temperature and Intensity*, Proceedings of the 16th IEEE Photovoltaic Specialists Conference (1982) 1133-37
- [4] D. A. Cusano *CdTe Solar Cells and Photovoltaic Heterojunctions in II-VI Compounds*, Solid State Electronics **6** (1963) 217-32
- [5] R. W. Aldrich and R. E. Halstead *New Work on CdTe Thin Film Solar Cells*, Proceedings of the 6th IEEE Photovoltaic Specialists Conference (1967) 188-200
- [6] A. L. Fahrenbruch, V. Vasilchenko, F. Buch, K. Mitchell and R. H. Bube *II-VI Photovoltaic Heterojunctions for Solar Energy Conversion*, Applied Physics Letters **25** (1974) 605-08
- [7] E. I. Adirovich, Y. M. Yuabov and G. R. Yagudaev *Investigation of n-CdS-p-CdTe Thin Film Heterojunctions*, Physica Status Solidi A **6** (1971) 311-22
- [8] D. Bonnet and H. Rabenhorst *New Results on the Development of a Thin-Film p-CdTe n-CdS Heterojunction Solar Cell*, Proceedings of the 9th IEEE Photovoltaic Specialists Conference (1972) 129-32
- [9] T. Aramoto, S. Kumazawa, H. Higuchi, T. Arita, S. Shibutani, T. Nishio, J. Nakajima, M. Tsuji, A. Hanafusa, T. Hibino, K. Omura, H. Ohyama and M. Murozono *16.0% Efficient Thin-Film CdS/CdTe Solar Cells*, Japanese Journal of Applied Physics Part 1 **36** (1997) 6304-05
- [10] A. De Vos, J. E. Parrott, P. Baruch and P. T. Landsberg *Bandgap Effects in Thin-Film Heterojunction Solar Cells*, Proceedings of the 12th European Photovoltaic Solar Energy Conference (1994) 1315-18
- [11] B. E. McCandless and S. S. Hegedus *Influence of CdS Window Layers on Thin-Film CdS/CdTe Solar Cell Performance*, Proceedings of the 22nd IEEE Photovoltaic Specialists Conference (1991) 967-70
- [12] C. S. Ferekides, K. Dugan, V. Ceekala, J. Killian and D. Oman *The Effects of CdS Processing and Glass Substrates in the Performance of CdTe Solar Cells*, Proceedings of the IEEE 1st World Conference on Photovoltaic Energy Conversion (1994) 99-102
- [13] P. V. Meyers *Design of a Thin Film CdTe Solar Cell*, Solar Cells **23** (1988) 59-67
- [14] E. Janik and R. Triboulet *Ohmic Contacts to p-Type Cadmium Telluride and Cadmium Mercury Telluride*, Journal of Physics D **16** (1983) 2333-40

- [15] A. Romeo, A. N. Tiwari, H. Zogg, M. Wagner and J. R. Günter *Influence of Transparent Conducting Oxides on the Properties of CdTe/CdS Solar Cells*, Proceedings of the 2nd World Conference on Photovoltaic Solar Energy Conversion (1998) 1105-08
- [16] J. Patterson *Surface Layers on CdTe*, Journal of Physics D 11 (1978) L83-L86
- [17] J. Sarlund, M. Ritala, M. Leskela, E. Siponmaa and R. Zilliacus *Characterization of Etching Procedure in Preparation of CdTe Solar Cells*, Solar Energy Materials and Solar Cells 44 (1996) 177-90
- [18] I. M. Kotina, L. M. Tukhkonen, G. V. Patsekina, A. V. Shchukarev and G. M. Gusinskii *Study of CdTe Etching Process in Alcoholic Solutions of Bromine*, Semiconductor Science and Technology 13 (1998) 890-94
- [19] H. Jäger and E. Seipp *Transition Resistances of Ohmic Contacts to p-Type CdTe and Their Time-Dependent Variation*, Journal of Electronic Materials 10 (1981) 605-18
- [20] D. Rose, D. S. Albin, R. J. Matson, A. B. Swartzlander, X. S. Li, R. G. Dhere, S. Asher, F. S. Hasoon and P. Sheldon *Effects of Oxygen During Close-Space Sublimation of CdTe Solar Cells*, Materials Research Society Symposium Proceedings 426 (1996) 337-48
- [21] B. M. Basol *Thin Film CdTe Solar Cells – a Review*, Proceedings of the 21st IEEE Photovoltaic Specialists Conference (1990) 588-94
- [22] M. P. R. Panicker, M. Knaster and F. A. Kroger *Cathodic Deposition of CdTe from Aqueous Electrolytes*, Journal of the Electrochemical Society 125 (1978) 567-72
- [23] R. W. Birkmire and E. Eser *Polycrystalline Thin Film Solar Cells: Present Status and Future Potential*, Annual Review of Materials Science 27 (1997) 625-53
- [24] I. Clemminck, M. Burgelman, M. Casteleyn, J. De Poorter and A. Vervaeet *Interdiffusion of CdS and CdTe in Screenprinted and Sintered CdS-CdTe Solar Cells*, Proceedings of the 22nd IEEE Photovoltaic Specialists Conference (1991) 1114-19
- [25] A. W. Brinkman, H. M. Al-Allak, G. R. Awan, P. D. Brown, K. Durose, C. Ercelebi, M. Y. Simmons and J. Woods *Epitaxial CdTe-Based Solar Cells*, International Journal of Solar Energy 12 (1992) 233-45
- [26] J. L. Boone, T. P. Vandoren and A. K. Berry *Deposition of CdTe by Spray Pyrolysis*, Thin Solid Films 87 (1982) 259-64
- [27] H. R. Moutinho, R. G. Dhere, M. M. Al-Jassim, D. H. Levi and L. L. Kazmerski *Investigation of Induced Recrystallization and Stress in Close-Spaced Sublimated and Radio-Frequency Magnetron Sputtered CdTe Thin Films*, Journal of Vacuum Science and Technology A 17 (1999) 1793-98
- [28] B. E. McCandless, L. V. Moulton and R. W. Birkmire *Recrystallization and Sulfur Diffusion in CdCl₂-Treated CdTe/CdS Thin Films*, Progress in Photovoltaics 5 (1997) 249-60
- [29] K. Durose, A. Turnbull and P. Brown *Structural Defects in Bulk and Epitaxial CdTe*, Materials Science and Engineering B 16 (1993) 96-102
- [30] M. M. Al-Jassim, F. S. Hasoon, K. M. Jones, B. M. Keyes, R. J. Matson and H. R. Moutinho *The Morphology, Microstructure and Luminescent Properties of CdS/CdTe Thin-Film Solar-Cells*, Proceedings of the 23rd IEEE Photovoltaic Specialists Conference (1993) 459-65

- [31] D. H. Levi, H. R. Moutinho, F. S. Hasoon, B. M. Keyes, R. K. Ahrenkiel, M. Aljassim, L. L. Kazmerski and R. W. Birkmire *Micro through Nanostructure Investigations of Polycrystalline CdTe – Correlations with Processing and Electronic Structures*, Solar Energy Materials and Solar Cells **41** (1996) 381-93
- [32] A. Romeo, D. L. Batzner, H. Zogg, A. N. Tiwari and C. Vignali *Influence of CdS Growth Process on Structural and Photovoltaic Properties of CdTe/CdS Solar Cells*, Solar Energy Materials and Solar Cells **58** (1999) 209-18
- [33] H. R. Moutinho, F. S. Hasoon, F. Abulfotuh and L. L. Kazmerski *Investigation of Polycrystalline CdTe Thin-Films Deposited by Physical Vapor-Deposition, Close-Spaced Sublimation, and Sputtering*, Journal of Vacuum Science and Technology A **13** (1995) 2877-83
- [34] Y. A. Cho, W. J. Nam, H. S. Kim, G. Y. Yeom, J. K. Yoon, K. H. Oho, S. H. Shin and K. J. Park *Effects of Rapid Thermal Annealing on CdTe/CdS Solar-Cell Fabrication*, Thin Films For Photovoltaic and Related Device Applications **426** (1996) 379-84
- [35] S. Seto, S. Yamada and K. Suzuki *Growth Kinetics and Structural Characterization of Polycrystalline CdTe Films Grown by Hot-Wall Vacuum Evaporation*, Solar Energy Materials and Solar Cells **50** (1998) 133-39
- [36] G. B. Harris *Quantitative Measurement of Preferred Orientation in Rolled Uranium Bars*, Philosophical Magazine Series 7 **43** (1952) 113-23
- [37] M. M. Al-Jassim, R. G. Dhere, K. M. Jones, F. S. Hasoon and P. Sheldon *The Morphology, Microstructure, and Luminescent Properties of CdS/CdTe Films*, Proceedings of the 2nd World Conference on Photovoltaic Solar Energy Conversion (1998) 1063-66
- [38] S. N. Al-Amri *Preparation and Characterisation of Thin-Film CdS/CdTe Solar Cells Produced by Close-Space Sublimation* (University of Durham Ph.D. thesis, 1999)
- [39] O. Zelaya, F. Sánchez-Sinencio, J. G. Mendoza-Alvarez, M. H. Farias, L. Cota-Araiza and G. Hirata-Flores *Large Grain-Size CdTe-Films Grown on Glass Substrates*, Journal of Applied Physics **63** (1988) 410-13
- [40] D. Kim, S. Pozder, B. Qi, Y. Zhu, T. E. Furtak, D. L. Williamson and J. U. Trefny *Thin Film CdS/CdTe Solar Cells Fabricated by Electrodeposition*, AIP Conference Proceedings **306** (1994) 320-28
- [41] D. Kim, B. Qi, D. L. Williamson and J. U. Trefny *Effect of Annealing on Microstructure, Residual Stress, and Photovoltaic Characteristics of Electrodeposited CdTe Films*, Proceedings of the 24th IEEE Photovoltaics Specialists Conference (1994) 338-41
- [42] A. Kampmann, P. Cowache, J. Vedel and D. Lincot *Investigation of the Influence of the Electrodeposition Potential on the Optical, Photoelectrochemical and Structural Properties of as-Deposited CdTe*, Journal of Electroanalytical Chemistry **387** (1995) 53-64
- [43] B. Qi, D. W. Kim, D. L. Williamson and J. U. Trefny *Effects of Postdeposition Heat-Treatment on Morphology and Microstructure of CdTe Grown by Electrodeposition*, Journal of the Electrochemical Society **143** (1996) 517-23
- [44] D. R. Johnson *Microstructure of Electrodeposited CdS/CdTe Cells*, Thin Solid Films **361** (2000) 321-26

- [45] H. R. Moutinho, F. S. Hasoon and L. L. Kazmerski *Studies of the Microstructure and Nanostructure of Polycrystalline CdTe and CuInSe₂ Using Atomic-Force and Scanning-Tunneling-Microscopy*, Progress in Photovoltaics **3** (1995) 39-46
- [46] T. L. Chu, S. S. Chu, C. Ferekides, J. Britt, C. Q. Wu, G. Chen and N. Schultz *Thin-Films of II-VI Compounds and Alloys*, Solar Cells **30** (1991) 123-30
- [47] P. R. Edwards, K. Durose, J. Beier, M. Campo and D. Bonnet *A Comparative Study of CdTe Solar Cell Activation Using CdCl₂ and Cl*, Proceedings of the 16th European Photovoltaic Solar Energy Conference (2000) *In Press*
- [48] N. W. Duffy, D. Lane, M. E. Ozsan, L. M. Peter, K. D. Rogers and R. L. Wang *Structural and Spectroscopic Studies of CdS/CdTe Heterojunction Cells Fabricated by Electrodeposition*, Thin Solid Films **361** (2000) 314-20
- [49] W. Stadler, D. M. Hoffman, H. C. Alt, T. Muschik, B. K. Meyer, E. Weigel, G. Muller-Vogt, M. Salk and K. W. Benz *Optical Investigations of Defects in Cd_{1-x}Zn_xTe*, Physical Review B **51** (1995) 10619-30
- [50] Y. S. Tyan, F. Vazan and T. S. Barge *Effect of Oxygen on Thin-Film CdS/CdTe Solar Cells*, Proceedings of the 17th IEEE Photovoltaic Specialists Conference (1984) 840-45
- [51] S. A. Galloway, A. J. Holland and K. Durose *A Study of CdS/CdTe Thin-Film Solar-Cells Using Beam Injection Techniques*, Journal of Crystal Growth **159** (1996) 925-29
- [52] P. R. Edwards, K. Durose, S. A. Galloway, D. Bonnet and H. Richter *Front-Wall Electron Beam-Induced Current Studies in Thin-Film CdS/CdTe Solar Cells*, Proceedings of the 2nd World Conference on Photovoltaic Solar Energy Conversion (1998) 472-76
- [53] D. Cahen and R. Noufi *Surface Passivation of Polycrystalline, Chalcogenide Based Photovoltaic Cells*, Solar Cells **30** (1991) 53-59
- [54] B. E. McCandless and R. W. Birkmire *Analysis of Post Deposition Processing for CdTe/CdS Thin Film Solar Cells*, Solar Cells **31** (1991) 527-35
- [55] R. W. Birkmire, B. E. McCandless and S. S. Hegedus *Effects of Processing on CdTe/CdS Materials and Devices*, International Journal of Solar Energy **12** (1992) 145-54
- [56] D. G. Jensen, B. E. McCandless and R. W. Birkmire *Thin Film Cadmium Sulphide/Cadmium Telluride Alloys*, Proceedings of the Materials Research Society Symposia **426** (1996) 325-30
- [57] N. Suyama, T. Arita, Y. Nishiyama, N. Ueno, S. Kitamura and M. Murozono *CdS/CdTe Solar Cells by the Screen-Printing-Sintering Technique*, Proceedings of the 21st IEEE Photovoltaic Specialists Conference (1990) 498-503
- [58] G. J. Conibeer, D. W. Lane, J. D. Painter, S. Hearne, G. Pekar, P. Capper and K. D. Rogers *Diffusion in CdTe and CdS Polycrystalline Thin Films and Single Crystal CdS*, Proceedings of the 2nd World Conference on Photovoltaic Solar Energy Conversion (1998) 1097-100
- [59] H. R. Moutinho, M. M. Al-Jassim, F. A. Abulfotuh, D. H. Levi, P. C. Dippo, R. G. Dhere and L. L. Kazmerski *Studies of Recrystallisation of CdTe Thin Films after CdCl₂ Treatment*, Proceedings of the 26th IEEE Photovoltaic Specialists Conference (1997) 431-34

- [60] E. L. Hall and J. B. Van Der Sande *Plastic Deformation Behaviour and Dislocation Structure of CdTe Single Crystals*, Journal of The American Ceramic Society **61** (1978) 417-25
- [61] J. B. Nelson and D. P. Riley *An Experimental Investigation of Extrapolation Methods in the Derivation of Accurate Unit Cell Dimensions of Crystals*, Proceedings of the Royal Society of London **57** (1945) 160-77
- [62] J. E. Burke and D. Turnbull *Recrystallization and Grain Growth*, Progress in Metallurgical Physics **3** (1952) 220-92
- [63] F. J. Humphreys and M. Hatherley "Recrystallization and Related Annealing Phenomena" (Pergamon, Oxford, 1995)
- [64] R. Ahmad-Bitar and D. E. Arafah *Processing Effects on the Structure of CdTe, CdS and SnO₂ Thin Films*, Solar Energy Materials and Solar Cells **51** (1998) 83-93
- [65] Y. Y. Loginov, K. Durose, H. M. Alallak, S. A. Galloway, S. Oktik, A. W. Brinkman, H. Richter and D. Bonnet *Transmission Electron Microscopy of CdTe/CdS Based Solar-Cells*, Journal of Crystal Growth **161** (1996) 159-63

Chapter 5

Experimental Techniques and Image Analysis

5.1 INTRODUCTION

To characterise the materials and devices used in this work a number of commonly reported techniques were used, such as current-voltage measurements, spectral response, and TEM. In addition to these, bevelling and defect etching techniques were used to study microstructure, with the resulting micrographs analysed using computational image analysis. Firstly, though, the details of the thin film growth are outlined.

5.2 MATERIAL GROWTH

The starting material, for studies into device quality material in this work, consisted of 25mm squares of CdTe/CdS solar cell, supplied by ANTEC

GmbH. Both the CdTe and CdS were grown by CSS at a growth temperature of 500°C. The technique of CSS is described in the review of thin-film growth methods in the previous chapter (section 4.2.2). The structure of the cells was 8-10 μm CdTe on ~100 nm CdS deposited on ITO coated glass. Some of the supplied material was supplied as-grown, whilst some was treated with the manufacturers proprietary CdCl_2 treatment. A description of the processing of this material with CdCl_2 to make the devices used in this work is deferred until section 7.2.1.

5.3 SOLAR CELL CHARACTERISATION

5.3.1 CURRENT-VOLTAGE MEASUREMENTS

Current-voltage measurement (J - V) is a standard way to extract the most relevant solar cell parameters— namely V_{oc} , J_{sc} , FF , and η — as discussed in section 2.3.1. This section will describe the apparatus used in this work to measure J - V curves.

A schematic of the J - V equipment is shown in figure 5.1. The whole apparatus, excluding measurement and recording equipment, is situated in a light-tight enclosure, to prevent stray light from affecting the readings. The lamp (1000W, quartz-tungsten-halogen) is chosen to mimic the general shape of the AM 0 spectrum. Sunlight's passage through water vapour in the atmosphere is a major contributing factor in modifying the AM 0 spectrum to become the AM 1.5. In order to model this effect in our equipment, a water tank is placed in the light-path. The depth of water required can be calculated from atmospheric water vapour density, and is ~25mm.

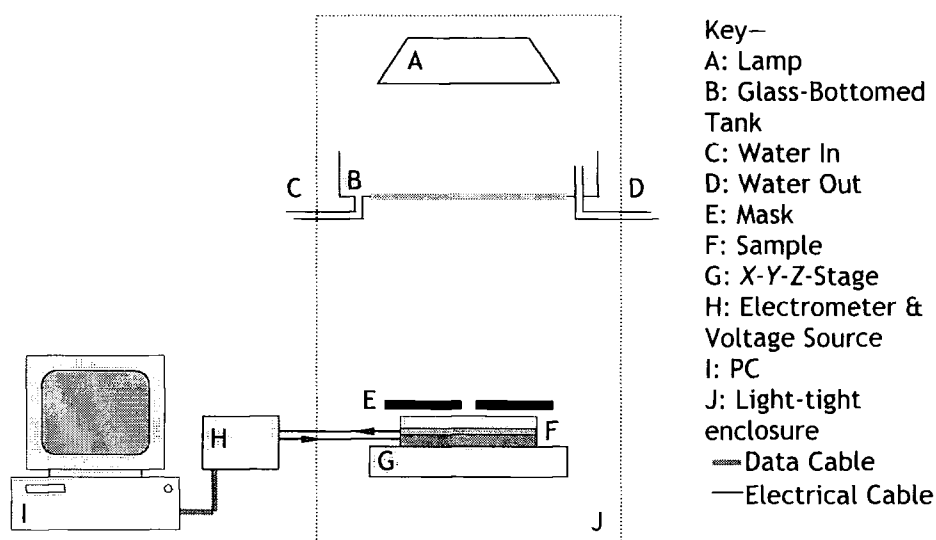


Figure 5.1: Schematic Diagram of the current-voltage measurement equipment used in this work.

Above the sample lies a mask, of the same shape and size as the back contact of the underlying sample. The purpose of this is to prevent light falling on un-contacted areas, and producing spurious current unrepresentative of a real device. The sample itself rests on an *x-y-z* stage. To standardise the light power, a photometer was used in place of the sample, and the height of the stage is adjusted in the *z*-direction until an incident power of 100 mW cm^{-2} is measured, a standard value in most reports of *J-V* measurements. This is a widely used power density standard for cell characterisation. When a cell has been installed in the apparatus, the *x*- and *y*-directions of the stage were adjusted until the maximum current is detected at a particular bias (typically zero). This ensures that the contact is centred beneath the mask aperture. A Keithley 617 electrometer was used to both supply a bias voltage and measure the current, with control of its voltage output and recording of the current it measures being performed using a PC via an IEEE 488 interface.

Although the mask prevents a large amount of extraneous heating of the cell from the high-power light source, the lamp was only turned on when a measurement was taking place. Thus, the cell temperature did not have time to rise appreciably above room temperature and thermally distort the readings, performed nominally at room temperature.

To test this equipment, J - V curves were taken for cells that had previously been measured in rigorously standardised J - V equipment. This comparison showed that current densities in the equipment described here were ~ 1.5 times greater than those measured elsewhere (probably due to the light spectrum being only an approximation to AM 1.5). However, by applying a linear correction factor to the data gave very good agreement between the two sets of apparatus. In addition to this, this work is not concerned with characterising cells, only with comparing cells to each other. This apparatus is sufficient for this purpose.

5.3.2 SPECTRAL RESPONSE

Spectral response measures the variation of quantum efficiency of a solar cell with wavelength. Comparison of this data with the expected values, calculated from theory, can give an indication of which parts of a device are behaving non-ideally, by where in the spectrum the difference occurs. Specific to the case of CdTe/CdS cells, it is possible to detect intermixing of the CdTe and CdS layers by a reduction of the long wavelength end of the spectrum: this is due to a bowing of the bandgap in the $\text{CdS}_{1-x}\text{Te}_x$ system. This type of measurement is widely reported [1, 2].

As part of this work, apparatus for spectral response measurements was built, and the control software written. A schematic of this equipment is shown in figure 5.2. The light source (a quartz-tungsten-halogen bulb) is

focussed roughly onto the input slits of a monochromator using a reflector. This type of lamp was chosen, as the spectrum is reasonably linear over the wavelength range of interest (400-900 nm). On exit from the monochromator, the beam is split, with one half of the light being focussed onto the sample, and the other falling on a calibrated photodiode. The output of this photodiode is used in a feedback loop controlling the lamp voltage, allowing a constant photon flux to be maintained at the sample. A mask is also placed above the sample, to prevent spurious current readings as described in the case of J - V in the previous section. The output from the cell is measured with an electrometer connected to a PC that controls all of the equipment. A typical iteration would be:

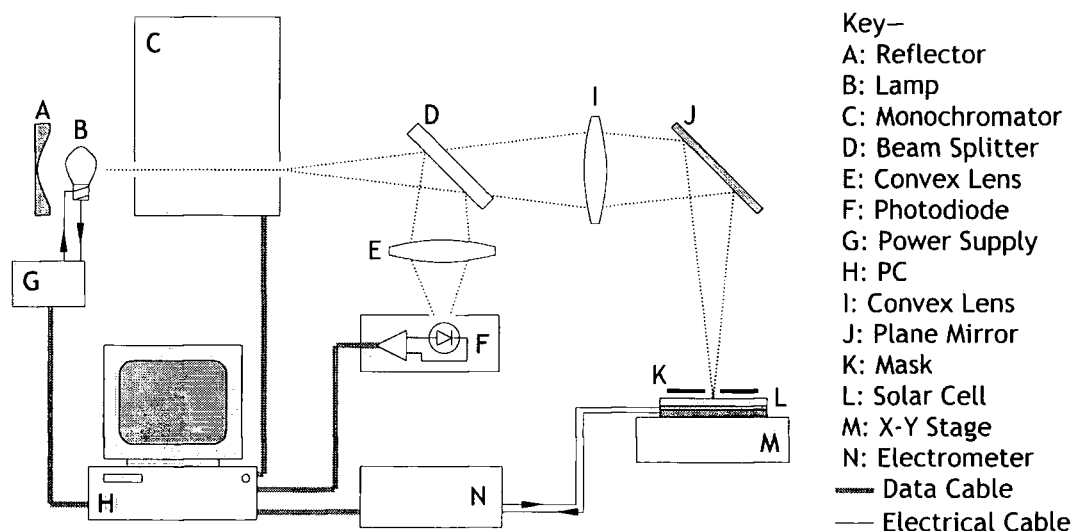


Figure 5.2: Schematic layout of the apparatus used for spectral response measurements.

- Set the monochromator to the desired wavelength;
- Measure the photodiode output and adjust the lamp output accordingly, and;
- Record the current.

In order to turn the current into quantum efficiency, a value for the photon flux is needed. This is found using a commercially calibrated photodiode in place of the sample. Data found in this way is then used to modify the data taken with a sample in place.

5.4 BEVELLING

In order to reveal the structure or composition throughout a thin film, bevelling can be performed. There are a number of different techniques reported for this but, in this work, the basic principle used is slow immersion of a film into an etchant solution (first reported by Huber *et al.* [3]). The end of the film immersed first will thus have more material removed, revealing the film near the substrate. The converse is obviously true for the other end of the film. This type of analysis is important for the CdTe films studied here, as various factors— discussed in section 4.3— cause the microstructure of the layer to vary considerably throughout its thickness.

In order to attain a controlled descent of the sample into the liquid, and thus a well-formed bevel, the equipment shown in figure 5.3 was used. The stationary reservoir contains the sample, typically mounted on a glass microscope slide with wax. The second, moveable reservoir can be raised or lowered to alter the level of etchant in the stationary reservoir. This is achieved using a motor with suitable gearing. A schematic diagram of the resulting shape is shown in figure 5.4. The unbevelled region is created by masking half of the sample with Lacomit varnish, and is created to give a possible reference surface for stylus profilometry.

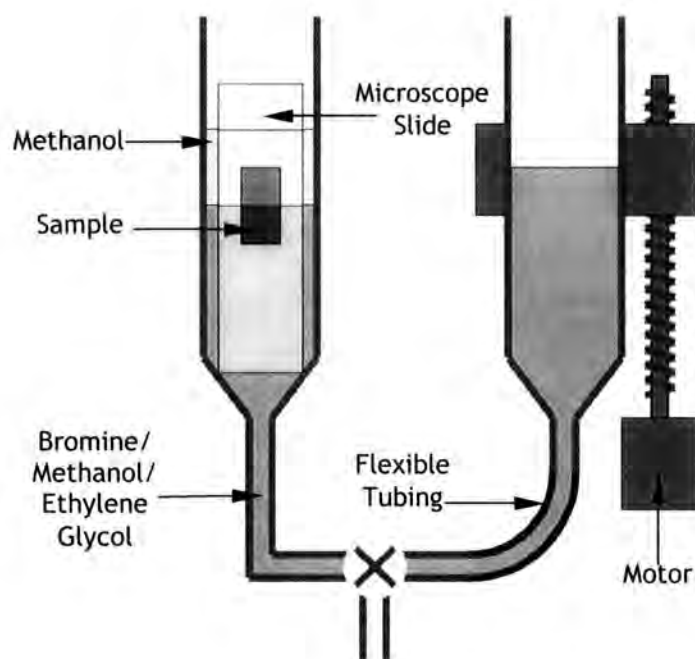


Figure 5.3: Schematic of the bevelling apparatus. The arrangement shown here is for use with bromine solution: a floating layer of methanol is used to prevent the sample being attacked by bromine vapour before immersion into the bromine solution.

The choice of etchant is of considerable importance: its etch rate needs to be suitably slow to allow a reasonable immersion rate to be used, and in many cases (this included) the final surface morphology is important. Two different etchants were tested; the specifics of their use, and the results achieved are detailed below.



Figure 5.4: Schematic diagram of a bevelled thin film bi-layer on a glass substrate

Citric Acid and Hydrogen Peroxide: A solution of citric acid in hydrogen peroxide (4g/100ml) is widely known to dissolve CdTe. It is simple to use, having none of the technical difficulties caused by evaporation or high viscosity (these two problems are discussed below in relation to the use of bromine/methanol). Using an immersion rate of ~5mm/minute gave a well-formed bevel. However, on examination of the resulting surface by optical microscopy, the morphology shown in figure 5.5 was seen. For the analysis to be performed on the bevelled films, a smooth, featureless morphology is needed (as discussed in section 7.2.4). Thus, this etchant was rejected for use in making bevels.

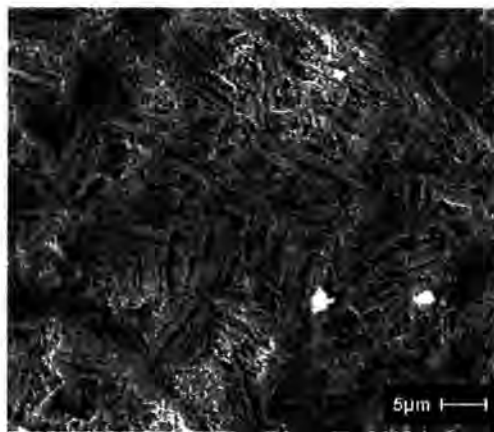


Figure 5.5: SEM image showing the resultant surface after immersion of a CdTe film in a citric acid/hydrogen peroxide solution. Preferential etching of twin boundaries can clearly be seen.

Bromine/Methanol: Using bromine methanol incurs a small experimental toll: used in the simple fashion described above, the bromine evaporates, attacking the film surface before immersion. To prevent this, a floating layer of methanol was used in the stationary reservoir to protect the sample. To control the etch rate and allow the methanol to float, ethylene

glycol* is added to the etchant solution to increase its density and viscosity. This increase in viscosity, though, does complicate the bevelling process: the viscous solution does not move smoothly when the reservoir is raised or lowered, leading to some uncertainty in the etch rate, and thus irreproducibility. However, the resulting surfaces were smooth and the bevels well enough formed. Thus, this solution was chosen for producing the bevels described in chapter 7, with volume fractions 0.05/0.25/0.7 of bromine, methanol and ethylene glycol respectively, at a dipping speed of ~2mm/minute.

5.5 DEFECT REVEALING ETCHES

For the studies of grain size of CdTe films, it was necessary to selectively etch the polished and bevelled surfaces, as no grain boundaries are visible after this treatment. A number of reported grain boundary etches were tested in order to find the most suitable. Desirable characteristics are high grain-boundary selectivity, reproducibility, and controllability. The results for each etchant trialed are reported below. All trials were performed at room temperature.

0.5% by Volume Bromine/Methanol: This is a photochemical etch reported by Williams [4] for CdTe. Small pieces of both CdCl₂ treated and untreated CdTe were mounted on a microscope slide and immersed in the solution for times between 30s and 5 minutes, and with different light sources (e.g. quartz-tungsten-halogen lamp, tungsten lamp). The etchant produced regions of well-defined grain boundaries after 2-3 minutes. However, inhomogeneities in the light source, and heating of the solution caused by the

* Also called 1,2 ethanediol

light source meant that this approach was not viable in an open beaker. Rather than build a more complicated piece of apparatus, other etchants were tried.

Saturated FeCl₃: This solution has been reported as a reliable defect etchant for CdTe by Watson *et al.* [5]. A saturated solution requires about 35g of FeCl₃·6H₂O per 10ml of water. The net reaction on dissolving the FeCl₃ is to produce concentrated hydrochloric acid and rust:



where,



CdTe samples of both as-deposited and untreated material were tested, in the same manner as above, to test its suitability as a grain boundary etch, and the optimum time for immersion.

Results from a typical trial are shown in figure 5.6. Extended treatment in this solution (~1min) led to etching of the grain boundaries in both as-deposited and CdCl₂ treated material. However, etch-pits were also created within the grains, probably at twin boundaries, as Watson *et al.* suggest. These pits would interfere with the computational image analysis, and so these surfaces are unsuitable. On reducing the treatment time to ~10s, CdCl₂ treated material still had etched grain boundaries, but the etch pits had not developed. Hence, saturated FeCl₃ is a very suitable etchant for CdCl₂ treated CdTe. However, as-grown material did not show any etching at the grain boundaries in this time. This suggests that the solution is not, in fact, acting on the CdTe in the way reported for bulk CdTe, but is probably dissolving the chlorine-rich material at the grain boundaries of the films.

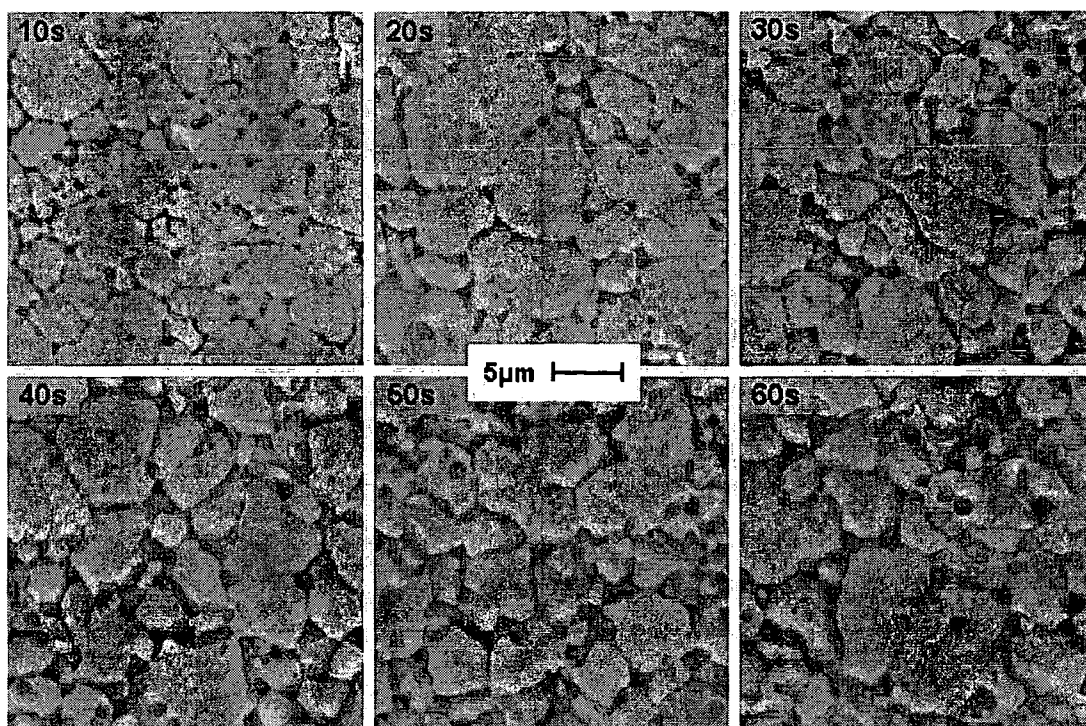


Figure 5.6: The effect of saturated FeCl_3 solution on CdTe over time. The grain boundaries are clearly visible long before etch pits begin to form, making the etching solution suitable for computational image analysis.

Inoue's EAg-1 Reagent: This etchant has long been recognised as a defect etch for CdTe [6], and is made in the following proportions: 20ml water, 10ml 70% HNO_3 , 4g $\text{K}_2\text{Cr}_2\text{O}_7$ and 1.5mg AgNO_3 . Trials were again performed on both treated and as-grown material. The resulting etched surfaces showed cleanly etched boundaries for both types of material, although it was found that the results were less reproducible than for saturated FeCl_3 .

It was therefore decided that the CdCl_2 treated samples should be etched in FeCl_3 , whilst Inoue's EAg-1 solution should be used on the as-grown samples. Use of these etchants to investigate grain size distributions is presented in chapter 7.

5.6 IMAGE ANALYSIS

In order to perform computational analysis of grain size distributions, a suite of procedures was written in C++ to perform the image analysis and processing required. In this section the computational techniques and their development are described and discussed, while the code itself is reproduced in appendix A.

Reducing an image of a real grain-boundary network to a data-file of grain sizes and positions is not a trivial problem, and any image analysis program designed to do this will require a number of critical steps. The different techniques used for this purpose here are outlined below in their general form: the specific details of their implementation are deferred until chapter 7. All arguments here are couched in terms of grey-scale (8-bit) images where applicable.

5.6.1 SPATIAL FILTERING

Spatial filtering is a simple technique for image enhancement or processing, not being subject to the computational constraints of Fourier transform methods, or image-size impositions of the fast Fourier transform. Although spatial filtering does not require the image to be translated into the frequency domain, it employs the same concepts. In figure 5.7, the stages of constructing a filter from knowledge of the desired frequency domain filter function are shown. This type of spatial filtering will cause the same transformation as altering the frequency components directly when applied as described below.

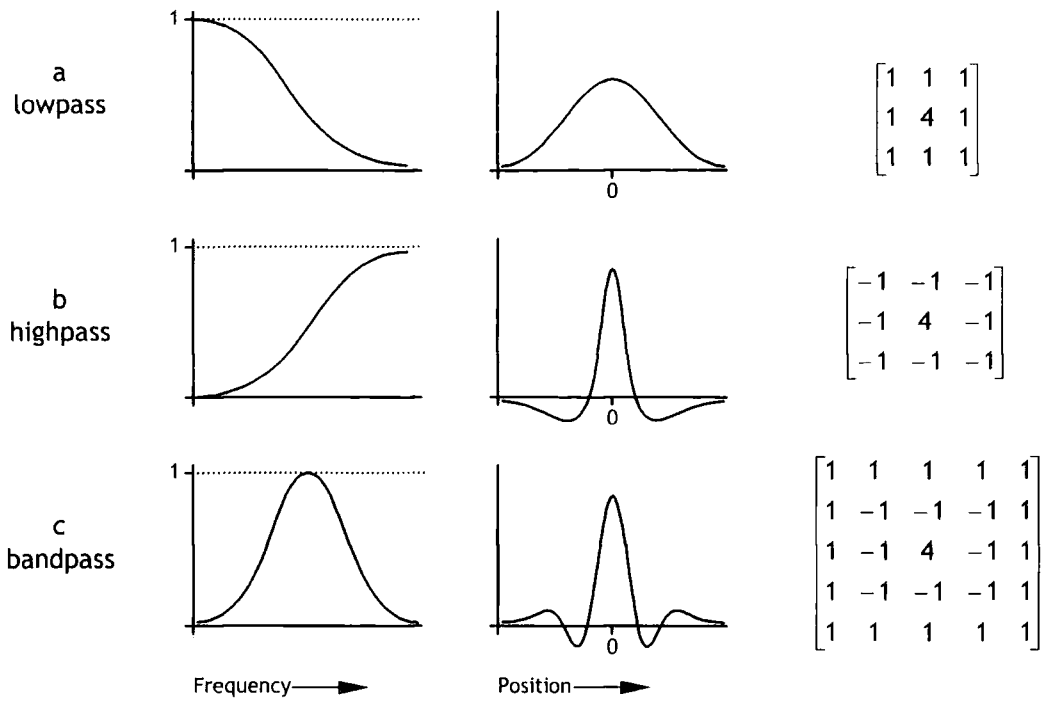


Figure 5.7: Examples of how spatial filters are constructed: the first column shows the filter function in the frequency domain; the second shows the inverse Fourier transform of this, i.e. the same filter in the spatial domain. The last column shows the spatial filtering function extended to two dimensions and approximated by the matrix filter (after Gonzalez and Woods [7]).

A filtering matrix is defined as,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad (5.3)$$

where n is an odd number greater than one. This matrix is then centred on each pixel of an image in turn, and each element of the array multiplied by the pixel over which it lies. The sum of these is the new pixel value (although it is usual to divide by the sum of matrix elements to return the pixel value to the 8-bit regime). This new value should be stored in a new image; otherwise, the matrix will be operating on an image other than the original.

In image analysis applications, it is usual for matrices used to be centro-symmetric: application of non-symmetric filters causes effects that are not in general mathematically meaningful. On applying these filters at the edges of images, where the matrix overlaps the edge of the image, the enhancement will be similar to the desired result, but not the same. In the case of the software used here, these edge pixels were simply removed after application of the filter. All filters used in this work were 3×3 and centro-symmetric, and the subroutine is capable only of handling this type of matrix.

5.6.2 FLOOD-FILLING

Flood-filling, whilst not an image analysis technique as such, forms an important part of many image analysis routines. A typical flood-filling routine will accept an image, a pixel coordinate, x , y , and a colour, C . If the pixel lies in a region of similarly coloured pixels, the routine will colour them all C . The other consideration for this technique is the question of what constitutes a boundary. The boundary between the two regions in figure 5.8a is obviously a boundary. However, that shown in figure 5.8b could be considered a boundary or not. A flood filling routine that considers this a boundary is called 'soft' and one that ignores it is termed 'hard'. The implementation of the two methods is, though, very similar, with the soft variant looking only at 4-neighbours, whilst the hard version looks at the 8-neighbours* of each pixel examined.

Two approaches to flood-filling exist, and will be discussed here. The first is compact, but limited by stack space, and the second requires a greater amount of— and more complex— code, but is considerably more robust.

* 4-neighbours of a pixel are those directly above, below and to either side. 8-neighbours of a pixel are the 4-neighbours, plus the four diagonal pixels at its corners.

A recursive flood filling routine consists of very few lines of code. All that is required in the routine is a *for*-loop to find a neighbour (either 4- or 8-) of same colour as those which need filling. It then passes the coordinates of this pixel to itself. Although elegant, this method is limited by the number of stacks available in the computer in question. In an image of 500×700 pixels (roughly typical for this work) it can easily be seen that a potential 350000 stacks could be needed; far more than the few thousand available in a conventional PC.

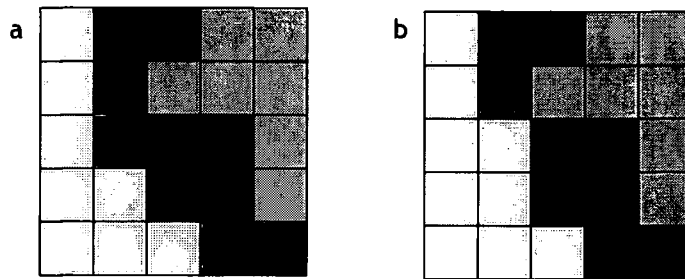


Figure 5.8: Two magnified digital images of a theoretical boundary (black) separating two grey regions. Shown in a) is a complete boundary. In b), however, the two grey regions have two corners touching: this is considered a boundary by a 'soft' flood-filling routine, but not by a 'hard' one.

The second approach to a flood filling routine is considerably more mechanical than the above recursive type, but does not have the limitations of the above routine. The same arguments are passed to the routine as above. Within the routine two arrays are then defined, 'current', *C*, and 'previous', *P*. To initialise the routine, the initial pixel is set to the fill colour and the coordinates are placed into *P*. Then, the program first searches around all of the pixels in *P* (initially only one) for neighbours of the required colour (again, 8- or 4- for hard or soft respectively). The coordinates of these neighbours are put into *C*, and the pixels at these coordinates filled. Once this is complete,

the pixels in C are then placed into P and the program iterates. It is simple to deduce that hard filling will fill in a roughly square pattern, and soft roughly circular. As only the pixels at the edge of the filled region are in either of the arrays at any one time, the largest size the array could be is $\sim 2(w + h)$ for an image of width of w and height h . This gives an array size of ~ 2400 entries for our 700×500 image. The final number will be considerably smaller than this as pixels at the edge of the image no longer feature in the arrays. This amount of memory is considerably smaller than that needed in stack space for the recursive routine.

Thus, although the code for the non-recursive implementation of the flood filling routine is more complicated to write, it is much more economical of memory, and more reliable, and so was used for this work.

5.6.3 SKELETONISATION

This process takes a noisy image of grain boundaries, and reduces it to a matrix in which each pixel is assigned to a particular grain. It is then simple to extract size and position information for grains from this type of data. The name 'skeletonisation' is used loosely: an alternative definition also in common usage is the process that reduces a series of thick lines in a monochrome image into lines a single pixel wide (for example, see Gonzalez and Woods [7]). In this implementation, the thick lines (grain boundaries) separate individually numbered regions (grains), and are reduced to zero thickness. It is quite reasonable to define the boundaries as having a negligible width at the image magnification used here.

The input into the routine is an image array consisting of noisy boundaries, having value zero, and separating grains whose pixels are individually numbered. Using unsigned 16 bit integers in the array

generously allows for ~64000 grains in the image. The process proceeds as follows: Each pixel is examined in turn. If its value is non-zero (i.e. within a grain), the program proceeds to the next pixel. If the pixel is on a boundary (zero), the program examines the 8-neighbours of that pixel, and tallies the number of each type of surrounding pixel (excluding boundary pixels). The pixel of interest is then assigned the value of its most frequently occurring neighbour in a secondary copy of the image. This use of a secondary copy of the image ensures that no bias creeps in due to the order in which the pixels are examined. After all of the pixels have been examined, the secondary image becomes the main image, and the process repeats until no pixels change when the image is examined. This implies that the grains now occupy the whole image, and the boundaries have disappeared.

Although most potential directional bias is removed from this method by use of the secondary image for storage of new pixel allocations, there is one source of bias. Examination of the 8-neighbours always proceeds in the same order, by means of two *for*-loops running from -1 to 1. Furthermore, if a boundary pixel is surrounded by two or more different pixels in equal quantities, it will pick that which it encountered first. The implication of this is that the whole image will be very slightly shifted in favour of the starting pixel.

There are a large number of solutions to this weighting, such as examining the 8-neighbours in random order, or picking the equally occurring neighbours at random. However, on modification of the routine to include this, no difference was detectable in the resultant image. Therefore, this modification was removed for the sake of computational pace.

5.7 TEM

Transmission Electron Microscopy is a well-established technique for the study of the structure of and defects in semiconducting materials. Use of this characterisation method was made in this work to study the region of CdTe/CdS cells within 1 μm of the metallurgical interface (the ‘near-interface’ region). This section describes the technique used for the preparation of plan view samples used to study this region.

Some researchers have had success with a ‘lift-off’ technique to make this kind of sample. This involves attaching a stub to the CdTe surface with a soluble adhesive, and simply pulling the stub off. The CdTe film should remain attached to the stub. This can then be removed with a suitable solvent and thinning undertaken in the usual way. This technique, which to the author’s knowledge has only been successful with PVD CdTe, was not found to be suitable for CSS-grown material; the adhesion of the layers to the substrate and to one another was found to be very high, resulting in only very small pieces of film being removed from the surface. Although these small pieces could have been mounted on a fine mesh grid, another technique was developed, which gave distinctly more satisfying results. This is detailed below.

A number of stages are required for the preparation of the samples for plan-view TEM. These are shown diagrammatically in figure 5.9, and are as follows:

- a) The surface morphology of the CdTe film is removed with $\frac{1}{4} \mu\text{m}$ diamond paste.
- b) The polished surface is stuck to a piece of microscope slide with glycol phthalate thermoplastic cement [8] (which is liquid at $\sim 150^\circ\text{C}$).

- c) The majority of the superstrate glass is ground off with 17 μm SiC (leaving a thickness of 50-100 μm).
- d) After masking the entire sample apart from the glass superstrate, the remainder of the glass is etched off with 40% HF. The ITO acts as an etch-stop. This leaves only the CdTe/CdS/ITO multilayer, stuck to the slide.
- e) Disks of material can then be drilled from the thin-film, and ring-grids attached with silver-loaded epoxy. The glycol phthalate cement is then dissolved in acetone, a solvent that leaves the epoxy resin intact.

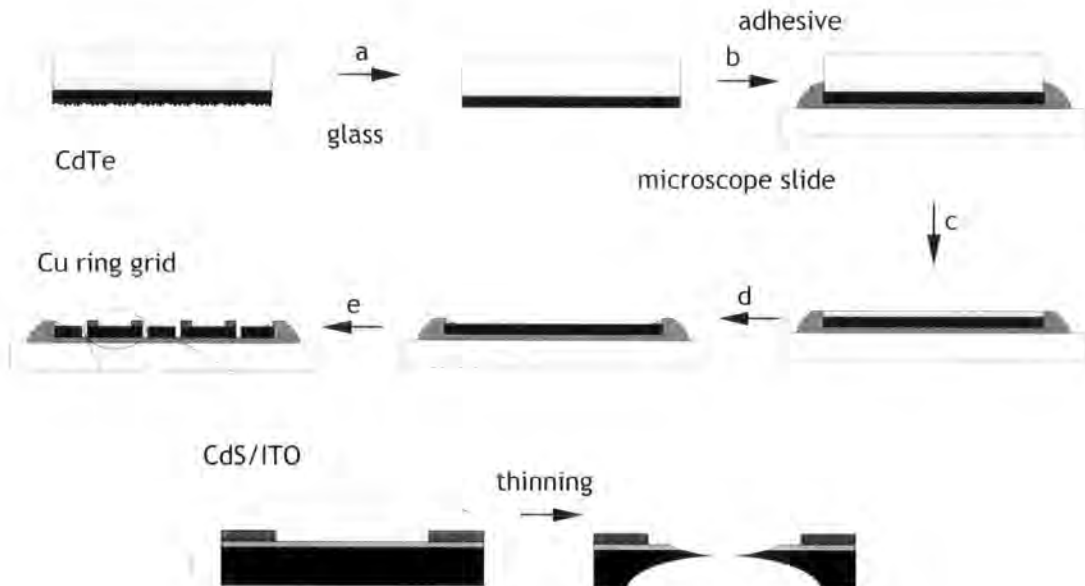


Figure 5.9: The stages in the construction of a plan-view TEM sample for examination of the near-interface region in CdTe/CdS solar cells. See text for details.

As the CdTe/CdS/ITO multilayer is at this stage only $\sim 10\ \mu\text{m}$ in thickness, a reasonably temperate form of ion beam thinning can be used to achieve electron transparency. At room temperature, solid source iodine has a sufficiently low vapour pressure to supply the thinning species. Although

more commonly encountered as I^- , the ion responsible for the thinning here is I^+ . The thinner used was manufactured by Ion Tech Ltd.

To confirm that thinning has revealed the near-interface region, the multilayers are thinned from the CdTe side until they cease to be opaque; 1 μm of CdTe absorbs 99% of visible light, and thus any degree of transparency in the films indicates that they are thinner than this value. The film is then thinned from the other side until holed, guaranteeing that the material close to this hole is free from ITO or CdS, in addition to being highly electron transparent.

— □ —

The next chapters will now describe the application of the techniques described in this chapter to thin-film and bulk CdTe, and discuss the results obtained. The first system to be examined is the behaviour of pressed pellets of CdTe powder on annealing.

5.8 REFERENCES

- [1] A. Rohatgi, S. A. Ringel, R. Sudharsanan and H. C. Chou *An Improved Understanding of Efficiency Limiting Defects in Polycrystalline CdTe/CdS Solar Cells*, Proceedings of the 22nd IEEE Photovoltaic Specialists Conference (1991) 962-66
- [2] D. M. Oman, K. M. Dugan, J. L. Killian, V. Ceekala, C. S. Ferekides and D. L. Morel *Reduction of Recombination Current in CdTe/CdS Solar Cells*, Applied Physics Letters **67** (1995) 1896-98
- [3] A. M. Huber, G. Laurencin and M. Razeghi *Study of Faults in Heteroepitaxial Films of III-V Materials by Metallographic Analysis on the Chemical Level*, Journal De Physique C **44** (1983) 409-14
- [4] D. J. Williams *Photochemical Etching of CdTe*, Proceedings of the 3rd II-VI Interaction Meeting (1986)
- [5] C. C. R. Watson, K. Durose, A. J. Banister, E. Okeefe and S. K. Bains *Qualification of a New Defect Revealing Etch for CdTe Using Cathodoluminescence Microscopy*, Materials Science and Engineering B **16** (1993) 113-17
- [6] M. Inoue, I. Teramoto and S. Takayanagi *Etch Pits and Polarity in CdTe Crystals*, Journal of Applied Physics **33** (1962) 2578-82
- [7] R. C. Gonzalez and R. E. Woods "Digital Image Processing- 3rd Edition" (Addison-Wesley, Reading, Mass., 1992)
- [8] G. W. Fynn and W. J. A. Powell "Cutting and Polishing Optical and Electronic Materials- 2nd Edition" (Adam Hilger, Bristol, 1988)

Chapter 6

Sintering Effects in CdTe Pressed Pellets

6.1 INTRODUCTION

As described in section 4.4, grain growth (section 3.3) is known to occur in thin-film CdTe/CdS solar cells on annealing. However, the precise mechanism by which this occurs is unclear, as are the effects for which it is responsible. In studying microstructural phenomena in materials, such as grain-growth, annealing pressed pellets of powdered material can be useful: the experimental technique is uncomplicated, and analysis is performed by measuring changes (often with time) in simple properties such as mass and physical size. Optical or secondary electron microscopy can also be revealing.

In this chapter, a pressed pellet study is used to try to probe the kinetic mechanism behind grain growth in CdTe (section 6.2), with a view to extrapolating how this behaviour may apply to thin-films of CdTe in CdTe/CdS solar cells. A study of the oxidation of the pellets is also presented

(section 6.3), oxidation being a factor that is known to influence device performance [1].

6.2 SINTERING OF PRESSED PELLETS

6.2.1 SINTERING KINETICS AND MECHANISM

Under the action of time and temperature, pressed pellets of many materials have been shown to exhibit densification and grain growth; using sintering to achieve this is a common aspect of materials processing, primarily in the field of ceramics. A flux is also routinely added to assist the migration of species between grains. As these pressed pellets are formed from compacted powder, a significant proportion of the volume is occupied by air and, as the grains grow, these voids are expelled from the body of the pellet. This reduces the overall volume of the pellet in a process known as densification. In this study the densification of CdTe powder was studied as a route to gaining understanding of grain-growth phenomena in CdTe, and how the addition of CdCl₂ (a common flux for CdTe) affects any processes.

Data on densification against sintering time can also be used to give information on the type of mechanism involved in the transport of material from one grain to another. Although a full derivation is outside the remit of this work, it can be shown that shrinkage, y , can be related to the time, t , by a power law of degree m [2]:

$$y = K'(T) t^m, \quad (6.1)$$

where $K'(T)$ is a constant for temperature T . The shrinkage is a one-dimensional equivalent of densification. This model arises from the consideration the interaction between two spherical grains: the different mechanisms of material transport to the 'neck' formed between the two

particles give rise to differing values of m . Some of these are presented in table 6.I. In real systems, however, these exponents are rarely measured, as grains are spherical only as an exception, and have a distribution of sizes. However, a power-law dependence is to be expected.

Mechanism	m	Model
Viscous Flow	1	Spheres in contact
Bulk Diffusion	2/5	
Intergranular Diffusion	1/3	
Bulk Diffusion	1/2	Overlapping spheres
Intergranular Diffusion	1/3	

Table 6.I: Values for the power dependence of shrinkage, m , (as defined in equation 6.1) for a two-sphere model, with the corresponding mechanism and model (from Koller [2]).

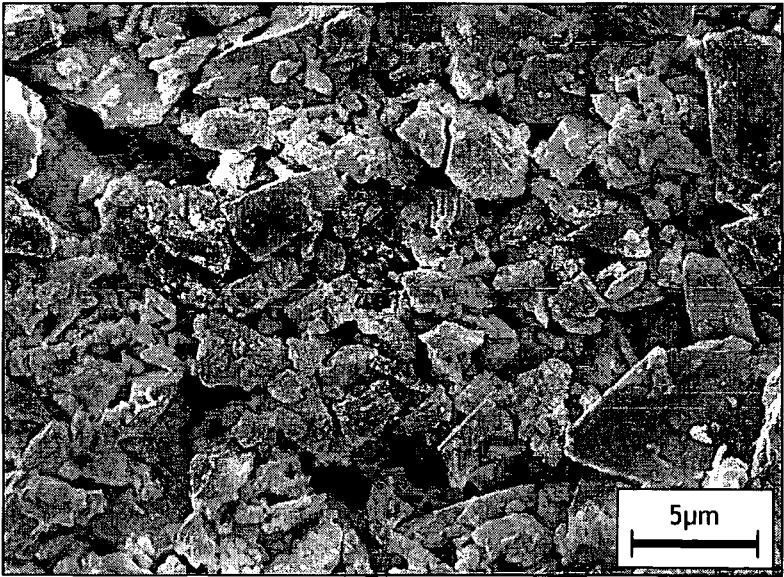


Figure 6.1: SEM micrograph of the powder used throughout the pressed pellet work described in this chapter. The powder shows a jagged morphology, with a large variance in particle sizes.

6.2.2 EXPERIMENTAL DETAILS

In order to allow consistent material to be used throughout the work described in this chapter, enough granular CdTe was ball milled to give sufficient supply for the whole study. Milling was performed for 24 hours using a Froude number [3] of ~ 0.7 (30% of the container body filled with the grinding material, and a water to solid ratio of 60% by volume). This powder was then dried, and briefly ground manually to remove any agglomeration caused by the drying. The resulting powder had a grain size of $\sim 1 \mu\text{m}$ and sharply faceted morphology shown in figure 6.1.

The pellets themselves were produced by placing a small amount of this powder (nominally 0.5g) into a 13mm diameter die and compressing the powder with a force of 30kN. The resulting pellets had a typical volume porosity, P , of $\sim 26\%$ defined as

$$P = \frac{V_{\text{pores}}}{V_{\text{pellet}}} = \frac{V_{\text{pellet}} - V_{\text{powder}}}{V_{\text{pellet}}} = 1 - \frac{m_{\text{powder}}}{\pi \rho_{\text{material}} r_{\text{pellet}}^2 h_{\text{pellet}}}, \quad (6.2)$$

for volume V , mass m , bulk density of the material in question, ρ , pellet radius r and pellet thickness h . The subscripts are self-explanatory. As mass is usually lost in the pellet making process, making the mass of powder used an uncertain quantity; this can be approximated by the mass of the pellet, if the mass of the air in the pores is neglected, giving,

$$P \approx 1 - \frac{m_{\text{pellet}}}{\pi \rho_{\text{material}} r_{\text{pellet}}^2 h_{\text{pellet}}}, \quad (6.3)$$

all of which variables are easily measurable, or are known.

The pellets were sintered at temperatures ranging between 400 and 900°C for times of up to 60 hours. CdCl_2 was included in proportions ranging from 0 to 10% by mass, introduced in powder form to the CdTe before compressing the pellet.

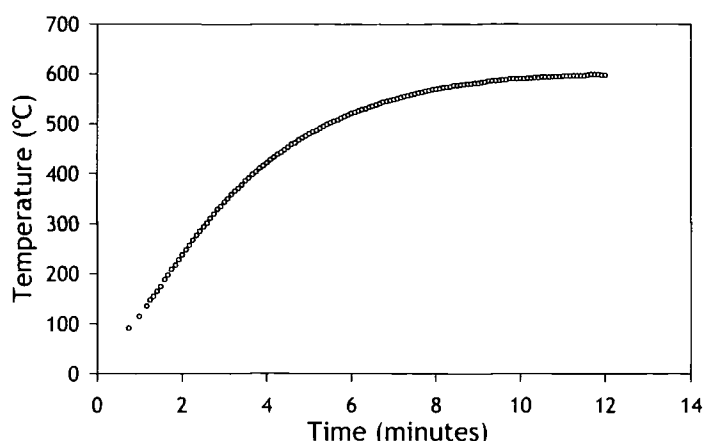


Figure 6.2: Temperature-time profile which the powder or pellet would experience on being inserted into the tube furnace

Before and after sintering, the pellets were weighed with an accuracy of $\pm 1\text{mg}$ and measured (using digital vernier callipers) to within $\pm 10\text{ }\mu\text{m}$. The pellets were then fired in an open tube furnace for the required time and at the requisite temperature. The furnace was pre-heated to reduce any thermal transients. On quantifying the heat-up time for a pellet, using a thermocouple in contact with a control pellet, a heat-up time of 10-12 minutes was found, as shown in figure 6.2. For the sintering times used here, the rate of temperature change is not significant. After annealing, any change in size can then be interpreted in terms of densification.

6.2.3 SINTERING RESULTS & DISCUSSION

The results of various pressed-pellet sintering experiments are shown in table 6.II. As can be seen, no evidence of densification was found at 400 to 450°C. Furthermore, at elevated temperatures a significant level of oxidation was seen, with an associated mass increase of the pellet. At still higher temperatures, the pellets were completely transformed, leaving only a hard yellow glassy phase, most probably a low melting-point oxide (for example, TeO_3 is a yellow substance with a melting point of 430°C [4]).

However, it was noted that the mechanical strength of the pellet increased even with low-temperature sintering, meaning that some structural effect had taken place. To study this and any other effects occurring in the grains upon annealing, SEM micrographs were taken of the fracture surface and of the face of those pellets indicated by shading in table 6.II. An untreated pellet was also examined, and a selection of the micrographs are shown in figure 6.3.

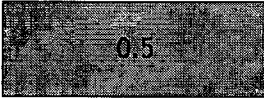
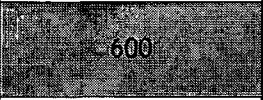
Time (hours)	Temperature (°C)	% CdCl ₂	Results
0.5	400	0	No Densification
			
4	450	0	
		5	
		10	
			No Densification. Heavy Oxidation and Mass Increase
60	400	5	No Densification. Mass Increase
17	900	0	Transformation into glassy phase
		5	

Table 6.II: Results of Sintering CdTe pressed pellets for differing times, temperatures, and percentages of CdCl₂. The shaded entries are those pellets examined by SEM, as described in the text.

Looking first at the outside of the pellets (the 'face'), it can be seen that considerable structural changes have taken place. The percentage of void is increased on annealing at 400°C without CdCl₂ (figures 6.3a and c); this may be indicative of small grained material that occupied this space being consumed by larger grains, although no extensive grain-growth is discernable. The sample treated with 5% CdCl₂ at 400°C (figure 6.3e) shows considerably different morphology again. The face has formed into a densified crust, and a

suggestion of a second phase also exists. Whilst this second phase makes the structure reminiscent of a partially recrystallised material, this assertion cannot be tested without considerable further study. This crust could also account for the higher mechanical strength of pellets after annealing.

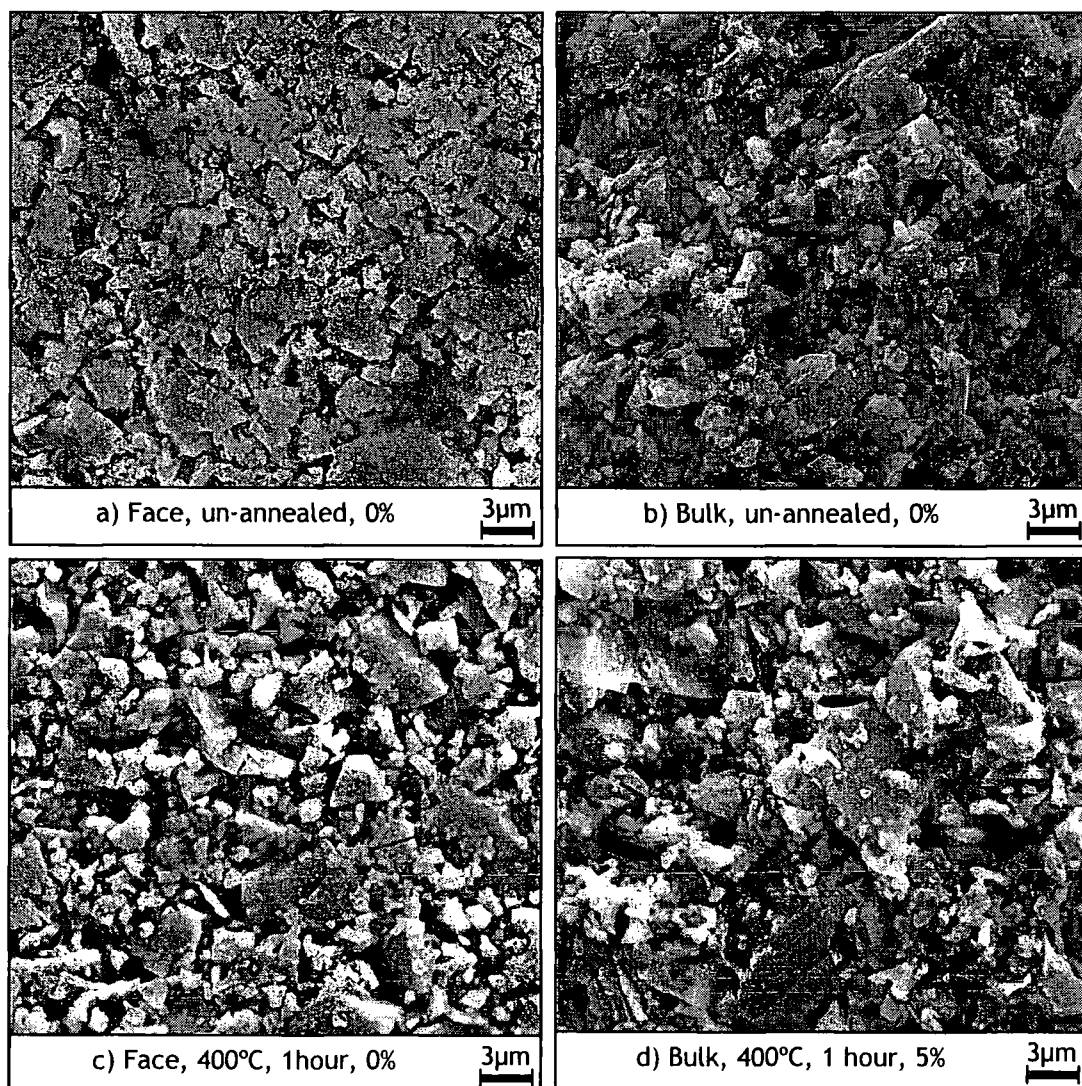


Figure 6.3: SEM micrographs pressed CdTe pellet surfaces and the bulk structure revealed on a fracture surface. Written below each image is (i) the area of the pellet examined, (ii) annealing temperature, (iii) annealing time, and (iv) amount of CdCl₂ included. (Continued overleaf).

Turning to the fracture surface of the pellets, annealing of the pellet at 400°C has very little effect on the microstructure. There is no discernible

difference between the un-annealed pellet (figure 6.3b), that annealed at 400°C without CdCl₂ (not shown), and that treated at 400°C with CdCl₂ (figure 6.3d). However, on increasing the temperature to 600°C (figure 6.3f), much of the small-scale microstructure is consumed forming a partially densified structure. In order to produce macroscopic densification, though, (and thus shrinkage of the pellet) the grains would need to grow by several more orders of magnitude, enough to eject voids from the body of the pellet.

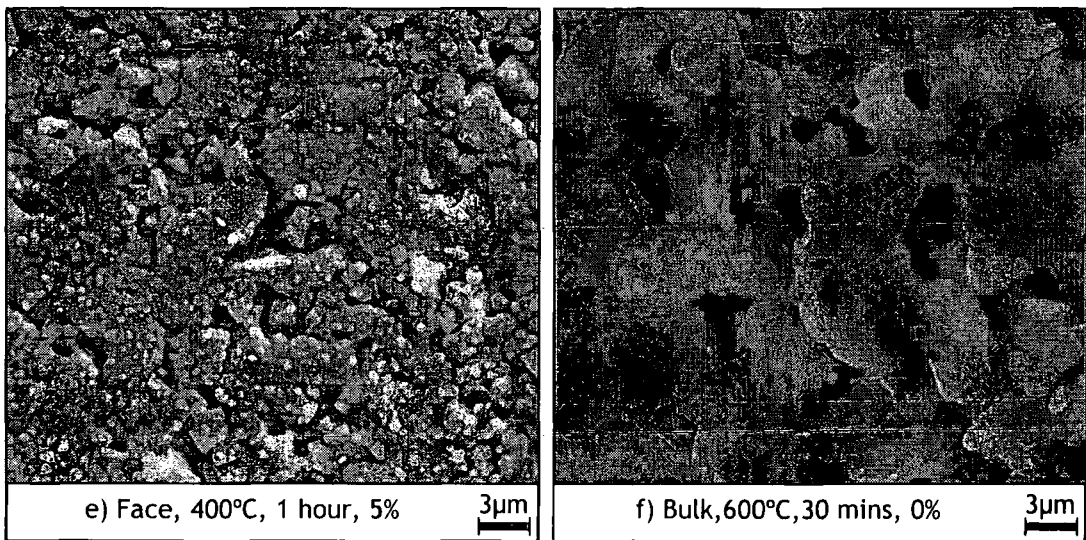


Figure 6.3: (continued)

From the above results, it seems clear why there is no densification observed on annealing pressed pellets: grain-growth is very limited indeed at 400°C, and is still insufficient at 600°C to cause densification. It is clear that a limiting grain size exists which is small at temperatures up to 600°C. The way in which the limits of grain-growth influence densification can be seen by considering equation 6.1. The densification is related to the time linearly by the factor $K'(T)$. This constant, however, is related to the grain radius by an inverse power-law [2]. Thus, as the radius of the grains increases, the capacity for densification drops off rapidly.

Although further energy could be supplied to the system by increasing the temperature (a *linear* function of $K'(T)$), this work has shown that oxidation quickly begins to occur at temperatures between 450 and 600°C. In addition, oxidation also occurs to a measurable degree at 400°C, if the samples are annealed for extended periods (60 hours in this case). The oxidation of CdTe is studied further in the following section.

Thus, this type of study, whilst able to detect limited grain-growth on sintering, cannot be used to predict any dynamics associated with this phenomenon in air. The grain-growth was also restricted by the existence of a limiting grain size, preventing densification. A further study would be best performed in inert gas or by sealing the pellets in evacuated ampoules, allowing high temperature sintering without chemical change.

6.3 OXIDATION STUDY OF CdTe

6.3.1 INTRODUCTION

The oxidation of CdTe pressed pellets seen in the last chapter affords a good opportunity to study the oxidation of CdTe in general terms. In order to study the reaction kinetics responsible for the oxidation of CdTe, a number of pellets were fired at 600°C for times varying between a few minutes and a few hours. These were prepared in exactly the same fashion as described in section 6.2.2. CdCl₂ flux was not used, as this would complicate the interpretation of such data by weight loss. However, for the shorter sintering times used here the heat-up time (figure 6.2) may well be significant. This is discussed later with reference to the results.

After sintering, any mass increase noted on annealing can then be interpreted as oxidation. To contrast this case for the pellets, a sample of loose powder was fired with the pellets.

6.3.2 THE DEAL-GROVE MODEL FOR OXIDATION

A model for the oxidation of solids was provided in the influential paper of Deal and Grove [5], who examined the creation of oxide layers on single-crystal silicon: it is described briefly below.

The transport of oxidising species in the system was broken down in to three stages, at a point in the process where oxidation has already begun:

- a) The species are transported from the surrounding atmosphere to the outer surface of the oxide;
- b) They travel across the oxide film to the silicon surface, and;
- c) They react with the silicon.

It can be shown that these three fluxes should be equal once the system reaches a steady state. Deal and Grove then proceeded to derive terms for each stage, and on equating these and solving of the resulting differential equation, they arrived at the relationship,

$$x_0 = \frac{A}{2} \left\{ \sqrt{1 + \frac{t + \tau}{A^2 / 4B}} - 1 \right\}, \quad (6.4)$$

which relates the thickness of the oxide layer , x_0 , to the time elapsed, t , via the constants τ , A and B . The time constant τ allows for the possibility of a native oxide layer present before annealing starts and A and B contain various system parameters. In the general form of this equation, shown in figure 6.4, we see linear behaviour in the lower limit:

$$x_0 = \frac{B}{A}(t + \tau), \quad (6.5)$$

changing to parabolic at higher values of t :

$$x_0 = \sqrt{Bt} . \quad (6.6)$$

This parabolic behaviour is often called ‘diffusion limited’; the rate-limiting step is the passage of species through the oxide layer.

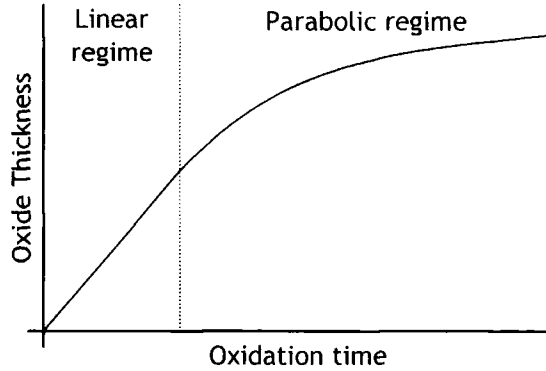


Figure 6.4: General form of the Deal-Grove equation (equation 6.4) for the thickness of an oxide layer with oxidation time. The two regimes of the curve (equations 6.5 and 6.6) are also shown (after Grosvenor [6]).

In applying this type of equation to the oxidation of CdTe pressed pellets, a number of points should be noted: the Deal-Grove equation was derived specifically for application to single-crystal silicon surfaces. Whilst the larger effective area associated with the pellets should not be an issue, once a grain is fully oxidised it will no longer take any part in the capture of oxygen molecules. In addition, the transport of oxygen to un-oxidised species will be influenced by transport through the pores of the pellet. Discussion of how this will affect the oxidation profile is deferred until later.

6.3.3 OXIDATION RESULTS AND DISCUSSION

The results for both pellets and powder are shown in figure 6.5. The measured mass change, Δm , for an initial mass of CdTe of m_i has been converted into a value for the completeness of the oxidation, f , using the relationship,

$$f = \frac{N_O}{N_{CdTe}} = \frac{\Delta m}{M_O} \cdot \frac{M_{Cd} + M_{Te}}{Cm_i} \quad (6.7)$$

The values M and N are the molar masses and number of moles of the components respectively, and C is the number of oxygen atoms required to oxidise CdTe. The latter is assumed equal to 3, as an oxide composition of $CdTeO_3$ is reported to be the most stable in the Cd-Te-O system [7], and also comprises the native oxide [8]. The values for f of close to unity for the powder sample (see figure 6.5) add further justification to the use of this chemical composition.

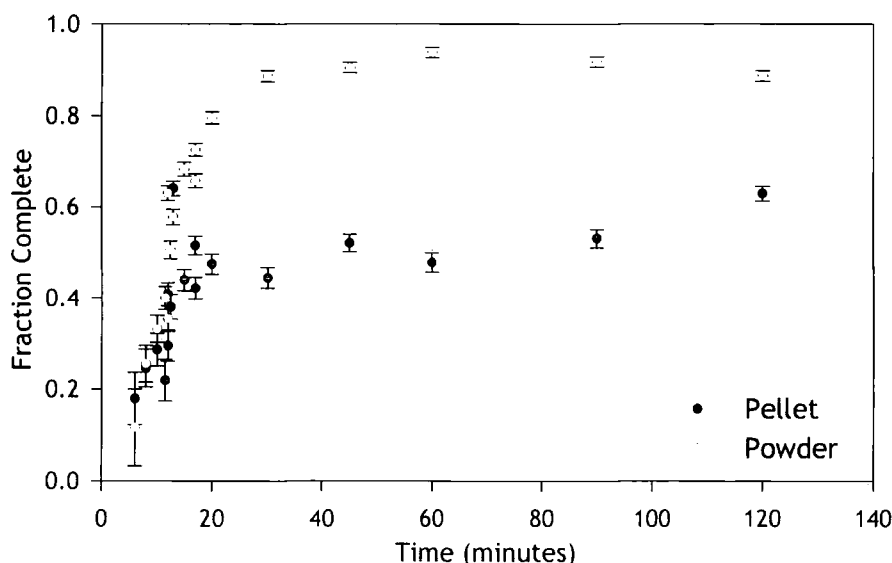


Figure 6.5: Oxidation profile for pellet and powder samples of CdTe annealed at 600°C. The units on the vertical axis assume the oxide composition to be $CdTeO_3$.

The first point of note in figure 6.5 is that the powder oxidises significantly faster than the pellets. This is to be expected, as the transport of oxygen to CdTe surfaces within the pellet will be limited by the pore size. Thus, once the outer surface of the pellets becomes oxidised, the oxidation profile for the pellet will begin to deviate from the powder case.

In order to compare this type of data to the Deal-Grove model (equation 6.4), the relationship between the fraction of the reaction complete and the oxide thickness needs to be ascertained. The former of these is directly proportional to the mass change. In addition, it can be assumed that the mass change is directly proportional to the thickness of oxide. Thus the data can be compared directly to the shape shown in figure 6.4.

Initial comparisons between this data and the Deal-Grove model appear limited, and any comparison is made with the assumption that the oxide thickness is proportional to the change in mass. For the data in the region $t < 10$ minutes, we see linear behaviour, which is the same for both powder and pellet. This can be attributed to the formation of an initial layer of oxide, as with silicon.

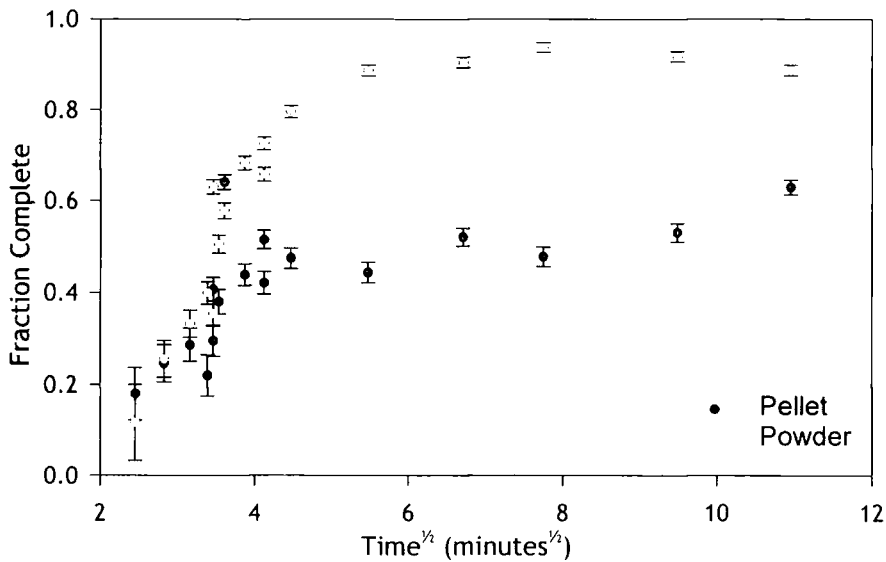


Figure 6.6: The same data as in figure 6.5, presented on different axes to examine whether diffusion limited behaviour can be inferred from the data.

To compare the data to the diffusion-limited regime of the Deal-Grove equation, the same data as in figure 6.5 is shown in figure 6.6 but with t replaced with $t^{1/2}$. This should force the parabolic portion of the curve to

become linear if the transport were diffusion limited (and $\tau \approx 0$, which is quite reasonable). The diagram clearly shows that simple diffusion limited transport is not the limiting factor in oxidation.

One anomaly in the behaviour of the oxidising pellets is the discontinuity in the powder curve at $t \sim 10$ minutes, representing a sudden increase in the oxidation rate. In addition, at this point, the data from the pellet samples deviates from the usual pattern. Although this is perhaps an artefact related to the heating of the sample after insertion into the furnace (see figure 6.2), it is difficult to generate a discontinuity in a model involving the two smooth functions involved, the reaction rate (an Arrhenius relationship) and the temperature-time profile of the form $T = y_0 + a\{1 - \exp(-bt)\}$. Some detailed measurements of the behaviour of the pellets in the region of interest may clarify the mechanism at work, however further investigation of what is nevertheless likely to be an artefact of measurement is beyond the scope of the investigation presented.

6.4 CONCLUSIONS

In this chapter, it has been shown that grain-growth can be observed in pressed pellets, and is encouraged by the presence of CdCl_2 . However, this type of measurement proved unable to predict the sintering mechanism of CdTe, as grain growth was limited by both the grain size of the material used, and oxidation of the CdTe surface on using higher temperatures and longer times.

In oxidation, powder and pellet samples of CdTe have been shown to conform to a two-regime model, the oxidation rate being initially constant before dropping off after longer times. However, classic diffusion-limited

behaviour of the Deal-Grove type was not observed. An anomalous change in reaction rate in the powder samples oxidised for short times could not be explained using a simple Arrhenius model but is probably nonetheless due to heat-up times.

In the following chapter, real CdTe/CdS solar cell structures will be examined for microstructural changes on annealing in comparison to the bulk behaviour characterised in this chapter. This will be performed by optical microscopy, throughout almost the full thickness of the CdTe layer.

6.5 REFERENCES

- [1] D. Rose, D. S. Albin, R. J. Matson, A. B. Swartzlander, X. S. Li, R. G. Dhere, S. Asher, F. S. Hasoon and P. Sheldon *Effects of Oxygen During Close-Space Sublimation of CdTe Solar Cells*, Materials Research Society Symposium Proceedings **426** (1996) 337-48
- [2] "Structure and Properties of Ceramics", Ed. A. Koller (Elsevier, Amsterdam, 1994)
- [3] "Concise Encyclopaedia of Advanced Ceramic Materials", Ed. R. J. Brook (Pergamon Press, Oxford, 1991)
- [4] "Handbook of Chemistry and Physics- 78th Edition", Ed. D. R. Lide (CRC Press, Boca Raton, 1997)
- [5] R. E. Deal and A. S. Grove *General Relationship for the Thermal Oxidation of Silicon*, Journal of Applied Physics **36** (1965) 3770-78
- [6] C. R. M. Grosvenor "Microelectronic Materials- 2nd Edition" (Institute of Physics Publishing, Bristol, 1989)
- [7] D. R. Rhiger and R. E. Kvass *Solid-State Quaternary Phase Diagram for the Hg-Cd-Te-O System*, Journal of Vacuum Science and Technology A **1** (1983) 1712-18
- [8] S. S. Choi and G. Lucovsky *Native Oxide Formation on CdTe*, Journal of Vacuum Science and Technology B **6** (1988) 1198-203

Grain Size and Performance of CdTe/CdS Solar Cells

7.1 INTRODUCTION

The purpose of this chapter is to describe the changes in grain structure of CSS CdTe layers in CdTe/CdS solar cells that occur on annealing with CdCl₂, and to establish whether grain-size itself determines performance. At the time this work was begun, there were conflicting reports on how the CdCl₂ anneal affected the grain structure in CdTe, as discussed previously in section 4.5.3. Significantly, there were no reported studies of grain-growth at the growth surface of CSS CdTe layers.

The work described in this chapter was designed to study the grain size of CSS CdTe absorber layers as a function of the severity of CdCl₂ treatment. This study differs from previous similar studies of CdTe/CdS devices in that the grain structure *throughout the layer* was studied for the first time, rather

than limiting itself to the layer surface. Similar work has been performed recently, using a similar depth profiling method to characterise layers by XRD and RBS [1, 2]. The grain structure information was then correlated with J - V and spectral response measurements for each sample. Analysis of the structural data was also undertaken, using radial and spatial distribution functions to study grain centroid distribution. To the author's knowledge this is the first time that such methods have been used to characterise polycrystalline structure.

As-grown cells were annealed with CdCl_2 for differing amounts of time as described in section 7.2.1, and the device responses measured (section 7.2.2). The structure throughout the layer was then revealed by polishing, bevelling, and selective etching (sections 7.2.3 to 7.2.5)

Optical micrographs were then taken of the surface so revealed, and image analysis performed. The procedure for image analysis, data extraction and manipulation of the data so acquired are described in section 7.3. Results are presented in section 7.4, and discussed in section 7.5

7.2 SAMPLE PREPARATION AND MEASUREMENT

This section describes the preparation of the series of samples used in this study. In order to examine the effect of the CdCl_2 anneal on the grain structure, each sample was annealed for a different amount of time, and then processed to reveal the intra-layer structure. The implementation of this procedure is detailed below. The starting material for the samples is described in section 5.2.

7.2.1 CdCl₂ TREATMENT

The severity of the CdCl₂ treatment was the major variable in this study, and this treatment proceeded as follows. Cadmium chloride was deposited onto the CdTe surface by simple evaporation with a source temperature of 450°C. The thickness of the deposit was monitored using a quartz thickness monitor, giving good reproducibility between samples, although the actual thickness (~60 nm) was known only approximately. The cells were then annealed in a pre-heated tube furnace at 400°C for the desired amount of time. Pre-heating the furnace and inserting the samples directly has the effect of minimising heat-up time. Finally, the samples were rinsed with de-ionised water in an ultrasonic bath. The anneal times used— 20, 30 and 60 minutes— were picked to bracket (and include) the commonly cited optimum treatment time. Another sample was left untreated.

After annealing, the samples were cleaved, with one half being used for *J-V* and spectral response measurements, and the other being used for the grain size analysis.

7.2.2 DEVICE FABRICATION

In order to perform *J-V* and spectral response measurements, the samples needed to be formed into working devices. The back contact was created by first treating the CdTe with a nitric/phosphoric acid mixture to leave a Te-rich layer on the CdTe surface (see section 4.2.1). Approximately 100 nm of gold was then evaporated onto the surface through a mask. The front contact (the ITO) was reached by abrading a small area of the CdTe and CdS. Fine copper lead wires were then attached to both contacts using silver paint. Spectral response and *J-V* could then proceed as outlined in section 5.3.

7.2.3 POLISHING

Polycrystalline CdTe layers have a very uneven surface morphology; the grain boundaries take the form of 'crevasses' up to $\sim 1\ \mu\text{m}$ deep, due to thermal grooving. The grains themselves are also domed and faceted. On bevelling, this structure would propagate, to some extent, throughout the bevel, making imaging of boundaries very difficult. Thus, the surface morphology has to be removed. This is achieved by manually polishing the samples with $0.25\ \mu\text{m}$ -diamond paste on a Leco Pan-W polishing pad. Polishing continues until the surface is featureless when examined at the magnification used for the microscopy ($100\times$ in this case).

7.2.4 BEVELLING

In order to reveal the grain structure throughout the layer, the sample must be bevelled. Bevelling of the CdTe (and CdS) layer was accomplished by immersion as described in section 5.4. Using a 5% bromine, 25% methanol and 70% ethylene glycol mixture, a bevel of $\sim 15\text{mm}$ in length was produced at an immersion speed of $\sim 2\text{mm/minute}$. One side of the sample was masked and left unbevelled to provide a possible height reference surface. The overall structure produced by this process is shown in figure 7.1b. The resulting surface morphology was smooth, flat, and ideal for the purposes here.

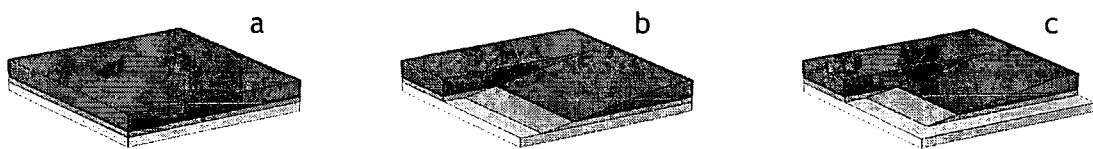


Figure 7.1: Schematic diagram of the steps for creating a sample for grain size analysis: a) the CdTe (upper), CdS and ITO (lower) structure (the glass is not shown); b) the sample after bevelling, and; c) with a reference surface etched off.

7.2.5 DEFECT SELECTIVE ETCHING

In order to reveal the grain boundaries for imaging, a grain-boundary sensitive etch is needed. A number of reagents were screened, the details of which are described in section 5.5, and the most suitable combination was found to be saturated FeCl_3 for the annealed samples, and Inoue's EAg-1 for the as-grown material. With the grain boundaries revealed throughout the layer, the sample was now ready for microscopy.

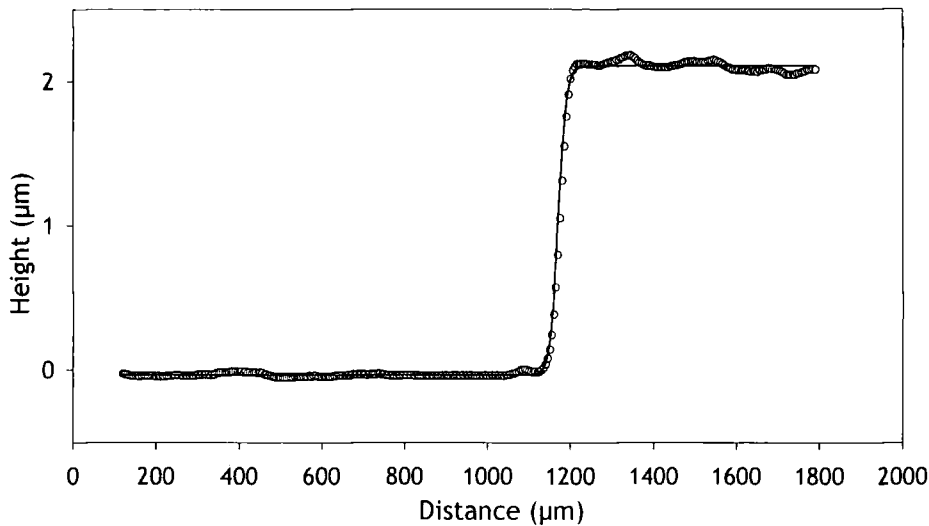


Figure 7.2: A typical stylus profile along the reference surface and onto the bevel. The points are the profilometry data, and the solid line a three-parameter step-function (equation 7.1) fitted to those points.

7.2.6 PROFILOMETRY AND MICROSCOPY

Further to taking micrographs of the grain structure along the bevel, the position in the layer from which the images come was determined. This was achieved by stylus profiling the CdTe bevel with respect to a reference surface. The unbevelled region of the CdTe layer (as shown in figure 7.1b) can be used as a reference surface. However, this surface is not necessarily flat, due to the polishing, and preferential bevelling at the interface of the bevelled and unbevelled region, producing topography difficult to profile. Thus, an

alternative method was used to provide a reference surface: a strip of CdTe and CdS was dissolved using strong Citric Acid/Hydrogen Peroxide solution (8g/100ml), allowing the very flat ITO surface to be used as a reference. This region is shown in figure 7.1c.

Reference marks were scribed at intervals along the bevel, allowing the same area to be both stylus profiled and examined by optical microscopy.

The ASCII files produced by the profilometer (a Tencor Alphastep 200) were analysed by fitting a 4-parameter (y_0 , x_0 , a and b) step function,

$$y = y_0 + \frac{a}{1 + \exp\left\{\frac{-(x - x_0)}{b}\right\}}, \quad (7.1)$$

using the Levenberg-Marquardt non-linear fitting technique. The overall implementation of this method was the same as described by Press *et al.* [3], but a more streamlined and stable version was written for fitting these curves. The source code is reproduced in appendix B. The parameter a can easily be shown to represent the step height. A typical result is shown in figure 7.2. Thus, the height above the CdS/ITO interface was ascertained for each area photographed. The micrographs themselves were taken using a Union Examet optical microscope with a 100× objective. The images were then analysed computationally as described in the following section.

7.3 DATA EXTRACTION AND INTERPRETATION METHODS

7.3.1 DATA EXTRACTION FROM THE IMAGES

In order to extract information concerning grain sizes and positions, the photographic images of the sample surface were processed computationally. The software used to perform this image analysis was specially written for the

task, and the C++ source code is reproduced in appendix A. An overview of the order in which various processing steps are performed is given in figure 7.3, with the specific computational implementation of each step given in section 5.6. The actual images analysed contained a very large number of grains— between 500 and 1500— the figure shows only a very small portion of an image. Applying the matrix filter,

$$\begin{bmatrix} 3 & 3 & 3 \\ 3 & 2 & 3 \\ 3 & 3 & 3 \end{bmatrix}, \quad (7.2)$$

to figure 7.3a produces the blurred image of figure 7.3b. This reduces the level of small-scale detail, but leaves the gross grain structure. Another image filter is then applied (figure 7.3c), which is gradient sensitive:

$$\begin{bmatrix} 8 & 8 & 8 \\ 8 & -1 & 8 \\ 8 & 8 & 8 \end{bmatrix}. \quad (7.3)$$

By flood filling each black region (the boundary plus regions of noise) and measuring it, and then selecting the largest region, figure 7.3d is produced. On skeletonising this image figure 7.3e can be produced, with the grey scaling artificially added to differentiate between grains. As can be seen, the grain boundary width is defined to be zero. Finally, grains located entirely within another grain are deleted, as these are usually an artefact of the processing at stage b). By taking each pixel that lies at the edge of a grain in figure 7.3e, and overlaying these onto the original image, the efficacy of the method can be tested. Figure 7.3f shows this type of image: an excellent correlation is seen. Also shown in this image is the guard-frame used, ensuring the inclusion of only grains that are fully within the image's boundaries and thus giving statistical validity [4]. It is a simple matter to extract grain size and position

data from figure 7.3e. The final output is an ASCII file containing the following for each grain:

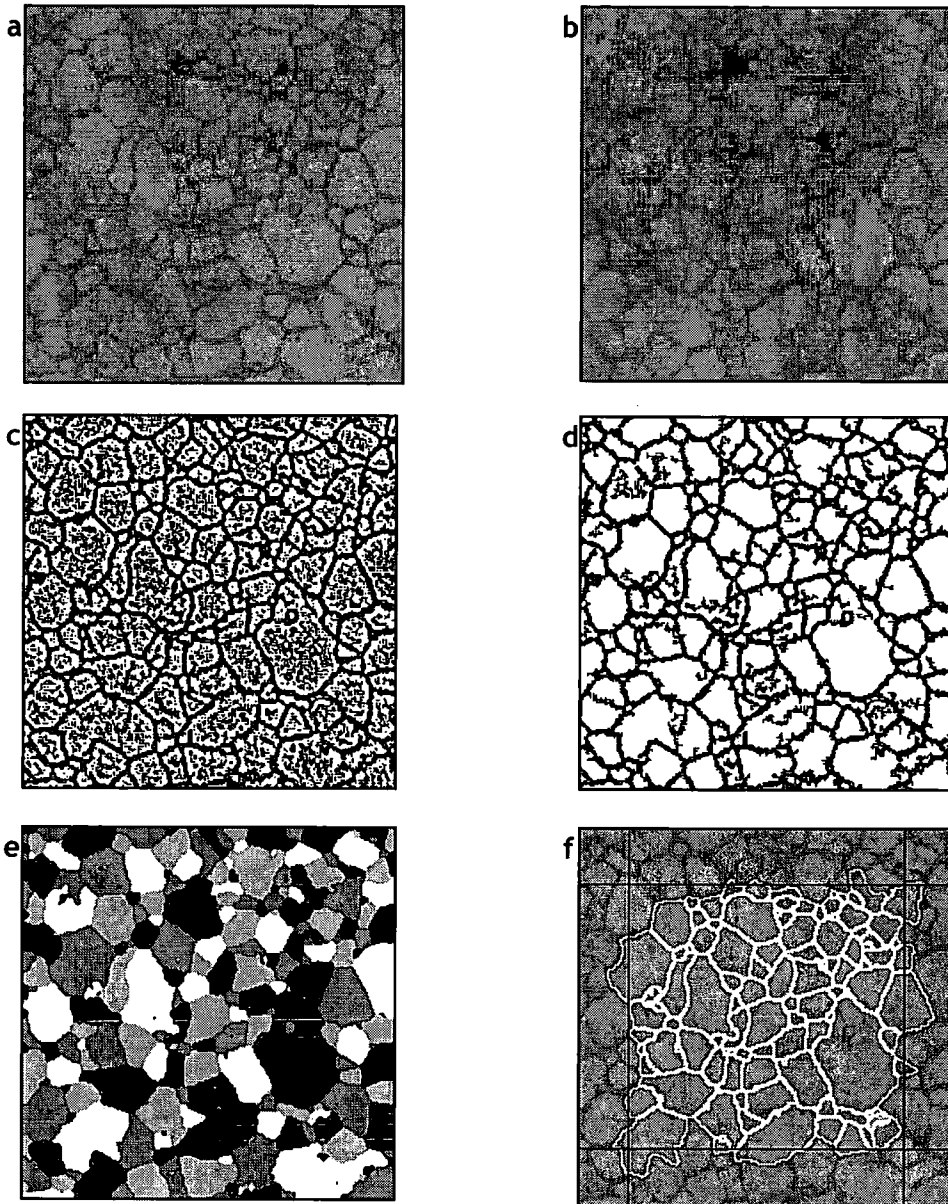


Figure 7.3: The stages used to reduce a grain-boundary image into data. The details of each step are given in the text.

- a) Grain centroid co-ordinates, defined by averaging the x and y coordinates of each pixel contained within the grain;
- b) Grain area, A , the total number of pixels in the grain; and,
- c) Grain size, defined as $\sqrt{A/\pi}$, the mean radius.

Appropriate conversion factors can then be applied to give dimensions in microns.

7.3.2 QUANTITATIVE INTERPRETATION OF GRAIN SIZE AND LOCATION

As well as the obvious measure of the grain size from each image, i.e. the average, histograms of grain radius can also be plotted, and curves fitted to them, again using the Levenberg-Marquardt non-linear fitting technique mentioned in section 7.2.6. The nature of these curves is discussed in the next section.

The coordinate data for the grain centroid position can be used in two further tests that can provide information on the homogeneity of the grains' distribution and also reveal trends in crystallographic relationships between grains.

The Radial Distribution Function is, in essence, the frequency of distances from each grain centroid to every other. It has been used for defect distribution analysis in the past [5], but not for grain centroids. A unit square was used, with the grain centroid positions normalised within it. The experimental frequency curve itself was then normalised to the curve generated from a very large number of random points. In this way, any deviation from unity can be interpreted as a sign of a non-random distribution of grains. Two such distributions are shown in figure 7.4, with the frequency both normalised and not.

The un-normalised plots, 7.4a and 7.4b, also show the theoretical ideal, generated from a random distribution of 10^5 points. In figures a) and c) is shown the type of plot which could be expected from a random distribution of centroids. The deviation at high values of normalised distance is caused by noise arising from the division of the small values of figure a). Figures b) and

d), for a hexagonal array also show this distribution. It is also very clear from the plots that the regularity of the distribution from which the plots are derived cause the generation of a large amount of structure in the radial distribution function.

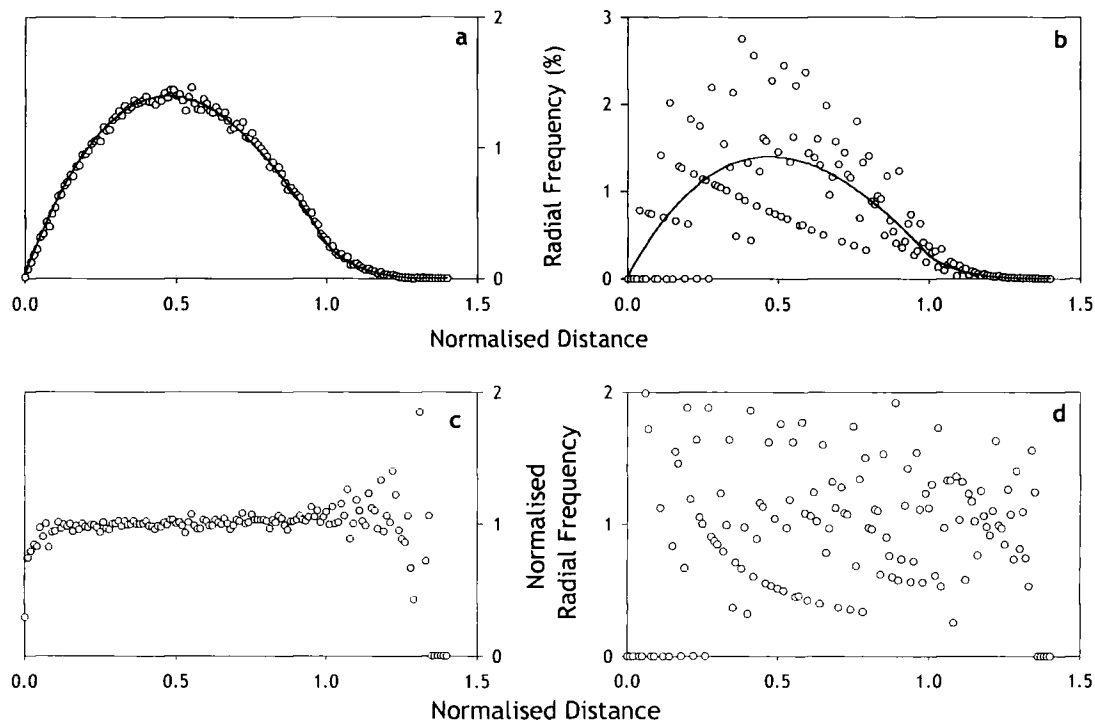


Figure 7.4: The first two figures show the un-normalised radial distribution functions (points), and the theoretical function required to normalise them (solid line). The first, a), shows the distribution arising from a random distribution, and b), that arising from a hexagonal array. Figures c) and d) show the same functions after normalisation.

The Spatial Distribution Function is a modified autocorrelation function and can be used to study any directional relationships in a set of points. In addition, it can supply information on the extent to which a grain structure has approached equilibrium. The plot is constructed thus: the vectors between each grain centroid are translated to the origin of the plot, with the absolute value of the y -direction being used. The frequency with

which a particular vector occurs can then be represented as a grey-scale value on a graph. Two such plots are shown in figure 7.5. The first (figure 7.5a) shows the distribution expected from a purely random distribution. The second (figure 7.5b) was generated from a hexagonal distribution, with a small Gaussian deviate added (using the Box-Muller function [3]). A hexagonal distribution is of interest, as this would be the distribution for an ideal grain-structure at equilibrium, as described in section 3.3. For example, the average distance between hexagonal cells is represented by the minimum spot separation, while the angle of 60° between the principal axes of the figure represent the hexagonal packing. The importance of this hexagonal distribution in grain studies is discussed below, with reference to the spatial correlation function. Thus, any sample displaying a spatial distribution like this can be assumed to be in a state near to equilibrium.

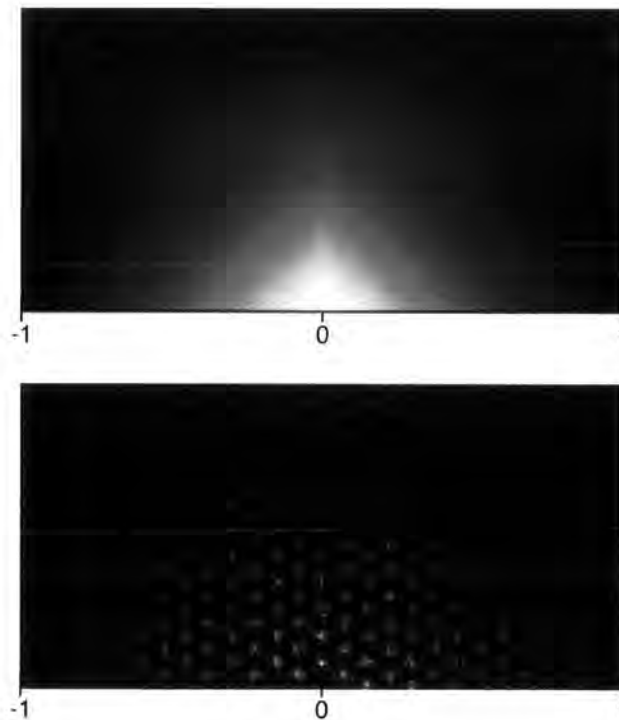


Figure 7.5: Spatial distribution (or modified autocorrelation) functions describing a) a purely random distribution of points, and b) an artificially created a hexagonal distribution of points.

7.4 RESULTS

7.4.1 SPECTRAL RESPONSE AND J - V

The J - V and spectral response curves for the different samples are shown in figures 7.6 and 7.7. From the J - V curves, and the data extracted from them (table 7.I) it is clear that CdCl_2 has the effect of improving each of the relevant cell parameters; J_{sc} , V_{oc} , FF , and η . The cell with the maximum efficiency, though, is that treated for 20 minutes, rather than the cell treated for 30 minutes, which is often considered the optimal time for CSS material. This is not necessarily significant, though, as the method of treatment used here is not standardised with those used by other authors. Differences in the procedure include:

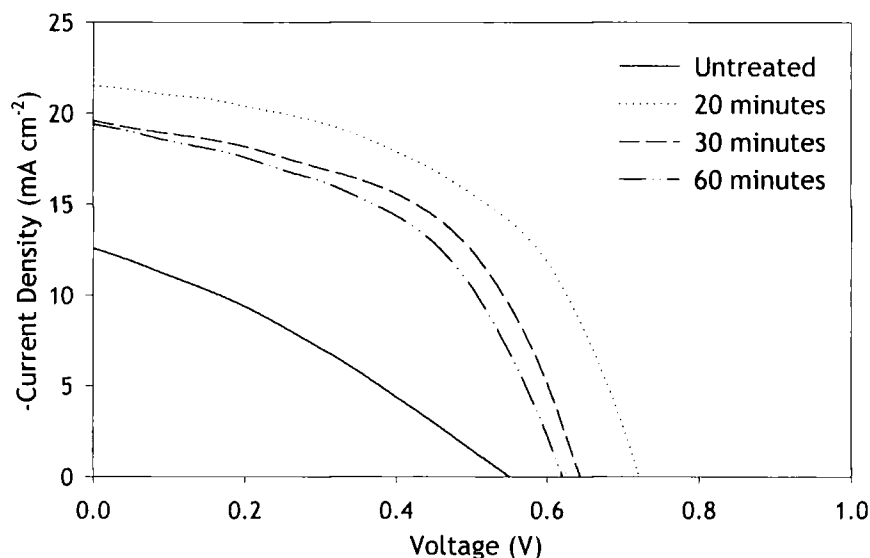


Figure 7.6: The AM 1.5 I-V curves for CdTe/CdS cells annealed in the presence of CdCl_2 for differing amounts of time.

- a) The CdCl_2 treatment: whilst evaporation of a solid layer has been used widely in the past, it is increasingly common in commercial processes to use a treatment in which the cells are annealed in the presence of



vapour phase CdCl_2 (for example see McCandless *et al.* [6]). Whilst it is accepted that chlorine will always influence efficiency on annealing regardless of the method of introduction [7], it is unclear as to how the specifics of the method modify the annealing process.

- b) The surroundings: the samples described here were annealed in a tube furnace, open to the atmosphere. It would seem likely that this would cause CdCl_2 vapour to be transported away from the CdTe surface on heating. An anneal in CdCl_2 vapour, as mentioned above, is obviously a closed system, and other reported treatments using solid CdCl_2 may also use such a system. This will allow better monitoring of the amount of chloride introduced.

Treatment time (mins)	0	20	30	60
Efficiency (%)	2.1	7.9	6.7	6.0
Fill Factor (%)	30.8	51.2	53.0	49.7
V_{oc} (V)	0.55	0.72	0.64	0.62
J_{sc} (mA cm^{-2})	12.6	21.5	19.6	19.4

Table 7.1 The AM 1.5 electrical characteristics of the CdTe/CdS solar cells for different treatment times. The data in this table was extracted from figure 7.6.

- c) The temperature. Although consistent between cells, the temperature was not known to great accuracy. Even small variations of temperature have been shown to greatly change the speed at which microstructural effects occur in polycrystalline CdTe treated with CdCl_2 . (for example see Moutinho *et al.* [8]).

The spectral response curves shown in figure 7.7 show an increase in efficiency corresponding to that seen in the J - V data. The short wavelength end was examined for any signs of intermixing— detectable as a red-shift due

to band-gap bowing (see section 4.4)— however, no statistically significant variation was detectable within the limits of the equipment’s sensitivity.

Thus, from the above results, it is clear that the CdCl₂ anneal effects the expected improvement in device performance for the CdTe/CdS solar cells treated for this study, which can be correlated, or otherwise, with any increase in grain size or morphological change.

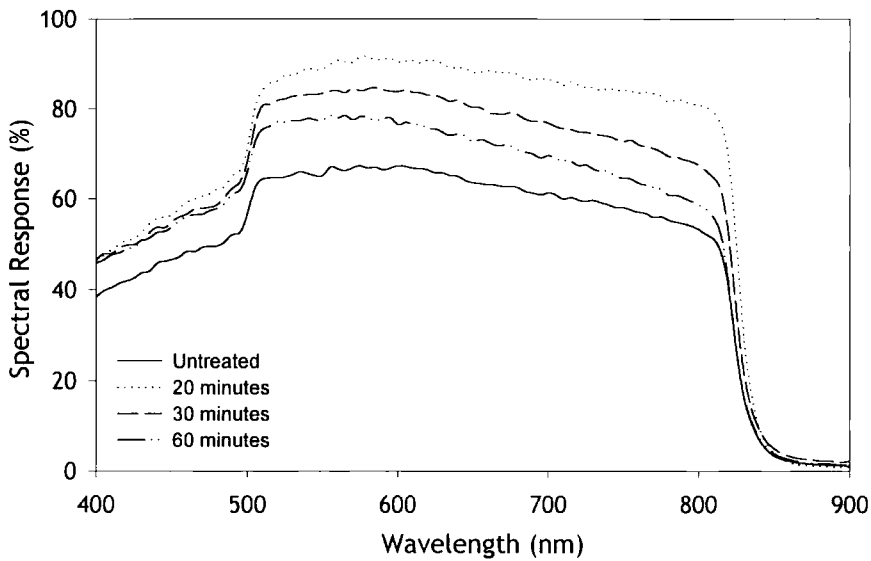


Figure 7.7: The spectral response curves for the samples examined in this chapter.

7.4.2 GRAIN SIZE DISTRIBUTIONS

As described above, the samples were polished, bevelled, and etched, then photo-micrographs were taken at 1mm intervals along the bevel (corresponding to ~0.5 μm through the layer), and analysed computationally. Shown in figure 7.8 is a series of micrographs from different positions within the untreated CdTe film. These images constitute small regions cropped from the actual micrographs analysed: the full micrographs contained between 500-1500 grains.

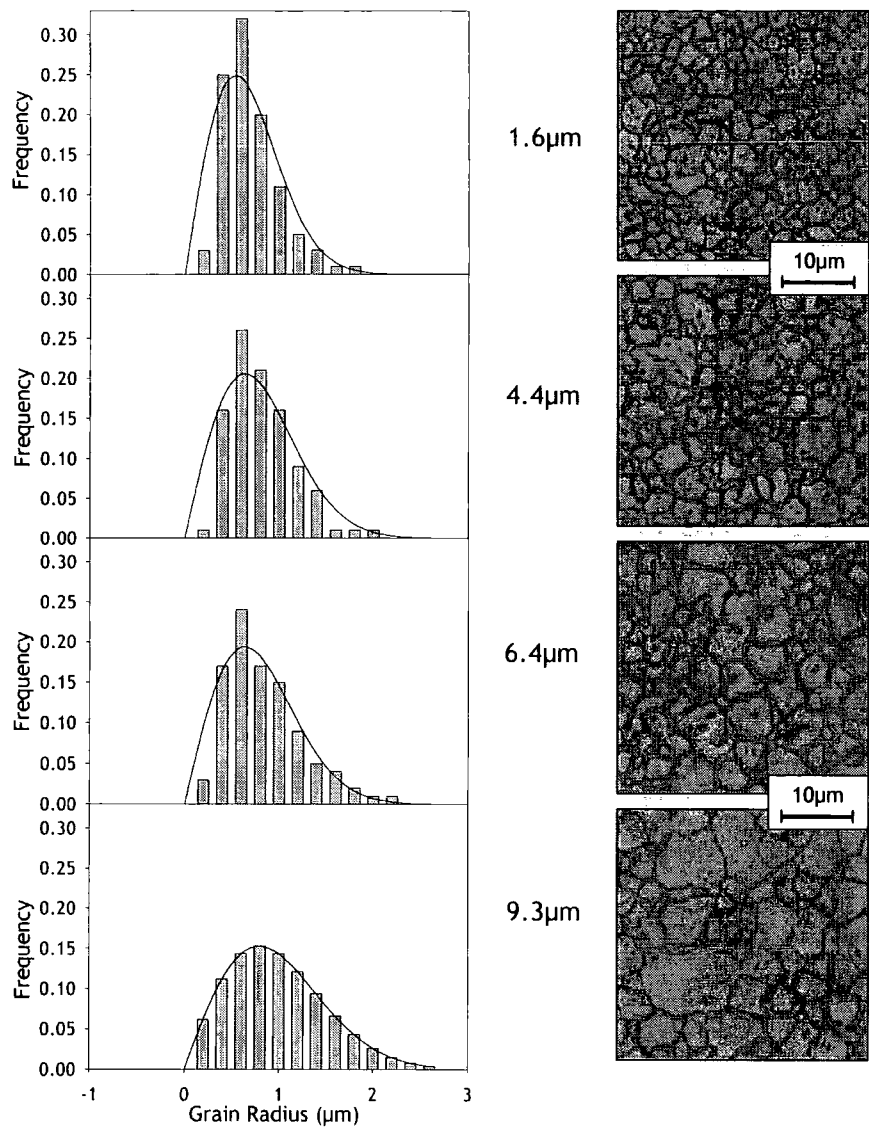


Figure 7.8: Small sections of some of the micrographs of the untreated material, and the accompanying grain size distributions (produced from the full image) and Rayleigh fits (solid lines). The numbers down the centre represent the distance from the CdS/ITO interface. The grain size can be seen becoming visibly larger, confirmed by the histograms.

The corresponding grain size distributions are also shown in this figure. From it, it can be seen that the grain size increases as the distance from the CdTe/CdS interface increases. In addition, we see that the distribution of grain sizes becomes wider as the layer thickness increases. These observations are discussed in section 7.5. In addition, to confirm the columnar nature of the CdTe layer, an untreated sample was cleaved and

examined in cross-section by SEM: the resulting image (figure 7.9) shows columnar behaviour.

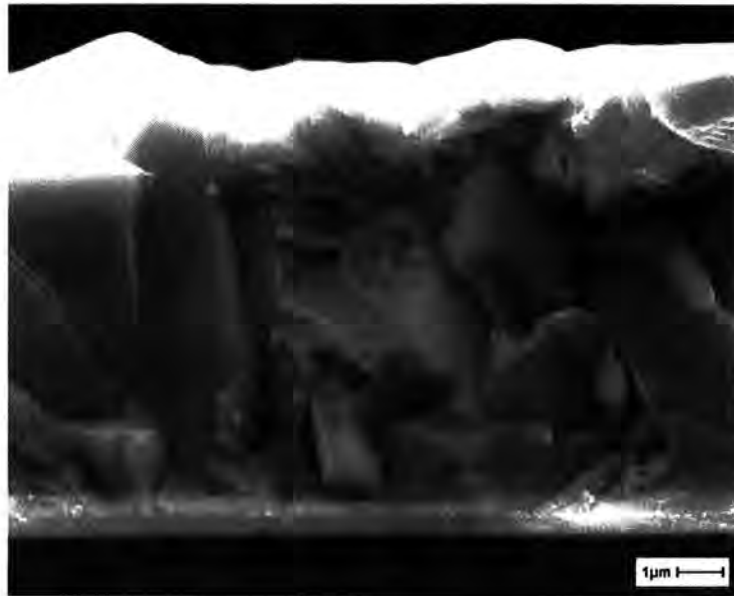


Figure 7.9: SEM cross-section of an untreated CSS-grown CdTe/CdS solar cell. The grains are clearly columnar in nature.

Grain size distributions, such as those shown in figure 7.8 can be modelled by a number of different distributions, discussed presently. It is worth mentioning, though, that there are other distributions that fit real grain sizes considerably better than those discussed here (for example the Aboav-Langdon model [9], or the gamma distribution [10]). These equations have, though, no physical basis, and the goodness-of-fit they show can be considered simply fortuitous. The equations described below, on the other hand, are derived from physical, theoretical bases— often being solutions of differential equations, describing grain size evolution over time (see section 3.3.2)— and as such their usage is justified. Three popular distributions were studied for goodness-of-fit to the experimental distributions, and are detailed below.

Hillert's Distribution: This model is based upon the assumption that grain-boundary velocity is inversely proportional to the radius of curvature of the grain [11]. It has the form,

$$f(r) = \frac{\beta R (2a)^\beta}{(2 - R)^{2+\beta}} \exp\left(\frac{-2\beta}{2 - R}\right), \quad (7.4)$$

where the radius, r , is represented as,

$$R = \frac{r}{r_0}. \quad (7.5)$$

The dimensionality of the distribution, β , is equal to either 2 or 3, and r_0 and a are constants.

The Rayleigh Distribution: This distribution arises out of Louat's analysis of grain boundary dynamics based on the random diffusion of sections of grain boundaries [12]:

$$f(r) = ar \exp\left\{-\left(\frac{r}{r_0}\right)^2\right\}. \quad (7.6)$$

The parameters a and r_0 are again constants.

Log-Normal Distribution: This distribution differs from the above two cases in that it is a starting assumption of Feltham's model of grain growth [13], and not a solution to a grain-growth equation. In the form used by Feltham the log-normal distribution is,

$$f(r) = \frac{a}{r_0} \exp\left\{-\frac{[\ln(r/r_0)]^2}{2\sigma^2}\right\}, \quad (7.7)$$

where the parameters a , r_0 and σ are constants.

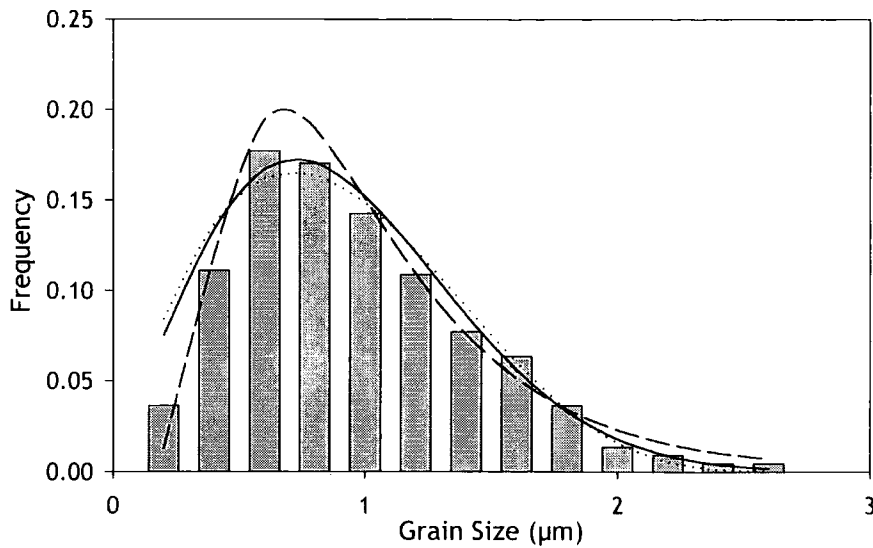


Figure 7.10: Histogram of actual grain size data showing three different types of fit used to describe such distributions; Rayleigh (solid line), Hillert (dotted), and log-normal (dashed). The Rayleigh curve provides statistically the best fit in the great majority of cases.

Figure 7.10 shows the three distributions fitted to actual data. In the majority cases it was found that the Rayleigh curve gave the best fit to the data, although the log-normal provided good fits in many cases. This is, perhaps, unsurprising as the equation contains three, rather than two variable parameters. However, the Rayleigh distribution was used in all cases to describe grain-size distributions as the form is simpler, and it is constrained to pass through the origin, which better reflects real distributions. The use of such curves to describe grain size distributions is discussed further in section 7.5.

7.4.3 THE AVERAGE GRAIN SIZE AS A FUNCTION OF DEPTH

If there is grain growth upon annealing at any one point within one of the CdTe layers studied, it should be detectable as a change in the average grain size. Thus, to detect any grain growth in the layer on CdCl₂ treatment, a plot of average grain size from each image, against the position in the layer at

which that image was taken, was made. This is shown in figure 7.11 for all of the samples. Each data point represents a considerable volume of data: one point represents one image analysed, each in turn containing 500-1500 grains. As can be seen, there is *no change in average grain size regardless of the treatment regime*. Also of note is the linearity of the data distribution, the points lying on the line $y = (0.107 \times x) + 1.06$, for distances expressed in microns.

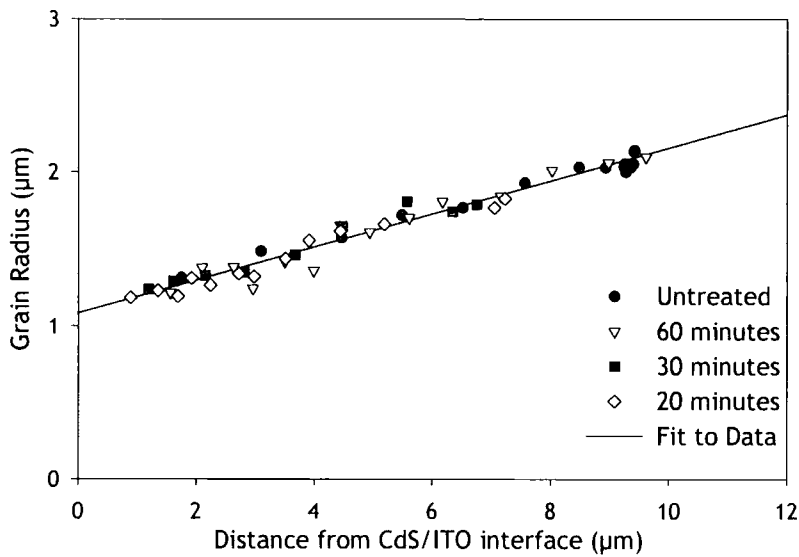


Figure 7.11: The mean grain radius as a function of position through CdTe layers subjected to different CdCl₂ anneal times. The way in which the value of each data point is calculated is described in section 7.4.3.

An alternative treatment of the data is to examine trends in the parameters extracted from the Rayleigh distribution fits, rather than from the numerical mean of the grain size. Figure 7.12 shows these parameters for the distributions here. The values are calculated from the standard integrals, and are given by,

$$\langle r \rangle = \frac{\sqrt{\pi}}{2} r_0, \quad (7.8)$$

and,

$$\sigma = r_0 \sqrt{1 - \frac{\pi}{4}}, \quad (7.9)$$

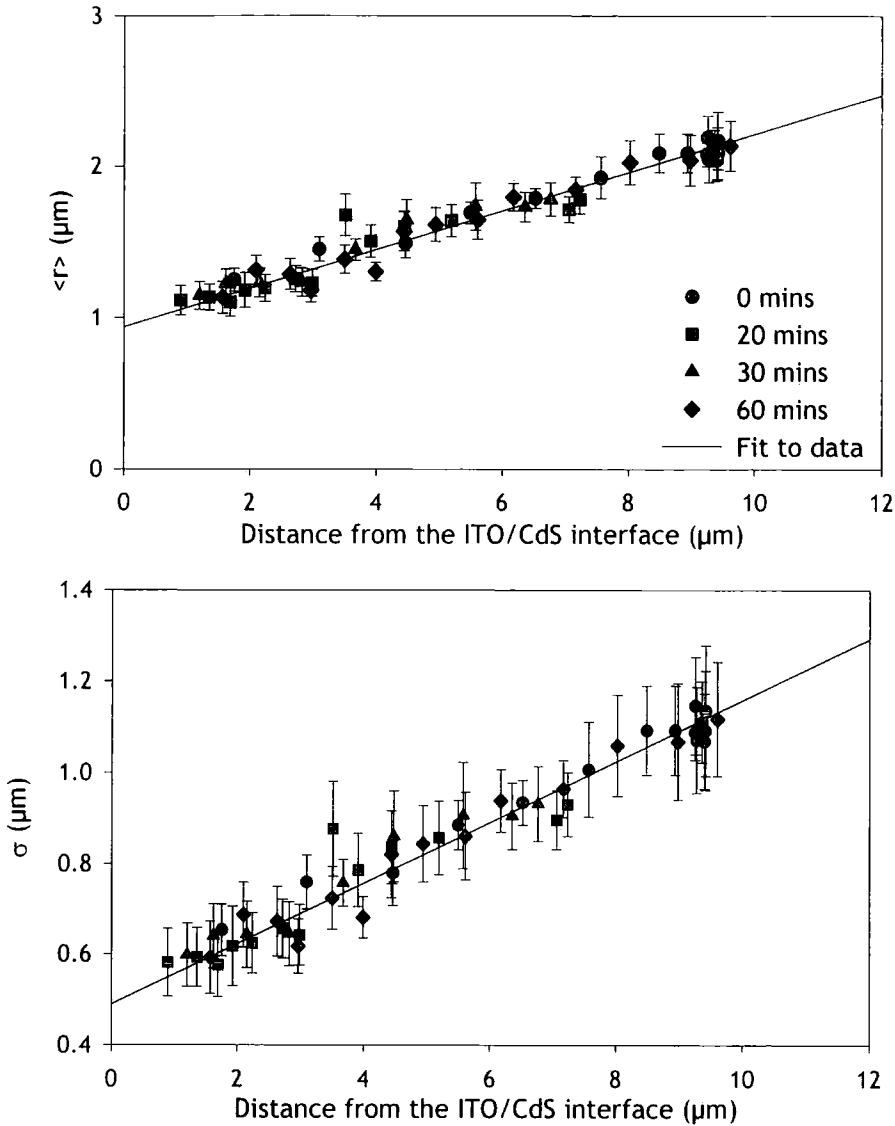


Figure 7.12: The average value and standard deviation of the Rayleigh fits to grain size distributions. There is no appreciable difference between the treated and untreated material. The error bars are the variances produced by the fitting procedure.

As can be seen, no grain-growth is detectable here either. Thus, within the limits of sensitivity of this experiment there is *no grain growth* in CSS CdTe layers on CdCl₂ treatment. However, the optical microscope used in this

experiment did not have sufficient resolution to study the smallest grained regions in the $h \leq 1\mu\text{m}$ region. Structural information from this region is presented in the following chapter.

7.4.4 SPATIAL ANALYSIS

All of the images taken (around 50 in total) were analysed using spatial and radial correlation functions and typical examples are shown in figure 7.13. However, none of these showed behaviour indicative of a specific distribution, being most similar to the random cases of figures 7.4c and 7.5a. Possible reasons for this will be discussed in the following section

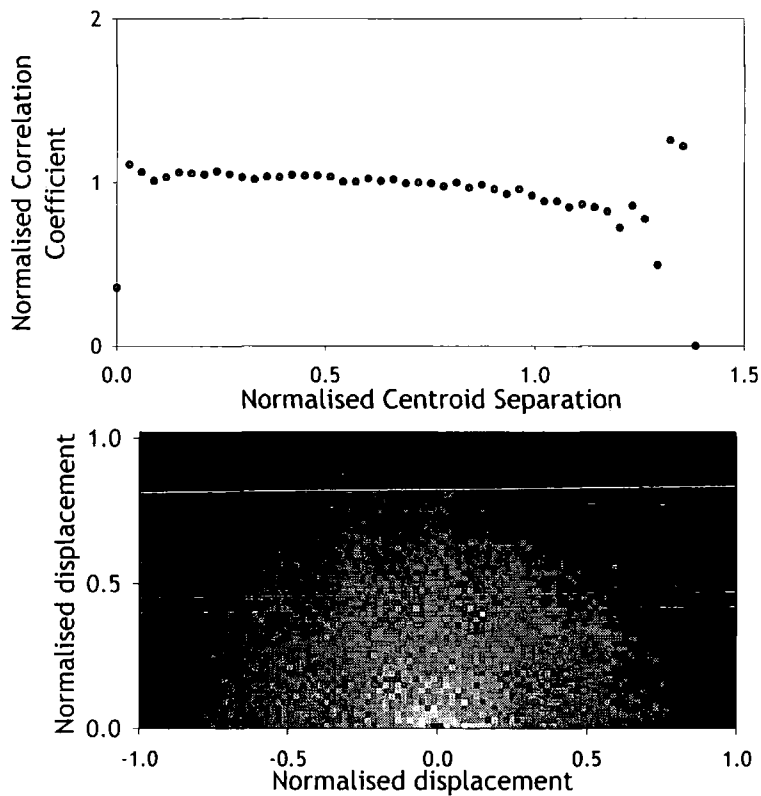


Figure 7.13: Typical radial and spatial distribution plots for the samples examined here. There is no evidence of the hexagonal pattern associated with full equilibration, both plots showing displaying the characteristics of a random distribution of centroids.

7.5 DISCUSSION

The grain-boundary distributions from the images (described in section 7.4.2) were found to be well fitted by the Rayleigh distribution, regardless of whereabouts in the layer the image came from. This distribution is widely reported to give the best fit to both real distributions of data [12] and Monte Carlo simulated data [14]. However, in fitting this type of distribution to real data, a number of points are of note.

- a) In the case of both Hillert's and the Rayleigh distribution, the grain size distributions are solutions of differential equations describing the growth rate of grains, and are consistent with the parabolic growth predicted by the model of Burke and Turnbull (see section 3.3.2) for normal grain growth. However, real systems rarely show this parabolic growth behaviour (see Atkinson [15] for a review). Moreover, describing a system in which the grains have clearly undergone some movement towards equilibrium (hence their lack of growth on CdCl₂ treatment) using an equation derived from models of grain growth itself should be examined critically.
- b) The models all assume isotropy of the distribution in three dimensions, but it is generally observed that the methods of CdTe growth used for fabricating solar cells give columnar grains.
- c) It could be thought that the goodness of fit of a particular curve could imply that the assumptions made in formulating the model were correct. It has been shown, though, that small adjustments to the grain-growth model can cause large changes to the predicted distribution, meaning that the model should not be used to confirm the mode of grain growth.

However, although the applicability of the Rayleigh curve may be thought of as 'un-physical' for the above reasons, it is a useful tool to describe the grain size distributions using only two parameters, and is used as such throughout this work.

In the results of section 7.4.3, it was shown that the CdCl_2 treatment caused no grain-growth in the CdTe films' bulk regions. However, section 7.4.1 showed the expected improvement in the cells' electrical characteristics. Thus, it can be inferred that grain-growth in the bulk of the layer is not a prerequisite for an enhancement of efficiency. It should be noted, however, that this method of grain size analysis is limited by the resolution of the optical microscope used, and so could not provide any information in the region closest to the CdS layer where the grain size is small. It is in this region, of course, that most of the photo-electronic interaction takes place, and so any grain growth in this interface region would be most significant. The following chapter (chapter 8) examines this region by TEM.

Although grain-growth in the bulk is clearly unnecessary to produce efficient cells, it must be asked why the CSS-grown material used here should be stable under CdCl_2 treatment, whereas material throughout the body of literature on CdTe/CdS cells displays grain growth. The reasons for the lack of grain growth in the region studied may be one of the following:

- a) Although heat and the CdCl_2 flux promote the expansion of grains, the driving force for this growth is the minimisation of the grain-boundary energy. There will come a point, though, where the heat supplied is no longer large enough to overcome the activation energy for this process even with the addition of a flux. McCandless *et al.* [16] reported this

phenomenon in small-grained material: without CdCl_2 , they saw a limiting grain size on annealing a sample. However, the addition of a CdCl_2 flux causes a large increase in the grain size with the same heat treatment.

- b) Whether the CdCl_2 , concentrated at the grain boundaries, causes solute pinning of these boundaries is unclear, but it is known that CdCl_2 has a very limited solubility in CdTe [17].

Thus, we can say that the CSS-grown material studied here has, to some extent, moved towards equilibrium during the growth process, which takes place at $\sim 500^\circ\text{C}$. This is in contrast to the small grained material grown at $\sim 300^\circ\text{C}$ and studied elsewhere. However, the radial and spatial distribution plots of section 7.4.4 show no sign of having actually reached equilibrium: the patterns from the data conform to the random case (figure 7.5a and 7.4c) rather than the hexagonal (7.5b and 7.4d). This is possibly because, like most real polycrystalline materials, the structure does not approach the theoretical hexagonal distribution due to pinning effects described in section 3.3.

Concerning the variation of grain size with distance from the CdTe/CdS interface, which shows linear variation, a number of factors need to be considered. These will be discussed in some detail below.

We first consider a kinetically limited model. One method by which the width of the columnar grain could increase would be if particular grains were to grow faster than others do. Some evidence for this is found in films deposited at low temperature. These usually exhibit $\{111\}$ preferred orientation, although the level of preference of crystallographic polarity has never been ascertained. It is also known that the $\{\bar{1}\bar{1}\bar{1}\}$ B face is the fastest growing in CdTe [18]. If, then, we consider the growth process to be

kinetically limited, with the growth rate dictated by the speed at which species can be removed from the surrounding atmosphere, then it may be the case that a $\{111\}$ oriented grain will grow faster than other orientations. This difference in growth rate will cause slower growing grains to be crowded out, and the surface to become populated by a progressively larger number of $\{111\}$ grains of progressively larger size. This type of behaviour has been reported: McCandless *et al.* [16] report that the surface grain size of predominantly $\{111\}$ oriented PVD films increases as the films are grown to greater thicknesses. This process exclusively, of course, can only cause an increase in the width of a columnar grain if there are other orientations. Once the film surface is entirely $\{111\}$, lateral grain expansion will cease.

A further mechanism which may occur in material which has become almost purely $\{111\}$ oriented at the growth surface is a variation of the growth rate *perpendicular* to the growth direction and the $\{111\}$ plane. It is the case that different faces of CdTe have different growth rates [19]. Thus if the plane which describes the side of one grain, abuts another slower growing plane of a neighbouring grain, it could be supposed that the grain with the faster growing plane will expand in that direction. Of course, whether this expansion leads to an increase in the size of this grain will depend on the motion of the other edges. This process and that detailed above, it should be remembered, describe processes that pertain directly to small-grained, low deposition temperature material. How such mechanisms could be applied to the randomly oriented CSS-grown CdTe studied here is unclear.

Another mechanism, perhaps more applicable to high growth temperature grains, involves a process analogous to classic grain growth. There is evidence that at temperatures around the CSS growth temperature, CdTe can restructure: Cho *et al.* [20] report recrystallisation of small-grained

CdTe films on air annealing at 550°C. This being the case, one could speculate that this temperature is sufficient for species to migrate between grains, as in grain-growth. If this is the case, an in-situ grain-growth process could occur whilst the CdTe layers are growing, in an attempt to reduce the grain boundary length to grain area ratio (in a 2-D analogue of the classical process). Pinning of the CdTe grain boundaries at the CdS interface should prevent the bottom of the grains from expanding, thus leading to the observed structure. This argument is supported by the increase in surface grain size seen on increasing the growth temperature [21]. It is reasonable to assume that higher temperatures will increase mobilities of diffusing species in the layers, enhancing the expansion of grains. That pinning occurs is also supported by the significant effect of the underlying CdS on the final surface structure of the CdTe [21, 22].

Overall, the above mechanisms propounded for the variation of grain size throughout the layer are speculative and considerable further work would have to be undertaken to pinpoint the exact mechanism. However, in chapter 9.2, a computer model is presented, which goes some way to predicting the behaviour seen in this material.

7.6 CONCLUSIONS

In this chapter, a method of examining the grain structure throughout the bulk of the CdTe layer in CSS-grown CdTe/CdS thin-film solar cells has been detailed. This was accomplished by polishing and bevelling of the CdTe film, followed by selective etching of the grain boundaries. This was performed on a series of samples that had been subjected to varying CdCl₂ treatments. To characterise the structure, micrographs were taken at

incremental points along the bevel, and stylus profilometry used to ascertain the points' position in the film. The resolution of the microscope was sufficient only to resolve grains more than 1 μm away from the CdTe/CdS interface. The images so obtained were then analysed using computational image processing methods giving data on grain size distributions and radial and spatial correlation functions of grain centroids. This is the first report of the latter method being applied to polycrystalline material in this way. SEM characterisation was used to examine the structure of the films in the direction perpendicular to the growth direction. Further to this structural investigation, the samples were characterised opto-electrically using J - V and spectral response measurements.

This investigation showed that the mean grain-size displays a linear variation with position in the CdTe layer, the size becoming progressively larger as the position in the layer studied becomes further from the CdTe/CdS interface. However, this variation was found to be *independent* of the degree of CdCl_2 treatment: the samples were identical within the limits of the experiment. This variation was quantified, and found to vary as $y = (0.107 \times x) + 1.06$. This lack of microstructural evolution notwithstanding, the J - V and spectral response measurements showed the expected increase in efficiency and related parameters. This work also revealed some fine details of the grain size and centroid distribution: at each point in each layer studied, the grain size was found to be well described by the Rayleigh distribution. The radial and spatial correlation analysis of the grain centroids revealed a random distribution of the grains. This is consistent with non-equilibrated grain structure.

In the discussion section of this chapter, it was suggested that two mechanisms might cause improvement in electrical characteristics found even

in the absence of grain growth. Firstly, changes in the electrical properties of the bulk grain boundaries may be responsible, and secondly, near-interface effects of a structural, electrical or optical nature. The latter of these effects is important as this region contains the metallurgical interface, and will be studied in the following chapter. The electrical activity of the boundaries in the bulk cannot be commented on within the context of these results. Further to this discussion, reports by other authors of measurable grain-growth effects on CdCl_2 treatment were considered. It was noted that these reports exclusively concern material grown at temperatures lower than the CdCl_2 treatment temperature, whereas CSS involves temperatures higher than this. Thus the absence of grain growth in the present samples is clearly due to the CSS-grown material being brought closer to equilibrium during growth, although radial and spatial correlation analysis revealed no classical indications of this. This implies the presence of a retarding force acting to reduce the grain-boundary mobility.

The variation of the grain size was discussed in this chapter with respect to a number of different effects. Pinning of the grain boundaries at the CdTe/CdS interface was cited to account for the smaller grain size closer to the interface, although this cannot account for the grain size's linear variation. The linear variation was considered in terms of how growth kinetics associated with the growth rates of different crystallographic faces could affect film growth. However, whilst this could, perhaps, account for behaviour in material that displays preferred crystallographic orientation of grains, CSS-grown material shows no such preference.

Thus, whilst this work shows effects distinct from those reported in the literature, these differences, in the main, can be attributed to the well-established metallurgical considerations arising from the differences in growth

conditions. In addition, whilst the variation of grain size is attributable to pinning at the CdTe/CdS interface, the reason for the *linear* nature of this variation was unclear. This is explored further in chapter 9. Clearly, however, near interface effects may play a significant role in clarifying the results of this chapter, and the following chapter describes a TEM study of this region.

7.7 REFERENCES

- [1] K. D. Rogers, D. A. Wood, J. D. Painter, D. W. Lane and M. E. Ozsan *Novel Depth Profiling in CdS-CdTe Thin Films*, Thin Solid Films **361-362** (2000) 234-38
- [2] K. D. Rogers, J. D. Painter, M. J. Healy, D. W. Lane and M. E. Ozsan *The Crystal Structure of CdS-CdTe Thin Film Heterojunction Solar Cells*, Thin Solid Films **339** (1999) 299-304
- [3] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery "Numerical Recipes in C: The Art of Scientific Computing- 2nd Edition" (Cambridge University Press, Cambridge, 1992)
- [4] R. Miles *The Sampling, by Quadrats, of Planar Aggregates*, J. Microsc. **113** (1978) 257-67
- [5] D. Rose, K. Durose, W. Palosz, A. Szczerbakow and K. Graszka *Methods of Dislocation Distribution Analysis and Inclusion Identification with Application to CdTe and (Cd,Zn)Te*, Journal of Physics D **31** (1998) 1009-16
- [6] B. E. McCandless, I. Youm and R. W. Birkmire *Optimization of Vapor Post-Deposition Processing for Evaporated Cdscdte Solar Cells*, Progress in Photovoltaics **7** (1999) 21-30
- [7] P. R. Edwards, K. Durose, J. Beier, M. Campo and D. Bonnet *A Comparative Study of CdTe Solar Cell Activation Using CdCl₂ and Cl*, Proceedings of the 16th European Photovoltaic Solar Energy Conference (2000) *In Press*
- [8] H. R. Moutinho, M. M. Al-Jassim, F. A. Abulfotuh, D. H. Levi, P. C. Dippo, R. G. Dhere and L. L. Kazmerski *Studies of Recrystallisation of CdTe Thin Films after CdCl₂ Treatment*, Proceedings of the 26th IEEE Photovoltaic Specialists Conference (1997) 431-34
- [9] N. P. Louat, M. S. Duesbery and K. Sadananda *On the Role of Random Walk in Grain Growth* in "Grain Growth in Polycrystalline Materials", Eds. G. Abbruzzese and P. Brozzo (Trans Tech Publications, Zurich, 1992)
- [10] M. Evans, N. Hastings and B. Peacock "Statistical Distributions" (Wiley, New York, 1993)
- [11] M. Hillert *On the Theory of Normal and Abnormal Grain Growth*, Acta Metallurgica **13** (1965) 227-38
- [12] N. P. Louat *On the Theory of Normal Grain Growth*, Acta Metallurgica **22** (1974) 721-24
- [13] P. Feltham *Grain Growth in Metals*, Acta Metallurgica **5** (1957) 97-105
- [14] M. P. Anderson, G. S. Grest and D. J. Srolovitz *Computer Simulation of Normal Grain-Growth in 3 Dimensions*, Philosophical Magazine B **59** (1989) 293-329
- [15] H. V. Atkinson *Theories of Normal Grain-Growth in Pure Single-Phase Systems*, Acta Metallurgica **36** (1988) 469-91

- [16] B. E. McCandless, L. V. Moulton and R. W. Birkmire *Recrystallization and Sulfur Diffusion in CdCl₂-Treated CdTe/CdS Thin Films*, Progress in Photovoltaics **5** (1997) 249-60
- [17] I. K. Andronik, Z. P. Kuleva and K. D. Sushkevich *Feasibility of Obtaining CdTe Crystals by Zone Refining from a Solution in Fused CdCl₂*, Inorganic Materials **12** (1976) 759-60
- [18] I. Teramoto and M. Inoue *Vapour Growth of Cadmium Telluride Crystals in the <111> Polar Directions*, Philosophical Magazine B **8** (1963) 1593-96
- [19] D. W. Snyder, E. I. Ko, S. Mahajan and P. J. Sides *Effect of Substrate Orientation on CdTe Film Grown by OMVPE*, Materials Research Society Symposium Proceedings **216** (1991) 41-46
- [20] Y. A. Cho, W. J. Nam, H. S. Kim, G. Y. Yeom, J. K. Yoon, K. H. Oho, S. H. Shin and K. J. Park *Effects of Rapid Thermal Annealing on CdTe/CdS Solar-Cell Fabrication*, Thin Films For Photovoltaic and Related Device Applications **426** (1996) 379-84
- [21] A. L. Fahrenbruch, V. Vasilchenko, F. Buch, K. Mitchell and R. H. Bube *II-VI Photovoltaic Heterojunctions for Solar Energy Conversion*, Applied Physics Letters **25** (1974) 605-08
- [22] A. Romeo, D. L. Batzner, H. Zogg, A. N. Tiwari and C. Vignali *Influence of CdS Growth Process on Structural and Photovoltaic Properties of CdTe/CdS Solar Cells*, Solar Energy Materials and Solar Cells **58** (1999) 209-18

TEM of the Near-Interface Region

8.1 INTRODUCTION

In the previous chapter, the grain structure of the CdTe layer in CdTe/CdS solar cells was examined using optical microscopy. However, it was found that the grain size became too small to resolve in the ‘near-interface’ region (defined here as to within 1 μm of the CdS/CdTe interface). As this is the region where both the metallurgical and electrical junctions lie, it could be expected that the material properties here have a large influence on device performance. Transmission Electron Microscopy (TEM) is a high-resolution technique, easily capable of resolving features on the scale of the near-interface grains. It has previously been used in several studies of microstructure in CdTe/CdS solar cells, both in cross-section [1-7] and plan-view [2, 7]. However, none of these studies have reported the grain size distribution and intra-grain microstructure before and after processing.

This chapter describes a plan-view TEM study of the near-interface region of CdTe/CdS solar cells, in both as-grown and CdCl₂ treated material, with a view to quantifying the grain size in this region and the effect that CdCl₂ treatment has upon it. In addition, the microstructure in this region was qualitatively examined for differences arising from the annealing procedure.

8.2 EXPERIMENTAL DETAILS

The CdTe/CdS solar cells used in this work were grown by CSS at 500°C by ANTEC GmbH, who supplied both as-grown and CdCl₂ treated material for this work. The CdCl₂ was introduced into the cell by a vapour-phase annealing technique (see section 4.4).

The preparation of samples for plan-view microscopy involves: a grinding and etching stage to remove the glass from a CdTe/CdS/ITO multiplayer; mounting on a ring grid; and, thinning with I⁺ ions. This latter step removes the ITO and CdS and reduces the CdTe to electron transparency. Full details of the sample preparation procedure are given in section 5.7. The finished samples had very large areas of sufficiently thin material, allowing many different areas to be imaged, and hence the images presented here can be taken as being representative of the material.

The microscope used was a JEOL 200CX, running at a voltage of 160kV.

8.3 RESULTS

8.3.1 OVERVIEW

Shown in figure 8.1 is a plan-view TEM overview of the near-interface grain microstructure in an as-grown CdTe/CdS solar cell. The structure clearly has a similar distribution of grain sizes to the surfaces examined by optical microscopy described in chapter 7. Also apparent is a large density of linear, planar defects, which can in fact be found in almost all grains above about $0.1\ \mu\text{m}$ in size (although many are out of contrast here). Figure 8.2 shows one of these boundaries, with the associated diffraction pattern, a pattern widely known to arise from twin boundaries. These have previously been reported to be highly prevalent in both bulk [8-11] and polycrystalline [2, 7] CdTe.



Figure 8.1: Plan-view TEM image showing the grain structure in the near-interface region of an as-grown CdTe/CdS solar cell. The high twin boundary density is clearly apparent.

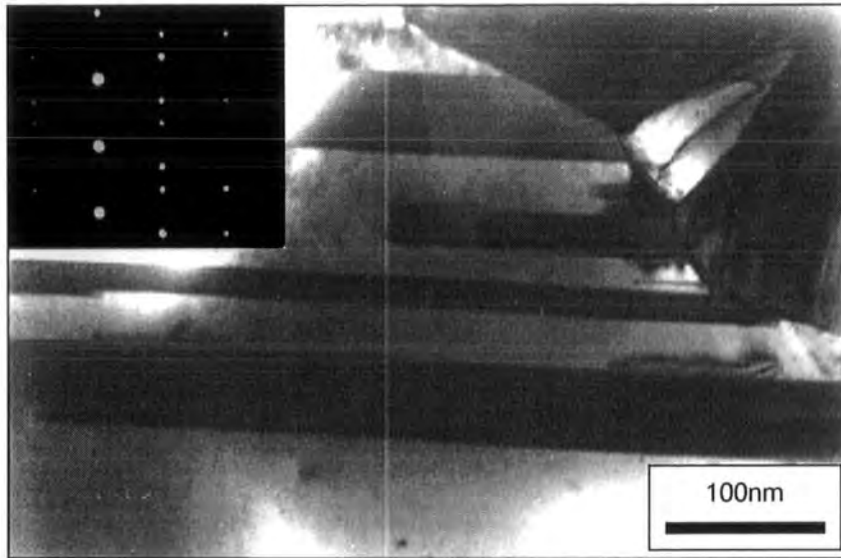


Figure 8.2: Plan-view TEM image showing a linear boundary in an as-grown CdTe/CdS solar cell. The inset diffraction pattern (corrected for relative rotation) confirms it as a twin boundary.



Figure 8.3: Plan-view TEM image showing the grain structure in the near-interface region of a CdCl₂-treated CdTe/CdS solar cell. Although this foil is slightly thicker than that shown in figure 8.1, the structure is clearly similar at this magnification.

The grain structure of near-interface CdCl₂ treated material (figure 8.3) was qualitatively similar to the as-grown material shown in figure 8.1, but

showed differences in the grain size and dislocation distribution, which are further discussed in sections 8.3.2 and 8.3.3 respectively.

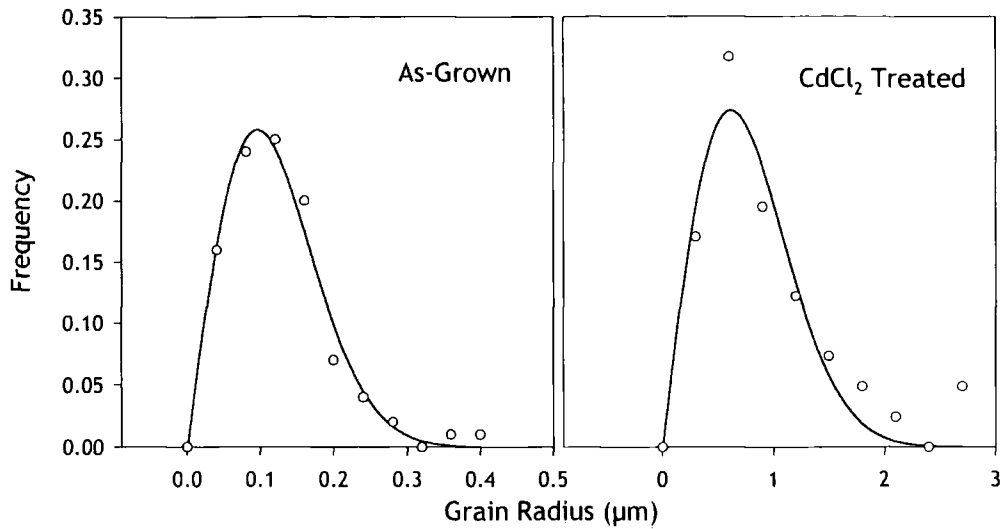


Figure 8.4: Grain size distributions for both as-grown and CdCl_2 treated CdTe from the near-interface region. The solid lines are Rayleigh fits to the data points. The treated sample has a considerably larger mean grain size.

8.3.2 GRAIN SIZE

The grain size distribution data was extracted from images similar to figures 8.1 and 8.3. This was effected by tracing the grain boundaries onto a transparency, and analysing the resulting pattern with the software described in section 7.3.1. The software could not be applied directly to the images, as it is incapable of distinguishing between grain boundaries and twin boundaries. It should be noted, however, that twin boundaries are known to be negligibly electrically active in comparison to grain boundaries, and can thus be safely discounted. For both the as-grown and CdCl_2 treated material, four micrographs of adjacent regions were analysed to improve the statistics. The resulting grain size distributions are shown in figure 8.4, with Rayleigh distributions fitted to them (see section 7.4.2). It should be noted that the

data for the as-grown sample is considerably smoother than that for the CdCl_2 treated sample due to the respective sample sizes of 177 and 41 grains. The fits are convincing, however, especially in the case of the as-grown material. In addition, and most importantly, the figure shows that the grain size in the treated material is approximately five times larger than in the untreated material. It is also interesting to note that the shape of the distribution is almost invariant under grain-growth, as is suggested by grain growth theories (see chapter 3 and Humphries and Hatherley [12]).

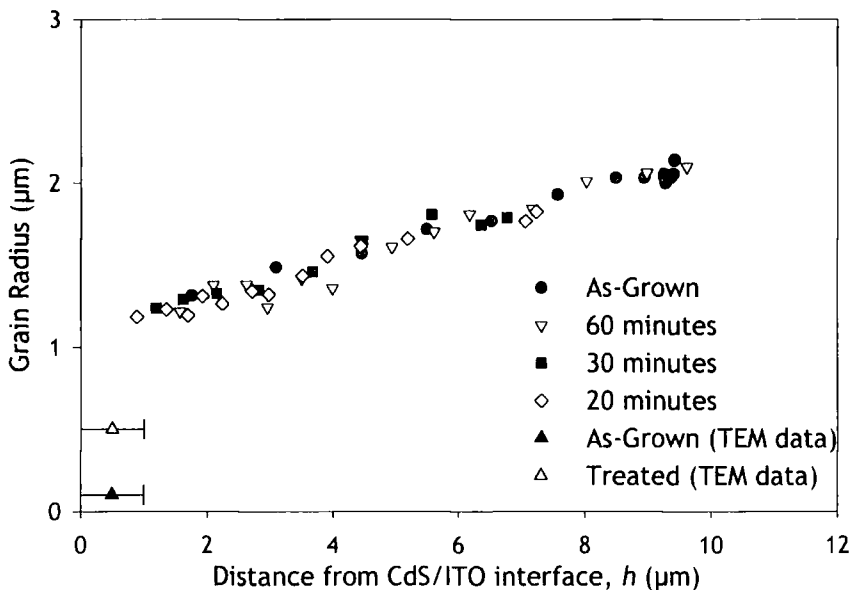


Figure 8.5: Grain size variation throughout the CdTe layer. The bulk data is that previously presented in section 7.4.3. The two points at $h < 1 \mu\text{m}$ are from the TEM work presented here. The treatment time for the CdCl_2 treated TEM data is not included as the processing is very different and thus not comparable.

The data extracted as described above can also give values for the mean grain size. The values for the as-grown material ($0.1 \mu\text{m}$) and CdCl_2 treated material ($0.5 \mu\text{m}$) are shown in figure 8.5 superimposed upon the data of section 7.4.3. The error bars on these points reflect the fact that it is known only that the micrographs come from the bottom $1 \mu\text{m}$ of the film: the exact

position is unknown. As shown in the figure, the variation of the grain size throughout the bulk of the material was shown to vary linearly with position in the layer. As is clearly apparent, the bulk linear trend does not extend all the way to the near interface region; a rapid drop-off in size occurs as the position in the CdTe layer approaches the CdS. At face value, the data appears to show small grained, near-interface material in the as-grown state undergoing grain-growth on CdCl_2 treatment. The validity of this conclusion is discussed in section 8.4.

8.3.3 FURTHER RESULTS

In addition to the difference in grain sizes of as-grown and CdCl_2 treated material, a number of other microstructural differences were observed on examination of individual grains:

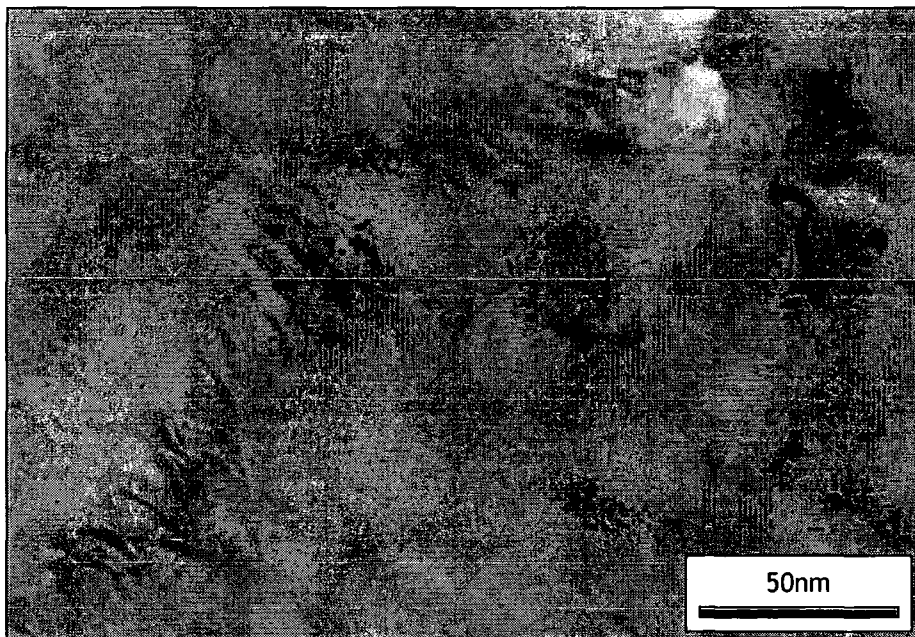


Figure 8.6: A phenomena specific to the CdCl_2 treated samples: large numbers of sub-grain boundaries were found in the treated material, but not in the as-grown.

- a) The CdCl_2 treated material contained a high density of sub-grain boundaries (such as in figure 8.6), which are largely absent in as-grown material. These are sometimes found in as-grown material (as reported by this author [13]) but subsequent comparative study has shown that their densities are much greater in CdCl_2 treated material.
- b) Where present, dislocations in the CdCl_2 treated material are associated with planar defects, such as twin boundaries (figure 8.7).
- c) The density of dislocations within grains is considerably higher in the as-grown material (e.g. figure 8.8) than the CdCl_2 treated material. Larger grains within the as-grown material also have a higher density of defects than smaller ones.

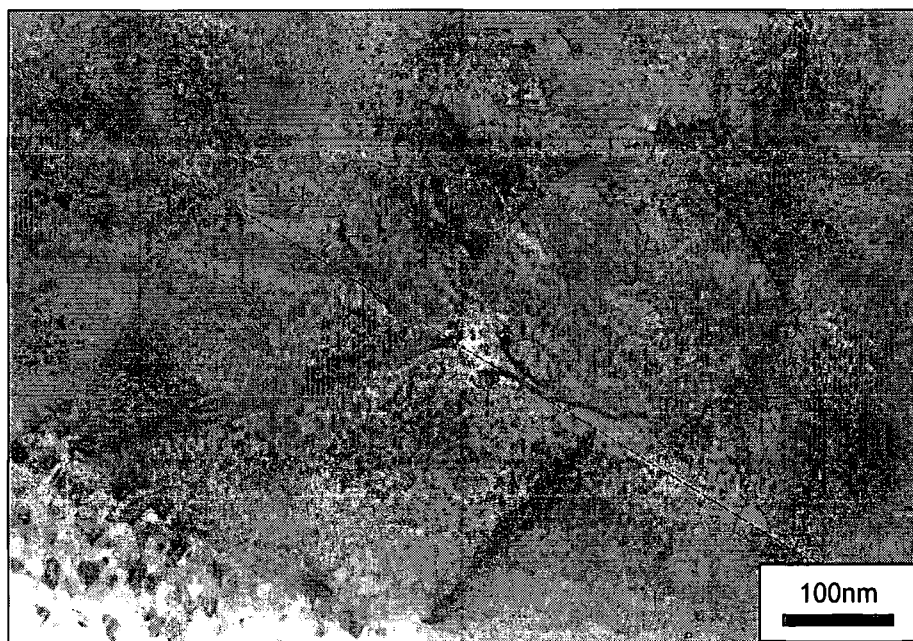


Figure 8.7: Dislocations in CdCl_2 treated material, associated with twin boundaries (in poor contrast, running top left to bottom right as indicated by the dotted line).

Unfortunately, dislocations could not be analysed by two-beam analysis due to the short time in which beam damage— visible as speckle in figure 8.7—

would obscure the defect. This is in addition to the usual difficulties in performing two-beam analysis on polycrystalline material.

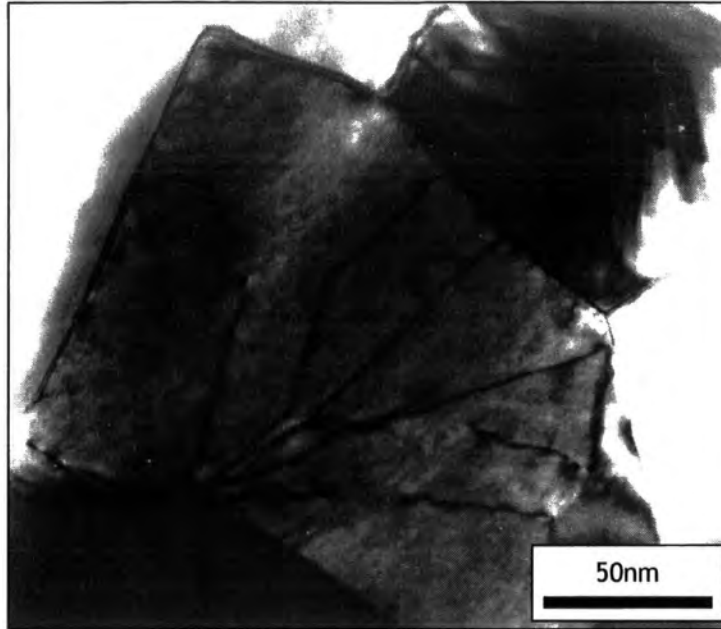


Figure 8.8: Dislocations in untreated near-interface CdTe.

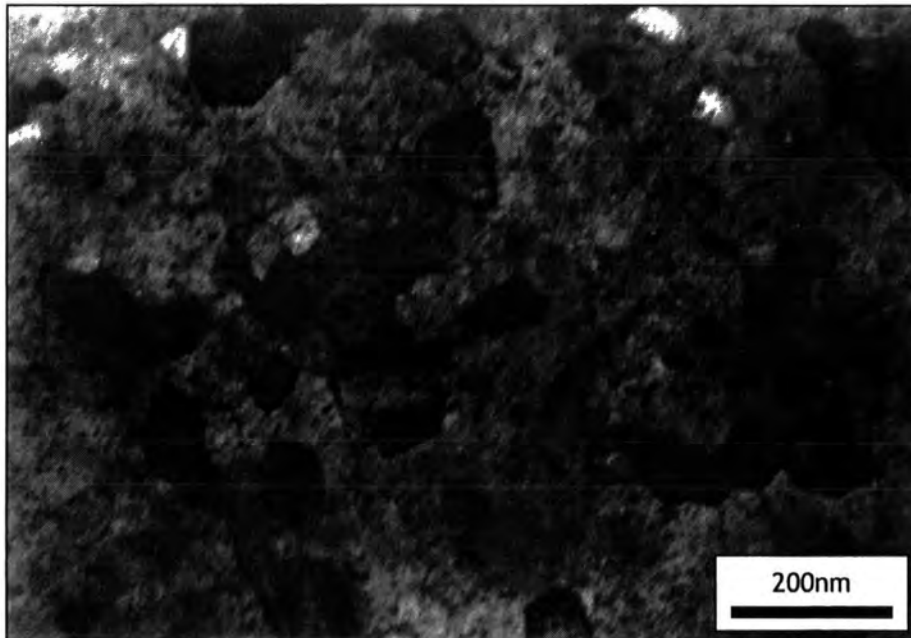


Figure 8.9: Plan-view TEM of the CdS layer.

By a slight modification of the thinning process, samples showing the CdS layer can be prepared. An overview of the grain structure of CdS is shown in figure 8.9. Calculating the grain size, by the tracing and image analysis technique described in section 8.3.2, gives a arithmetical mean value of 49 nm.

8.4 DISCUSSION

In section 3.2, the classic mechanisms for microstructural evolution were discussed, namely recovery, recrystallisation and grain growth. This section will discuss the TEM observation of dislocation and grain boundary behaviour (section 8.3.3) in terms of this ternion, after first considering the grain size distribution data of section 8.3.2. In addition, relationships between near-interface grain growth and solar cell efficiency are explored by means of a simple model.

Section 8.3.2 and figure 8.5 show that whilst the grain size varies linearly with depth in the bulk, it drops off rapidly in the near-interface region. In order to explain this variation a number of arguments must be invoked. Firstly, Al-Jassim *et al.* [1] have shown by cross-sectional TEM (albeit on PVD CdTe) that CdTe grows epitaxially on CdS on a grain-by-grain basis: Grain boundaries were seen to propagate across the CdTe/CdS interface. Secondly, in the same paper, the above correlations between the two layers were found to disappear on annealing with CdCl₂. It is well established that the reduction of the free energy associated with grain boundaries acts as a driving force for grain growth [12]. Thus, it seems reasonable to speculate that during layer growth, the near interface CdTe grain size (~100 nm) is pinned, and thus limited by the grain size of the

underlying CdS (~50 nm). However, as the CdTe layer becomes thicker, this constraint rapidly disappears, and the grain size approaches the equilibrium value dictated by the grain boundary energetics. This relationship between grain size and layer thickness is investigated further by modelling in the following chapter (chapter 9).

Turning to microstructural development on CdCl₂ treatment, evidence for recovery, recrystallisation, and grain growth is presented in turn in the following paragraphs, starting with recovery.

Recovery: Sub-grain boundaries are widely known to be created during the progression of recovery on annealing a material containing defects (see section 3.2.1). In the material studied here, sub-grain boundaries (also called polygonisation walls) were found to be common in CdCl₂ treated material, but were almost absent in the as-grown material (see section 8.3.3). This suggests the occurrence of a recovery-like process in the CdTe/CdS system on CdCl₂ treatment. This idea is supported by (i) the association of defects with one another, typically dislocations themselves and planar boundaries (as in figure 8.7), and (ii) the decreased density of dislocations found after annealing. The material is showing classic recovery behaviour, with defects migrating to lower energy positions. To the author's knowledge, this is the first direct evidence of recovery in CdTe/CdS solar cells. Moreover, as the growth temperature (500°C) is greater than the annealing temperature (400°C), it seems certain that the presence of CdCl₂ at the annealing stage is playing a vital role in the defect migration. However, the ability of CdCl₂ to directly influence defect mobility is dependent upon its solubility in CdTe. The phase diagrams for the CdTe-CdCl₂ system of Andronik *et al.* [14] show CdCl₂ to have zero solubility in CdTe. This is, perhaps, a limitation of the measurements, though: it is well known that Cl is a soluble dopant in CdTe

and indeed is added as CdCl_2 in the synthesis of high resistivity CdTe , for applications such as X-ray detectors [15, 16]. The material can easily accommodate 50ppm of Cl, for example [15].

In addition to the above solubility issues, it has also been suggested that CdCl_2 has the effect of producing Te vacancies in CdTe via the following mechanism (in Kroger-Vink notation): The Cd from the CdCl_2 produces a Te vacancy in the CdTe lattice,



This vacancy can then be occupied by one of the Cl ions:



but the other will require a second Te site:



Combining these gives a net gain in Cd vacancies:



The introduction of such species could significantly affect the mobilities of dislocations, and encourage recovery*.

Recrystallisation: The next of the three microstructural development phenomena to be discussed is recrystallisation. A number of observations have been made in CdTe/CdS solar cells that suggest recrystallisation events, including some explicitly in the near-interface region. (It should be noted, however, that those studies were performed on PVD material). As described

* Incidentally, these reactions show that the addition of CdCl_2 creates cadmium vacancies, which may form an 'A-centre' compensated by the $\text{Cl}_{\text{Te}}^{\bullet}$ i.e. $\text{Cl}_{\text{Te}}^{\bullet} + [V_{\text{Cd}}^{\prime\prime} + \text{Cd}_{\text{Te}}^{\bullet}]$. This can only be seen as compatible with the literature assertion that doping is via the A centre if additional Cd vacancies are supplied. This is possible due to Cd loss during heating or the supply of excess Te.

earlier in this section, Al-Jassim *et al.* [1] have reported local epitaxial growth of CdTe on CdS, and thus the propagation of grain boundaries from the CdS into the CdTe. However, this inter-layer relationship is lost on annealing. The disappearance of the epitaxial correlation implies that the crystal structure of the overlying material must have been entirely restructured: this can be viewed as a localised recrystallisation event. Recrystallisation is known to be driven by an excess of volume strain energy. In CdTe/CdS bilayers, this can arise from the following sources:

- a) Strain at the metallurgical interface, from a combination of lattice mismatch and differences in thermal expansion coefficients;
- b) Strain present at defects, such as grain boundaries or dislocations;
- c) Small grain size, as shown by Moutinho *et al.* [17].

In the case of a) above, this work offers no evidence that CSS CdTe on CdS shows local epitaxial growth: the lack of preferred orientation in the as-grown state may indeed suggest its absence (see section 4.3.2). However, cases b) and c) are very possibly applicable to as-grown CSS material: a large number of grains contain dislocations, and the near-interface grain size is small. Vitally, though, it must be remembered that recrystallisation *consumes* the old microstructure. If this were the case here, then the sub-grain boundaries discussed with reference to recovery above would not be found in CdCl₂ treated material. Thus, we can clearly discount the occurrence of recrystallisation on CdCl₂ treatment of CSS-grown CdTe layers.

Grain Growth: Turning finally to microstructural change by grain growth, the result of figure 8.4 shows clearly that grain-growth occurs in the near-interface region on CdCl₂ treatment. It can only be assumed that this growth is a result of the addition of the CdCl₂ flux that encourages the

migration of species between grains to lower the overall grain-boundary free energy.

Considering that the measured five-fold increase in grain size implies a reduction in the grain boundary density, it could be hypothesised that this would also reduce the level of carrier recombination in the absorber layer, as grain boundaries can act as sinks for carriers. The extent to which this may be plausible is now assessed using a simple model with the following assumptions:

- a) The grain boundaries have a 100% collection/recombination probability over their width, t ;
- b) The width of the boundaries is the same before and after CdCl_2 treatment.
- c) The material within the grains is the same before and after treatment, having a collection/recombination probability of zero.
- d) The photogenerated current within the grains is assumed to be the same before and after treatment.

The cross-sectional grain structure is approximated by the tessellating regular hexagons shown in figure 8.10, and a photogenerated current density, J_p , is assumed to travel perpendicular to this plane. Thus, in travelling through this plane, a certain number of carriers, dictated by the effective area of the grain boundaries, will be trapped. If we concern ourselves with an area A of this structure, then it is clear that the collected current J_c will be given by:

$$J_c = \frac{A - A_{\text{boundary}}}{A} J_p, \quad (8.5)$$

where A_{boundary} is the total area in A occupied by grain boundaries. Ignoring edge effects, this can just as well be expressed in terms of the dimensions of

individual grains:

$$J_c = \frac{\sigma}{A_{HEX}} J_p, \quad (8.6)$$

where σ is the area within a single grain not occupied by boundary. A_{HEX} is simply the area of a hexagon, easily shown to be,

$$A_{HEX} = 2\sqrt{3}r^2. \quad (8.7)$$

From the diagram, it can be seen that the expression for σ is the same as that for A_{HEX} (equation 8.7)), but with r replaced by $(r - t)$. Substituting A_{HEX} and σ into equation 8.6) gives:

$$\frac{J_c}{J_p} = \frac{2\sqrt{3}(r-t)^2}{2\sqrt{3}r^2} = 1 - \frac{2t}{r} + \frac{t^2}{r^2}. \quad (8.8)$$

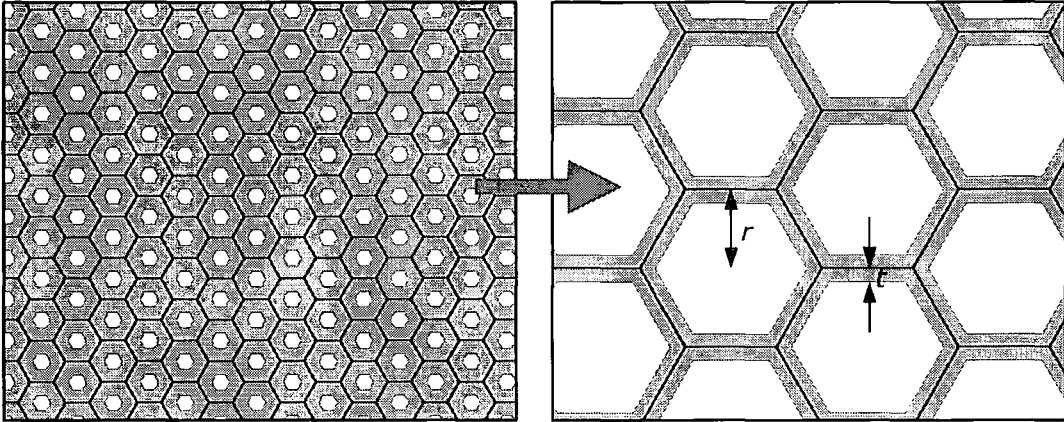


Figure 8.10: Simple geometry used to ascertain how grain expansion and the effective width of grain boundaries for carrier recombination on annealing might affect carrier recombination and thus collected photocurrent. The shaded area represents the effective width of the grain boundaries for carrier capture.

To compare the as-grown and CdCl_2 treated cases, appropriately subscripted values of J_c and r can be substituted into two copies of this equation, and the ratio of the two taken. This process assumes that t remains constant during annealing, a conjecture which will be discussed later in this

section. This process, with some re-arrangement gives the quadratic equation,

$$t^2 \left(\frac{1}{r_f^2} - \frac{R}{r_i^2} \right) + 2t \left(\frac{R}{r_i} - \frac{1}{r_f} \right) + (1 - R) = 0, \quad (8.9)$$

where r_i and r_f are the initial and final grain sizes, and $R = J_{cf}/J_{ci}$, the ratio of final to initial short circuit current values.

Values are known for how annealing affects both grain size, and short circuit current: this section has already reported a fivefold increase in grain size, and section 7.4.1 reports an approximate doubling of current density after CdCl_2 treatment. Substituting these values into equation 8.9 and solving for t gives a value of 32 nm for t , (plus another un-physical root) relative to values for r_i and r_f of 0.1 and 0.5 μm respectively. Hence, if the efficiency gain were to arise solely from near-interface grain-growth, the grain boundaries act as minority carrier traps with a depletion width of 32 nm on each side. However, this should be discussed with reference to changes in the material upon annealing and literature models of grain boundary recombination.

Calculation of this value of t assumes the material within the grains to be unchanged on annealing: this has shown not to be the case by the evidence of recovery given in this chapter, although precisely how it affects transport is unknown. Further to this, the model assumes that the grain boundaries are identical in band structure before and after treatment: this is not the case. Various authors have reported a distinct passivating effect of the grain boundaries on CdCl_2 treatment [18, 19]. In addition, reports exist which conflict with the simple constant capture cross-section model for grain boundaries presented here:

- a) Woods *et al.* [20] report the grain-boundary band structure of CdCl_2 treated material to be as shown in figure 8.11a, using in-plane transport and impedance spectroscopy measurements. As is clear from the shapes of the band-edges, electrons at a certain distance from the boundary will be repelled, whereas those very close to the boundary will become trapped. Whilst this model predicts a well-defined width for carrier capture, no width is specifically stated in the work.
- b) Durose *et al.* [21] used Remote EBIC measurements to study the electrical activity of lateral high angle grain boundaries in bulk untreated CdTe , and showed the band structure of figure 8.11b. This form of potential will repel electrons. It may be the case however, that this structure is a resolution-limited version of that of case a) above.

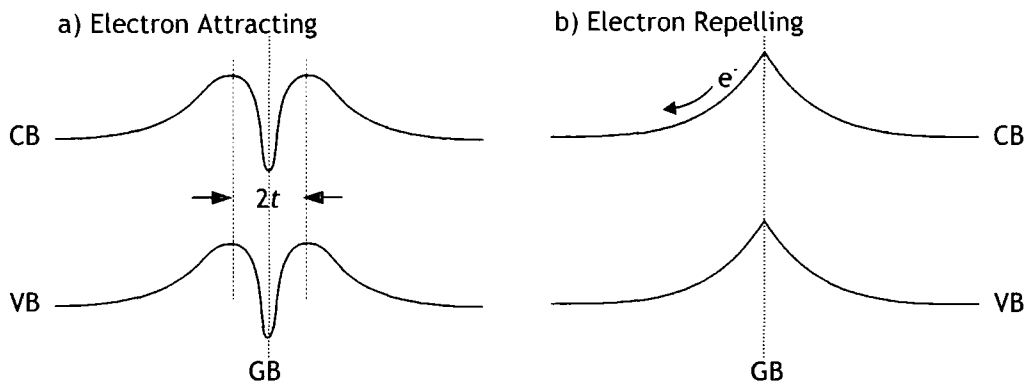


Figure 8.11: Two different band-bending schemes proposed for CdTe grain boundaries (GB's), due to a) Woods *et al.* [20] in CdCl_2 treated material, and b) Durose *et al.* [21] in as-grown material. The conduction and valence bands are denoted by CB and VB respectively.

It is clear that these two cases do not reflect the full story: if they did represent the 'before' and 'after' cases of a single system, it is unlikely that the CdCl_2 treated material would have better electrical characteristics. However,

they also make clear that some of the basic assumptions of the model may be simplistic at best.

8.5 CONCLUSIONS

This chapter has documented a plan-view TEM study of polycrystalline CdTe in the near interface region of CdTe/CdS bi-layers. Whilst the bulk of the CdTe layer was shown to be microstructurally stable under the influence of CdCl₂ and heat in the preceding chapter, this chapter has shown that the near interface region undergoes a considerable crystallographic upheaval on annealing. In addition to a five-fold increase in the grain size, this chapter has reported the first direct evidence of recovery in this system. Furthermore, CdCl₂ treatment was shown to reduce the dislocation density within grains, and it was speculated that the annihilation of dislocations would improve the electrical characteristics of a CdTe/CdS device to some extent, as might the increase in interface grain size.

A simple model in which improvements in efficiency were attributed solely to reduction in the grain boundary capture cross-section indicated an electron capture width of 32 nm for each grain boundary. However, this assumes that grain boundaries in as-deposited and CdCl₂ treated material are electrically identical. This is unlikely to be the case, CdCl₂ treatment being the doping mechanism in these films. Other authors have also produced evidence that CdCl₂ treatment passivates the grain boundaries to some extent although the details of the models differ. Seen together with the observation that the microstructure of the bulk of the films is unchanged by CdCl₂ and yet the efficiency increases, it might be speculated overall that microstructural

changes in the material are perhaps less important than the electrical structure of the grain boundary interfaces in a given system.

In the following chapter, the variation of grain size throughout the CdTe layer ascertained in this and the previous chapter will be described using a simple computer model.

8.6 REFERENCES

- [1] M. M. Al-Jassim, F. S. Hasoon, K. M. Jones, B. M. Keyes, R. J. Matson and H. R. Moutinho *The Morphology, Microstructure and Luminescent Properties of CdS/CdTe Thin-Film Solar-Cells*, Proceedings of the 23rd IEEE Photovoltaic Specialists Conference (1993) 459-65
- [2] M. M. Al-Jassim, F. S. Hasoon, K. M. Jones, B. M. Keyes, R. J. Matson and H. R. Moutinho *The Characterisation of CdS/CdTe Thin Films for Solar Cell Applications*, Proceedings of the Microscopy of Semiconducting Materials Conference (1993) 411-16
- [3] R. G. Dhere, S. E. Asher, K. M. Jones, M. M. Al-Jassim, H. R. Moutinho, D. H. Rose, P. Dippo and P. Sheldon *Characterization of Intermixing at the CdS/CdTe Interface in CSS Deposited CdTe*, AIP Conference Proceedings **353** (1996) 392-99
- [4] Y. Y. Loginov, K. Durose, H. M. Alallak, S. A. Galloway, S. Oktik, A. W. Brinkman, H. Richter and D. Bonnet *Transmission Electron Microscopy of CdTe/CdS Based Solar-Cells*, Journal of Crystal Growth **161** (1996) 159-63
- [5] B. E. McCandless, L. V. Moulton and R. W. Birkmire *Recrystallization and Sulfur Diffusion in CdCl₂-Treated CdTe/CdS Thin Films*, Progress in Photovoltaics **5** (1997) 249-60
- [6] M. M. Al-Jassim, R. G. Dhere, K. M. Jones, F. S. Hasoon and P. Sheldon *The Morphology, Microstructure, and Luminescent Properties of CdS/CdTe Films*, Proceedings of the 2nd World Conference on Photovoltaic Solar Energy Conversion (1998) 1063-66
- [7] D. R. Johnson *Microstructure of Electrodeposited CdS/CdTe Cells*, Thin Solid Films **361** (2000) 321-26
- [8] A. W. Vere, S. Cole and D. J. Williams *The Origins of Twinning in CdTe*, Journal of Electronic Materials **12** (1983) 551-61
- [9] K. Durose and G. J. Russell *Structural Defects in CdTe Crystals Grown by Two Different Vapor- Phase Techniques*, Journal of Crystal Growth **86** (1988) 471-76
- [10] K. Durose, A. Turnbull and P. Brown *Structural Defects in Bulk and Epitaxial CdTe*, Materials Science and Engineering B **16** (1993) 96-102
- [11] K. Durose and G. J. Russell *Twinning in CdTe*, Journal of Crystal Growth **101** (1990) 246-50
- [12] F. J. Humphreys and M. Hatherley "Recrystallization and Related Annealing Phenomena" (Pergamon, Oxford, 1995)
- [13] M. A. Cousins and K. Durose *TEM of the near-Interface Region in CdTe/CdS Solar Cells*, Proceedings of the 16th European Photovoltaic Solar Energy Conference (2000) *In Press*
- [14] I. K. Andronik, Z. P. Kuleva and K. D. Sushkevich *Feasability of Obtaining CdTe Crystals by Zone Refining from a Solution in Fused CdCl₂*, Inorganic Materials **12** (1976) 759-60

- [15] S. Seto, A. Tanaka, Y. Masa and M. Kawashima *Chlorine-Related Photoluminescence Lines in High-Resistivity Cl-Doped CdTe*, Journal of Crystal Growth **117** (1992) 271-75
- [16] K. Saminadayar, J. M. Francou and J. L. Pautrat *Electron-Paramagnetic Resonance and Photoluminescence Studies of CdTe-Cl Crystals Submitted to Heat-Treatment*, Journal of Crystal Growth **72** (1985) 236-41
- [17] H. R. Moutinho, M. M. Al-Jassim, F. A. Abulfotuh, D. H. Levi, P. C. Dippo, R. G. Dhere and L. L. Kazmerski *Studies of Recrystallisation of CdTe Thin Films after CdCl₂ Treatment*, Proceedings of the 26th IEEE Photovoltaic Specialists Conference (1997) 431-34
- [18] P. R. Edwards, S. A. Galloway, P. R. Wilshaw and K. Durose *A Study of the Activation of CdTe/CdS Thin Film Solar Cells Using OBIC*, Institute of Physics Conference Series **157** (1997) 583-86
- [19] P. R. Edwards, S. A. Galloway and K. Durose *EBIC and Luminescence Mapping of CdTe/CdS Solar Cells*, Thin Solid Films **361-362** (2000) 364-70
- [20] L. M. Woods, D. H. Levi, V. Kaydanov, G. Y. Robinson and R. K. Ahrenkiel *Electrical Characterization of CdTe Grain-Boundary Properties from as Processed CdTe/CdS Solar Cells*, Proceedings of the 2nd World Conference on Photovoltaic Solar Energy Conversion (1998) 1043-46
- [21] K. Durose, J. R. E. Sadler, A. J. W. Yates and A. Szczerbakow *R-EBIC Study of the Electrical Activity of Grain Boundaries in CdTe and Cd(S,Te)*, Proceedings of the 28th IEEE Photovoltaic Specialists Conference (2000) *In Press*

Computer Simulation of Absorber-Layer Growth

9.1 INTRODUCTION

In chapters 7 and 8, the variation of grain size throughout the CdTe absorber layers of CSS-grown CdTe/CdS solar cells was determined. It was shown that, whilst the grain size varies linearly throughout the bulk of the layer, it drops off rapidly as the position in the CdTe layer approaches the CdS. This gives two distinct regions as shown in figure 8.5. It has also been shown that the grains are columnar in nature (figure 7.9). These observations were discussed in their respective chapters. In order to elucidate further the mechanism responsible for this behaviour, a computer model was developed, which forms the subject of this chapter.

A number of computer studies have been performed for describing the growth of polycrystalline material. The development of grain boundaries over

dimensions of ~ 100 monolayers was studied by Smith [1], although the work results in layers with preferred orientation, which is not observed in the CSS material of interest here. Romeo *et al.* [2] have studied the behaviour of Cd and Te species in the vapour phase during CSS growth, but not the resulting materials. Various aspects of growth have been studied for polycrystalline forms of both silicon [3] and diamond [4], as well as studies for general materials. Levine *et al.* [3] briefly review a number of different methods and reports of their usage.

The model used in this work was designed to explore the possibility of simulating the measured grain size variation using a simple algorithm, which dictates how species attach themselves at the growth surface, and thus form a polycrystalline film. This chapter will first describe the rules governing this model (section 9.2), before presenting some results of simulations (section 9.3), and associated discussion (section 9.4).

9.2 PRECEPTS OF MODEL BEHAVIOUR

This section describes the rules governing the behaviour of the model created to describe grain size development during growth in CdTe absorber layers. The source code itself, written in C++, is reproduced in appendix C.

The growth is performed in a three dimensional array— of height h , and width and depth w — at the ‘bottom’ of which is placed an initial uniform layer on which species will subsequently nucleate. This is referred to as the ‘zeroth’ layer. Species are introduced at the ‘top’ of the array at random positions and then descend until they hit either the zeroth layer or the growth surface.

At the beginning of the simulation, arriving species are used to seed the surface with nuclei of grains: the number of these species and hence grains is

given by w^2/G_A , where G_A is a program parameter representing the mean area of the grains in the first layer. These attach themselves upon the zeroth layer, with the condition that any which find themselves in the same position as a previous nucleus are rejected. When w^2/G_A nuclei are in place no further nucleation of grains is then permitted.

The behaviour of arriving species then changes, to allow them to incorporate themselves into the nucleated grains. This proceeds as follows: a species that descends and finds itself adjacent (in any direction) to a previously nucleated grain or grains can incorporate itself into this grain. The probability of incorporation at a point (x, y, z) is represented by the value $P_A(x, y, z)$: if this is greater than a randomly generated value between 0 and 1, the grain will attach to the growth surface, else it will migrate and repeat the process. The means of calculating $P_A(x, y, z)$ is described below, a process that also dictates the grain into which the new species will be incorporated should it arrive at a grain boundary.

As previously mentioned, migration of a species involves it taking a unit step across the growth surface. In this model, the direction taken is random, with the limitations that the species cannot go up steps in the surface. If it migrates and finds itself over a hole in the surface, it will fall down it, negating the possibility of any voids in the final film. In addition, a grain attempting to leave the bounds of the array will be 'wrapped', reappearing at the opposite side of the array. This negates any edge effects, although corner effects may become significant at large grain sizes.

As mentioned above, the calculation of $P_A(x, y, z)$ dictates not only whether an arriving species will be incorporated at that point or not but also the designation of the grain into which the new species will be incorporated. In essence, the species is assigned to the grain to which it is most strongly

bound, a consideration of importance if the point (x, y, z) is positioned adjacent to more than one grain.

The calculation of $P_A(x, y, z)$ proceeds thus: if the 26-neighbours of pixel (x, y, z) are considered, each is either empty, in which case it is ignored, or belongs to a particular grain. If different grains are denoted by different values of the subscript n , then we can say that X_n is the number of the 26-neighbours of (x, y, z) which belong to grain n . We can then construct a normalised sum,

$$\sigma_n = \frac{\sum_{c=1}^{X_n} d_c^{-B}}{\sum_i \text{all neighbours} d_i^{-B}}, \quad (9.1)$$

for each n . The parameter d_c represents the distance from the centre of array element c to the centre of (x, y, z) , and B represents a bonding term, to modify the drop-off in attractive force between species. A value of $B = 2$ was used throughout this work.

The sum of the values of σ_n , when multiplied by a program parameter P_S , (i.e. an overall attractive term, and scaling factor for bond strength) gives $P_A(x, y, z)$. The grain with the highest value of σ_n is that into which the new species will be incorporated, unless the comparison of $P_A(x, y, z)$ with the random value disallows this.

In order for films considerably thicker than the array height to be produced, layers are output to file on being completed. Each layer can then be moved down by one in the array. The chosen file format was the bitmap; this allows easy viewing on a PC and the direct application of the analysis software described in section 7.3.1. Working in this way, the model can continue growing layers until either a pre-set limit is reached, or unfinished

layers completely fill the array, although this undesirable latter limit is seldom reached with careful choice of h . When either limit is reached, the remaining incomplete layers are also outputted to file. This gives information on the behaviour of species at the growth surface.

Typical values used here are 100 for w , and 25 for h , with a maximum of 100 layers output.

9.3 RESULTS

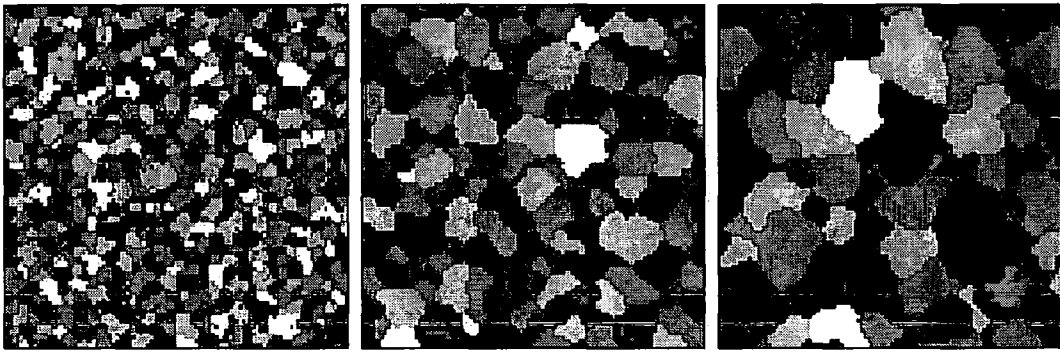


Figure 9.1: From left to right, the first, 50th and 100th layers produced by the modelling program, with parameters $P_s = 1$, $G_A = 10$, $w = 100$ and $h = 20$. The structure is clearly comparable to a real polycrystalline cross-section, with an increase in grain size as the film grows thicker.

9.3.1 OVERVIEW

Shown in figure 9.1 is a series of images from a typical run of the modelling program described above: shown are the first layer, the middle layer (50th) and the last layer (100th). The microstructure is clearly reminiscent of the real structures reported earlier in this work (for example in figures 7.3 and 8.1). In addition, it is quite clear that the mean grain size is increasing as the layer becomes thicker. This expansion will be quantified in the following section.

Some other general observations are of note: Figure 9.2a shows a cross-section through one of the films, showing the columnar nature of the growth (compare to figure 7.9). Also pertinent is the predominance of angles between grain boundaries of $\sim 120^\circ$ (figure 9.2b), found after growth has progressed some way. In a real solid, this is indicative of a structure near to equilibrium, as described in section 3.3.1. Furthermore, it was found that on running a small ($w = 50$) model over a large number of layers (~ 2000), no limiting grain size was found, with a single grain finally occupying the whole layer.

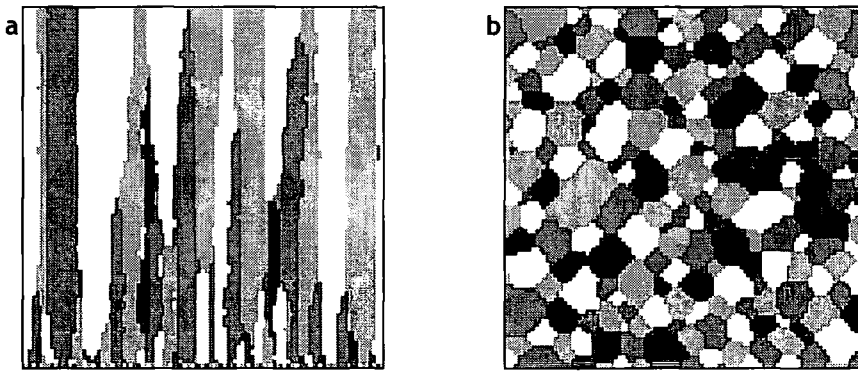


Figure 9.2: Images from simulated data showing: a) a typical cross-section generated from a full set of layers and, b) an image showing the tendency of the grain boundary vertices to approach 120° .

9.3.2 QUANTIFICATION OF STRUCTURE

By analysing the grain size distributions of the images produced for each layer, a picture of how the grain size varies throughout the layer quickly emerges. Shown in figure 9.3 is a plot showing the grain size distribution of all the layers for a film with growth parameters $P_S = 10^{-1}$, $G_A = 10$, $w = 100$ and $h = 20$. During the growth of the first few layers, a tall, narrow peak, arising from the nucleation of grains on the substrate, dominates the distribution of grain sizes. As growth progresses, the distribution quickly becomes lower and wider, with an increased average value. In addition, at

high values of layer number, the distribution becomes weakly bimodal with peaks at grain sizes of about 4 and 6 pixels.

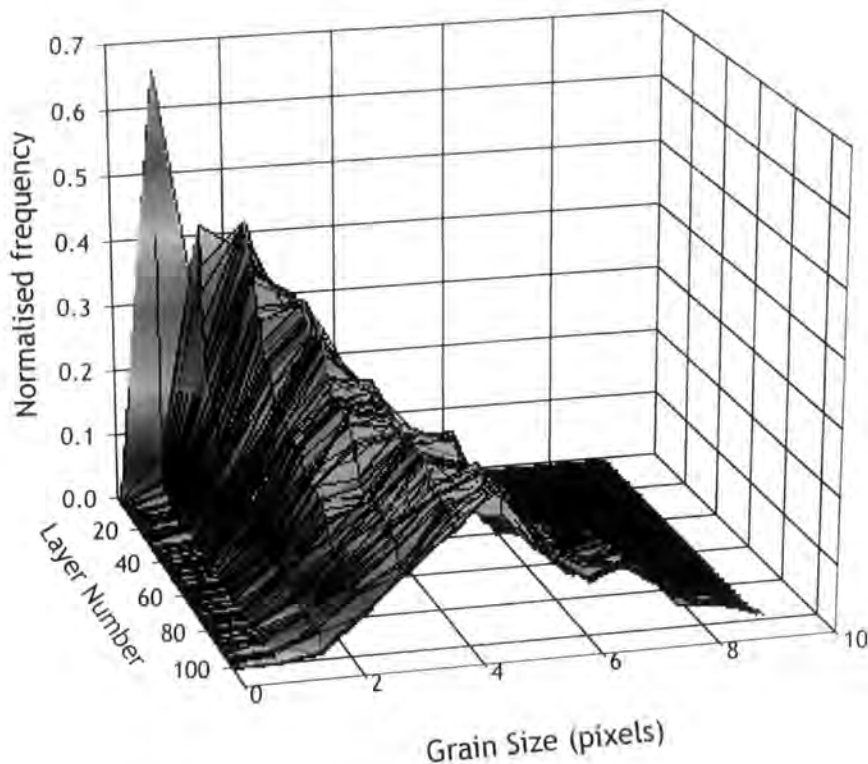


Figure 9.3: The distribution of grain sizes throughout a film consisting of 100 layers. The z-axis data has been normalised for each layer using the number of grains in that particular layer. The large peak in the background arises from the small nucleation size

In sections 7.4.2 and 8.3.2, the Rayleigh function proved efficient at describing grain size distributions extracted from images of real polycrystalline materials. Figure 9.4 shows the grain size distributions arising from the layer structure shown in figure 9.1, with Rayleigh functions fitted to it. It is clear that this particular distribution is not appropriate for describing this data: The distribution is clearly of a more centro-symmetric nature than that arising from experimental measurements.

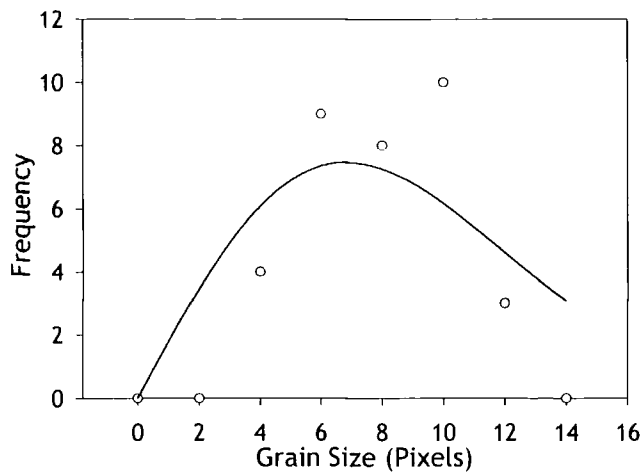


Figure 9.4: Grain size distributions for the central image in figure 9.1 (points), with a Rayleigh distribution fitted to it (line). As is typical for the model data, the fits are extremely poor.

By plotting the mean grain size of each layer against the layer number for various values of G_A and P_S , figure 9.5 is produced. Especially in the case of small G_A , the model displays the behaviour noted in the experimental case of figure 8.5; a rapid increase in grain size early on, followed by a transition to a more measured rate of increase. Inset into this figure is the experimental data presented in the previous chapter (figure 8.5). Comparison between the modelled data and this experimental data shows a clear similarity in distribution shape, implying that to some extent the model may reflect the physical processes in real materials. The validity of any such comparison is discussed in the following section, as are the trends caused by the variation of the parameters.

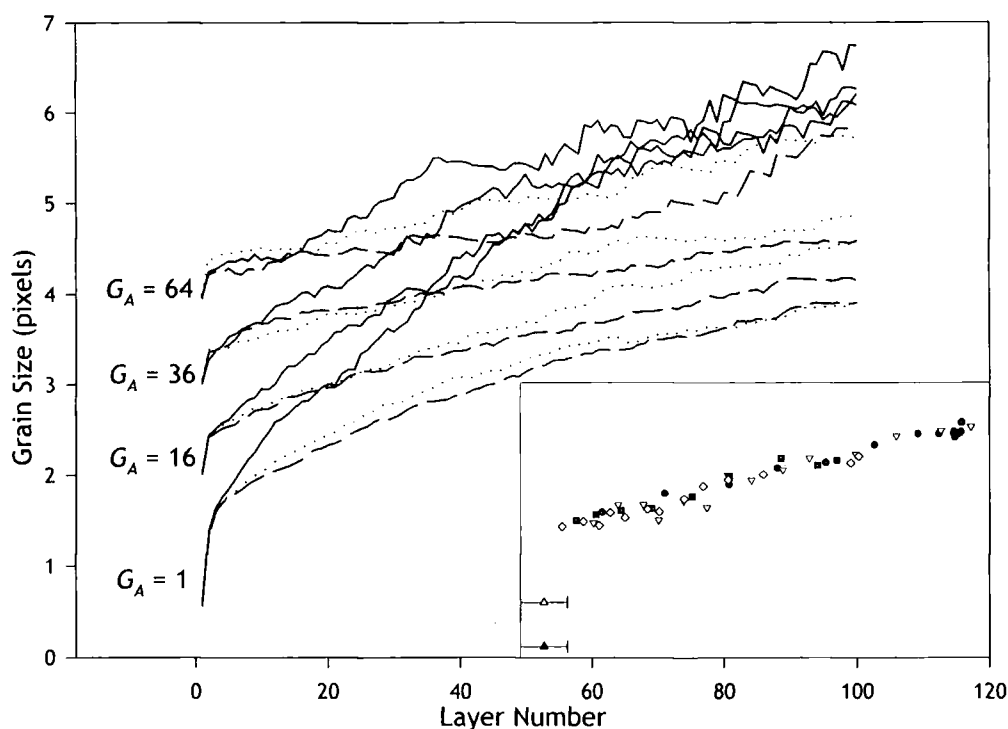


Figure 9.5: The effect of varying G_A (the mean area per nucleus) and P_S (the bonding strength) on the mean grain size variation. In each set of different G_A , $P_S = 10^{-2}$ is the dashed curve, $P_S = 10^{-1}$ is dotted, and $P_S = 1$ is solid. In addition, the experimentally obtained data of sections 7.4.3 and 8.3.2 is shown inset for comparison (refer to original (figure 8.5) for legend). The shapes of the simulated curves are clearly comparable to the experimental data.

Finally, figure 9.6 shows some of the curves of figure 9.5 run for 1000 rather than 100 layers (computational constraints negated this procedure for all the curves). Fitted to this data are curves of the form,

$$y = y_0 + ax^b \quad (9.2)$$

showing reasonable correlation in most cases. In the case of material with a small initial grain size, the power term, b , is ~ 0.5 .

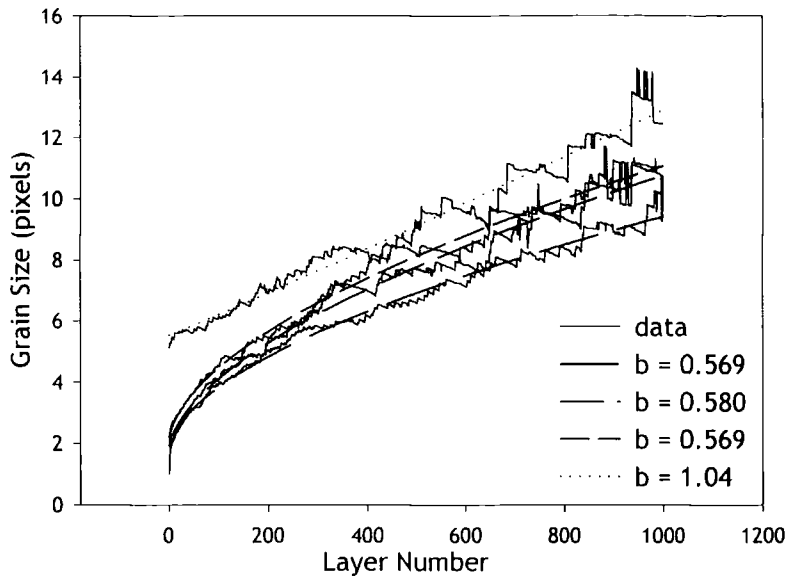


Figure 9.6: Modelled data run over 1000 layers, to investigate any overall trend in mean grain size. The curves fitted to the data are of a power-law form (equation 9.2), and show reasonable fits.

9.4 DISCUSSION

It is clear from the results of the previous section that comparison can be drawn between the modelled data and that from experiment: the grain size data shows the same two-regime variation, most notably in the case of material with a small initial grain size; and, the growth is columnar. The two regimes of grain size are characterised by a steep ‘near-interface’ region near the substrate surface, followed by a shallower ‘bulk’ region, in which the distribution is either linear or gently curved. In addition, the model displays the columnar growth characteristic of CdTe grown by many techniques discussed in this work (see section 4.3). However, whilst qualitative comparisons such as this can clearly be made, it is still necessary to decide whether this is simply fortuitous or whether the model represents a facsimile of some or all of the growth processes occurring in polycrystalline CdTe. This

section shall address these issues separately for columnar growth and the grain size variation.

9.4.1 COLUMNAR GROWTH

As mentioned above, both the modelled data, and that for real CSS-grown CdTe show distinct columnar growth. However, the aspect ratio of the grains seen in figure 9.2a is very different to that for CSS-grown material (figure 7.9) being more reminiscent of material grown by low temperature techniques such as PVD or EBD. Al-Jassim *et al.* [5] show TEM micrographs of comparable microstructure, albeit only over a small distance of the near-interface region. Nonetheless, the comparatively needle-like growth of the model is, perhaps, an issue of scale, as the grain size variation shows the required shape.

By examination of the model algorithm, the explanation for columnar growth in the model can quickly be accounted for. It arises from the constraint on the arriving species, which prevents them from nucleating anywhere but on the underlying zeroth layer. Thus, as new grains never appear after the first layer, columnar growth is dictated. In real material, it is conceivable that a similar type of mechanism is responsible for columnar growth. For grains of size ~2000 atomic spacings (typical for mid-film CSS CdTe), it is hard to imagine an arriving Cd or Te₂ molecule finding it energetically favourable to nucleate a new grain after a number of layers have already been deposited. Even if it new grains were to nucleate in such a way, they would be unlikely to grow sufficiently to avoid being subsumed by their larger neighbours almost immediately.

By considering the grain size quoted above, it is clear that issues of scale are important in this model: for a reasonable number of grains to be

examined, the model would have to be of size $w \sim 10^5$, a computationally prohibitive size. Thus, the 'species' used in the model represent agglomerations of atoms or molecules etc. However, the similarities between the experimental and synthetic grain size variation data, suggest these species may be capable of predicting mean atomistic behaviour. This is now discussed.

9.4.2 GRAIN SIZE VARIATION

As mentioned previously, the opacity of the mechanism behind the two-stage variation in grain size formed the rationale for this modelling study. However, this does not preclude speculations upon the mechanisms behind the model's success. A number of attributes are clear from figure 9.5.

- a) The variation in grain size has two distinct regimes.
- b) Larger values of the sticking coefficient, P_s , cause more grain growth.
- c) Variation in P_s has a greater effect for small values of G_A , the initial grain size. This is apparent as a reduction in the divergence of the sets of curves with the same P_s , as initial grain size increases, after the initial near-interface region.
- d) In figure 9.5 there appears to be a strong tendency for the curves to converge, most noticeably in the high P_s (low migration) material. From this data, it is unclear whether this is asymptotic behaviour or approach to a single function: however figure 9.7 shows no evidence of convergence for thicker films.
- e) Material with faster growing grains shows a less smooth variation in grain size, i.e. the curves are rougher.

From the simulations it is clear that there exists a driving force for grain size increase, and that it is a function of grain size. Moreover, the

mechanisms in the bulk and near-interface region also appear different. These phenomena can be discussed in terms of three mechanisms, outlined below.

Migration Effects: Variation of P_s is tantamount to directly varying the propensity of arriving species to migrate. This has a distinct effect on the way in which surfaces grow. In the case of this model, a species which migrates for a long time without attaching (low P_s) is in effect testing a larger number of sites for bonding strength. This implies that it is considerably more likely to fill both step-sites, as the bonding is stronger, and holes, as they are more likely to encounter such sites during migration. This has the overall effect of giving a smoother growth surface for lower P_s (The surface roughness could be extracted from the three-dimensional array of data generated by the model). The roughness of grain boundaries themselves is a factor that is discussed below with reference to species mobility and the likelihood of grain growth events.

Boundary Effects: An effect that has a large influence on the model's behaviour is the nature of the boundaries. In order for a grain to become larger during film growth, the plane of the boundary must be shift as layers are progressively deposited. Shown in figure 9.7a is an example of a stable boundary, which will not move from layer to layer: a species at the boundary, as shown in the diagram, will see values of $\sigma_n = 8.7$ for the light grain, compared with $\sigma_n = 4.9$ for the dark grain (see equation 9.1). It will thus join the light grain (if it joins a grain at all, of course).

In figures 9.7b and c are shown two boundaries which can move as growth progresses, allowing a grain to expand as the film becomes thicker. However, it is clear that these constitute high entropy cases, which become

progressively more unlikely as P_s decreases, due to the migration effects discussed above.

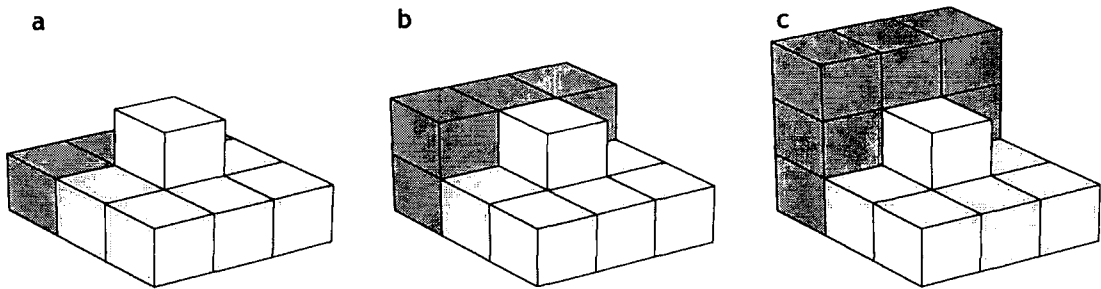


Figure 9.7: Three cases of a migrating species (white) arriving at a planar boundary between two extant grains (light and dark grey). In a) the species will join the light grey grain, in b) the grain it joins is random, and in c) it must join the dark grey grain, making the plane of the boundary shift, causing the dark grain to grow.

This, then, demonstrates why models with high values of P_s exhibit greater grain growth with increasing film thickness: their boundaries show greater disorder, and are less likely to have achieved any form of stability: they therefore have higher mobility. It should be noted that intersections between boundaries are never stable, thus there must be some boundary migration regardless of P_s .

A very similar argument can account for the roughness of the grain size versus layer number plots, shown in figure 9.5, in cases of high P_s . Not only are these conditions more likely to form high entropy boundaries, but also rough ones (in three dimensions), due to inferior hole filling during layer growth. This leads to greater boundary mobility, but if the driving force for growth is small, this will lead to large changes in structure between layers, but not necessarily an overall increase in grain size.

Small Grain Effects: As was mentioned earlier in this section, it appears that a different mechanism appears to be responsible for the rapid

near-interface grain growth, which only affects small grains. Such a mechanism is now described. In the extreme case of a flat growth surface, consisting of unit grains (shown in figure 9.8a), at any point on this substrate, a migrating species will experience σ_n of $1/\sqrt{2}$ for all grains apart from that directly below it (with $\sigma_n = 1$). Thus, wherever it finally attaches, it will be part of the same grain as the pixel directly below it (figure 9.8b). Figure 9.8c shows the behaviour of a second arriving species. If this species attaches itself adjacent to the first, it *must* join the same grain as it. Arriving species will continue to behave in this fashion, subsuming unit grains (figure 9.8d), until the grain encounters another behaving in the same fashion (figure 9.8e).

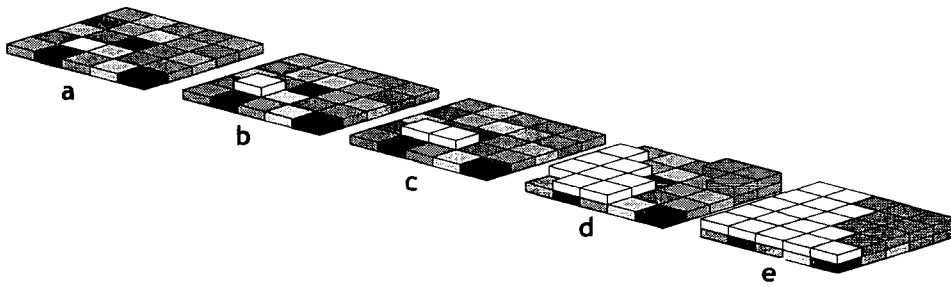


Figure 9.8: A series of diagrams demonstrating the behaviour of species arriving at a unit-grain substrate (a). An arriving species *must* attach itself to the grain directly below it (b). Similarly, subsequent species migrating to positions adjacent to the first species must attach to the original species (c and d). This continues until this outgrowth encounters another grain behaving in the same fashion (e).

Thus for very small grains, the rate of increase of mean grain size is very rapid, with the amount of increase proportional to the number of very small grains. It should be noted that, although this can be regarded as a different mechanism to that responsible for grain growth in the bulk region, it could be perceived as an extreme case of the grain boundary linearity argument proposed above.

9.4.3 THE PHYSICALITY OF THE MODEL

Although the model reflects the behaviour of the experimental data, and the way in which the model functions has been speculated upon, how these mechanisms relate to reality needs to be discussed.

In the near-interface region, where the growth is dictated by the requirement of arriving species to bond directly above small grains, little parallel can be drawn with reality. If this mechanism were to be extended to real systems, it would imply an amorphous substrate (which could be regarded as polycrystalline, but with one atom per grain); the difference in scale between the 'species' of the model and atoms has already been emphasised— the grain pixels and arriving species do not represent atoms but represent 'blocks' of material. Thus, whilst this mechanism indicates a very powerful driving force for grain-growth, its applicability to real systems cannot be guaranteed.

Turning to the bulk of the layer, some reasonable comparisons can be drawn. It has been shown that, in the model, boundaries with high curvature have a tendency to move, whereas linear boundaries are stable. Section 3.3 describes how this behaviour occurs in reality to minimise boundary free energy, albeit in terms of grain growth. Whilst the processes in the model are on a different scale, these could be scaled directly to atomic dimensions. However, it should be noted that the failure of the grain size distributions to adhere to the Rayleigh distribution (figure 9.4), implies the presence of other growth processes in real material.

9.5 CONCLUSIONS

In this chapter, a simple model was described and implemented in order to help explain the measured variation in grain size seen in

polycrystalline layers of CdTe. Overall, the model displayed many of the gross characteristics of grain size variation resulting from growth in polycrystalline material. The form of the predictions agree remarkably well with experiment. Moreover the model provides some pointers to the physical processes that govern grain size development during growth, namely small grain effects and the relationship of sticking and hence boundary roughness to growth. Nevertheless, it is recognised that the success of the model does not constitute proof that the model is atomistically correct, as the pixels of the grain structure represent blocks of material.

9.6 REFERENCES

- [1] R. W. Smith *A Kinetic Monte Carlo Simulation of Fiber Texture Formation During Thin-Film Deposition*, Journal of Applied Physics **81** (1997) 1196-203
- [2] N. Romeo, R. Tedeschi, L. Ferrari, S. Pasquali, A. Bosio and V. Canevari *Monte Carlo Computer Simulation of the Deposition of CdTe Thin Films by Close-Spaced Sublimation*, Materials Chemistry and Physics **66** (2000) 259-65
- [3] S. W. Levine, J. R. Engstrom and P. Clancy *A Kinetic Monte Carlo Study of the Growth of Si on Si(100) at Varying Angles of Incident Deposition*, Surface Science **401** (1988) 112-23
- [4] S. Barrat, P. Pigeat and E. BauerGrosse *Three-Dimensional Simulation of CVD Diamond Film Growth*, Diamond and Related Materials **5** (1996) 276-80
- [5] M. M. Al-Jassim, F. S. Hasoon, K. M. Jones, B. M. Keyes, R. J. Matson and H. R. Moutinho *The Morphology, Microstructure and Luminescent Properties of CdS/CdTe Thin-Film Solar-Cells*, Proceedings of the 23rd IEEE Photovoltaic Specialists Conference (1993) 459-65

Chapter 10

Conclusions

This thesis has been concerned primarily with determining the microstructure of CdTe polycrystalline thin films for use in CdTe/CdS solar cells. Some attention has been paid to correlations between the changes in microstructure and improvements in electrical characteristics that occur during CdCl₂ annealing. In addition to reviewing both general and microstructural work by other authors, results have been presented giving detailed information on the microstructure of CdTe layers grown by CSS.

In chapter 6 a method was described for examining grain growth phenomena in bulk CdTe using pressed pellets of powdered CdTe. Studies of how annealing changes simple properties of pellets, such as mass and size, can provide an experimentally simple route to understanding the mechanisms behind grain growth. Thus, by producing pressed pellets of powdered CdTe, and subsequently sintering these, information was sought on the mechanisms that facilitate grain growth, with a view to applying these findings to thin-film material. A number of developments in grain structure were seen, such as the conglomeration of grains, and the formation of a densified 'crust' on some

pellets. These effects were enhanced by the addition of a CdCl_2 flux. However, no macroscopic densification effects were seen, and on elevating the temperature, to promote further structural change, significant oxidation was found to occur. It was inferred that at the temperatures below which oxidation occurred, a limiting size existed for grain-growth for the particle size ($\sim 1 \mu\text{m}$) used. Thus, this 'open system' approach to sintering gave limited information.

On further study of the oxidation of CdTe, it was found that some parallels could be drawn between standard models of oxidation (derived for single crystal silicon) and the oxidation of CdTe powder. However, the transport of oxygen to the reaction surface rapidly became dominated by transport through the pores in the pressed pellet. At this point comparisons with analytical models of surface oxidation broke down.

Following the above study, chapter 7 examined how annealing with CdCl_2 changes the microstructure of CdTe-CdS bi-layers suitable for devices. CdCl_2 is known to effect a large increase in efficiency in devices manufactured from such bi-layers. The study involved the treatment of the bi-layers (grown by CSS) for varying CdCl_2 anneal times, producing a series of samples. The samples were then bevelled, to allow the microstructure throughout the films to be examined, photographed along the bevel by optical microscopy, and the resultant micrographs analysed using computational image analysis techniques. This method of examination was suitable only for material more than $\sim 1 \mu\text{m}$ from the CdTe/CdS interface, due to the microscope's inability to resolve the small grains of the near-interface region. In addition to this, devices were manufactured from the annealed bi-layers, and characterised electrically.

A number of features were observed in this study. At all points throughout the layers, the grain size distributions were found to conform to the Rayleigh distribution, commonly observed in systems displaying normal grain growth. The mean of these distributions was found to vary linearly throughout the thickness of the region studied. However, within the limit of the technique, no grain growth was detectable in any sample, regardless of the treatment times, even on comparison to an untreated sample: the variations of grain size with position through the layer were identical. This result points to the system being driven towards equilibrium during growth. To examine further the level of equilibration, the radial and spatial correlation of the grain centroids was examined in the first report of this type of study performed on polycrystalline material. This study, however, showed no evidence of an approach to the ideal equilibrium structure, implying the existence of a limiting mechanism for grain boundary movement under the growth and sintering conditions used here. Although the CdCl_2 treatment was not found to affect the microstructure in the bulk of the layer, the J - V and spectral response measurements showed the expected increase in efficiency. This result in isolation implies that grain size in the bulk in no way dictates the performance of these particular devices, a result that is further evaluated with reference to near-interface TEM later in this chapter.

Whilst the smaller grain-size nearer to the interface could be explained simply by grain boundary pinning by the CdS , this argument cannot be used to clarify the cause of the *linear variation* in mean grain size. A number of arguments were proposed to account for this, concentrating on kinetic factors dictated by the disparate growth rates of differing crystallographic faces. However, in CSS-grown CdTe , which shows no preferred orientation, the applicability of such arguments was questioned.

As the above study had discounted bulk microstructural events as being responsible for increases in efficiency, attention was turned to the near interface region, inaccessible by the above optical method. To examine this region, a plan-view transmission electron microscopy study was undertaken, described in chapter 8. The material studied was again thin-film CdTe/CdS, CSS-grown by ANTEC GmbH. This was supplied in both the as-grown state and CdCl₂ treated by the growers' vapour-phase method. A detailed description of the sample preparation is given in the text. In contrast to the body of the film, clear evidence of grain growth was found in this study: the mean grain size increased by a factor of approximately five on CdCl₂ treatment (from 0.1 to 0.5 μm), although the distributions were still concomitant with the Rayleigh distribution. In addition to this, the size of the grains was found to be smaller by a factor of ten (for the as-grown material) than predicted by extrapolation of the linear trend of grain size found in the bulk of the films. This rapid drop-off in grain size was discussed in terms of pinning of the CdTe grain size to that of the underlying CdS. The grain growth found in this region was explained by consideration of the differences between the bulk and the near interface grain size: small-grained material has a higher propensity to grow due to the higher free energy arising from the higher density of grain boundaries.

In addition to the grain size considerations, a high number of both planar defects and dislocations were found in the material. The density of the latter, however, was found to be dramatically lower in the treated material, but with a corresponding increase in sub-grain boundaries. This behaviour is indicative of the action of a recovery-like mechanism and is the first report of this mechanism in CdTe/CdS solar cells. The presence of the recovery-formed defects in the treated material discounted the occurrence of recrystallisation

on annealing of this material: the sub-grain boundaries would be consumed in such a process.

By using a simple model, the way in which the increase in grain size might cause the increase in device efficiency was considered. This assumed that the capture cross-section of each grain boundary was independent of treatment, and examined how the total capture area due to grain boundaries varied on grain growth, in order to explain the measured change in J_{sc} . By using grain sizes and J_{sc} values measured in this work, a value for grain-boundary electron capture width of 32 nm was obtained. This appears to be physically reasonable, although no comparative value exists within the literature. However, this simplified view of the grain-boundary behaviour is at odds with a number of literature reports: whilst the model has boundaries unchanged by CdCl_2 treatment, a number of authors have reported considerable passivation of the boundaries on annealing. Moreover, the band-structure associated with grain boundaries has been reported upon and, whilst not forming a complete picture the literature refutes the simplified constant capture cross-section model depicted here. Thus, it was concluded that the model was perhaps simplistic. This TEM study has shown, however, that considerable microstructural change occurs in the near-interface region on CdCl_2 treatment, even in material that was considerably stabilised during growth. It may be considered likely that such changes in interface structure could not be responsible, in some part, for the noted change in device parameters.

In the final chapter of experimental results in this work (chapter 9), the measured variation of grain size throughout the CdTe layer— the steady, linear decrease in the body of the film, followed by a rapid drop at the

interface— was studied using a simple computer model. This was designed to attempt to give insight into the processes that give the two-regime behaviour, by dictating how arriving species attached to the growth surface (and initially the substrate). On running this model, a structure highly comparable to real material was developed. In addition to displaying columnar growth, with the correct choice of parameters, a high degree of correlation with experiment was found, in terms of the shape of the simulated grain-size distribution throughout the layer. The model showed both the two-regime behaviour expected and lateral growth of grains as the layer became thicker. It was speculated that the rate of lateral grain expansion in the bulk region was dictated by the roughness and linearity of grain boundaries, which was in turn dictated by the sticking coefficient associated with the arriving species. However, in considering the probable mechanisms behind the model's behaviour in the near-interface region, it was noted that this material was grown in a manner not extensible to a physical process. Furthermore, it was found that the grain size distribution in the simulated layers was not concordant with the Rayleigh distribution, which described real grain size distribution as well. Overall, this model was found to be able to describe, but not predict, phenomena found in real materials.

Further to the work described in this thesis, a number of similar studies may prove fruitful in further revealing whether CdCl_2 is modifying the electrical characteristics of the grain boundaries or the grain material, or causing microstructural effects that effect changes in carrier transport. A high temperature treatment of a CdTe layer in an inert atmosphere could be performed, without the addition of the CdCl_2 flux. This should cause microstructural change in the near-interface region, and any change in

efficiency would be caused purely by microstructural factors. A similar, converse avenue of exploration would be to characterise CdTe film conductivity in the direction of the columnar grains. If thin-films could be manufactured with a large and invariant grain-size throughout, changes in conductivity on CdCl₂ treatment could be ascribed solely to bulk electrical effects; this work has shown that large grained microstructure is invariant on annealing. However, this type of study does not address other effects of the CdCl₂ treatment, such as intermixing of the CdTe and CdS, which has a pronounced optical and electrical effect.

Finally, it is hoped that this work has reflected the complexity of the interactions between materials and their processing in thin-film polycrystalline CdTe/CdS solar cells. Furthermore, the author would be glad to think that this work had made these interworkings, in some small way, more transparent.

Appendix **A**

C++ Code: Image Analysis

A.1 INTRODUCTION

This and the following two appendices reproduce the C++ routines used in this work. The present appendix describes and reproduces the code used for the image analysis software of chapter 7. Appendix B describes the Levenberg-Marquardt non-linear least-squares fitting method and the associated code, used in sections 7.2.6 for fitting step functions, and throughout for fitting Rayleigh functions and similar. Appendix C then does similar for the modelling software of chapter 9.

In all three cases the code is reproduced in full for the sake of reproducibility and completeness, but the density of annotations depends on the salience of the routine in question: some utility routines have only comments as to their purpose. In addition, some routines common to all three programs described in these appendices are listed only in this appendix.

The code was compiled in Microsoft Visual Studio, although efforts have been made to ensure the code is reasonably portable.

A.2 IMAGE ANALYSIS

A.2.1 SPATIAL FILTERING

The spatial filtering technique is described in section 5.6.1. As mentioned there, this implementation of a spatial filter will only apply a centro-symmetric 3×3 matrix:

$$\begin{bmatrix} f_2 & f_2 & f_2 \\ f_2 & f_1 & f_2 \\ f_2 & f_2 & f_2 \end{bmatrix}$$

where the values f_1 and f_2 shown here correspond to the variables passed to the routine. This version of the routine truncates values that do not lie in the range $0 \leq x \leq 255$ (via the subroutine `charize`), rather than storing all the values as floating-point, and compressing these into the 8-bit integer range later on: this choice is arbitrary, depending on the desired functionality.

```
#define CHAR_DATA
#include "matrix.h"

//-----

//Routine to apply a centro-symmetric spatial
//filter to a grey-scale image.
//f1 is the value of the central filter element,
//f2 the value of the surrounding elements.
void gen_filter (c_matrix **transin,
                char f1, char f2, usc bias){
    usc charize(float z);
    usc i, j;
    char ip, jp;
    float avgrad = 0;
    usc charize (float);
    c_matrix *in = *transin;

    //define working space for the routine two
    //pixels smaller than the input, as the edge
    //pixels are ignored.
    c_matrix *out = new c_matrix(in->lr,
                                in->hr - 2, in->lc, in->hc - 2);
    if (out->errmsg != NULL)

        mainerror("Memory error: gen_filter");

    //Raster through the image (not the
    //edges, though).
    for (j = in->lr + 1; j <= in->hr - 1; ++j)
        for (i = in->lc + 1; i <= in->hc - 1; ++i){
            //For each pixel, take the sum of the
            //filter element times the pixel value...
            for (jp = -1; jp <= 1; ++jp)
                for (ip = -1; ip <= 1; ++ip)
                    avgrad += (float)((ip == 0 && jp == 0)?
                                   f1: f2) * (float)in->get
                                   (i + ip, j + jp);

            //...and put this into the working space.
            out->set(i - 1, j - 1, charize(avgrad /
                                           (8 * f2 + f1 + 1)) + bias);
            avgrad = 0;
        }

    // The working space becomes the output
    //matrix, and the input matrix is deleted.
    *transin = out;
    delete in;
}
```

A.2.2 SKELETONISATION

The following code is that described in section 5.6.3. It takes as its input a noisy boundary image, and reduces this to a map consisting only of grains, with boundaries of zero width. This is accomplished in two stages:

i) assigning unique values to each grain in the image (get_map), and ii) dilating the grains iteratively until the boundaries are gone (skel_map).

```
#define CHAR_DATA
#include "matrix.h"
#define USHORT_DATA
#include "matrix.h"
enum FILL_TYPE{FFHARD, FFSOFT};

//-----
//Main routine for the skeletonisation which
//calls the two subroutines.
uss skeletonise(c_matrix *in, s_matrix *out){

    uss grain_map(c_matrix *in, s_matrix *out);
    void skel_map(s_matrix *transin);
    uss i, j;
    uss grain_count;

    //Output matrix is zeroed.
    for(j = out->lr; j <= out->hr; ++j)
        for (i = out->lc; i <= out->hc; ++i)
            out->set(i, j, 0);

    //First part of skeletonisation.
    grain_count = grain_map(in, out);

    //Second part of skeletonisation.
    skel_map(out);

    return grain_count;
}

//-----
//This subroutine takes a monochrome image of noisy
//black boundaries on white, and outputs the same
//image, but with the white replaced by unique
//values for each different grain.
uss grain_map(c_matrix *in, s_matrix *out){

    usl flood_fill(c_matrix *pic,
        s_matrix *map, short i, short j, uss no,
        FILL_TYPE type, usc BND, usc ABND);
    uss i, j, no = 0;

    //Iterate through the image...
    for(j = in->lr; j <= in->hr; ++j)
        for (i = in->lc; i <= in->hc; ++i)

            //...if a pixel is white...
            if (in->get(i, j) == 255){

                //...fill the region with a unique colour.
                no++;
                flood_fill(in, out, i, j,
                    no, FFSOFT, 255, 0);
            };

    //Return the number of grains for use later.
    return no;
}

//-----
//This subroutine takes the output of the above
//routine, and dilates the grains until the
//boundaries are of zero width.
void skel_map(s_matrix *transin){

    uss x, y, temp, maxno, maxgno;
    char xp, yp;
    bool IsFinished;
    usc n, k;
    s_matrix *swap;

    //Define pointer to original array.
    s_matrix *in = transin;

    //Define new array like the input picture as
    //working space.
    s_matrix *skl = new s_matrix(
        in->lr, in->hr, in->lc, in->hc);

    //Define matrix used to store colour of the
    //8-neighbours and the number of each.
    s_matrix *surs = new s_matrix(1, 8, 0, 1);

    //Main loop.
    do{
        IsFinished = true;

        //Iterate through the image.
        for (y = in->lr; y <= in->hr; ++y)
```

```
        for (x = in->lc; x <= in->hc; ++x){

            //If the pixel is not boundary, set as
            //such in work space, and go to next
            //pixel...
            if (in->get(x, y) != 0) {
                skl->set(x, y, in->get(x, y));
                continue;
            }

            //...otherwise initialise required
            //variables...
            IsFinished = false;
            n = 0;
            for (k = 1; k <= 8; ++k)
                surs->set(1, k, 1);

            //...and look at the pixel's 8-neighbours,
            //checking that they lie within
            //the image...
            for (yp = -1; yp <= 1; ++yp)
                for (xp = -1; xp <= 1; ++xp){
                    if (x + xp < in->lc ||
                        x + xp > in->hc ||
                        y + yp < in->lr ||
                        y + yp > in->hr)
                        continue;

                    //...ignore it if it's boundary...
                    if ((temp = in->get(
                        x + xp, y + yp)) == 0)
                        continue;

                    //...otherwise check whether there's
                    //already one recorded of that colour,
                    //and if so, increment the record...
                    for (k = 1; k <= n; ++k)
                        if (temp == surs->get(0, k)){
                            surs->set(1, k,
                                surs->get(1, k) + 1);
                            break;
                        }

                    //...else create a new entry.
                    if (k == n + 1){
                        ++n;
                        surs->set(0, n, temp);
                    }
                }

            //Find the most frequently occurring of
            //the non-boundary 8-neighbours. If
            //there are none of these, then it
            //remains as boundary.
            maxgno = 0;
            if (n != 0)
                for (maxno = 0, k = 1; k <= n; ++k)
                    if (surs->get(1, k) > maxno){
                        maxno = surs->get(1, k);
                        maxgno = surs->get(0, k);
                    }

            //Store this in the working space.
            skl->set(x, y, maxgno);
        }

        //After examining the whole image, swap
        //the two matrices, and iterate if any
        //pixels were changed.
        swap = in;
        in = skl;
        skl = swap;
    }while(!IsFinished);

    //Make sure 'in' points to the input/output
    //matrix otherwise the data matrix will be
    //deleted below (both matrices should contain
    //the same image now).
    if (in != transin){
        swap = in;
        in = skl;
        skl = swap;
    }

    //The input/output matrix now contains the
    //skeletonised image.
    delete skl;
    delete surs;
}
```

A.2.3 FLOOD FILLING

The flood-filling routine described in section 5.6.2 is now reproduced. This implementation of the basic algorithm has considerably more variables

passed to it than is necessary for a basic routine of this type: this is, however, to allow the same routine to be used at a number of points in the program.

```
#define USHORT_DATA
#include "matrix.h"
#define CHAR_DATA
#include "matrix.h"

//Variable to describe fill type.
enum FILL_TYPE{FFHARD, FFSOFT};

//-----
//A non-recursive flood-filling routine,
//specialised for use with this program.
usi flood_fill(c_matrix *pic, s_matrix *map,
              short i, short j, uss no,
              FILL_TYPE type, usc BND, usc ABND){
    short ip, jp;
    usi k;
    struct coord{
        short x;
        short y;
    };
    coord prev[100000];
    coord cur[100000];
    usi n, nc;

    //Initialise variables.
    usi N = 1, np = 1;
    prev[1].x = i; prev[1].y = j;
    pic->set(i, j, ABND);
    map->set(i, j, no);

    //Main loop.
    do{
        n = 0;

        //Iterate through previously changed pixels.
        for (k = 1; k <= np; ++k)

            //Examine their neighbours (4- or 8- for
            //soft or hard respectively), to see if
            //they need filling. Ensure examined pixel
            //lies within the image
            for (jp = -1; jp <= 1; ++jp)
                for (ip = -1; ip <= 1; ++ip){
                    if (type == FFSOFT)

                        if (abs(ip) + abs(jp) != 1) continue;
                        if ((ip == 0 && jp == 0)
                            || prev[k].x + ip < pic->lc
                            || prev[k].x + ip > pic->hc
                            || prev[k].y + jp < pic->lr
                            || prev[k].y + jp > pic->hr)
                            continue;

                        //If pixel requires changing, log it in
                        //the 'cur' matrix, and change it in
                        //the image arrays.
                        if (pic->get(
                            prev[k].x + ip, prev[k].y + jp) == BND
                            && map->get(
                                prev[k].x + ip, prev[k].y + jp) == 0){
                            ++n;
                            ++N;
                            cur[n].x = prev[k].x + ip;
                            cur[n].y = prev[k].y + jp;
                            map->set(
                                prev[k].x + ip,
                                prev[k].y + jp, no);
                            pic->set(
                                prev[k].x + ip,
                                prev[k].y + jp, ABND);
                        }
                    }

                //Put the 'cur' values into the 'prev' array
                //and iterate unless no pixels have
                //been changed.
                np = n;
                nc = 1;
                do{
                    prev[nc].x = cur[n].x;
                    prev[nc].y = cur[n].y;
                }while (++nc <= np);

            }while(n != 0);

        return N;
    }
}
```

A.3 DRIVER ROUTINE

This section reproduces a typical driver routine for the image analysis routines. The Sort_Params routine performs parsing of the command-line switches and calls the main analysis subroutine, Reduce_Image. This in turn calls the subroutines to perform the image analysis steps (outlined in section 5.6 and listed in the sections above), plus some further routines for general purposes: these are reproduced in later sections.

The 'mixed file' referred to in the code is a copy of the original image with the grain-structure from the analysis overlaid. This is a very useful tool for ascertaining the efficacy of the analysis performed.

```
const char Intro[] =
    "Grain Image Analysis Program."
    " (c)2000 by M.A.Cousins.\n\n";
const char Helpstuff[] =
    "This Program is suitable for greyscale "
    "tessellated grain images.\n"
    "Syntax:\n"
    "\t[executable path]cgs2"
    "\t'Input File Name' [/switch1] [/switch2] [etc...\n"
    "\tSwitches :-\n"
    "\t\t/o<output filename> (default = c:\\gg.bmp)\n"
    "\t\t/d<data filename> (default = c:\\gg.dat)\n"
    "\t\t/x<mixed file> (def = c:\\mix.bmp)\n"
    "\t\t/r<percentage size reduction> (def = 100)\n"
```

```

"\t/c<pixel size in metres> (def = 1)\n"
"\t/g (turns on guard frame)\n"
"\t/m<minimum allowed grain size> (def = 0)\n"
"\t/? (displays this help)\n";

#include "dtype.h"
#include <string.h>
#include "bitmap.h"
#include <time.h>
#define USHORT_DATA
#include "matrix.h"

char *DEF_PICOUT = "c:\\gg.bmp";
char *DEF_DATOUT = "c:\\gg.dat";
char *DEF_MIXOUT = "c:\\mix.bmp";

//-----
//Main routine.
int main(int argc, char* argv[]){

    void Sort_Params(int argc,
        char **argv[], char **infile,
        char **outfile, char **datfile,
        char **mixfile, usc *redfac,
        usc *minsize, float *con_fac,
        bool *DoHelp, bool *IsGuarded);

    void Reduce_Image(char imagein[],
        char imageout[], char datfile[],
        char mixfile[], usc perc_redn,
        usc min_size, float con_fac,
        bool IsGuarded);

    char *inf = NULL, *outf = NULL, *mixf = NULL;
    bool DoHelp = false;
    bool IsGuarded = false;
    usc redn_fac = 0, min_size = 0;
    float con_fac = 0.0F;

    printf(Intro);

    Sort_Params(argc, &argv, &inf, &outf, &datf,
        &mixf, &redn_fac, &min_size,
        &con_fac, &DoHelp, &IsGuarded);

    if (DoHelp)
        printf(Helpstuff);
    else
        Reduce_Image(inf, outf, datf, mixf, redn_fac,
            min_size, con_fac, IsGuarded);
    return 0xfec;
}

//-----
//Command-line parser, which returns default
//values if switch is unused.
void Sort_Params(int argc,
    char **argv[], char **infile,
    char **outfile, char **datfile,
    char **mixfile, usc *redfac,
    usc *minsize, float *con_fac,
    bool *DoHelp, bool *IsGuarded){

    int i;
    usc c1,c2;

    if (argc == 1) mainerror(
        "Input file required. \nUse /? for help.");

    *infile = (argv[0][1]);
    if (strlen(*infile) == 0)
        mainerror(
            "Input file is a zero length string. "
            "\nUse /? for help.");

    for (i = 2; i <= argc - 1; ++i){
        int tmp;
        double dtmp;
        c1 = *argv[0][i];
        if (c1 != '/') mainerror(
            "Invalid Argument. \nUse /? for help.");
        c2 = *(argv[0][i] + 1);

        switch (c2){
            case 'o':
            case 'O':
                *outfile = (argv[0][i] + 2);
                if (strlen(*outfile) == 0)
                    mainerror(
                        "Output file (/o) is a zero length"
                        "string. \nuse /? for help.");
                break;

            case 'd':
            case 'D':
                *datfile = (argv[0][i] + 2);
                if (strlen(*datfile) == 0)
                    mainerror(
                        "Data output file (/d) is a zero length"
                        "string. \nuse /? for help.");
                break;

            case 'x':
            case 'X':
                *mixfile = (argv[0][i] + 2);
                if (strlen(*mixfile) == 0)
                    mainerror(
                        "Mix file (/x) is a zero length string."
                        "\nUse /? for help.");
                break;

            case 'r':
            case 'R':
                tmp = atoi(argv[0][i] + 2);
                *redfac = (usc) tmp;

```

```

        if (tmp < 10 || tmp > 100)
            mainerror (
                "The reduction parameter (/r) must be"
                "between 10 and 100 percent.\nUse /? "
                "for help.");
        break;

    case '?':
        *DoHelp = true;
        break;

    case 'm':
    case 'M':
        tmp = atoi(argv[0][i] + 2);
        *minsize = (usc) tmp;
        if (tmp < 0 || tmp > 255)
            mainerror (
                "Minimum grain area (/m) must lie "
                "between 0 and 255 (pixels). "
                "\nuse /? for help.");
        break;

    case 'c':
    case 'C':
        dtmp = atof(argv[0][i] + 2);
        *con_fac = (float) dtmp;
        if (*con_fac == 0.0F)
            mainerror (
                "Pixel width cannot be zero."
                "\nuse /? for help.");
        break;

    case 'g':
    case 'G':
        *IsGuarded = true;
        break;

    default:
        mainerror(
            "Invalid Argument."
            "\nUse /? for help.");
    }
}

if (*infile == NULL && *DoHelp == false)
    mainerror("Input file required."
        "\nuse /? for help.");

if (*outfile == NULL) *outfile = DEF_PICOUT;
if (*datfile == NULL) *datfile = DEF_DATOUT;
if (*mixfile == NULL) *mixfile = DEF_MIXOUT;
if (*redfac == 0) *redfac = 100;
if (*minsize == 0) *minsize = 0;
if (*con_fac == 0.0F) *con_fac = 0.0F;
}

//-----
//The analysis subroutine.
void Reduce_Image(char imagein[],
    char imageout[], char datfile[],
    char mixfile[], usc perc_redn,
    usc min_size, float con_fac,
    bool IsGuarded){

    void gen_filter (C_matrix **transin,
        char f1, char f2,usc bias);
    void negative(C_matrix *in);
    usc skeletonise(C_matrix *in, s_matrix *out);
    void resize(C_matrix **transin, usc red_fac);
    void get_boundaries(C_matrix *in,
        usc BND,usc ABND);
    void kill_minors(C_matrix *mono,
        s_matrix *grainmap, usc min_size, usc gno);
    void clean(s_matrix *in, usc gno);
    void save_to_bitmap(C_matrix *in, char* fname);
    void Get_Bitmap(char *fname, C_matrix **out);
    void get_data(s_matrix *in, usc no,
        C_matrix *out, char* datfile,
        float con_fac, bool IsGuarded);

    long start = (long) clock();
    srand( (unsigned)time( NULL ) );

    C_matrix *ws;
    C_matrix *wstmp;

    //Open bitmap.
    printf("Opening %s...\n", imagein);
    Get_Bitmap(imagein, &ws);

    //Reduce size.
    if (perc_redn != 100){
        printf(
            "Reducing size by %u%%...\n",
            perc_redn, '%');
        resize( &ws, perc_redn);
    }

    //If a mixed file is required, make a copy of
    //the original image.
    if (mixfile != NULL){
        short i,j;
        wstmp = new C_matrix (ws->lr, ws->hr,
            ws->lc, ws->hc);
        for (j = ws->lr; j <= ws->hr; ++j)
            for (i = ws->lc; i <= ws->hc; ++i)
                wstmp->set(i, j, ws->get(i, j));
    }

    //Apply softening filter.
    printf("Applying softening filter...\n");
    gen_filter(&ws, 3, 2, 0);

```

```
//Apply edge-finding filter.
printf("Applying edge finding filter...\n");
gen_filter(&ws, 8, -1, 0);

//Find main boundary structure.
printf("Finding boundary region...\n");
negative(ws);
get_boundaries(ws, 255, 0);

//Skeltonise and extract data.
s_matrix *sm = new s_matrix(ws->lr, ws->hr,
ws->lc, ws->hc);
if (sm->errmsg != NULL) mainerror(
"Memory error: sm");
printf("Skeltonising...\n");
uss gno = skeletonise(ws, sm);

//Remove small grains if a minimum size
//was defined.
printf("Cleaning up...\n");
if (min_size != 0) kill_minors(ws, sm,
min_size, gno);

//Get rid of any grains within other grains
//((Usually arising from image noise).
clean(sm, gno);

//Reduce image to data file and output.
printf("Producing data file %s...\n", datfile);
get_data(sm, gno, ws, datfile, con_fac, IsGuarded);

//Print output image to file.
printf("Producing output image %s...\n", imageout);
save_to_bitmap(ws, imageout);

//Create mixed file, if required.
if (mixfile != NULL){
printf("Producing mixed image %s...\n", mixfile);
```

```
short i,j;
bitmap *mixop = new bitmap(
ws->hc - wstmp->lc + 1,
wstmp->hr - wstmp->lr + 1,
BMP_24BIT);
for (j = wstmp->lr; j <= wstmp->hr; ++j)
for (i = wstmp->lc; i <= wstmp->hc; ++i){
mixop->setpixel(i + 1, j + 1,
RED, wstmp->get(i, j));
mixop->setpixel(i + 1, j + 1,
GREEN, wstmp->get(i, j));
mixop->setpixel(i + 1, j + 1,
BLUE, wstmp->get(i, j));
}
for (j = ws->lr; j <= ws->hr; ++j)
for (i = ws->lc; i <= ws->hc; ++i)
if (ws->get(i, j) == 0){
mixop->setpixel(i + 3, j + 3, RED, 255);
mixop->setpixel(i + 3, j + 3, GREEN, 0);
mixop->setpixel(i + 3, j + 3, BLUE, 0);
}
mixop->save(mixfile);
delete mixop;
}

//Clean up and exit.
if (mixfile != NULL)
delete wstmp;
delete sm;
delete ws;

long stop = (long) clock();
printf("Analysis complete in %0.2f seconds.\n",
(double)(stop-start)/CLOCKS_PER_SEC);
}
```

A.4 MISCELLANEOUS ROUTINES

A.4.1 IMAGE MANIPULATION ROUTINES

This section reproduces subroutines, which, whilst not strictly falling into the category of image analysis tools, are vital for general image manipulation.

```
#include <math.h>
#define CHAR_DATA
#include "matrix.h"
#define USHORT_DATA
#include "matrix.h"

enum FILL_TYPE{FFHARD, FFSOFT};
const double ROOT2 = sqrt(2);

//-----
//Routine to resize a grey-scale image in an array.
void resize(c_matrix **transin, usc red_fac){
uss i,j,m,n;
float x1,x2,y1,y2;
float abovegive1(float f);
float abovegive0(float f);
float fraction;
float total, divtotal;
c_matrix *in = *transin;
c_matrix *out = new c_matrix(in->lr,
(uss)((float) in->hr *
(float) red_fac / 100.0F),
in->lc
(uss)((float) in->hc *
(float) red_fac / 100.0F));
for(j = out->lr; j <= out->hr; ++j)
for (i = out->lc; i <= out->hc; ++i){
x1 = ((float) in->hc /
(float) out->hc) * (float) i;
x2 = ((float) in->hc /
(float) out->hc) * (float) (i + 1);
y1 = ((float) in->hr /
(float) out->hr) * (float) j;
y2 = ((float) in->hr /
(float) out->hr) * (float) (j + 1);
total = 0.0F;
divtotal = 0.0F;
for (n = (uss) y1; n <= (uss) y2; ++n)
for (m = (uss) x1; m <= (uss) x2; ++m){
fraction =
(abovegive1( (float) m + 1.0F - x1 ) -
```

```
abovegive0( (float) m + 1.0F - x2 ) ) *
(abovegive1( (float) n + 1.0F - y1 ) -
abovegive0( (float) n + 1.0F - y2 ) );
if (m <= in->hc && n <= in->hr){
divtotal += fraction;
total +=
(float) in->get(m, n) * fraction ;
}
}
out->set(i, j, (usc) (total / divtotal));
}

*transin = out;
delete in;
}

//-----
//Inverts a greyscale image
void negative(c_matrix *in){
uss i,j;
for (j=in->lr; j <= in->hr; ++j)
for (i = in->lc; i <= in->hc; ++i)
in->set(i,j,255-in->get(i,j));
}

//-----
//Flood fills all regions which are white, then
//rejects all filled regions except the largest
void get_boundaries(c_matrix *in, usc BND,usc ABND){
uss1 flood_fill(c_matrix *pic, s_matrix *map,
short i, short j, usc no,
FILL_TYPE type, usc BND, usc ABND);
uss i,j;
s_matrix *tmp = new s_matrix(in->lr, in->hr,
in->lc, in->hc);
```

```

uss no = 0;
usl temp, maxval=0, maxpos=0;

for(j = in->lr; j <= in->hr; ++j)
  for(i = in->lc; i <= in->hc; ++i)
    if (in->get(i, j) == BND){
      no++;
      if ((temp = flood_fill(in, tmp, i, j, no,
        FFSOFT, BND, ABND)) > maxval){
        maxval = temp;
        maxpos = no;
      }
    }

  }

for(j = in->lr; j <= in->hr; ++j)
  for(i = in->lc; i <= in->hc; ++i){
    if (tmp->get(i, j) == maxpos)
      in->set(i, j, 0);
    else
      in->set(i, j, 255);
  }

delete tmp;
}

//-----
//Subroutine to get rid of grains below the
//minimum allowed size.
void kill_minors(c_matrix *mono,
  s_matrix *grainmap, usc min_size, usc gno){

  short i, j, k, nr = 0;
  usc n;
  float rnd01(void);
  void get_plusses(char_coord in[9]);
  char_coord plusses[9];
  bool GotOne = true;

  s_vector *rejects = new s_vector(1, gno);
  get_plusses(plusses);
  s_vector *area = new s_vector(1, gno);

  for(j = grainmap->lr; j <= grainmap->hr; ++j)
    for(i = grainmap->lc; i <= grainmap->hc; ++i)
      area->incr(grainmap->get(i, j));

  nr = 0;

  for(i = 1; i <= gno; ++i)
    if (area->get(i) <= min_size && area->get(i) > 0)
      rejects->set(++nr, i);

  delete area;

  while(true){
    for(j = grainmap->lr; j <= grainmap->hr; ++j)
      for(i = grainmap->lc; i <= grainmap->hc; ++i){
        for(k = 1; k <= nr; ++k)
          if (grainmap->get(i, j) == rejects->get(k))
            break;
        if (k == nr + 1) continue;
        usc os = (usc) (rnd01() * 8.0F);
        for(n = 0; n <= 7; ++n){
          if (i + plusses[(n + os) % 8].ip <
            grainmap->lc ||
            j + plusses[(n + os) % 8].jp <
            grainmap->lr ||
            i + plusses[(n + os) % 8].ip >
            grainmap->hc ||
            j + plusses[(n + os) % 8].jp >
            grainmap->hr)
            continue;
          if (grainmap->get(
            i + plusses[(n + os) % 8].ip,
            j + plusses[(n + os) % 8].jp) !=
            rejects->get(k)){
            grainmap->set(i, j, grainmap->get(
              i + plusses[(n + os) % 8].ip,
              j + plusses[(n + os) % 8].jp));
            break;
          }
        }
      }
  }

  area = new s_vector(1, gno);
  for(j = grainmap->lr; j <= grainmap->hr; ++j)
    for(i = grainmap->lc; i <= grainmap->hc; ++i)
      area->incr(grainmap->get(i, j));

  uss onr = nr;
  nr = 0;
  for(i = 1; i <= gno; ++i)
    if (area->get(i) <= min_size && area->get(i) > 0)
      rejects->set(++nr, i);
  delete area;
  if (nr == 0) break;
}

delete rejects;
}

//-----
//Subroutine to delete any grains within grains.
void clean(s_matrix *in, usc gno){
  short i, j, n;
  enum {X1, X2, Y1, Y2, A};
  uss tempval;
  usl flood_fill(c_matrix *pic, s_matrix *map,
    short i, short j, usc no,
    FILL_TYPE type, usc BND, usc ABND);
  void SAB(c_matrix *in, char* fname);
  s_matrix *posits = new s_matrix(1, gno, X1, A);
  c_matrix *ws = new c_matrix(in->lr, in->hr,
    in->lc, in->hc);
  s_matrix *temp = new s_matrix(in->lr, in->hr,
    in->lc, in->hc);

  for(n = 1; n <= gno; ++n){
    posits->set(X1, n, in->hr);
    posits->set(Y1, n, in->hc);
  }

  for(j = in->lr; j <= in->hr; ++j)
    for(i = in->lc; i <= in->hc; ++i){
      tempval = in->get(i, j);
      posits->incr(A, tempval);
      if (i < posits->get(Y1, tempval))
        posits->set(Y1, tempval, i);
      if (i > posits->get(Y2, tempval))
        posits->set(Y2, tempval, i);
      if (j < posits->get(X1, tempval))
        posits->set(X1, tempval, j);
      if (j > posits->get(X2, tempval))
        posits->set(X2, tempval, j);
    }

  for(n = 1; n <= gno; ++n){
    if (posits->get(A, n) == 0)
      continue;
    c_matrix *ws = new c_matrix(
      -1, posits->get(X2, n) -
        posits->get(X1, n) + 1,
      -1, posits->get(Y2, n) -
        posits->get(Y1, n) + 1);
    s_matrix *temp = new s_matrix(
      -1, posits->get(X2, n) -
        posits->get(X1, n) + 1,
      -1, posits->get(Y2, n) -
        posits->get(Y1, n) + 1);

    for(j = 0; j <= posits->get(X2, n) -
      posits->get(X1, n); ++j)
      for(i = 0; i <= posits->get(Y2, n) -
        posits->get(Y1, n); ++i)
        if (in->get(i + posits->get(Y1, n),
          j + posits->get(X1, n)) == n)
          ws->set(i, j, 255);

    flood_fill(ws, temp, -1, -1, 128, FFSOFT, 0,
      255);

    for(j = 0; j <= posits->get(X2, n) -
      posits->get(X1, n); ++j)
      for(i = 0; i <= posits->get(Y2, n) -
        posits->get(Y1, n); ++i)
        if (temp->get(i, j) != 128)
          in->set(i + posits->get(Y1, n),
            j + posits->get(X1, n), n);
  }
}

```

A.4.2 GENERAL AND FILE I/O ROUTINES

This section reproduces, in a cursory fashion, those routines needed for miscellaneous tasks.

```

#include <math.h>
#define CHAR_DATA
#include "matrix.h"
#define USHORT_DATA
#include "matrix.h"
#define FLOAT_DATA
#include "matrix.h"
#include "bitmap.h"

```

```

#include <string.h>
enum FILL_TYPE{FFHARD, FFSOFT};
const double pi = 4.0 * atan(1.0);
//-----
//Routine to reduce the processed image

```

```
//to a data file, and produce an image of the
//processed file.
void get_data(s_matrix *in, uss no,
             c_matrix *out, char* datfile,
             float con_fac, bool isGuarded){

void get_frame(s_matrix *in, uss gno,
              float *row_min, float *row_max,
              float *col_min, float *col_max){

uss rc, cc, n;
char rp, cp;
uss temp;
uss gno;
float avarea=0;
float row_min = in->lr, row_max = in->hr;
float col_min = in->lc, col_max = in->hc;
enum{INCL, CP, RP, AREA, R, INGF};
const float PI = 3.141592F;

f_matrix *dat = new f_matrix(1,no,INCL,INGF);

for (rc = in->lr; rc <= in->hr; ++rc)
  for (cc = in->lc; cc <= in->hc; ++cc){
    gno = in->get(cc,rc);
    dat->incr(AREA, gno);
    dat->set(CP, gno, dat->get(CP, gno) + cc);
    dat->set(RP, gno, dat->get(RP, gno) + rc);
  }

for (n = 1; n <= no; ++n){
  if (dat->get(AREA,n) == 0) continue;
  dat->set(CP, n, dat->get(CP, n) /
    dat->get(AREA, n));
  dat->set(RP, n, dat->get(RP, n) /
    dat->get(AREA, n));
  dat->set(R, n,
    (float) sqrt(dat->get(AREA, n) / PI));
}

if (isGuarded)
  get_frame (in, gno, &row_min, &row_max,
             &col_min, &col_max);

bool tmplogic;

FILE *fp = fopen(datfile,"w");
if (fp == NULL) mainerror (
  "Data file error: get_data.\n");

uss cnt = 0;
if (con_fac == 0.0F) con_fac = 1.0F;
fprintf(fp, "Image width: %8.7e\n",
  con_fac * (float) (in->hc - in->lc) );
fprintf(fp, "Image height: %8.7e\n",
  con_fac * (float) (in->hr - in->lr) );
fprintf(fp, "Guard Frame x: %8.7e\t%8.7e\n",
  con_fac * (float) col_min, con_fac *
    (float) col_max);
fprintf(fp, "Guard Frame y: %8.7e\t%8.7e\n",
  con_fac * (float) row_min, con_fac *
    (float) row_max);

if (con_fac == 1.0F)
  fprintf(fp, "No\tx-pos(pels)\ty-pos(pels)\t"
    "Area(pels^2)\tradius(pels)\tIn Frame");
else
  fprintf(fp, "No\tx-pos(m)\ty-pos(m)\t"
    "Area(m^2)\tradius(m)\tIn Frame");

for (n = 1; n <= no; ++n){
  if (dat->get(AREA,n) == 0) continue;
  tmplogic = dat->get(RP, n) > row_min &&
    dat->get(RP, n) < row_max &&
    dat->get(CP, n) > col_min &&
    dat->get(CP, n) < col_max;
  dat->set(INGF, n, (float) tmplogic);
  if (isGuarded && !tmplogic) continue;
  fprintf(fp, "%8.7e\t%8.7e\t%8.7e\t%8.7e\t",
    (float) ++cnt, dat->get(CP, n) * con_fac,
    dat->get(RP, n) * con_fac,
    (uss) dat->get(AREA, n) * con_fac * con_fac,
    dat->get(R, n) * con_fac, (float) tmplogic);
}

//Produce image file.
for(rc = in->lr; rc <= in->hr; ++rc)
  for (cc = in->lc; cc <= in->hc; ++cc)
    out->set(cc,rc,255);

bool isBord;

for (rc = in->lr; rc <= in->hr; ++rc)
  for (cc = in->lc; cc <= in->hc; ++cc){
    isBord = false;
    temp = in->get(cc,rc);
    for (rp = -1; rp <= 1; ++rp)
      for (cp = -1; cp <= 1; ++cp){
        if ((cp==0 && rp == 0) ||
          cc+cp < in->lc ||
          cc+cp > in->hc ||
          rc+rp < in->lr ||
          rc+rp > in->hr) continue;
        if (in->get(cc + cp,rc + rp) != temp){
          isBord = true;
          goto breakout;
        }
      }
    }
breakout: if (isBord){
  if (dat->get(INGF, temp) < 1.0F
    && isGuarded) continue;
  out->set(cc,rc,0);
}

}

delete dat;

```

```
}

//-----
//Routine to find the edges of the guard frame
//from the processed image.
void get_frame(s_matrix *in, uss gno,
              float *row_min, float *row_max,
              float *col_min, float *col_max){

short n, rc, cc;
s_matrix *edges = new s_matrix(0, gno, 1, 4);
bool dostuff;

for (cc = in->lc; cc <= in->hc; ++cc){
  dostuff = true;
  for (n = 1; n <= edges->get(1,0) + 1; ++n)
    if (edges->get(1, n) == in->get(cc, in->lr)){
      dostuff = false;
      break;
    }
  if (!dostuff) continue;
  edges->incr(1, 0);
  edges->set(
    1, edges->get(1, 0), in->get(cc, in->lr));
}

for (cc = in->lc; cc <= in->hc; ++cc){
  dostuff = true;
  for (n = 1; n <= edges->get(2, 0) + 1; ++n)
    if (edges->get(2, n) == in->get(cc, in->hr)){
      dostuff = false;
      break;
    }
  if (!dostuff) continue;
  edges->incr(2, 0);
  edges->set(
    2, edges->get(2, 0), in->get(cc, in->hr));
}

for (rc = in->lr; rc <= in->hr; ++rc){
  dostuff = true;
  for (n = 1; n <= edges->get(3, 0) + 1; ++n)
    if (edges->get(3, n) == in->get(in->lc, rc)){
      dostuff = false;
      break;
    }
  if (!dostuff) continue;
  edges->incr(3, 0);
  edges->set(
    3, edges->get(3, 0), in->get(in->lc, rc));
}

for (rc = in->lr; rc <= in->hr; ++rc){
  dostuff = true;
  for (n = 1; n <= edges->get(4, 0) + 1; ++n)
    if (edges->get(4, n) == in->get(in->hc, rc)){
      dostuff = false;
      break;
    }
  if (!dostuff) continue;
  edges->incr(4, 0);
  edges->set(
    4, edges->get(4, 0), in->get(in->hc, rc));
}

bool gotone = true;

for (rc = in->lr; rc <= in->hr; ++rc){
  if (gotone == false) break;
  gotone = false;
  for (cc = in->lc; cc <= in->hc; ++cc)
    for (n = 1; n <= edges->get(1, 0) + 1; ++n)
      if (edges->get(1, n) == in->get(cc, rc) &&
        rc >= *row_min){
        *row_min = rc;
        gotone = true;
      }
}

gotone = true;

for (rc = in->hr; rc >= in->lr; --rc){
  if (gotone == false) break;
  gotone = false;
  for (cc = in->lc; cc <= in->hc; ++cc)
    for (n = 1; n <= edges->get(2, 0) + 1; ++n)
      if (edges->get(2, n) == in->get(cc, rc) &&
        rc <= *row_max){
        *row_max = rc;
        gotone = true;
      }
}

gotone = true;

for (cc = in->lc; cc <= in->hc; ++cc){
  if (gotone == false) break;
  gotone = false;
  for (rc = in->lr; rc <= in->hr; ++rc)
    for (n = 1; n <= edges->get(3, 0) + 1; ++n)
      if (edges->get(3, n) == in->get(cc, rc) &&
        rc >= *col_min){
        *col_min = cc;
        gotone = true;
      }
}

gotone = true;

for (cc = in->hc; cc >= in->lc; --cc){
  if (gotone == false) break;
  gotone = false;
  for (rc = in->lr; rc <= in->hr; ++rc)
    for (n = 1; n <= edges->get(4, 0) + 1; ++n)
      if (edges->get(4, n) == in->get(cc, rc) &&

```

```

        *col_max = cc;
        gotone = true;
    }
}
delete edges;
}
//-----
//Routine to open a bitmap and store it in an array
void Get_Bitmap(char *fname, c_matrix **out){
    unsigned i,j;

    bitmap *bmp = new bitmap(fname);
    if (bmp->errmsg != NULL) mainerror(bmp->errmsg);

    if (bmp->type != BMP_256) mainerror(
        "Only 256 colour bitmaps supported.");
    *out = new c_matrix(
        0, bmp->ys - 1, 0, bmp->xs - 1);
    if ((*out)->errmsg != NULL) mainerror(
        "Memory error: Get_Bitmap");
    if ((*out)->errmsg != NULL) exit(13);

    for (j = 1; j <= bmp->ys; ++j)
        for (i = 1; i <= bmp->xs; ++i)
            (*out)->set(i - 1, j - 1, bmp->getpixel(i, j));
    delete bmp;
}
//-----
//Routine to save an image in an array to a bitmap
void save_to_bitmap(c_matrix *in, char * fname){
    int i,j;

    bitmap *bmp = new bitmap(in->hc - in->lc + 1,
        in->hr - in->lr + 1,
        BMP_GS);

    for (j = in->lr; j <= in->hr; ++j)
        for (i = in->lc; i <= in->hc; ++i)
            bmp->setpixel(i - in->lc + 1,
                j - in->lr + 1,
                in->get(i, j));

    bmp->save(fname);
    delete bmp;
}

```

```

//-----
//Routine to assign and store indexes for the
//8-neighbours locations
void get_plusses(char_coord in[9]){
    double rads;
    int i;

    for (i = 0; i <= 7; ++i){
        rads = (double) i * 2.0 * pi / 8.0;
        in[i].ip = (char) (cos(rads) * 1.8);
        in[i].jp = (char) (sin(rads) * 1.8);
    }
}
//-----
//This routine returns 1 for number > 1,
//or the number itself otherwise
float abovegive1(float f){
    if (f > 1.0F) return 1.0;
    return f;
}
//-----
//This routine returns 0 for number > 1
//or < 0, or the number itself otherwise
float abovegive0(float f){
    if (f > 1.0F || f < 0.0F) return 0.0;
    return f;
}
//-----
//Returns a random number >=0 and < 1
float rnd01(){
    return (float) rand()/((float) RAND_MAX + 1.0F);
}
//-----
//Routine to truncate numbers which do not
//fit in the 1 byte range.
usc charize(float z){
    if (z < 0)
        return (usc) 0;
    else if (z > 255)
        return (usc) 255;
    return (usc) z;
}

```

A.5 UTILITY ROUTINES AND HEADER FILES

In this last section are reproduced general utility routines for instantiating arrays, and manipulating bitmaps. These routines are also used in programs described in later appendices.

A.5.1 BITMAP UTILITY ROUTINE

The following routines form the body of the bitmap class, which is designed to open, close and create bitmaps and read and write to their pixels. It does not, however work for run-length encoded or 32-bit bitmaps. The header containing the class definition is given in the following section.

```

#include "bitmap.h"
//-----
//Routine to open named bitmap
bitmap::bitmap(char *fname):c_matrix(){
    FILE *fp;
    unsigned filetype;

    if (!(fp = fopen(fname, "rb"))){
        bmperror("Cannot find file...\n");
    }
}

```

```

fread(&filetype, sizeof(unsigned), 1, fp);
if (filetype != BITMAP_ID)
    bmperror("File format unrecognised...\n");

if (fread((void *) &header,
    sizeof(bitmap_header), 1, fp) != 1u)
    bmperror("Unexpected end of file...\n");
if (header.isRLE) bmperror("Bitmap is RLE...\n");
type = (bitmap_types) header.bitsperpixel;

```

```

if (type != BMP_24BIT){
    palette = (palentry*) calloc(
        2<<(type-1), sizeof(palentry));
    if (palette == NULL)
        bmperror("Memory error...\n");
}

datax = get_datax((uss) header.xsz, type);

if ((header.xsz <= 0)
    || (header.xsz > BMP_SZ_MAX)
    || (header.ysz <= 0)
    || (header.ysz > BMP_SZ_MAX))
    bmperror("File format error...\n");

if (type != BMP_24BIT)
    if (fread(palette, sizeof(palentry),
        2<<(type-1), fp) !=
        (usl)(2<<(type-1)))
        bmperror("Unexpected end of file...\n");

c_matrix::constr(
    0, (uss) header.ysz - 1, 0, datax - 1);

if (fread(data[0], sizeof(usc),
    datax * header.ysz, fp) !=
    (usl)(datax * header.ysz))
    bmperror("Unexpected end of file...\n");

xs = (uss) header.xsz;
ys = (uss) header.ysz;
mask = NULL;
if (type != BMP_24BIT)
    create_mask(type);

fclose(fp);
}

//-----
//Routine to create blank bitmap
bitmap::bitmap(uss xsize, uss ysize,
    bitmap_types bmp_type):c_matrix(){

    uss i;

    xs = xsize;
    ys = ysize;
    isgs = false;
    type = bmp_type;
    if (type == BMP_GS){
        isgs = true;
        type = BMP_256;
    }
    datax = get_datax(xs, type);
    c_matrix::constr(0, ys - 1, 0, datax - 1);
    if (errmsg != NULL)
        bmperror("Memory error: constructor.\n");
    create_header();
    if (type != BMP_24BIT){
        palette =
            (palentry*) calloc(
                1<<type, sizeof(palentry));
        if (palette == NULL)
            bmperror("Memory error: constructor.\n");
    }
    if (isgs)
        for (i = 0; i <= 255; ++i){
            palette[i].R = (usc) i;
            palette[i].G = (usc) i;
            palette[i].B = (usc) i;
        }
    mask = NULL;
    create_mask(type);
}

//-----
//Routine to calculate the padded line length
uss bitmap::get_datax(uss x, bitmap_types type){

    uss minbytes;

    switch (type){
    case BMP_MONO:
        minbytes = ((uss) ((float)(x - 1) / 8.0F)) + 1;
        break;
    case BMP_16:
        minbytes = ((uss) ((float)(x - 1) / 2.0F)) + 1;
        break;
    case BMP_256:
        minbytes = x;
        break;
    case BMP_24BIT:
        minbytes = x*3;
    }

    return ((uss) ((minbytes-1) / BYTEQUANT) + 1)
        * BYTEQUANT;
}

//-----
//Routine to create binary mask used to simplify
//pixel setting/getting.
void bitmap::create_mask(bitmap_types type){
    usc i;
    if (mask != NULL) delete(mask);

    mask = new c_matrix(0, 7, 0, 1);
    if (errmsg != NULL) ("Memory error: mask.\n");

    for (i = 0; i <= BPB / type; ++i){
        mask->set(0, i, (8/type - 1 - i) * type);
        mask->set(

```

```

        1, i, ((1<<type) - 1) << mask->get(0, i));
    }
}

//-----
//Routine to build a header structure for a new
//bitmap.
void bitmap::create_header(){

    switch (type){
    case BMP_24BIT:
        header.datastart = HEADER_LENGTH;
        break;
    default:
        header.datastart =
            HEADER_LENGTH + 4 * (1<<type);
    }

    header.datalength = datax * ys;
    header.filesize =
        header.datastart + header.datalength;
    header.xsz = xs;
    header.ysz = ys;
    header.dpix = header.dpiy = DEFDIMS;
    header.gap[0] = header.gap[1] = 0;
    header.gap2[0] =
        header.gap2[1] = header.gap2[2] = 0;
    header.q28 = 0x28;
    header.planes = 1;
    header.gapl = 0;
    header.bitsperpixel = type;
    header.isrle = false;
    header.q0 = 0;
}

//-----
//Routine to save bitmap
void bitmap::save(char *fname){

    uss usstmp = BITMAP_ID;

    FILE *fp = fopen (fname, "wb");
    if (fp == NULL)
        bmperror("Output file error...\n");
    fwrite (&usstmp, 2, 1, fp);
    fwrite (&header, sizeof(header), 1, fp);
    if (type != BMP_24BIT)
        fwrite (palette,
            sizeof(palentry), 1<<type, fp);
    fwrite (data[0],
        sizeof(usc), header.datalength, fp);
    fclose( fp);
}

//-----
//Routine to steal the palette from another bitmap
void bitmap::steal_palette(bitmap *inp){

    uss i;

    if (inp->type == BMP_24BIT || type == BMP_24BIT)
        bmperror("Invalid call for 24 bit bitmap: "
            "steal_palette.\n");
    for (i = 0; i <= (1<<type)-1; ++i)
        palette[i] = inp->palette[i];
}

//-----
//Pair of routines to set/get RGB components of
//24-bit bitmap pixels.
usc bitmap::getpixel(uss x, uss y, RGB col){
    return get(--x * 3 + (uss) col, --y);
}

void bitmap::setpixel(
    uss x, uss y, RGB col, usc dat){
    set(--x * 3 + (uss) col, --y, dat);
}

//-----
//Pair of routine to set/get bitmap pixels
//of non-24-bit types.
usc bitmap::getpixel(uss x, uss y){
    usc sigbyte =
        get((uss)((usl)(--x * type) / 8.0F), --y);
    usc masknum = x % (8 / type);
    return (sigbyte & mask->get(1, masknum)) >>
        mask->get(0, masknum);
}

usc bitmap::setpixel(uss x, uss y, usc dat){

    usc sigbyte =
        get((uss)((usl)(--x * type) / 8.0F), --y);
    usc masknum = x % (8 / type);
    return set((uss)((usl)(x * type) / 8.0F), y,
        (sigbyte & ~(mask->get(1, masknum))) |
        (dat << mask->get(0, masknum)));
}

//-----
//Routine to set a 24-bit pixel using a long value
void bitmap::setpixel(uss x, uss y, usl dat){

    usl mask = 0xFF;

```

```

setpixel(x,y,RED, (usc) (dat & mask));
setpixel(x,y,GREEN, (usc) ((dat >>= 8) & mask));

```

```

} setpixel(x,y,BLUE, (usc) ((dat >>= 8) & mask));
}

```

A.5.2 BITMAP HEADER

This section reproduces the definitions for the routines of the previous section, named as `bitmap.h`.

```

#ifndef BITMAPHEAD
#define BITMAPHEAD

#include "dtype.h"
#define CHAR_DATA
#include "matrix.h"
#define HEADER_LENGTH 0x36
#define BITMAP_ID 0x4d42
#define DEFDIMS 0xec4
#define BMP_SZ_MAX (usc) 0x7fff
#define BYTEQUANT 4
const int BPB = 8;
enum bitmap_types{BMP_MONO = 1, BMP_16 = 4,
                  BMP_256 = 8, BMP_24BIT = 24,
                  BMP_GS};
enum RGB{BLUE, GREEN, RED};

#define bmperror(a) {mainerror(a);}

struct palentry{
    usc R;
    usc G;
    usc B;
    usc spacer;
};

struct bitmap_header{
    usl filesize;
    usl gap1;
    usl datastart;
    usl _q28;
    usl xsz;
    usl ysz;
    usc planes;
    usc bitsperpixel;
    usc _g0;
    bool isSRLE;
    usc gap2[3];
};

```

```

    usl datalength;
    usl dpix;
    usl dpiy;
    usl gap[2];
};

class bitmap:public c_matrix{
private:
    usc datax;
    usc get_datax(usc x,bitmap_types type);
    bitmap_header header;
    void create_header();
    bool isGS;
    c_matrix *mask;
    void bitmap::create_mask(bitmap_types type);

public:
    bitmap(usc xsize, usc ysize, bitmap_types type);
    bitmap(char *fname);
    void save(char *fname);
    palentry *palette;
    usc xs, ys;
    bitmap_types type;
    usc getpixel (usc x,usc y);
    usc setpixel (usc x, usc y, usc dat);
    usc getpixel (usc x, usc y, RGB col);
    void setpixel(usc x, usc y, RGB col, usc dat);
    void setpixel(usc x, usc y, usl dat);
    void bitmap::steal_palette(bitmap *inp);
};

#endif

```

A.5.3 ARRAY HEADER

Used extensively throughout all the code in this work, the following code forms a definition for a flexible array, named `matrix.h`. All of the necessary code is contained within this header file. It is used by defining a data type in the code (`FLOAT_DATA`, `CHAR_DATA`, etc.) followed by the `#include` statement containing this header

```

#include "dtype.h"
#include <stdlib.h>

#if !(defined(DTYPE) && defined(MCLASS) &&
defined(VCLASS))

    #if defined(FLOAT_DATA) && !defined(FD)
        #define FD
        #define DTYPE float
        #define MCLASS f_matrix
        #define VCLASS f_vector
        #undef FLOAT_DATA

    #elif defined(INT_DATA) && !defined(FD)
        #define ID
        #define DTYPE int
        #define MCLASS i_matrix
        #define VCLASS i_vector
        #undef INT_DATA

    #elif defined(CHAR_DATA) && !defined(CD)
        #define CD
        #define DTYPE usc
        #define MCLASS c_matrix
        #define VCLASS c_vector
        #undef CHAR_DATA

    #elif defined(ULONG_DATA) && !defined(LD)
        #define LD
        #define DTYPE usl
        #define MCLASS l_matrix

```

```

        #define VCLASS l_vector
        #undef ULONG_DATA

    #elif defined(USHORT_DATA) && !defined(SD)
        #define SD
        #define DTYPE usc
        #define MCLASS s_matrix
        #define VCLASS s_vector
        #undef USHORT_DATA

    #elif defined(DOUBLE_DATA) && !defined(DD)
        #define DD
        #define DTYPE double
        #define MCLASS d_matrix
        #define VCLASS d_vector
        #undef DOUBLE_DATA

    #endif

#endif

#if defined(DTYPE)
class MCLASS{
public:
    char *errmsg;
    short lr, hr, lc, hc;
    MCLASS();
};

```

```

MCLASS(short r1, short r2, short c1, short c2){
    constr(r1, r2, c1, c2);
}

void constr(short r1, short r2,
             short c1, short c2){

    short i;
    lr = r1; hr = r2; lc = c1; hc = c2;

    pointarray=(DTYPE **) calloc(hr-lr+1,
                                  sizeof(DTYPE*));
    if (pointarray == NULL)
        mainerror("Memory error: matrix");
    data = pointarray-lr;
    mainarray=(DTYPE *)
        calloc((hr-lr+1)*(hc-lc+1), sizeof(DTYPE));
    if (mainarray == NULL)
        mainerror("Memory error: matrix");
    for(i = 0; i <= hr - lr; i++)
        pointarray[i] =
            mainarray + i*(hc - lc + 1) - lc;
    errmsg = NULL;
};

~MCLASS(){
    free((void*) pointarray);
    free((void*) mainarray);
};

DTYPE get(short x, short y){
    return data[y][x];
}

DTYPE set(short x, short y, DTYPE val){
    return (data[y][x] = val);
}

DTYPE incr(short x, short y){
    return ++data[y][x];
}

protected:
    char ERR[100];
    DTYPE **data;

```

```

private:
    DTYPE **pointarray ;
    DTYPE *mainarray ;
};

class VCLASS{
public:
    char *errmsg;
    DTYPE *data;
    short l,h;

    VCLASS(short c1, short c2){
        l = c1; h = c2;
        freearg =
            (DTYPE *) calloc(h - l + 1, sizeof(DTYPE));
        if (freearg == NULL)
            mainerror("Memory error: vector");
        data = freearg - l;
        errmsg = NULL;
    };

    ~VCLASS(){
        free(freearg);
    };

    DTYPE get(short x){
        return data[x];
    }

    void set(short x, DTYPE val){
        data[x] = val;
    }

    void incr(short x){
        ++(data[x]);
    }

private:
    char ERR[100];
    DTYPE* freearg;
};

#undef MCLASS
#undef VCLASS
#undef DTYPE

#endif

```

A.5.4 DATA TYPES HEADER

Finally, this section reproduces a header containing definitions for often-used data structures, and abbreviated data-types.

```

#if !defined(DATATYPES)
#define DATATYPES
#include <stdio.h>

#define mainerror(a) {printf((a));\
                      printf("\nTerminating.\n");\
                      exit(0xFEC);}

typedef unsigned char usc;
typedef unsigned short uss;
typedef unsigned long usl;

struct char_coord{
    char ip;
    char jp;
};

#endif

```

Appendix **B**

C++ Code: Levenberg-Marquardt Non-Linear Fitting

B.1 INTRODUCTION

This routine was used for both the fitting of a step function to profilometer data (section 7.2.6) and fitting of grain size distributions (i.e. Rayleigh functions) throughout this work. A rigorous derivation is out of the scope of this work, and only the results of interest will be reproduced. A complete derivation is give by Press *et al.**

* W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery "Numerical Recipes in C: The Art of Scientific Computing- 2nd Edition" (Cambridge University Press, 1992)

B.2 THE LEVENBERG-MARQUARDT METHOD

The basis of the method is to solve the matrix equation,

$$\sum_{l=1}^M \alpha'_{kl} \delta a_l = \beta_k, \quad (\text{B.1})$$

for δa_l , which is a matrix, $\mathbf{a}$, containing augmentations to the parameters of the equation to be fitted. M is the number of independent variables in the equation to be fitted. The matrix α' is given by:

$$\alpha'_{jj} = \alpha_{jj}(1 + \lambda), \quad (\text{B.2})$$

where λ is a non-dimensional 'fudge-factor', and,

$$\alpha'_{jk} = \alpha_{jk} \quad (j \neq k). \quad (\text{B.3})$$

The matrices α and β are the slightly modified gradient and Hessian of the chi-squared function for a function with parameters $\mathbf{a}$:

$$\chi^2(\mathbf{a}) = \sum_{i=1}^N \left[\frac{y_i - y(x_i; \mathbf{a})}{\sigma_i} \right]^2, \quad (\text{B.4})$$

where x and y are the data points being fitted to, and σ is their standard deviation. This gives,

$$\beta_k = -\frac{1}{2} \frac{\partial \chi^2}{\partial a_k} = \sum_{i=1}^N \frac{[y_i - y(x_i; \mathbf{a})]}{\sigma_i^2} \frac{\partial y(x_i; \mathbf{a})}{\partial a_k}, \quad (\text{B.5})$$

and,

$$\alpha_{kl} = \frac{1}{2} \frac{\partial^2 \chi^2}{\partial a_k \partial a_l} = \sum_{i=1}^N \frac{1}{\sigma_i^2} \left[\frac{\partial y(x_i; \mathbf{a})}{\partial a_k} \frac{\partial y(x_i; \mathbf{a})}{\partial a_l} - \{y_i - y(x_i; \mathbf{a})\} \frac{\partial^2 y(x_i; \mathbf{a})}{\partial a_l \partial a_k} \right]. \quad (\text{B.6})$$

It is usual, though, to ignore the second term in α_{kl} as it does not significantly affect the model, giving:

$$\alpha_{kl} = \sum_{i=1}^N \frac{1}{\sigma_i^2} \left[\frac{\partial y(x_i; \mathbf{a})}{\partial a_k} \frac{\partial y(x_i; \mathbf{a})}{\partial a_l} \right]. \quad (\text{B.7})$$

The technique for using these equations proceeds as follows:

- a) The program is initialised by calculating a value for $\chi^2(\mathbf{a})$ for an arbitrary value of λ
- b) Equation B.1 is solved for $\delta\mathbf{a}$, and $\chi^2(\mathbf{a} + \delta\mathbf{a})$ found.
- c) If $\chi^2(\mathbf{a} + \delta\mathbf{a}) \geq \chi^2(\mathbf{a})$ (i.e. the new value is worse than the old), λ is increased by a factor of, say, ten, and the program iterates.
- d) If $\chi^2(\mathbf{a} + \delta\mathbf{a}) < \chi^2(\mathbf{a})$ (i.e. the new value is better than the old), λ is decreased by a factor of ten, and the program iterates but with $\mathbf{a}$ replaced by $\mathbf{a} + \delta\mathbf{a}$.

The program iterates until the desired tolerance in the change in chi squared from one iteration to the next is reached.

B.3 THE CODE

The code falls into four parts: The main class, which performs the fitting and associated header file; the driver routine (in this case that for fitting step functions); the actual functions to be fitted, and; the matrix solving routine. Each is discussed and reproduced in turn below along with a simple file I/O routine used.

B.3.1 THE DRIVER ROUTINE

The driver routine reproduced here is that used for fitting the step functions defined in a following section (B.3.4). No parsing of command line is performed.

```
#include "levmar.h"
#include <stdio.h>
#include "DataGet.h"

//-----
void main(int argc, char*argv[]){
    usc npt = -1;
    usc i;
    FILE *fp;
    float xmin = -1e38F, xmax = -1e38F;
    float ymin = 1e38F, ymax = -1e38F;

    //Declarations of the function to be fitted...
    float sigmoid(float xq, f_vector *p);
    //...and its derivative
    float sigderiv(float xq, usc n, f_vector *p);
```

```
//Matrix into which the above functions'
//variables will be put
f_vector *a = new f_vector(1,4);

//Matrix into which the data to be
//fitted will be put
f_matrix *dat;

//Load data from file
dat = Load_Data(argv[1], 3, 0, TAB, 1);
npt = dat->nr;

//Get min & max values for initial values
//of coefficients. Also set sigma values to unity
for (i = 1; i <= npt; ++i){
    dat->set(x, i, dat->get(x, i) / 1000);
    dat->set(s, i, (float)1.0F);
    if (dat->get(x, i) > xmax)
```

```

        if(dat->get(X, i) < xmax) xmax = dat->get(X, i);
        if(dat->get(X, i) < xmin) xmin = dat->get(X, i);
        if(dat->get(Y, i) > ymax) ymax = dat->get(Y, i);
        if(dat->get(Y, i) < ymin) ymin = dat->get(Y, i);
    }

    //Set initial values for variables
    a->set(1, (xmax - xmin) / 2.0F); //x0
    a->set(2, ymin); //y0
    a->set(3, ymax - ymin); //a
    a->set(4, 1.0F); //b

    // initialise the fitting class
    levmarq *lm = new levmarq();

    //Pass the fitting class the data, the parameters
    //and the function (and its derivatives) to fit
    lm->set_data(dat);
    lm->set_vars(a);
    lm->set_functions(sigmoid, sigderiv);

```

```

//Iterate the fitting routine until
//tolerance is reached
do{
    lm->iteration();
}while(lm->chidiff > 0 || lm->chidiff < -0.01);

delete lm;
delete dat;

// Print the parameters to file
fp = fopen(argv[2], "w");
if (fp == NULL) mainerror ("Output file error")

fprintf(fp, "x0:\ty0:\ta:\tb:\n");
fprintf(fp, "%F\t%F\t%F\t%F",
        a->get(1), a->get(2), a->get(3), a->get(4));

fclose(fp);

delete a;
}

```

B.3.2 THE FITTING CLASS

This section reproduces the main fitting class and associated routines.

```

#include "levmar.h"
//-----
// Constructor
levmarq::levmarq(){
    xy = NULL;
    a = NULL;
    np = 0;
    na = 0;
    yq = NULL;
    dyda = NULL;
    alpha = new f_matrix(1, na, 1, na);
    beta = new f_vector(1, na);
    isFirst = true;
    lamda = 0.001F;
}
//-----
// The following three functions are called to set
// pointers to the data array, variable array, and
// functions (and derivatives) to be fitted.
void levmarq::set_data(f_matrix *dat){
    xy = dat;
    np = dat->nr;
}
void levmarq::set_vars(f_vector *vars){
    a = vars;
    na = vars->h - vars->l + 1;
}
void levmarq::set_functions(
    float (*f)(float, f_vector *),
    float (*dfda)(float x, usc n, f_vector *)){
    yq = f;
    dyda = dfda;
}
//-----
// Destructor
levmarq::~levmarq(){
    delete alpha;
    delete beta;
}
//-----
//Called for each iteration
void levmarq::iteration(){
    int i;

    // In the first iteration, check necessary has
    // been initialised, and calculate starting values.
    if (isFirst){
        if (xy == NULL
            || a == NULL
            || np == 0
            || na == 0
            || yq == NULL
            || dyda == NULL)
            mainerror("LM variable unset.")

        calc_chisq();
        calc_coeffs();
        augment();
        isFirst = false;
    }

    // Define working space
    f_vector *da = new f_vector(1, na);

    // Solve the equation
    m_solve(alpha, beta, da, na);

```

```

// Add new values to old
for (i = 1; i <= na; ++i)
    a->set(i, a->get(i) + da->get(i));

// Store old value of chi squared and
// calculate new value.
pchisq = chisq;
calc_chisq();
chidiff = chisq - pchisq;

// If fit is worse, take old lamda values out
// of alpha, increase lamda, and return parameters
// to original. Re-augment alpha with
// new lamda value.
if (chisq >= pchisq){
    deaugment();
    lamda *= 10;
    for (i = 1; i <= na; ++i)
        a->set(i, a->get(i) - da->get(i));
    chisq = pchisq;
    augment();
}

// If fit is better, reduce lamda, keep the
// current parameter values and initialise
// variables for the next loop.
else{
    lamda /= 10;
    calc_coeffs();
    augment();
}

delete da;
}
//-----
// Subroutine to calculate values for alpha and beta
void levmarq::calc_coeffs(){
    usc i, j;
    usc k;
    float grad, sigsq;

    //Set matrices alpha and beta to zero
    for (j = 1; j <= na; ++j){
        for (i = 1; i <= na; i++){
            alpha->set(i, j, 0.0);
            beta->set(j, 0.0);
        }
    }

    //Calculate the values of upper triangular
    //matrix elements.
    for (k = 1; k <= np; ++k){
        sigsq = xy->get(S, k) * xy->get(S, k);
        for (j = 1; j <= na; ++j){
            grad = dyda(xy->get(X, k), j, a);
            for (i = 1; i <= j; ++i)
                alpha->set(
                    i, j, alpha->get(i, j) + grad *
                    (dyda(xy->get(X, k), i, a) / sigsq));
            beta->set(
                j, beta->get(j) + (xy->get(Y, k) -
                yq(xy->get(X, k), a)) * (grad / sigsq));
        }
    }

    //Copy upper triangular to lower
    for (j = 1; j <= na; ++j)
        for (i = j+1; i <= na; i++)
            alpha->set(i, j, alpha->get(j, i));
}
//-----
// Subroutine for calculating Chi Squared
void levmarq::calc_chisq(){

```

```

uss k;
float delta_y;
chisq = 0;
for (k = 1; k <= np; ++k){
    delta_y = (xy->get(y, k)-yq(xy->get(x, k), a));
    chisq += delta_y * delta_y /
        (xy->get(s, k) * xy->get(s, k));
}
}
//-----
//subroutines to augment or de-augment
//alpha with lamda values

```

```

void levmarq::augment(){
    int i;
    for (i = 1; i <= na; ++i)
        alpha->set(i, i, alpha->get(i, i) *(1 + lamda));
}
void levmarq::deaugment(){
    int i;
    for (i = 1; i <= na; ++i)
        alpha->set(i, i, alpha->get(i, i) /(1 + lamda));
}

```

B.3.3 THE FITTING CLASS HEADER

The following code contains the class definition for the routines in the above section. Some other types are also defined.

```

#ifndef LEVMARHEAD
#define LEVMARHEAD

#define FLOAT_DATA
#include "matrix.h"

enum{X = 1,Y,S};

class levmarq{
public:
    levmarq();
    ~levmarq();
    void iteration();
    void levmarq::set_data(f_matrix *dat);
    void levmarq::set_vars(f_vector *vars);
    void levmarq::set_functions(
        float (*f)(float, f_vector *),
        float (*dfda)(float x, usc n, f_vector *));
    float chidiff;
private:

```

```

    float lamda;
    f_vector *a;
    usc na;
    usc np;
    f_matrix *xy;
    void calc_coeffs();
    void calc_chisq();
    float (*dyda)(float, usc, f_vector *);
    float (*yg)(float, f_vector *);
    f_matrix *alpha;
    f_vector *beta;
    void augment();
    void deaugment();
    void print_as();
    float pchisq, chisq;
    bool IsFirst;
    friend bool m_solve(f_matrix *, f_vector *,
        f_vector *, int );
};
#endif

```

B.3.4 THE FUNCTIONS

The following functions return the step function to be fitted and the partial derivatives. Although a numerical means to calculate derivatives can be used, for a simple function such as this it is as easy to supply the function.

```

#define FLOAT_DATA
#include "matrix.h"
#include <math.h>
#define x0 (double) p->get(x0)
#define y0 (double) p->get(y0)
#define a (double) p->get(A)
#define b (double) p->get(B)
const float float_max = 3.402823466e+38F;

enum{x0=1,Y0,A,B};
//-----
float sigmoid(float xq, f_vector *p){
    double rval;

    rval = exp((x0 - (double) xq) / b);
    rval = a / (rval + 1);
    rval += y0;

    return (float) rval;
}

```

```

}
//-----
float sigderiv(float xq, usc n, f_vector *p){
    double rval;
    double E;

    rval = 1.0;
    if (n == Y0) goto out;
    E = exp((x0 - (double) xq) / b);
    rval /= (1.0 + E);
    if (n == A) goto out;
    rval *= (-E / b * a / (1.0 + E));
    if (n == X0) goto out;
    rval *= ( (double)xq - x0) / b );

out:
    return (float) rval;
}

```

B.3.5 THE MATRIX SOLVER

This routine solves a matrix equation by simple elimination. Although not fast, it is very stable.

```

#include "dtype.h"
#define FLOAT_DATA
#include "matrix.h"

```

```

#define DOUBLE_DATA
#include "matrix.h"
#include <math.h>

```

```
//-----
//Routine to solve a matrix equation by simple
//elimination, returning the boolean value
//true to indicate success.
bool m_solve(f_matrix *A, f_vector *B,
             f_vector *Q, int sz){
    int i,j,k;
    double divfac;
    double temp;
    int maxpos;
    double maxval;

    d_matrix *I = new d_matrix(1, sz, 1, sz);
    d_vector *S = new d_vector(1, sz);

    for (i = 1; i <= sz; ++i){
        for (j = 1; j <= sz; ++j){
            I->set(i, j, (double) A->get(i, j));
            S->set(i, (double) B->get(i));
        }
    }

    for (i = 1; i <= sz-1; ++i){
        maxval = 0.0;
        for (j = i+1; j <= sz; ++j){
            if ((float) fabs((float) I->get(i, j)) >=
                maxval){
                maxval = (float) fabs((float) I->get(i, j));
                maxpos = j;
            }
        }
        if (maxval == 0.0) return false;
        for (j = 1; j <= sz; ++j){
            temp = I->get(j, maxpos);
            I->set(j, maxpos, I->get(j, i));
            I->set(j, i, temp);
        }
    }
}
```

```

    temp = S->get(maxpos);
    S->set(maxpos, S->get(i));
    S->set(i, temp);

    for (j = i+1; j <= sz; ++j){
        if (I->get(i, j) == 0) continue;
        if (I->get(i, j) != I->get(i, i)){
            divfac = I->get(i, j)/I->get(i, i);
            for (k = i; k <= sz; ++k)
                I->set(k, j, I->get(k, j) / divfac);
            S->set(j, S->get(j) / divfac);
        }
        for (k = i; k <= sz; ++k)
            I->set(k, j, I->get(k, j) - I->get(k, i));
        S->set(j, S->get(j) - S->get(i));
    }

    for (i = sz; i >= 1; --i){
        temp = 0;
        for (j = i+1; j <= sz; ++j)
            temp += S->get(j)*I->get(j, i);
        S->set(i, (S->get(i) - temp)/I->get(i, i));
    }

    for (i = sz; i >= 1; --i){
        Q->set(i, (float) S->get(i));
    }
    return true;
}
```

B.3.6 FILE I/O HEADER

This header defines functions for file I/O.

```
#define FLOAT_DATA
#include "matrix.h"

#define ASCII_LF 10
enum delimiters {TAB, COMMA};
const char delims[2][4] = {"\t", ","};

//-----
f_matrix * Load_Data(char *fname, usc columns,
                      usc ignore, delimiters delim,
                      usc os){
    int n = 0;
    int i = 0, j;
    float tmpfloat;

    FILE *fp = fopen(fname, "r");
    if (fp == NULL)
        mainerror("File error: load_data")

    while (!feof(fp)){
        if (fgetc(fp) == ASCII_LF) ++n;
    }

    n -= ignore + 1;
    rewind(fp);

    while (ignore != 0){
        if (fgetc(fp) == ASCII_LF)
            if (++i == ignore) break;
    };

    f_matrix *localdat = new f_matrix
        (os, n - 1 + os, 1, columns);

    for (i = os; i <= n - 1 + os; ++i)
        for (j = 1; j <= columns; ++j){
            fscanf(fp, "%e", &tmpfloat);
            localdat->set(j, i, tmpfloat);
            localdat->get(j, i);
        }

    fclose(fp);
    return localdat;
}

//-----
void Save_Data(char *fname, f_matrix *data,
               delimiters delim){
    int n = -1;
    int i = 0, j;
    float tmpfloat;

    FILE *fp = fopen(fname, "w");
    if (fp == NULL)
        mainerror("File error: Save_Data.");

    for (i = data->lr; i <= data->hr; ++i)
        for (j = data->lc; j <= data->hc; ++j){
            tmpfloat = data->get(j, i);

```

```

            fprintf(fp, j == data->hc ?
                "%e\n": delims[delim], tmpfloat);
        }
    }
    fclose(fp);
}
```

C++ Code: Deposition Model

C.1 INTRODUCTION

This chapter reproduces the code for the deposition model described and used in chapter 9. As with previous appendices, the full code is reproduced, excluding some utility routines given in section A.5. Annotations to the code are more abundant for the more relevant parts of the program.

C.2 THE DRIVER ROUTINE

The following code consists of four subroutines. The main routine, first calls `Sort_Params` to parse the command-line parameters, and then repeatedly calls the model class to generate new layers. The subroutine `NextName` simply increments the string which stores the output name for the next layer image file, by means of `increment_string`. What is, perhaps, unclear from the code

is that the attachment probability parameter is the integer $\log_{10}$ of the probability required.

```
#include "ggmod.h"

#define DEF_MODSIZE 200
#define DEF_MODHEIGHT 20
#define DEF_NO_LAYERS 100

//Main driver routine.
void main( int argc, char *argv[]){
    gg_model *model;
    Bitmap *bmp;
    usc MODSIZE;
    usc MODHEIGHT;
    char P_ATA;
    usc G_AREA;
    usc NO_LAYERS;
    char *outpath;
    char fullpath[500];
    usc LayersDone = 0;

    void Sort_Params(int argc, char **argv[],
                    char **outpath, usc *MODSIZE,
                    usc *MODHEIGHT, usc *G_AREA,
                    char *P_ATA, usc *NO_LAYERS);
    void NextName(char *outpath, char fullpath[]);

    //Parse command line.
    Sort_Params(argc, &argv, &outpath, &MODSIZE,
                &MODHEIGHT, &G_AREA, &P_ATA,
                &NO_LAYERS);

    //Initialise model with parameters.
    model = new gg_model(MODSIZE, MODHEIGHT,
                        G_AREA, P_ATA);

    //Save these parameters to the output directory.
    model->saveparams(outpath, "params.txt");

    printf("Creating first layer...\n");

    while(true){
        //Main call to get next complete model layer.
        model->get_next_layer(&bmp);

        //check that model has not run out of space.
        if (bmp == NULL) break;

        //Output layer to bitmap of incremented name.
        NextName(outpath, fullpath);
        bmp->save(fullpath);
        printf("outputted ");
        printf(fullpath);
        printf("\n");
        LayersDone++;

        //If model is complete, terminate.
        if (LayersDone >= 100)
            model->terminate();
    }

    //-----
    //This routine increments and constructs the
    //output filename.
    void NextName(char *outpath, char fullpath[]){
        char tmp[500];
        void increment_string(char **filename, usc cpos);
        static char *filename = "00000000";
        const char *extn = ".bmp";

        increment_string(&filename, 1);
        strcpy(tmp, outpath);
        strcat(tmp, filename);
        strcat(tmp, extn);
        strcpy(fullpath, tmp);
    }

    //Recursive routine to increment the filename.
    void increment_string(char **filename, usc cpos){
        ((*filename + (8-cpos))) == '9' ?
        ((*filename + (8-cpos))) = '0' :
        ++((*filename + (8-cpos)));

        if ((*filename + (8-cpos))) == '0')
            increment_string(filename, cpos + 1);
    }

    //command-line parser.
```

```
void Sort_Params(int argc, char **argv[],
                char **outpath, usc *MODSIZE,
                usc *MODHEIGHT, usc *G_AREA,
                char *P_ATA, usc *NO_LAYERS){
    int i;
    usc c1, c2;
    *outpath = NULL;
    *MODSIZE = 0;
    *MODHEIGHT = 0;
    *NO_LAYERS = 0;
    *P_ATA = 1;
    *G_AREA = 0;

    if (argc == 1) mainerror("output path "
                            "required.");

    *outpath = (argv[0][1]);
    if (strlen(*outpath) == 0)
        mainerror("output path is a "
                "zero length string.");

    for (i = 2; i <= argc - 1; ++i){
        int itmp;
        char ctmp;
        usc usctmp;

        c1 = *argv[0][i];
        if (c1 != '/') mainerror("Invalid Argument.");

        c2 = *(argv[0][i] + 1);
        switch (c2){
            case 's':
            case 'S':
                itmp = atoi(argv[0][i] + 2);
                *MODSIZE = (usc) itmp;
                if (itmp < 10 || itmp > 250)
                    mainerror(
                        "The model size (/s) must "
                        "be between 10 and 250.");
                break;

            case 'h':
            case 'H':
                usctmp = atoi(argv[0][i] + 2);
                *MODHEIGHT = usctmp;
                if (usctmp < 0 || usctmp > 50)
                    mainerror("Model height (/h) must lie "
                            "between 5 and 50.");
                break;

            case 'o':
            case 'O':
                itmp = atoi(argv[0][i] + 2);
                *NO_LAYERS = itmp;
                if (usctmp < 0 || usctmp > 5000)
                    mainerror("output Layers (/O) must lie "
                            "between 1 and 5000.");
                break;

            case 'g':
            case 'G':
                usctmp = (usc) atoi(argv[0][i] + 2);
                *G_AREA = usctmp;
                if (*G_AREA <= 0 || *G_AREA > 250)
                    mainerror("Grain Size must be "
                            "between 1 and 250.");
                break;

            case 'a':
            case 'A':
                ctmp = (char) atoi(argv[0][i] + 2);
                *P_ATA = ctmp;
                if (*P_ATA > 0)
                    mainerror("Attachment probability "
                            "cannot be > 1 (i.e. 10^0).");
                break;

            default:
                mainerror("Invalid Argument."
                        "Use ?? for help.");
        }
    }

    if (*outpath == NULL)
        mainerror("output path required.");
    if (*MODSIZE == 0)
        *MODSIZE = DEF_MODSIZE;
    if (*MODHEIGHT == 0)
        *MODHEIGHT = DEF_MODHEIGHT;
    if (*NO_LAYERS == 0)
        *NO_LAYERS = DEF_NO_LAYERS;
    if (*P_ATA == 1)
        mainerror("Attachment probability required.");
    if (*G_AREA == 0)
        mainerror("Grain size must be specified.");
}
```

C.3 THE MAIN MODEL CLASS

This section is dedicated to the modelling class that performs the simulation itself. It consists of a number of different subroutines.

```
#include "ggmod.h"

//Constructor: initialisation of program and growth
//space with program parameters, and also seeding
//of substrate with nuclei.
gg_model::gg_model(uss MS, usc MH,
                  usc GS, char PA){

    ModSize = MS;
    ModHeight = MH;
    P_Atta = PA;
    G_Area = GS;
    ModelInactive = false;
    MaxLayer = 0;

    grid = short3dgrid(0, ModHeight + 1, 0,
                      ModSize - 1, 0, ModSize - 1);
    filled =
        (usl*) calloc((ModHeight + 2), sizeof(usl));
    if (filled == NULL)
        mainerror("Memory error in "
                  "constructor: gg_model.");

    get_surrs();
    op_bmp = new bitmap(ModSize, ModSize, BMP_24BIT);
    seed();
}

//Routine to seed nuclei in first layer.
void gg_model::seed(){
    usl i;

    usl nograin = (uss) ((float)
        (ModSize * ModSize) / ((float) G_Area));

    for (i = 1; i <= nograin; ++i){
        do{
            generate_new();
        }while( grid[1][cp.x][cp.y] != 0 );
        grid[1][cp.x][cp.y] = i;
    }

    filled[1] = nograin;
}

//Public class member called to get another
//complete layer. It also tests to see
//whether the model should finish.
void gg_model::get_next_layer(bitmap **rt_bmp){
    if (!ModelInactive)
        activelayer();

    if (MaxLayer == 0){
        *rt_bmp = (bitmap *) NULL;
    }
    else{
        bottom_layer_to_bmp();
        *rt_bmp = op_bmp;
    }
}

//Public routine to terminate model.
void gg_model::terminate(){
    ModelInactive = true;
}

//Main subroutine: Introduces species until a
//complete layer is formed, then returns.
void gg_model::activelayer(){
    inline bool exp_chance( char n );

    float chance;
    usl nlocal;
    usl no = 0;

    //Each iteration in this loop represents a
    //new particle
    while(true){
        generate_new();

        //Each iteration of this loop checks to see
        //if the particle will attach, and
        //moves it if not ('migrate').
        while(true){
            //Drops the particle through 'free space'
            //until it hits a filled space.
            while(cp.z > 1){
                if (grid[cp.z - 1][cp.x][cp.y] == 0)
                    --cp.z;
                else
                    break;
            }

            chance = rnd01();

            //Calculates probability of attachment,
            //and tests it against 'chance'. If it
            //attaches, it's position is registered.
            if (chance < bind(&nlocal)
                && exp_chance(P_Atta)){
                if (nlocal == 0) mainerror("Arses");
                grid[cp.z][cp.x][cp.y] = nlocal;
                ++filled[cp.z];
                if (cp.z > MaxLayer) MaxLayer = cp.z;
                break;
            }

            wander();
        }

        //tests if the array is filled up.
        if (MaxLayer == ModHeight){
            ModelInactive = true;
            return;
        }

        //tests whether a layer has been completed.
        if (filled[1] >=
            (usl) ModSize * (usl) ModSize){
            return;
        }
    }
}

//Calculates binding probability, and the
//grain to which the species will attach.
float gg_model::bind(uss *nlocal){
    usc n,m;

    bool gotone = false;
    usl tmp_gno;
    usl loc_gno[26];
    float loc_bind[26];
    usc grains_so_far = 0;
    usc pos;

    // Following constant ONLY valid for
    // BINDDROPOFF = 2;
    const float DIV_FAC = 13.0F + (2.0F/3.0F);
    float tot_bind = 0, tmp_bind = 0.0;

    //Zero matrices which store the binding from
    //each grain, and the number of each.
    for (n = 0; n <= 25; ++n){
        loc_gno[n] = 0;
        loc_bind[n] = 0.0F;
    }

    //offset for iterating through the neighbours:
    //eliminates directional effects
    usc os = (usc) (26.0F * rnd01());
    *nlocal = 0;

    //Iterates through the 26-neighbours.
    for (n = 0; n <= 25; ++n){
        pos = (n + os) % 26;
        tmp_gno = grid
            [cp.z + surrs3d[pos].z]
            [getpos(cp.x + surrs3d[pos].x)]
            [getpos(cp.y + surrs3d[pos].y)];

        //Tests if this type of neighbour has so far
        //been encountered, and updates the binding
        //term if it has...
        for (m = 0; m <= grains_so_far; ++m)
            if (tmp_gno == loc_gno[m]){
                loc_bind[m] += bind_term[pos];
                break;
            }
    }
}
```

```

//...and creates a new record for
//that type if not
if (m > grains_so_far){
    ++grains_so_far;
    loc_gno[grains_so_far] = tmp_gno;
    loc_bind[m] = bind_term[pos];
}

}

tmp_bind = 0.0;

//works out highest binding grain,
//and total binding.
for (m = 0; m <= grains_so_far; ++m){
    if (loc_gno[m] == 0)
        continue;
    if (loc_bind[m] > tmp_bind){
        tmp_bind = loc_bind[m];
        tmp_gno = loc_gno[m];
    }

    tot_bind += loc_bind[m];
}

//Returns above two figures.
*local = tmp_gno;
return tot_bind / DIV_FAC;
}

//Moves species one step in random direction,
//after testing if that square is empty.
void gg_model::wander(){
    usc i;
    scoord3d tmp = cp;
    usc mpos = (usc) (8.0F * rnd01());

    for (i = 0; i <= 7; ++i){
        tmp.x = getpos
            (tmp.x + surrs2d[(mpos + i) % 8].x);
        tmp.y = getpos
            (tmp.y + surrs2d[(mpos + i) % 8].y);
        if (grid[tmp.z][tmp.x][tmp.y] == 0){
            cp = tmp;
            return;
        }
    }
}

//Generates a new particle in a random position
//at the top of the growth space
void gg_model::generate_new(){
    cp.x = rnd0sz();
    cp.y = rnd0sz();
    cp.z = MaxLayer + 1;
}

//Two functions to return random numbers.
inline float rnd01(){
    return (float) rand() / ((float) RAND_MAX + 1.0F);
}

uss gg_model::rnd0sz(){
    return (uss) (rnd01() * (float) ModSize);
}

//takes an argument, n, which is the
// (integer) log10 of the probability, and
// returns a boolean saying if the chance
// came off.
bool exp_chance( char n ){
    float rnd01();
    char i;

    for (i = 0; i > n; --i)
        if (rnd01() >= 0.1F){
            return false;
        }

    return true;
}

//Routine which wraps positions which fall
//out of model bounds
inline usc gg_model::getpos(short n){
    return ((n % ModSize) + ModSize) % ModSize;
}

```

```

//Routine which generates the plus/minus positions
//of the 26-neighbours, and 8-neighbours, and the
//binding strength associated with
//each 26-neighbour.
void gg_model::get_surrs(void){
    char i,j,n = 0;
    double rads;

    for (i = 0; i <= 7; ++i){
        rads = (double) i * 2.0 * pi / 8.0;
        surrs2d[i].x = (char) (cos(rads) * 1.8);
        surrs2d[i].y = (char) (sin(rads) * 1.8);
    }

    surrs3d[0].z = -1;
    surrs3d[0].x = 0;
    surrs3d[0].y = 0;

    for (j = -1; j <= 1; ++j){
        for (i = 0; i <= 7; ++i){
            rads = (double) i * 2.0 * pi / 8.0;
            surrs3d[i+n].z = j;
            surrs3d[i+n].x = (char) (cos(rads) * 1.8);
            surrs3d[i+n].y = (char) (sin(rads) * 1.8);
        }

        surrs3d[25].z = 1;
        surrs3d[25].x = 0;
        surrs3d[25].y = 0;

        for (j = 0; j <= 25; ++j){
            bind_term[j] =
                (float) (1.0 / pow(ROOTS
                    [ surrs3d[j].x * surrs3d[j].x
                      + surrs3d[j].y * surrs3d[j].y
                      + surrs3d[j].z * surrs3d[j].z
                    , BINDDROPOFF));
        }
    }

    //Copies bottom layer of array to a bitmap, and
    //shifts all other layers down by one (leaving
    //outputted layer in zeroth layer).
    void gg_model::bottom_layer_to_bmp(){
        int i,j;

        for (j = 1; j <= ModSize; ++j)
            for (i = 1; i <= ModSize; ++i){
                grid[0][i-1][j-1] = 0;
                op_bmp->
                    setpixel(i, j, (usl) grid[1][i-1][j-1]);
            }

        uss **temp = grid[0];
        for (j = 0; j <= ModHeight; ++j){
            filled[j] = filled[j+1];
            grid[j] = grid[j+1];
        }

        filled[ModHeight] = 0;
        grid[ModHeight] = temp;
        --MaxLayer;
    }

    //Copies program parameters to file.
    void gg_model::saveparams(char *fpath,
                               char *fname){
        char tmpstr[500] = "";
        strcpy(tmpstr, fpath);
        strcat(tmpstr, fname);

        FILE *fp = fopen(tmpstr, "w");
        fprintf(fp, "Model Size:\t%d\n", ModSize);
        fprintf(fp, "Model Height:\t%d\n", ModHeight);
        fprintf(fp, "Initial Grain Area:\t%d\n", G_Area);
        fprintf(fp, "Sticking Parameter: 10^{\t%d", P_Atta);
        fclose(fp);
    }

    //Destructor.
    gg_model::~gg_model(){
        free_short3dgrid(grid, 0, ModHeight + 1, 0,
                          ModSize - 1, 0, ModSize - 1);
    }
}

```

C.4 THE MODEL HEADER

This header contains the definitions for the class in the above section, along with some other useful definitions.

```
#include "bitmap.h"
#include "dtype.h"
#include <math.h>
#include <string.h>
#include <stdlib.h>
const double pi = 3.141592653589;
const double ROOTS[4] = {0.0,
                          1.0,
                          1.414213562373,
                          1.732050807568};

#define BINDDROPOFF 2

uss ***short3dgrid(long nr1, long nrh,
                  long ncl, long nch,
                  long nd1, long ndh);
void free_short3dgrid(uss ***t,
                    long nr1, long nrh,
                    long ncl, long nch,
                    long nd1, long ndh);

float rnd01();

struct scoord3d{
    uss x;
    uss y;
    uss z;
};

struct scoord2d{
    uss x;
    uss y;
};

struct char3d{
    char x;
    char y;
    char z;
};

struct char2d{
    char x;
    char y;
};

class gg_model{
public:
    gg_model(uss MS, uss MH,
            uss GS, char PA);
    void gg_model::get_next_layer
        (bitmap **rt_bmp);
    void gg_model::terminate();
    void gg_model::saveparams
        (char *fpath, char *fname);

    bitmap *op_bmp;

private:
    void gg_model::generate_new();
    void gg_model::wander();
    float gg_model::bind(uss *nolocal);
    void gg_model::seed();
    void gg_model::get_surrs();
    uss gg_model::getpos(short n);
    void gg_model::bottom_layer_to_bmp();
    void gg_model::activelayer();
    uss gg_model::rnd0sz();
    ~gg_model();

    uss ModSize;
    uss ModHeight;
    uss G_Area;
    char P_Atta;
    uss ***grid;
    uss *filled;
    short MaxLayer;
    scoord3d cp;
    char3d surrs3d[25];
    char2d surrs2d[8];
    float bind_term[25];
    bool ModelInactive;
};
```

C.5 THE 3D ARRAY CLASS

This is routine to initialise the 3D array for the growth space. It is based on the tensor structure of Press *et al.**

```
#include "dtype.h"
#include <stdlib.h>
#define NR_END 1
#define FREE_ARG char*

uss ***short3dgrid(long nr1, long nrh,
                  long ncl, long nch,
                  long nd1, long ndh){

    long i, j;
    nrow = nrh - nr1 + 1;
    ncol = nch - ncl + 1;
    ndep = ndh - nd1 + 1;
    uss ***t;

    t = (uss***) calloc((size_t) (nrow + NR_END),
                      sizeof(uss**));

    if (!t) mainerror("Memory error: 3dGrid()");
    t += NR_END;
    t -= nr1;

    t[nr1] = (uss**) calloc
        ((size_t) (nrow * ncol + NR_END), sizeof(uss*));
    if (!t[nr1]) mainerror("Memory error: 3dGrid()");
    t[nr1] += NR_END;

    t[nr1][nc1] = (uss *) calloc
        ((size_t) (nrow * ncol * ndep + NR_END), sizeof(uss));
    if (!t[nr1][nc1]) mainerror("Memory error: 3dGrid()");
    t[nr1][nc1] += NR_END;

    for(j = nc1 + 1; j <= nch; j++)
        t[nr1][j] = t[nr1][j - 1] + ndep;

    for(i = nr1 + 1; i <= nrh; i++) {
        t[i] = t[i - 1] + ncol;
        t[i][nc1] = t[i - 1][nc1] + ncol * ndep;
        for(j = nc1 + 1; j <= nch; j++) t[i][j] = t[i][j - 1] + ndep;
    }

    return t;
}

void free_short3dgrid(uss ***t,
                    long nr1, long nrh,
                    long ncl, long nch,
                    long nd1, long ndh)
{
```

* W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery "Numerical Recipes in C: The Art of Scientific Computing— 2nd Edition" (Cambridge University Press, 1992)

```
free((FREE_ARG) (t[nr1][inc1]+nd1-NR_END));  
free((FREE_ARG) (t[nr1]+nc1-NR_END));  
free((FREE_ARG) (t+nr1-NR_END));  
}
```

List of Publications

M. A. Cousins and K. Durose *Grain Structure of CdTe in CSS-Deposited CdTe/CdS Solar Cells*, Thin Solid Films **361** (2000) 253-57

M. A. Cousins and K. Durose *TEM of the near-Interface Region in CdTe/CdS Solar Cells*, Proceedings of the 16th European Photovoltaic Solar Energy Conference, Glasgow, UK (2000) 835-38

M. J. Hogan, A. W. Brinkman, T. Hashemi and M. A. Cousins *Preparation and Humidity Sensitive Impedance of the Spinel Ceramic Ni_2GeO_4* in "Growth and Processing of Electronic Materials", Ed. N. M. Alford (I.O.M. Communications, London, 1998)

K. Durose, M. D. G. Potter, M. A. Cousins and D. P. Halliday *Microstructure and Impurities in Thin Film CdTe/CdS Solar Cells*, Submitted to Journal of Microscopy

M. D. G. Potter, D. P. Halliday, M. Cousins and K. Durose *A Study of the Effects of Varying Cadmium Chloride Treatment on the Luminescent Properties of CdTe/CdS Thin Film Solar Cells*, Thin Solid Films **361** (2000) 248-52

(contd...)

M. D. G. Potter, M. Cousins and D. P. Halliday *Effect of Interdiffusion and Impurities on Thin-Film CdTe/CdS Photovoltaic Junctions*, Journal of Materials Science: Materials in Electronics **11** (2000) 525-30

M. D. G. Potter, D. P. Halliday, M. Cousins and K. Durose *Analysis of CdCl₂ Annealing Process in CdTe/CdS Solar Cells*, Proceedings of the 16th European Photovoltaic Solar Energy Conference, Glasgow (2000) 847-50

