



**University of
Nottingham**

UK | CHINA | MALAYSIA

Machine Learning for the Automation and Optimisation of Optical Coordinate Measurement

Thesis submitted to the University of Nottingham for the degree of
Doctor of Philosophy, September 2022.

Joe Eastwood

04342322

Supervised by

Dr Samanta Piano
Professor Richard K. Leach

Signature _____

Date ____ / ____ / ____

Knowledge is porridge.

Dr Stewart Pearson,
Director of Communications for the Opposition, 2009

Abstract

Camera based methods for optical coordinate metrology are growing in popularity due to their non-contact probing technique, fast data acquisition time, high point density and high surface coverage. However, these optical approaches are often highly user dependent, have high dependence on accurate system characterisation, and can be slow in processing the raw data acquired during measurement. Machine learning approaches have the potential to remedy the shortcomings of such optical coordinate measurement systems. **The aim of this thesis is to remove dependence on the user entirely by enabling full automation and optimisation of optical coordinate measurements for the first time.** A novel software pipeline is proposed, built, and evaluated which will enable automated and optimised measurements to be conducted. No such automated and optimised system for performing optical coordinate measurements currently exists. The pipeline can be roughly summarised as follows:

intelligent characterisation → *view planning* → *object pose estimation* → *automated data acquisition* → *optimised reconstruction*.

Several novel methods were developed in order to enable the embodiment of this pipeline. Chapter 4 presents an intelligent camera characterisation (the process of determining a mathematical model of the optical system) is performed using a hybrid approach wherein an EfficientNet convolutional neural network provides sub-pixel corrections to feature locations provided

by the popular OpenCV library. The proposed characterisation scheme is shown to robustly refine the characterisation result as quantified by a 50 % reduction in the mean residual magnitude. The camera characterisation is performed before measurements are performed and the results are fed as an input to the pipeline. Chapter 5 presents a novel genetic optimisation approach is presented to create an imaging strategy, ie. the positions from which data should be captured relative to part's specific geometry. This approach exploits the computer aided design (CAD) data of a given part, ensuring any measurement is optimal given a specific target geometry. This view planning approach is shown to give reconstructions with closer agreement to tactile coordinate measurement machine (CMM) results from 18 images compared to unoptimised measurements using 60 images. This view planning algorithm assumes the part is perfectly placed in the centre of the measurement volume so is first adjusted for an arbitrary placement of the part before being used for data acquisition. Chapter 6 presents a generative model for the creation of surface texture data is presented, allowing the generation of synthetic but realistic datasets for the training of statistical models. The surface texture generated by the proposed model is shown to be quantitatively representative of real focus variation microscope measurements. The model developed in this chapter is used to produce large synthetic but realistic datasets for the training of further statistical models. Chapter 7 presents an autonomous background removal approach is proposed which removes superfluous data from images captured during a measurement. Using images processed by this algorithm to reconstruct a 3D measurement of an object is shown to be effective in reducing data processing times and improving measurement results. Use the proposed background removal on images before reconstruction are shown to benefit from up to a 41 % reduction in data processing times, a reduction in superfluous background points of up to 98 %, an increase in point density on

the object surface of up to 10 %, and an improved agreement with CMM as measured by both a reduction in outliers and reduction in the standard deviation of point to mesh distances of up to 51 μm . The background removal algorithm is used to both improve the final reconstruction and within stereo pose estimation. Finally, in Chapter 8, two methods (one monocular and one stereo) for establishing the initial pose of the part to be measured relative to the measurement volume are presented. This is an important step to enabling automation as it allows the user to place the object at an arbitrary location in the measurement volume and for the pipeline to adjust the imaging strategy to account for this placement, enabling the optimised view plan to be carried out without the need for special part fixturing. It is shown that the monocular method can locate a part to within an average of 13 mm and the stereo method can locate a part to within an average of 0.44 mm as evaluated on 240 test images. Pose estimation is used to provide a correction to the view plan for an arbitrary part placement without the need for specialised fixturing or fiducial marking.

This pipeline enables an inexperienced user to place a part anywhere in the measurement volume of a system and, from the part's associated CAD data, the system will perform an optimal measurement without the need for any user input. Each new method which was developed as part of this pipeline has been validated against real experimental data from current measurement systems and shown to be effective.

In future work given in Section 9.1, a possible hardware integration of the methods developed in this thesis is presented. Although the creation of this hardware is beyond the scope of this thesis.

Acknowledgements

I would like to thank all my family and friends, particularly my parents Wendy and Richard, my brother Lloyd, and my partner Mairi, for their constant support. I would also like to thank, in no particular order:

Adam Thompson and Luke Todhunter for convincing me to join MMT, one of the better decisions I have made.

Danny Sims-Waterhouse for teaching me everything I know about photogrammetry, for providing much of the inspiration for the work in this thesis, and for guiding me through the early days of my project.

My supervisors; Richard Leach for turning me into a metrologist, and Samanta Piano for her unending guidance and encouragement.

All my co-authors: Isa, George, Hui, Lewis and Sofia for their excellent inputs.

All the housemates I have had the pleasure of living with during my PhD: Guy and Iz, Grace, Emma and Roper, and T{Woc, Ross}. Thanks for putting up with me.

And finally, Albert and Geordie Matt - for starting and ending this thing together, see you at Sat Bains <3.

Publication list

Journal papers

Zhang H, **Eastwood J**, Isa M A, Sims-Waterhouse D, Leach R K, Piano S 2020 Optimisation of camera positions for optical coordinate measurement based on visible point analysis *Precision Engineering* **67** 178-188.

Eastwood J, Newton L, Leach R K, Piano S 2021 Generation and categorisation of surface texture data using a modified progressively growing adversarial network *Precision Engineering* **74** 1-11.

Eastwood J, Leach R K, Piano S 2022 Autonomous image background removal for accurate and efficient close-range photogrammetry *Measurement Science and Technology* **34** 035404.

Catalucci S, Thompson A, **Eastwood J**, Zhang M, Branson D T, Leach R K, Piano S 2022 Smart optical coordinate and surface metrology *Measurement Science and Technology* **34** 012001.

Eastwood J, Gayton G, Leach R K, Piano S 2023 Improving the localisation of features for the calibration of cameras using EfficientNets *Optics Express* **31** 7966-7982.

Book chapters

Eastwood J, Sims-Waterhouse D, Piano S 2020 Machine learning approaches *in* Leach R K *Advances in Optical Form and Coordinate Metrology* (IoP Publishing: Bristol).

Conference papers

Eastwood J, Sims-Waterhouse D, Piano S, Leach R K 2019 Autonomous close-range photogrammetry using machine learning *Proc. ISMTII* (Niigata, Japan).

Eastwood J, Zhang H, Isa M A, Sims-Waterhouse D, Leach R K, Piano S 2020 Smart photogrammetry for three-dimensional shape measurement *Proc. SPIE* (Strasbourg, France).

Eastwood J, Sims-Waterhouse D, Piano S, Leach R K 2020 Pose estimation from a monocular image for automated photogrammetry *Proc. euspen Int. Conf.* (Geneva, Switzerland).

Eastwood J, Newton L, Leach R K, Piano S 2021 Generation of simulated additively manufactured surface texture data using a progressively growing generative adversarial network *Proc. euspen Int. Conf.* (Copenhagen, Denmark).

Eastwood J, Leach R K, Piano S 2022 Efficient close-range photogrammetry reconstruction through autonomous image background removal *Proc. euspen Int. Conf.* (Geneva, Switzerland).

Eastwood J, Gayton G, Leach R K, Piano S 2022 Improving camera calibration with machine learning for the measurement of additively manufactured parts *Proc. ASPE Adv. Precis. in Addit. Manuf.* (Knoxville, TN).

Contents

Abstract	i
Acknowledgements	iv
Publication list	iv
List of Tables	xiii
List of Figures	xiv
Chapter 1 Introduction	2
1.1 Coordinate metrology	3
1.2 Problem statement	6
1.3 Description of work	8
1.3.1 Aims and objectives	9
1.3.2 Thesis outline	11
1.3.3 Limitations and scope	13
1.4 Summary of novelty	14
Chapter 2 Background theory and state of the art	17
2.1 Optical coordinate metrology	18
2.1.1 Close range photogrammetry	19
2.1.2 Digital fringe projection	32
2.1.3 Camera characterisation	36
2.2 Machine learning	41
2.2.1 Artificial neural networks	43
2.2.2 Convolutional neural networks	47
2.2.3 Support vector machines	53
2.2.4 Genetic algorithms	56

2.3	State of the art in machine learning for optical coordinate metrology	58
2.3.1	Machine learning for stereo matching	58
2.3.2	Machine learning for phase unwrapping	62
2.3.3	Machine learning for view planning	64
2.3.4	Machine learning for camera characterisation	65
2.3.5	Machine learning for point cloud analysis	68
2.3.6	Full automation of the measurement pipeline	78
2.4	Summary	80
Chapter 3	Methods	82
3.1	Measurement systems	83
3.1.1	Photogrammetry	84
3.1.2	Fringe projection	86
3.1.3	Texture measurement	87
3.1.4	MMT-LS system	88
3.1.5	FV system	89
3.1.6	Tactile measurement	90
3.2	Measurement data analysis	90
3.3	Computational methods	91
3.3.1	Data acquisition	92
3.3.2	Photogrammetric reconstruction	92
3.3.3	Machine learning methods	93
3.3.4	Rendering methods	93
3.4	Artefacts	94
3.4.1	Characterisation target	94
3.4.2	Measurement artefacts	95
Chapter 4	Improving camera and projector characterisation	98
4.1	Introduction to camera characterisation	100

4.2	Dataset creation	102
4.3	Line-spread function approach	106
4.4	Machine learning approach	108
4.5	Characterisation results	109
4.5.1	Model training results	109
4.5.2	Results on real characterisation data	111
4.6	Discussion of characterisation results	116
4.7	Characterisation conclusions	119
Chapter 5	Automated and optimised view planning from CAD	121
5.1	Introduction to view planning	123
5.2	Proposed view planning approach	123
5.2.1	Visible points analysis	125
5.2.2	Optimisation scheme	131
5.3	View planning results	137
5.3.1	Photogrammetry using optimised camera positions .	137
5.3.2	Comparison of equally spaced and optimised camera positions	139
5.3.3	Comparison with CMM data	143
5.4	View planning conclusions	144
Chapter 6	Automated background removal	147
6.1	Introduction to background removal	149
6.1.1	Previous work	149
6.2	Background removal technique	151
6.2.1	Algorithm detail	152
6.2.2	Experimental procedure	157
6.3	Background removal results	157
6.3.1	Impact on reconstruction efficiency and point density	158
6.3.2	Comparison to CMM	160

6.4	Background removal discussion	162
6.5	Future work on background removal	165
6.6	Background removal conclusions	166
Chapter 7	Generating surface texture data	168
7.1	Introduction to texture generation	170
7.2	Description of the surface generation model	172
7.3	Creating training datasets	175
7.3.1	Industrial coating dataset	175
7.3.2	AM surface dataset	176
7.4	Surface generation results	178
7.4.1	Surface categorisation results	180
7.4.2	Quantitative comparison to real surface data	186
7.5	Discussion of surface texture generation	189
7.6	Example application - producing renders of AM parts	190
7.6.1	AM material shader	191
7.7	Surface texture generation conclusions	193
7.7.1	Future work on surface generation	196
Chapter 8	Pose estimation	198
8.1	Introduction to pose estimation	200
8.2	Monocular method	202
8.2.1	Generation of training data	203
8.2.2	Model	206
8.3	Stereo method	208
8.3.1	Calculating initial alignment	209
8.3.2	Pose optimisation	211
8.4	Monocular pose estimation results	215
8.4.1	Monocular model training results	216
8.4.2	Results on real data	218
8.5	Stereo pose estimation results	219

8.5.1	Results on synthetic data	220
8.5.2	Results on real data	222
8.6	Discussion of both pose estimation approaches	224
8.7	Pose estimation conclusions	229
Chapter 9	Conclusions and future work	232
9.1	Future Work	238
9.2	Summary	241
Bibliography		242
Abbreviations		264
Nomenclature		265
Appendices		268
Appendix A	GOM system performance verification	269
Appendix B	CMM calibration results.	274
B.1	Calibration certificate	275
B.2	Method of calibration	277
B.3	Length error graphs	280
Appendix C	Stereo baseline characterisation	282
C.1	Baseline characterisation results	283
Appendix D	Motion characterisation	287
Appendix E	EfficientNet-B5 architecture	290
Appendix F	GAN based AM material shader.	292
Appendix G	Functions	294
G.1	The closest point between two rays	294
G.2	Custom 'improvement' metric	295

List of Tables

4.1	Parameter distributions used when creating the simulated characterisation dataset.	104
4.2	EfficientNet models evaluated.	109
4.3	Validation results for each model, best performing model shown in bold.	110
4.4	Mean residual magnitude.	114
4.5	Estimated parameters from each dataset.	115
5.1	Performance comparison of three visible point analysis methods: triangulation-based, HPR and enhanced HPR. Including reduction in misclassified points when using enhanced HPR.	131
5.2	Comparison of reconstruction performance for equally spaced and optimised camera locations. Shown in bold are the values for sixteen optimised image positions and twenty-two un-optimised image positions which were shown to perform similarly in Figure 5.11b	141
6.1	Impact of applying background masks on dense reconstruction.	158
6.2	Comparison of time expended at each reconstruction step, averaged across three reconstructions of the Tomas artefact.	159
6.3	Comparison of memory usage at each reconstruction step, averaged across three reconstructions of the Tomas artefact.	160

6.4	Comparison of points reconstructed, averaged across three reconstructions of the Tomas artefact.	160
8.1	Error in position and translation estimate on real images. . .	219
8.2	Pose optimisation results on the synthetic dataset using both minimisation methods.	221
8.3	Results of the stereo pose estimation method on both artefacts over 120 images of each artefact.	223
C.1	Sphere fitting to CMM data results.	283
C.2	Sphere-to-sphere distances extracted from repeated CMM measurements.	284
C.3	Sphere fitting results to a photogrammetric measurement performed by the Taraz system.	284
C.4	Sphere-to-sphere distances compared and used to calculate the scale factor between the photogrammetric point cloud and the CMM data.	284
C.5	Determination of the baseline distance for each stereo pair. .	286

List of Figures

1.1	Renishaw CMM probe design.	4
1.2	Smoothing caused by CMM probe-tip diameter.	5
1.3	Proposed software pipeline indicating where each novel contribution fits in the overall scheme.	10
2.1	Phase difference between the source signal and reflected signal detected by a sensor in a phase difference time of flight system.	19
2.2	Close range photogrammetry measurement pipeline.	20
2.3	Camera coordinate system $[x, y, z]$ within the global coordinate system $[X, Y, Z]$	20
2.4	Parameters required to define the intrinsic orientation of a camera.	22
2.5	Non-linear lens distortion model.	24
2.6	Set of filtered images separated by scale and their respective feature maps produced from the difference between each layer.	25
2.7	Epipolar geometry. The epipolar line (shown in red) is used to match features (shown as crosses) P^1 and P^2 which are then triangulated to localise 3D point P^{xyz}	28
2.8	Example reconstruction results.	31
2.9	Diagram of an example DFP system.	33
2.10	2D phase unwrapping; showing the received image, the wrapped phase, and the unwrapped phase.	34

2.11	Common camera characterisation targets.	37
2.12	Basic ANN architecture. Input nodes shown in green, hidden nodes shown in black (with bias nodes shown in grey) and output node shown in red.	43
2.13	Detail of an ANN neuron (specifically $h_{1,2}$ from Figure 2.12). Here $w_{i,j}^l$, is the weight of a connection, l, j and k are indices representing the layer, input node and output node respectively, and $a(\mathbf{z})$ is a non-linear activation function which must be differentiable at all \mathbf{z}	44
2.14	Kernel convolution on an n -dimensional tensor.	47
2.15	Residual block from a residual neural network (ResNet) based architecture - skip connection highlighted in red.	49
2.16	EfficientNet MBConv block.	50
2.17	A generic generative adversarial network (GAN) showing the generator model, the discriminator model and the zero-sum optimisation through which they are trained. A variety of loss functions are available for calculating the real and fake losses but it is common to use the binary cross-entropy.	51
2.18	Gato for control. A sequence of tokenised observations, separators and previous actions are consumed to predict the next action token in an autoregressive manner. The action is applied, new observations taken, and the process repeats.	53
2.19	The same data separated by different hyper-planes (shown as a solid line). Support vectors highlighted in red in (b).	55

2.20	The process of two selected parent models producing a child for a simple model consisting of two parameters each represented by four binary "genes". The red dotted line in (b) represents the random crossover point determining how each parent contributes to the resultant offspring. The red value in (c) represents a mutated gene, the mutation rate is normally set such that mutation occurs at a relatively low rate.	57
2.21	CNN to predict the 'confidence' that two stereo image patches are correctly matched. The convolutional layers (conv) consist of kernel convolution with the shown kernel size, batch normalisation and ReLU activation.	59
2.22	Example of a twin CNN for learned stereo matching. In contrast to Figure 2.21, here the stereo pair are input to parallel copies of the same network with shared weights. . .	60
2.23	Overview of a LSM, based on Kar <i>et al.</i> [1]	61
2.24	ANN for per-pixel phase unwrapping (Nguyen et al 2019), corresponding pixels from the three wrapped phase maps are passed through the network which produces a single output depth prediction.	63
2.25	Types of scene segmentation. Can be applied to both many data types including images and point clouds.	70
2.26	Multi-class voxel grid segmentation using a 3D CNN.	71
2.27	PointNet architecture [2]. The classification network takes n input points, applies the learned input and feature transforms, pools the results and provides the global classification. The segmentation network takes the global result and local feature maps to perform per-point classification.	72

2.28	Auto-encoder architecture. Training is conducted such that the output can be decoded from the compact representation with minimum difference when compared to the input, therefore, ensuring the compact representation captures the input as fully as possible.	74
2.29	ML approach for multi-view point cloud registration [3]. Features are extracted from the input point clouds and then iteratively matched to find the best alignment.	75
2.30	Point cloud completion network Liu <i>et al.</i> [4]. This approach to point cloud completion uses an encoder to create a compact representation of the input, this is then combined with a random 2D set of points to predict a mapping of those points onto unknown 3D surfaces, a set of these predicted surface are combined to create a coarse completion estimation. This estimation is then further densified through feed-forward information from the input and ResNet layers. . . .	76
3.1	The MMT system, used to collect initial photogrammetry data.	84
3.2	The Taraz system, a Taraz Metrology P2 system.	85
3.3	The DFP system, used to validated system characterisation methods. Dotted red line indicates that the height of the target platform can be adjusted prior to, but not controlled during, a measurement.	86
3.4	The GOM system, a GOM ATOS core 300. Used for comparison to commercial optical CMS. Red line shows controllable motion, dotted red line shows adjustable position prior to measurement.	87

3.5	The MMT-LS system, including laser speckle projector for large scale surface texture measurement.	88
3.6	Focus variation schematic.	89
3.7	Focus variation operating principle.	90
3.8	ICP algorithm.	91
3.9	Characterisation target used to validate the proposed characterisation approach.	94
3.10	CAD data for the four simple artefacts.	96
3.11	CAD for Tomas artefact.	96
3.12	CAD for bracelet.	97
4.1	Camera characterisation shown within the overall proposed measurement pipeline.	99
4.2	An example image of the characterisation target used in this paper. In blue, a zoomed section of the target is shown with OpenCV feature locations shown in red. In green, an example of the cropped sub-images formed around each detected feature is shown.	101
4.3	Ellipse geometry parameters.	103
4.4	Effect of changing the specular parameters on randomly sampled ellipses with all other parameters set to be constant.	104
4.5	Feature sub-image simulation method.	105
4.6	Comparison of real and simulated feature sub-images.	106
4.7	LSF approach to ellipse centre localisation refinement.	107
4.8	EfficientNetB5 metric evolution during the training period.	110
4.9	Example characterisation target images included in the cooperative dataset.	113

4.10	Example characterisation target images included in the uncooperative dataset.	113
4.11	Pixel distributions internal to each ellipse from the cooperative and uncooperative datasets as a distance from a threshold determined by Otsu's method [5].	114
4.12	Distribution of residual errors in the reprojection of features for each characterisation method for each data set	115
4.13	Showing some features from the uncooperative dataset, where blue dots have been discarded by the RANSAC algorithm and red dots have been kept as estimated boundary points. (a) and (b) show cases where ellipse fitting failed to produce a good outcome and (c) and (d) show cases where ellipse fitting was successful despite some outliers.	117
5.1	View planning shown within the overall proposed measurement pipeline.	122
5.2	Outline of the view planning method and camera pose parameterisation	124
5.3	Enhanced visible points analysis technique.	129
5.4	Near-edge point classification based on different threshold values for four objects.	129
5.5	Misclassified points when using HPR which are correctly classified when using the proposed enhanced visible point analysis. Visible points classified as invisible are shown in blue, and invisible points classified as visible are shown in orange. Scale is in millimetres.	130
5.6	View optimisation scheme.	132
5.7	Camera positions from which the maximum number of surface points are visible for each artefact.	133

5.8	Global optimisation for the four test artefacts showing the number of camera views required to pass the objective function threshold.	136
5.9	Dense colourised reconstructions for the pyramid artefact using the proposed optimised camera positions.	138
5.10	Deviations in measurement results from the reference measurement given by the GOM system.	139
5.11	Comparison of the standard deviation in PTM distances for both optimised and equally spaced camera imaging positions.	141
5.12	Comparison of the PTM deviations of the pyramid and pillar point clouds from GOM results.	142
5.13	PTM distances for a CMM comparison of the reconstructions of the pyramid artefact.	143
6.1	Background removal shown within the overall proposed measurement pipeline.	148
6.2	Example image and image contours, open contours shown in red, maximum closed contour shown in green.	151
6.3	The proposed background removal algorithmic pipeline.	152
6.4	Impact of bilateral filtering on Canny edge detection.	154
6.5	Background removal pipeline.	156
6.6	Example results of background removal across a range of artefacts.	156
6.7	Imaging positions used for every scan.	157
6.8	Comparison of dense reconstruction of the pyramid artefact.	158
6.9	Comparison of dense reconstruction of the tomas artefact.	159
6.10	Comparison of PTM distances for the pyramid artefact.	160
6.11	Comparison of the distribution of PTM distances for the pyramid artefact.	161

6.12	Comparison of PTM distances for the Tomas artefact.	162
6.13	Comparison of the distribution of PTM distances for the Tomas artefact.	162
6.14	Comparison of the dense reconstruction of cylindrical features.	163
6.15	Example failure case of the background removal, masking contour shown in red.	164
7.1	Surface texture generation shown within the overall pro- posed measurement pipeline.	169
7.2	The process of doubling the resolution smoothly in a PG- GAN using the α parameter	173
7.3	Twelve example encoded images taken from the industrial coatings dataset, showing the range of different surface types present in the training data.	176
7.4	Bracelet artefact used to create the AM dataset.	177
7.5	Twelve example encoded images taken from the AM dataset showing the large variation present between surfaces included the dataset.	177
7.6	Surface types and their locations on the bracelet artefact with examples taken from the AM surface dataset.	178
7.7	A comparison between a random sample of twenty-eight coating surface images, both real and generated.	180
7.8	A comparison between a random sample of twenty-eight AM surface images, both real and generated.	181
7.9	Surface topography height maps of real measurement data from the AM surface dataset compared to decoded fake data from the generator.	182

7.10	Categorisation CNN, which takes an input image, extracts features through a sliding window kernel convolution, flattens the feature maps, feeds through a fully connected layer and produces a predicted class label (T: top,U: up-skin, D: down-skin).	183
7.11	Categorisation CNN training history. To prevent overfitting, ten percent of the dataset was used for cross validation and model weights restored to the maximum validation accuracy.	184
7.12	Comparison of encoded depth data for an example of a common misclassified surface with real surfaces around the borderline of the up-skin/down-skin categories	185
7.13	An unrepresentative image showing encoded depth data produced by the generator showing clear weld tracks and noise from agglomerated particles.	186
7.14	Comparison of the mean and 95% confidence intervals of ISO height parameters for real and generated AM surfaces. .	187
7.15	Comparison of the mean and 95% confidence intervals of ISO spatial parameters for real and generated AM surfaces.	188
7.16	Visualisation of how the Poliigon UberMapping node can be used to create an infinite and visually seamless image texture. (b) and (c) shown at 10:1 scale compared to (a). . .	192
7.17	Material shader overview, full shader model included in Appendix F.	193
7.18	Material shader applied to flat plane at a range of scales. . .	194
7.19	Material shader applied to complex 3D model.	195
8.1	Pose estimation shown within the overall proposed measurement pipeline.	199
8.2	Real and simulated versions of the MMT system.	203

8.3	Intermediate material shader.	204
8.4	Example of weld-line artefact visible at sharp corners on PBF parts.	205
8.5	Comparison of real and synthetic image of the pyramid measurement artefact. The dashed connection allows information flow during the forward pass only, not during back propagation.	206
8.6	Monocular pose estimation CNN. The dashed connection allows information flow during inference only, not during back propagation.	207
8.7	Initial alignment of CAD data using stereo binary image masks.	209
8.8	Stereo image pair taken by the Taraz system.	210
8.9	Example initial alignment result. Predicted alignment shown in yellow wireframe overlaid over each image.	211
8.10	Loss function visualised on the initial pose prediction given by the rough alignment procedure. Correctly classified pixels shown in green, misclassified background pixels shown in blue, misclassified object pixels shown in red.	212
8.11	Example result of refined pose estimation.	215
8.12	Training and validation losses of the initial network at each training epoch.	216
8.13	Separated loss contributions.	217
8.14	Example pose estimations for each artefact.	217
8.15	Example result on real image.	218
8.16	Suzanne mesh.	220
8.17	Pose predictions for each artefact, each prediction shown is overlaid in yellow wireframe on the input stereo image pair.	222

8.18	Pose predictions for each artefact, each prediction shown is overlaid in yellow wireframe on the input stereo image pair.	223
8.19	Local minima optimisation result on the Tomas artefact. The local z axis rotation can be seen to be 180° incorrect while the translation is still relatively accurate.	224
8.20	Loss function visualisation for the rough and optimised alignments of high performing samples taken from the real and synthetic datasets.	227
8.21	Effect of an imperfect mask on the optimised pose prediction.	228
8.22	Top view showing the optimised camera positions. (a) About the centre of the instrument; (b) corrected for the placement of the artefact using the CNN pose prediction	231
9.1	The now complete automated and optimised software pipeline.	233
9.2	MCDDM demonstrator CAD.	239
9.3	MCDDM demonstrator data flow.	240
C.1	Sphere arrangement for the characterisation of the stereo baseline distance.	282
D.1	Global axis alignment for the Taraz system. Each circle indicates an imaging position in the dataset.	288
E.1	Building blocks of the EfficientNet model.	290
E.2	Full EfficientNetB5 model	291

Chapter 1

Introduction

Current approaches to optical coordinate metrology are highly dependant on a human operator and can be slow in both data acquisition and data processing. Inexperienced operators often capture too much data which further delays both data acquisition and data processing and they often capture these data from sub-optimal positions, harming the measurement outcome. **It is the aim of this thesis to develop a software pipeline to enable, for the first time, a fully automated coordinate measurement system which conducts measurements in an optimised manner.** The thesis will consider all parts of the measurement pipeline including developing an accurate model of the imaging system, planning the measurement and capturing and processing the measurement data. Automation is desirable to remove any operator dependence, allowing inexperienced users to achieve good measurement results. Optimisation in this case refers to the desire to reconstruct high quality and complete measurement data of an object while minimising time, computational cost and unwanted background information. Both automation and optimisation are achieved by exploiting *a priori* information, particularly through the use of machine learning (ML).

In this Chapter, a brief introduction to coordinate metrology is given and current issues with optical approaches are summarised. The proposed pipeline to enabling full automation of an optical measurement system is then presented along with a summary of the novel contributions developed in this thesis which enable this pipeline to be realised.

1.1 Coordinate metrology

According to the international vocabulary of metrology, metrology is "the science of measurement and its application" [6]. Coordinate metrology is

a branch of dimensional metrology which aims to construct a set of three-dimensional (3D) points representing the surface of some object of interest. When a surface is measured it can be thought of in two components; form and texture, which together make the surface topography. Form is the underlying shape of a part or fit to a measured surface [7]. Texture comprises the geometrical irregularities at the surface which do not contribute to the form of the surface [8]. The output of a form measurement is typically a 3D point cloud, an unordered data structure which is a list of (x, y, z) coordinates paired with additional data such as colour and local surface normals. In contrast, the output of a texture measurement is a 2.5D height map, a regular grid of (x, y) locations with height data (z) at each location. In a coordinate measurement, it is normally the form that is captured by a measurement, although some coordinate measurement systems (CMSs) can also be used to extract large scale texture as is demonstrated in Section 7.3.1. Traditionally form is measured using tactile probing systems, so called co-ordinate measurement machines (CMMs). Figure 1.1 shows the touch trigger probe used in Renishaw plc CMMs [9].

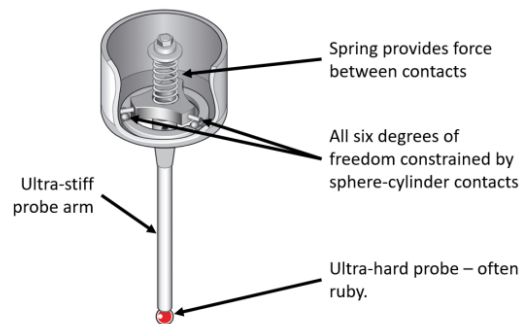


Figure 1.1: Renishaw CMM probe design.

CMMs can be calibrated to provide metrologically traceable measurement results, meaning that measurements provided by a CMM can be related to a reference through an unbroken chain of calibration, each con-

tributing to the uncertainty [6]. However, a contact probe such as that shown in Figure 1.1 has many disadvantages [10]: due to the need to record many points which must all be individually contacted it takes a large amount of time to complete a measurement, the physical contact of the probe can cause the measurement itself to influence the result, some complex geometries may be unreachable by the probe, and any surface texture detail which has feature sizes smaller than the diameter of the probing tip will be smoothed out as shown in Figure 1.2.

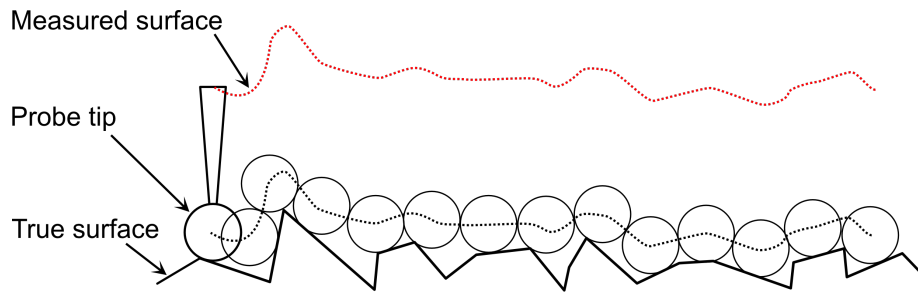


Figure 1.2: Smoothing caused by CMM probe-tip diameter.

The advent of additive manufacturing (AM) [11] has enabled huge design freedom in modern parts [12] - the kinds of complex freeform surfaces enabled by AM are often unsuitable for contact measurement. For this reason, non-contact optical CMSs are growing in popularity within this sector. In particular, camera based triangulation methods can offer high point density and surface coverage with low data acquisition times. These methods have the added benefit of having inexpensive "off-the-shelf" hardware requirements. Arguably the biggest drawback of using optical camera based coordinate measurements is that they cannot be calibrated and thus their measurements cannot be paired with a statement of uncertainty. Although calibration of optical CMSs is an active area of research [13], it falls outside the purview of this thesis.

The most popular camera based triangulation methods are close range

photogrammetry (from hereon referred to as photogrammetry), and digital fringe projection (DFP). The details of photogrammetry and DFP are covered in Section 2.1.1 and Section 2.1.2 respectively but, at a high level, both operate by triangulating 3D points from extracted 2D image features between a minimum of two imaging positions using a projective camera model. Photogrammetry approaches coordinate measurement in a passive way through extracting, matching and triangulating features present in a set of overlapping raw images around a part. DFP, in contrast, uses an active sensing approach where a projector is used to project a known pattern onto the surface of a part, a camera records the deformation of the known pattern caused by the surface form and depth information is extracted from this deformation.

1.2 Problem statement

As stated previously, photogrammetry and DFP suffer from a high dependence on the operator, and humans are prone to error and sub-optimal operation. For example, a requirement for producing useful measurement results is the development of a mathematical model of any cameras and projectors in the system. The generation of these models is referred to as characterisation, as is covered in Section 2.1.3. Characterisation of the camera and projector models is a manual process wherein the operator must capture images of a target artefact at a range of positions and orientations within the field of view (FOV) of the imaging system. The choice of characterisation locations can cause poor measurement results if conducted sub-optimally as the mathematical model of the optical systems may not accurately represent the real system. This issue can be addressed in two ways, removing the user entirely and automating the characterisation pro-

cedure or by making the characterisation algorithms more robust to poor imaging conditions caused by the inclusion of suboptimal imaging locations in the characterisation dataset.

Additionally, the measurement result is highly sensitive to the imaging strategy (ie. how many images to capture and from which positions to capture them). For example, complex parts may cause self occlusions from some viewing angles leading to data loss. This issue is often addressed Naïvely by simply capturing data from a very high number of positions, minimising the probability of self occlusion and poor surface coverage. However, this leads to huge inefficiencies in data processing. In DFP point clouds are reconstructed individually from each view and stiched together to create the overall form of the part, in creasing the number of point clouds and therefore stitching operations which must be computed leads to large computational costs. In the case of photogrammetry, features detected in each image are matched putatively putatively (ie. each feature in every image is a potential match with all features in every other image) which leads to a combinatorial explosion with each additional image in the dataset. The desirable approach would be to remove the user from both planning the imaging strategy and performing the data capture. Instead, the imaging locations would be optimised based on the specific geometry of the object in such a way that maximises surface coverage and reconstruction quality while minimising the number of imaging positions. This optimised plan would then be conducted autonomously in a computer numerical control (CNC) measurement system.

A further inefficiency is caused because there is often a large amount of background data captured within the system's FOV. These background pixels are not distinguished from the pixels that contain information about the object of interest and are reconstructed alongside the object. Not only does this lead to computational cost spent reconstructing 3D points which

must be manually removed at a later stage but can also impact the quality of the final measurement outcome. The measurement outcome can be impacted for a few reasons, first the background is often beyond the focal plane of the imaging system leading to blurred data which can cause erroneous matches to be triangulated. Furthermore, it is common practice to automate data collection by placing the object on a rotation stage so that it can be rotated relative to the camera and images captured around the part. The use of a rotation stage means that between images the part moves but the background remains static. Any matches detected in the background therefore have a conflicting spatial relationship to those detected on the part's surface. In seeking to minimise the overall triangulation error, these static matches can lead to the triangulation algorithms incorrectly localising the object points in an attempt to account for the conflicting information provided by the static matches.

In this thesis, a software pipeline is developed for the first time which can enable full automation to remove user dependence completely, with the long term goal of integrating the proposed solutions into a hybrid photogrammetry/DFP multi-view system. Optimisations are built in to the pipeline to directly address the inefficiencies outlined above. Many novel contributions to science were required to achieve this goal and these are outlined in the following section.

1.3 Description of work

This thesis considers the entire measurement procedure from camera characterisation, through view planning, to image capture and processing. An information rich metrology (IRM) approach is taken where *a priori* information about both the measurement system and the object to be measured

is exploited. Of particular value to the aims of this thesis is the ubiquity of computer aided design (CAD) data which almost all manufactured parts will have associated with them. Although any manufactured part will deviate from its design data to some extent due to manufacturing errors and tolerances the CAD should represent the overall form and shape of the part well enough to inform the measurement strategy.

The main contribution of this thesis is the pipeline shown in Figure 1.3 which will enable automation and optimisation to be realised. Each novel development required to deploy this pipeline in software is highlighted with a red superscript number which refers to the chapter which details that contribution.

In summary, known information about the object, characterisation target and measurement system (from their respective CAD and data-sheets) are combined with an initial data acquisition. An optimal view plan is generated in the form of a list of imaging positions. The object pose in the measurement volume is estimated and the measurement plan adjusted to account for this alignment without the need for specialist fixturing or user input. The cameras are characterised using a new procedure which can account for adverse imaging conditions. The measurement data can then be autonomously collected with a CNC measurement system. Background data are removed autonomously before reconstruction which leads to the final measurement outcome.

1.3.1 Aims and objectives

The aims of this project can be simply summarised into three main objectives:

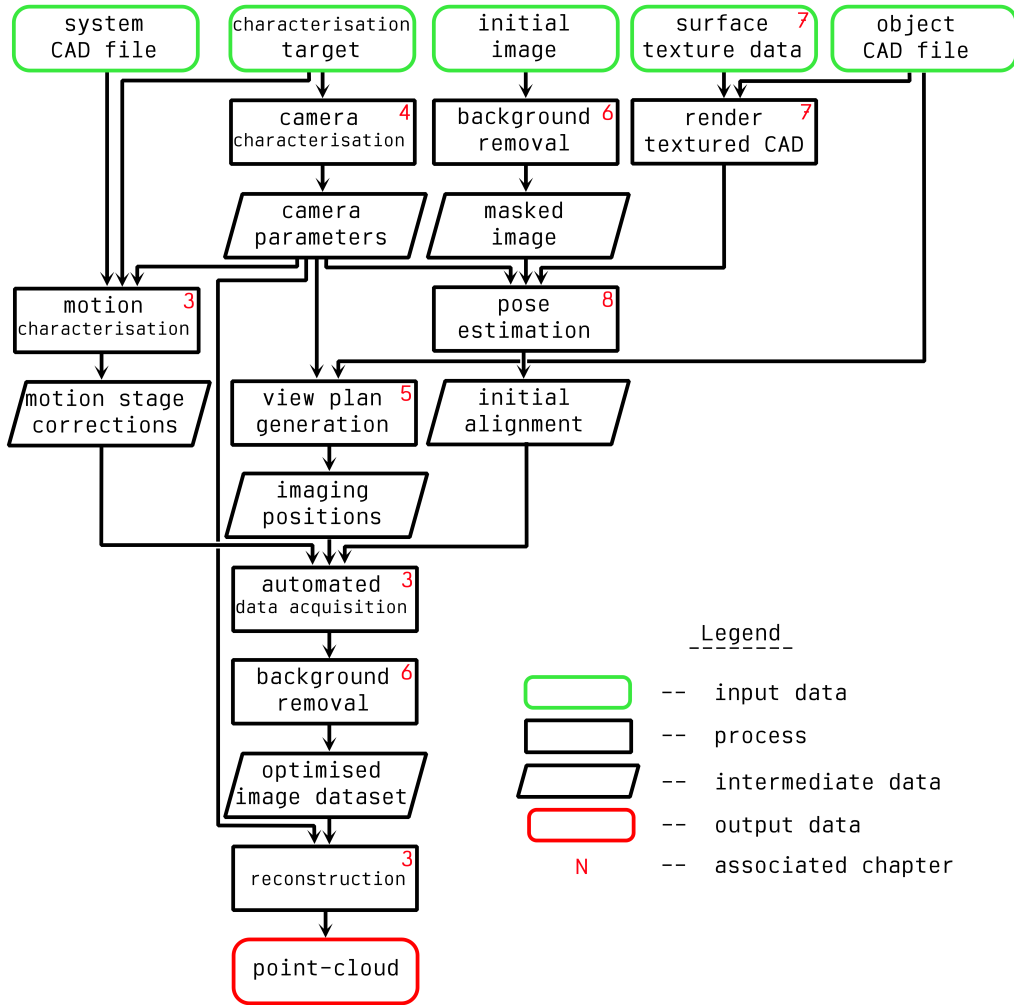


Figure 1.3: Proposed software pipeline indicating where each novel contribution fits in the overall scheme.

1. To create a software pipeline which will enable the creation of a fully automated measurement system.
2. To develop algorithms as part of this pipeline to perform measurements in a way which maximises surface coverage and reconstruction quality while minimising computational expense and time.
3. To allow for arbitrary placement of the measurement object within the measurement volume, ie. no fixturing or fiducial marking required.

The work performed to achieve these goals is summarised in the following

section.

1.3.2 Thesis outline

Each results chapter constitutes a self contained part of the pipeline which is, in itself, a novel and valuable contribution to the field. It is made clear in each case where the given chapter fits into the overall pipeline and how it contributes to overall goals of automation and optimisation.

Chapter 2 provides all necessary background theory needed to understand the methods presented in this thesis. This includes photogrammetry, DFP and camera characterisation. Also presented is the ML theory required to implement the solutions to automation and optimisation of optical CMSs proposed later in the thesis. This chapter then presents the current state of the art in machine learning for optical coordinate metrology with particular focus on gaps in the literature which this thesis aims to fill.

Chapter 3 presents the any common methods used across the results chapters. This includes descriptions of all experimental setups, reconstruction methods, measurement artefacts, characterisation artefacts, and software used.

Chapter 4 presents a new approach for intelligent camera characterisation with the aim of improving overall performance while being more robust to unoptimal imaging conditions such as those caused by specular reflection. Specifically, a new approach to refining the localisation of characterisation targets within an image. A convolutional neural network (CNN) based on the EfficientNet-B5 architecture is trained to produce sub-pixel corrections to the location of target features. It is shown that using this model provides a large quantitative improvement to the characterisation result as measured by the reprojection error. Further, it is shown that the proposed

model is more robust to adverse imaging conditions such as noise and specular reflections than a competing refinement method based on traditional image processing.

Chapter 5 focuses on the optimisation of the imaging strategy. First, an improved method for evaluating which surface points are visible from a given viewing position is proposed. This visible point analysis method is used in a genetic optimisation to find the minimum number of imaging positions which can produce a high quality measurement result as assessed by a custom global objective function. This proposed procedure is conducted on a range of test artefacts and shown to produce high quality scans from a very low number of images as assessed through comparison to other photogrammetry measurements, commercial DFP measurements, and tactile CMM measurements.

Chapter 6 proposes a solution for autonomous removal of background data from the images comprising a photogrammetric scan. This is shown to have numerous benefits including reduced processing time, improved memory usage, decreased numbers of background points reconstructed, and increased object point density. It is also shown that the measurement result is improved quantitatively through greater agreement to CMM and improved reconstruction of surface features.

Chapter 7 presents a method for producing synthetic surface texture data. A progressively growing generative adversarial network (PG-GAN) is trained to produce a wide range of surface types which are shown to be representative, but distinct, from real measurement data. This model was developed to enable photo-realistic renders of manufactured parts from their CAD data, an approach to using the surface generation model in a material shader is presented. These simulated images are used to produce synthetic photogrammetry data used to train models in Chapter 8.

Chapter 8 Finally, two approaches to object pose estimation are presented.

A monocular method, which relies on simulated data to train a CNN to directly regress the six degrees-of-freedom (DoF) pose of the object relative to a single camera. A stereo method, which uses binary masks generated by the algorithm presented in Chapter 6 alongside predicted masks generated by raycasting the CAD data through a characterised camera model to minimise a loss function defined between the real image masks and the predicted masks. It is shown that both of these methods can be used on real photographic data, with the stereo method making the lowest error predictions on average. Either of these techniques, depending on the system requirements, can then be used to establish the spatial relationship between the camera system and the object. This spatial relationship is then used to perform an automated measurement based on the optimised view plan, to be conducted without specialised fixturing or fiducial marking.

1.3.3 Limitations and scope

To keep the scope of this project feasible, the following limitations are placed upon it:

- It is assumed that all parts which would be conceivably measured in a manufacturing metrology setting will have associated CAD data that is freely available to the measurement system. Given the particular focus taken to AM where CAD data is an implicit requirement, this assumption seems reasonable.
- The project will focus on applications to photogrammetry specifically, but applications to DFP will be highlighted when relevant.
- This thesis will feed into a separate project to embody the proposed

pipeline in a physical measurement system, but will not itself contain any hardware development or integration. The proposed hardware solution is summarised in future work Section 9.1.

1.4 Summary of novelty

The novel contributions made in this thesis are:

1. A first of its kind software pipeline, allowing fully automated and optimised coordinate measurement of generic geometry 3D parts for the first time. Presented in Figure 1.3 and shown again at the start of each results chapter to contextualise that chapter's place in the overall scope of the project.
2. A new hybrid ML approach to characterisation with a specific goal of increasing robustness to adverse imaging conditions, thus reducing the impact of poor choice of target locations and allowing a greater range of target positions to be included in the characterisation dataset. Compared to the current standard characterisation software the approach is shown to robustly offer improvements of 50% as measured by the mean magnitude residual. Additionally the approach is shown to outperform a state of the art characterisation refinement approach in uncooperative imaging scenarios leading to more accurate camera modelling.
3. An improved method for analysing visible surfaces of a given object from a given view using CAD data is presented in Chapter 5. This is used during view planning to select optimal imaging locations. The proposed method combines the best qualities of two pre-existing

methods and is shown to reduce the number of misclassified visible points by up to 57% as compared to the previous state of the art while maintaining fast operation.

4. A procedure for the global optimisation of the imaging strategy for a given geometry is also presented in Chapter 5 and is shown to provide good quality reconstruction results from very few images. Optimised reconstructions using as few as 18 images are shown to have greater agreement to CMM measurement than unoptimised reconstructions (using current standard industrial practice) with 60 images, while also having much lower data processing times.
5. A generative ML model for the generation and categorisation of synthetic surface texture data is presented in Chapter 7 which can be used to produce realistic renderings of a manufactured part from its CAD data. A completely novel approach to surface generation, being much less expensive than physics based models and more representative than pure mathematical representations. The model is used to create large datasets to train further ML models more robustly. The generative results are shown to be quantitatively representative of real measured surface data, securing this model as a highly useful tool.
6. A method to autonomously segment background pixels from object pixels within an image and to remove the background pixels is presented in Chapter 6. This is used to during both reconstruction and for object pose estimation to enable fixture-less measurements. Using masked images directly in reconstruction is shown to improve the measurement result agreement with CMM, reduce the number of reconstructed background points by up to 98 %, and lead to reduced reconstruction times by up to 41 %. The proposed method is more

robust than the previous state of the art due to the exploitation of known properties of the imaging system.

7. A monocular method for the 6 DoF pose estimation of CAD data from a single image is presented in Chapter 8. . This pose is then used to adjust the measurement plan to account for an arbitrary placement of a given part. Both the model architecture and dataset generation method are novel contributions. The model is shown to generalise well onto real image data, accurately localising the object to within an average of 13 mm with a low rotational residual error. Previous state of the art was restricted to predicting bounding boxes or cuboids, often with performance evaluated on benchmark datasets which may not be representative of true objects and imaging conditions.
8. Also presented in Chapter 8 is a stereo approach to pose estimation which uses the image background removal algorithm presented in Chapter 6 to iteratively refine an initial estimation. The stereo model is shown to locate artefacts to within an average of 0.44 mm when evaluated on a set of 240 test photographic images. This approach does not require any pre-training, a large advantage over the previous state of the art. The stereo model also benefits from improved certainty in prediction due to consensus across multiple views.

Chapter 2

Background theory and state of the art

Some parts of this chapter have been previously published as the author's contributions to an Institute of Physics book chapter and a topical review paper:

Eastwood J, Sims-Waterhouse D, Piano S 2020 Machine Learning Approaches *in* Leach R K *Advances in Optical Form and Coordinate Metrology* (Bristol: IoP Publishing).

Catalucci S, Thompson A, Eastwood J, Zhang Z M, Piano S, Branson D T, Leach R K 2022 Smart optical coordinate and surface metrology *Meas. Sci. Technol.* **34** 012001.

This chapter is comprised of three sections: the background theory needed to understand camera-based optical coordinate metrology, the background theory required to understand the machine learning methods used to address the shortcomings of these measurement methods, and a state of the art review of the current literature regarding applications of machine learning approaches to optical coordinate metrology.

2.1 Optical coordinate metrology

As was discussed in Section 1.1, optical methods of coordinate measurement are growing in popularity. Initial optical CMSs were based on time of flight methods such as light detection and ranging (LiDAR) [7]. In the case of LiDAR, a laser is directed at a surface of interest, the light is reflected from the surface and detected when it returns to a sensor located near the laser emitter. As the speed of light is a known universal constant ($c \approx 3 \cdot 10^8$ ms⁻¹), the distance to the surface point can be calculated from $d = \frac{ct}{2}$ where t is the time of the round trip from emitter to detector. However, to measure a height change of 1 mm requires the time of flight to be resolved on the order of picoseconds. While some optical CMSs directly use the time of flight [14] often, for improved precision, depth is instead calculated from the phase change in the returning signal compared to the reference signal $S(t)$ [15],

$$S(t) = A(t) \cdot e^{i \cdot \phi_s}, \quad (2.1)$$

where $A(t)$ is the amplitude and ϕ_s is the phase. The depth d can be calculated from the phase difference,

$$d = \frac{c}{4\pi f_m} \cdot (\phi_r - \phi_s), \quad (2.2)$$

where f_m is the frequency modulation and ϕ_r is the reflected phase. Figure 2.1 shows how this phase difference is detected at the time of flight sensor.

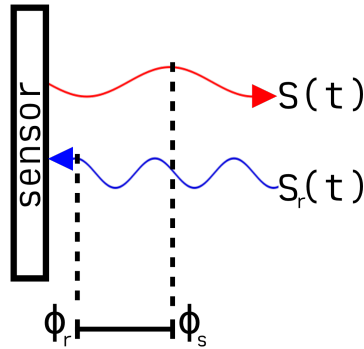


Figure 2.1: Phase difference between the source signal and reflected signal detected by a sensor in a phase difference time of flight system.

Despite this phase difference approach, time of flight CMSs are still limited by spatial and depth resolutions when compared to triangulation based methods. Furthermore, they still require the laser point to scan a rasterised path across the surface which can take a relatively long time. In comparison, camera based triangulation methods can acquire data covering the entire visible surface very quickly.

2.1.1 Close range photogrammetry

From Luhman *et al.* [16], "close range photogrammetry ... uses accurate imaging techniques to analyse the three-dimensional shape of a wide range of manufactured and natural objects." . Close range photogrammetry is a triangulation based optical coordinate measurement method which extracts features from a set of photographic images, matches corresponding features between images, and triangulates these features to reconstruct the object. After triangulation, a coarse point cloud representing the object's surface will have been produced, these coarse measurements are then often densified to capture more surface detail. Figure 2.2 summarises the main

stages in a photogrammetric measurement.

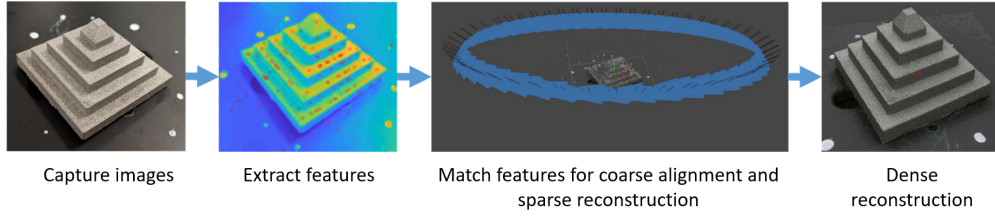


Figure 2.2: Close range photogrammetry measurement pipeline.

2.1.1.1 Camera model

to implement the pipeline shown in Figure 2.2 a mathematical model of a camera is required. This model transforms 3D world coordinate points to 2D pixel coordinates and can be defined using two transformation matrices referred to as the camera's intrinsic and extrinsic matrices [17]. The extrinsic matrix defines a coordinate transform from the world reference frame $[X, Y, Z]$ to the camera's reference frame $[x, y, z]$. The camera reference frame is shown in Figure 2.3, the z -axis is co-linear to the camera's principal axis and the x, y plane is parallel with the imaging plane.

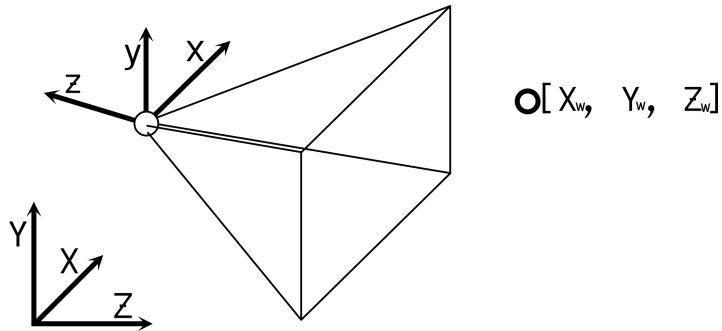


Figure 2.3: Camera coordinate system $[x, y, z]$ within the global coordinate system $[X, Y, Z]$.

The world-to-camera coordinate transform can therefore be defined as,

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = [\mathbf{R}, \mathbf{T}]_{(4 \times 4)} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2.3)$$

where \mathbf{T} is a translation vector and \mathbf{R} is a rotation matrix which together form the extrinsic matrix E as,

$$\mathbf{E} = [\mathbf{R}, \mathbf{T}] \quad (2.4)$$

where,

$$\mathbf{R} = \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad (2.5)$$

where θ, ϕ, κ are the Euler angle rotations about the x, y, z axes respectively; and,

$$\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix}, \quad (2.6)$$

where t_x, t_y, t_z are the x, y, z translation components. Once the 3D coordinates have been transformed from world space to camera space they can be projected onto the 2D imaging plane. To enable a camera-to-image transformation the parameters which define the intrinsic matrix must be known which are visualised in Figure 2.4. The intrinsic parameters include; the focal length f which is the distance of the imaging plane from the camera principal point (the point on the image plane which the perspective center is projected onto), the principle point offset parameters c_x and c_y

which define the offset of the principal point relative to the origin of the image coordinate system $[u, v]$, the pixel pitch p_u, p_v defined in meters, and the skewness parameter s which is defined as,

$$s = \frac{-f}{p_u} \cdot \cot(\gamma), \quad (2.7)$$

where γ is the skew angle; the parameter s is often assumed to be zero.

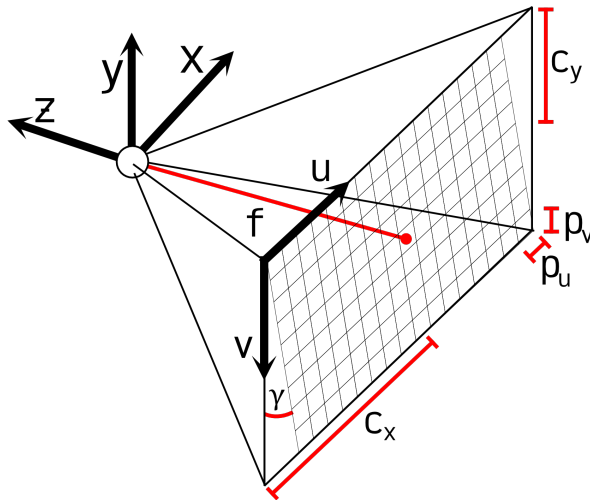


Figure 2.4: Parameters required to define the intrinsic orientation of a camera.

The intrinsic matrix (\mathbf{K}) can be constructed from the intrinsic parameters in the form,

$$\mathbf{K} = \begin{bmatrix} \frac{f}{p_u} & s & c_x & 0 \\ 0 & \frac{f}{p_v} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.8)$$

The intrinsic matrix describes the coordinate transform from the camera coordinate system to image coordinates i.e. the camera-to-image transform. Composing the world-to-camera transform given in Equation 2.3 with this camera-to-image transform results in the final world-to-image transforma-

tion given by,

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{p_u} & s & c_x & 0 \\ 0 & \frac{f}{p_v} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2.9)$$

where s is an arbitrary scalar and the combination of the intrinsic and extrinsic matrices is referred to as the projection matrix \mathbf{A} . The linear camera model given in Equation 2.9 is enough to describe an idealised pin-hole camera. However, in order for the camera model to generalise to real cameras the model must be extended further to consider non-linear lens distortion. Lens distortion is typically modelled by considering two distortion components, radial and tangential. Figure 2.5 shows an exaggerated representation of these two distortion components independently and then, in Figure 2.5d, combined in the full distortion model.

The distortions shown in Figure 2.5 are modelled using the popular Brown-Conrady model [18]. The Brown-Conrady model parameterises lens distortion with five coefficients,

$$\mathbf{dist} = [k_1, k_2, k_3, p_1, p_2], \quad (2.10)$$

where k_n are radial parameters, and p_n are tangential parameters. The radial distortion component can then be corrected for using,

$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \quad (2.11)$$

$$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \quad (2.12)$$

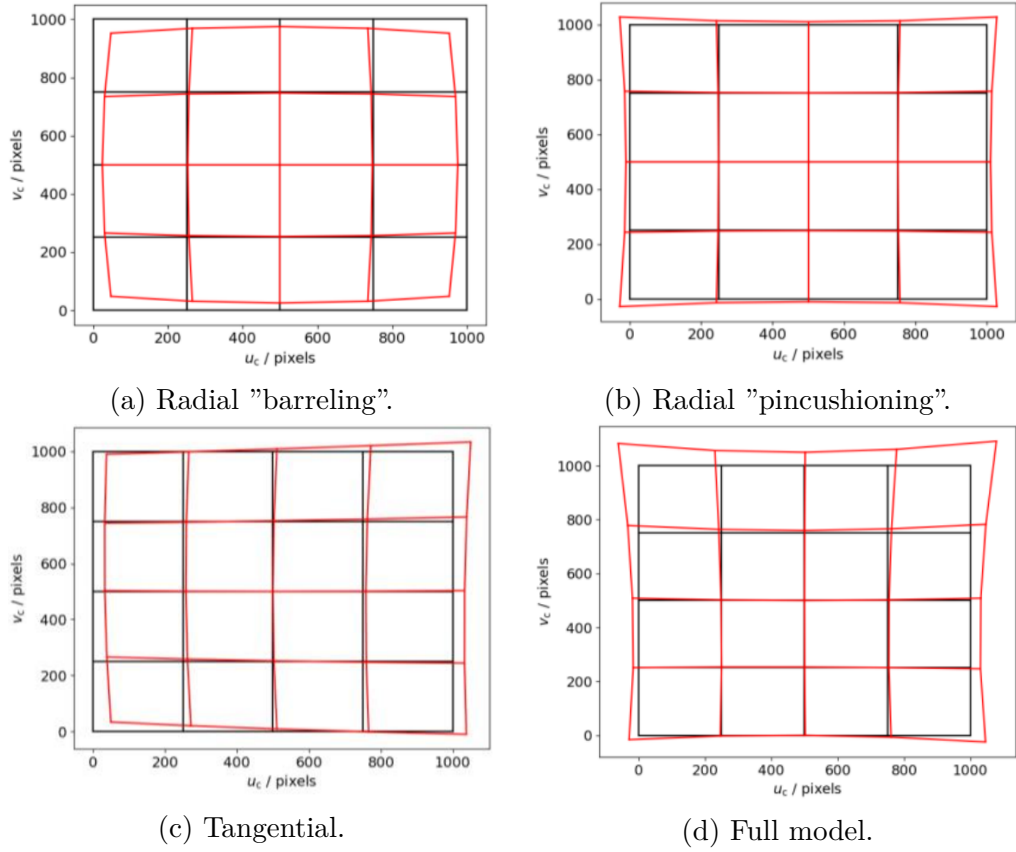


Figure 2.5: Non-linear lens distortion model.

where (x', y') are the corrected coordinates, (x, y) are the imaged coordinates before correction, and $r = \sqrt{x^2 + y^2}$ is the radial distance relative to the distortion centre (assumed to be in the center of the image). The tangential distortion can also be corrected for using,

$$x' = x + 2p_1xy + p_2(r^2 + 2x^2), \quad (2.13)$$

$$y' = y + p_1(r^2 + 2y^2) + 2p_2xy, \quad (2.14)$$

The determination of the intrinsic and extrinsic matrices, alongside any distortion parameters, is the process of camera characterisation (referred to often in the literature as camera calibration) [19] which is described in Section 2.1.3. Accurate characterisation of the camera parameters is key to producing accurate measurement results. The stages of the photogrammet-

ric pipeline, described in the following sections, assume that all cameras in the network have been accurately characterised.

2.1.1.2 Feature extraction

The first stage of the photogrammetric pipeline is to extract features from the image data collected. There are many possible approaches for feature extraction, early work often used the Movarec operator [20] or the Förstner operator [21]. It is important that any extracted features are invariant to scale, rotation, translation and affine transformations. Furthermore, it is desirable that features be minimally variant under illuminations changes, noise and small distortions. These invariances are important to a robust feature matching step. The most commonly used approach for both feature extraction and matching is the scale invariant feature transform (SIFT) algorithm [22]. SIFT features are extracted using a difference of Gaussians (DoG) approach. In DoG an image pyramid is formed by convolving Gaussian kernels of different scales over the image as is shown in Figure 2.6.

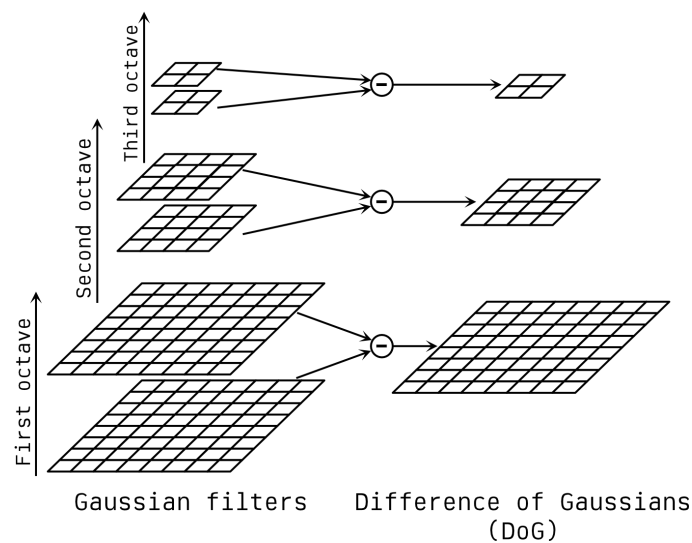


Figure 2.6: Set of filtered images separated by scale and their respective feature maps produced from the difference between each layer.

A Gaussian kernel with a standard deviation of $\sigma = \sqrt{2}$ is applied to the captured image to create a smoothed image A , the same kernel is convolved with A to produce an image B which has been incrementally smoothed by a further factor of $\sigma = \sqrt{2}$ giving a total smoothing compared to the original image of $\sigma = 2$. When the smoothing has doubled the resolution of the image can be downsampled by a factor of 2, this creates the second octave of Gaussians shown in Figure 2.6. The DoG pyramid is then created by taking the difference between each smoothed image. SIFT features are detected as the local extrema over both scale and space. Each pixel intensity is compared to its neighbours, at the current scale - if it is a local maxima or minima it is a feature candidate. The pixel is then compared to its equivalent neighbouring pixels at the next lowest layer of the pyramid, if the pixel is still a maxima or minima this process is repeated for each layer of the pyramid. This process is efficient as most pixels will be discounted as potential features with few comparisons.

To improve stability, each SIFT feature location is paired with a rotation histogram calculated from local gradient variations in the image. Image gradients and gradient directions can efficiently be calculated from an image \mathbf{I} using the Sobel operator [23],

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \mathbf{I} \text{ and } \mathbf{G}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{I} \quad (2.15)$$

where \mathbf{G}_x and \mathbf{G}_y are the x and y components of the local image gradient. From this the gradient magnitude \mathbf{G} and direction of the local gradient Θ

can be simply calculated as,

$$\mathbf{G} = \sqrt{\mathbf{G}_x + \mathbf{G}_y}, \quad (2.16)$$

$$\Theta = \text{atan2}(\mathbf{G}_y, \mathbf{G}_x). \quad (2.17)$$

It has been shown that these features fit the invariances that are desired, the original paper tests the robustness of the feature localisation under a large set of transforms and distortions with favourable results.

2.1.1.3 Feature matching

Once a set of features has been successfully extracted from a set of images correspondences between these features must be found to enable matching points in different images to be triangulated. Each detected feature is paired with a description vector which is constructed from the $16 \text{ pixel} \times 16 \text{ pixel}$ region surrounding the feature. This region is split into (4×4) square pixel sub-regions and the local gradient direction is calculated for each sub-region and stored in an eight-bin histogram. The sub-region rotations are given relative to the rotation of the feature, which was calculated previously, to maintain rotational independence. Appending the eight-bin rotation histograms for each sub region creates the 128 value description vector for each SIFT feature. Feature matching is conducted by calculating the Euclidian distance between the description vectors of features in different images. Description vectors with sufficiently small distances from each other are taken to be potential corresponding features. The probability that a potential correspondence is correct is calculated by taking the ratio between the distance between the closest and second closest neighbouring description vectors. In Lowe's original implementation [22] any correspondences with a distance ratio greater than 80% were rejected.

Correspondences are further refined through Hough transform voting [24], here clusters of corresponding features which agree on a consistent overall pose of the scene are interpreted as having a higher probability of being a true correspondence. A least squares linear regression is performed over these clusters to minimise the error in the affine transform between the 3D points and image points, this allows outlying correspondences to be discarded.

A further useful tool for correspondence analysis is epipolar geometry which requires the exterior orientation of the camera network to have been determined. Determining the exterior parameters of the camera network can be achieved using the SIFT detected points through space resectioning [17]. Figure 2.7 shows how epipolar geometry can be used to both, find matching image features and localise the corresponding 3D point.

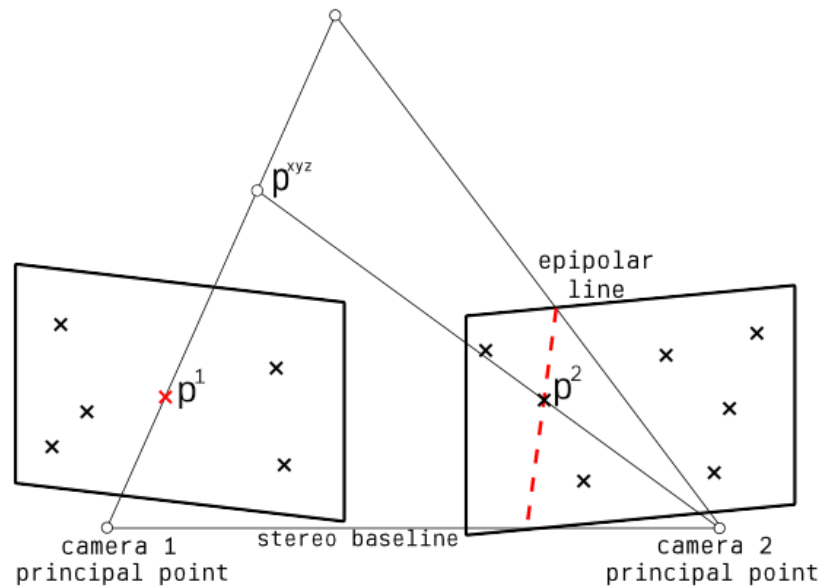


Figure 2.7: Epipolar geometry. The epipolar line (shown in red) is used to match features (shown as crosses) P^1 and P^2 which are then triangulated to localise 3D point P^{xyz} .

In a pair of stereo images there may be many candidate point correspondences which lie on the same epipole, this introduces ambiguity in which

point is the true match. In the case of a multi-view close range photogrammetric measurement where many imaging positions are used this ambiguity is much reduced, and the same correspondence refinement procedures as were described above can also be employed to increase confidence in the detected correspondences.

Correspondences found using the techniques described above can be back-projected into 3D space through the camera model given in Equation 2.9 and triangulated to produce both a sparse reconstruction of the scene and to provide estimates of the extrinsic parameters of the camera network.

2.1.1.4 Bundle adjustment

Bundle adjustment is the process of globally refining the entire reconstructed scene, which includes: the triangulated surface points, the camera network as defined by the cameras' extrinsic parameters and, optionally, the cameras' intrinsic matrices along with any distortion parameters. The refinement of camera parameters along with surface point coordinates is conducted simultaneously in a global minimisation problem which aims to minimise the reprojection error of the scene. The reprojection error is the difference between the pixel coordinates of extracted image features and the pixel coordinates of triangulated 3D object points 'reprojected' from 3D space, through the camera model, back onto the imaging plane.

If the scene contains n object points and k cameras and \mathbf{x}_{ij} is the image coordinate of point i as projected onto image j where camera j is parameterised by \mathbf{c}_j and point i exists at 3D vector \mathbf{p}_i , then the bundle adjustment minimisation problem can be formulated as,

$$\min_{\mathbf{c}_j, \mathbf{p}_i} \sum_{i=1}^n \sum_{j=1}^k \left[\text{vis}_{ij} \cdot d(Q(\mathbf{c}_j, \mathbf{p}_i), \mathbf{x}_{ij})^2 \right], \quad (2.18)$$

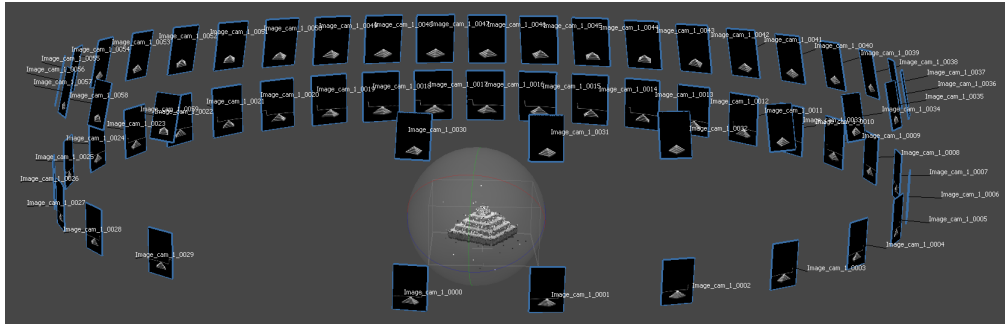
where $\text{vis}_{ij} = 1$ when point i is visible from view j and $\text{vis}_{ij} = 0$ otherwise, $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between two vectors, and $Q(\mathbf{c}_j, \mathbf{p}_i)$ is the reprojected coordinate of point \mathbf{p}_i through camera model \mathbf{c}_j . This minimisation can be achieved using a variety of non-linear least-squares algorithms, of which sparse Levenberg-Marquardt has become the most successful due to its fast convergence which is robust to poor initial scene estimations, and efficient computation due to the sparse interaction matrix between parameters [25].

If robust camera characterisation has been performed prior to measurement, bundle adjustment can be simplified by fixing the camera models \mathbf{c}_j and only adjusting the location of surface points \mathbf{p}_i . Alternately the bundle adjustment optimisation of interior and distortion parameters can be utilised to ensure reconstruction results can be obtained despite poor camera characterisation - this process is referred to as self-characterisation [26].

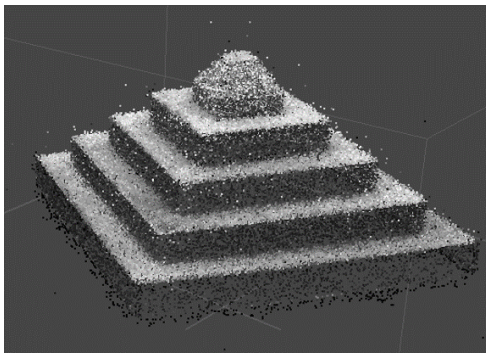
2.1.1.5 Dense reconstruction

The output of bundle adjustment is an optimised, but sparse, scene. For surface measurement applications, where point spacing on the order of tens of microns is required, the sparse scene must be densified. Densification often raises the number of points from the order of thousands to the order of millions or tens-of-millions of points depending on scene complexity, reconstruction quality parameters, and number of cameras in the network. A popular approach for densification is patch-based multi-view stereopsis (PMVS) [27–29]. PMVS takes the refined, sparse scene as input and outputs a dense set of small rectangular patches of points covering the visible surfaces contained in the image set. PMVS segments the surface into patches where each patch contains a list of all images in which it is visible,

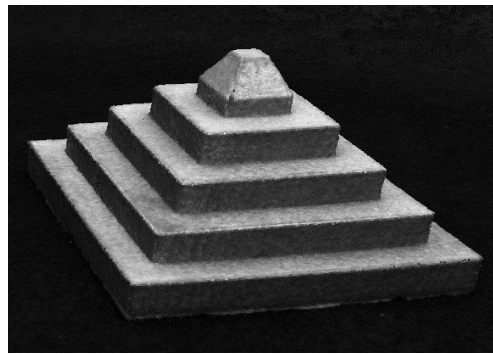
each patch is then projected onto the imaging plane of each image which views it. Images are segmented into cells (2×2 pixels in the original PMVS paper [27]) where each image cell contains a list of which patches are projected within that cell's bounds. PMVS iteratively matches, expands, and filters the patches, starting with the initial sparse points, with the goal reconstructing at least one patch within every cell. Each patch is expanded into its neighbouring cells unless there is a large depth discrepancy (i.e. due to a step height change) or the cell already contains a patch which neighbours the current patch. Figure 2.8 shows a comparison between a refined sparse scene and a densified scene.



(a) Sparse scene, showing imaging locations.



(b) Sparse point cloud.



(c) Dense point cloud.

Figure 2.8: Example reconstruction results.

The example reconstruction in Figure 2.8 is a measurement of a metal AM part, the scan data consists of 120 images taken by a stereo imaging head at 60 equally spaced radial positions around the object, this is typical of 3D form measurement applications. The sparse point cloud shown in

Figure 2.8b consists of 75766 points which was densified to a point cloud containing 4518146 points shown in Figure 2.8c. To give a sense of the time scales and processing power required for reconstructions of this kind, sparse reconstruction and camera alignment took approximately 5 minutes while densification took approximately 25 minutes. Reconstruction was performed using the commercial software Agisoft Metashape [30] on a Windows 10 PC with an Intel(R) Xeon(R) W-2123 CPU @ 3.60GHz CPU, 32.0 GB of RAM, and a Nvidia Quadro P400 GPU.

2.1.2 Digital fringe projection

DFP is an alternative to close-range photogrammetry [31]. In addition to a camera, DFP uses a digital projector to project a sequence of known sinusoidal patterns onto an object, the deformation of the projected pattern as detected by a camera encodes information about the surface profile of the object. Specifically the depth information is encoded in the phase of the received patterns. DFP has some advantages over photogrammetry; "one shot" measurements are made possible which can be very useful for applications such as powder bed monitoring in AM, the reconstruction speed compared to photogrammetry is often much lower while still producing high density point clouds which could have applications for real time monitoring, and DFP can be used on relatively featureless surfaces which could not be measured by photogrammetric means. However, DFP requires at least one projector which can be expensive, projectors also introduce complex non-linearities which must be accounted for. Photogrammetry is naturally multi-view which can lead to higher surface coverage without the need for data fusion. Furthermore, the process of phase-unwrapping (described below) can introduce $2k\pi$ phase ambiguities in any step height changes.

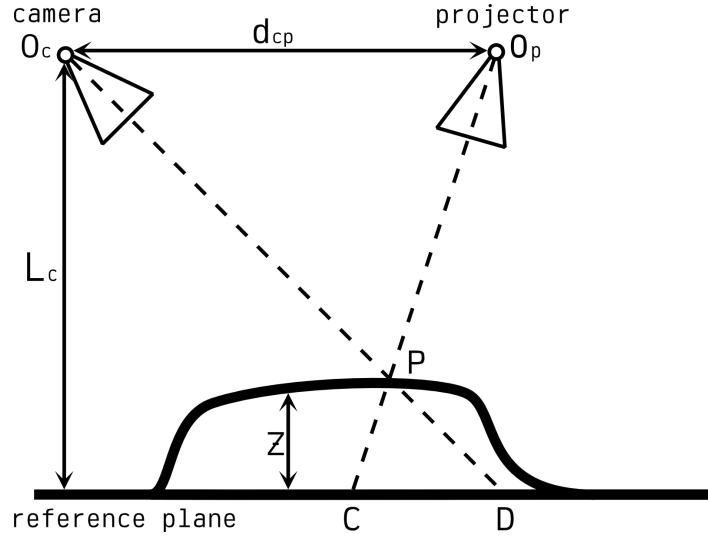


Figure 2.9: Diagram of an example DFP system.

Figure 2.9 shows a typical DFP system. For n projected fringe patterns, the pixel intensity values of the i^{th} image can be represented by,

$$\mathbf{I}_i(u, v) = \mathbf{I}_a(u, v) + \mathbf{I}_b(u, v) \cdot \cos[\phi(u, v) - \delta_i], \quad (2.19)$$

where $\mathbf{I}_a(u, v)$ is the illumination component arising from background light, $\mathbf{I}_b(u, v)$ is the intensity modulation from the projected fringe pattern, δ_i is the projected phase shift given by $(2\pi i/n)$ and $\phi(u, v)$ is the received phase. The phase can be calculated from the detected image as,

$$\phi(x, y) = \tan^{-1} \left(\frac{\sum_i^{n-1} \mathbf{I}_i(u, v) \sin(2\pi i/n)}{\sum_i^{n-1} \mathbf{I}_i(u, v) \cos(2\pi i/n)} \right). \quad (2.20)$$

This phase information is wrapped between $[-\pi, \pi]$ and must be unwrapped to obtain the true phase. The true phase is given by,

$$\psi(u, v) = \phi(u, v) + 2k\pi, \quad (2.21)$$

where k is the fringe number which is an index between zero and the total

number of fringes projected. The phase unwrapping process for a single sinusoid projected unto a flat plane is shown in Figure 2.10 where Figure 2.10b shows the wrapped phase (ϕ) and 2.10c shows the unwrapped phase (ψ).

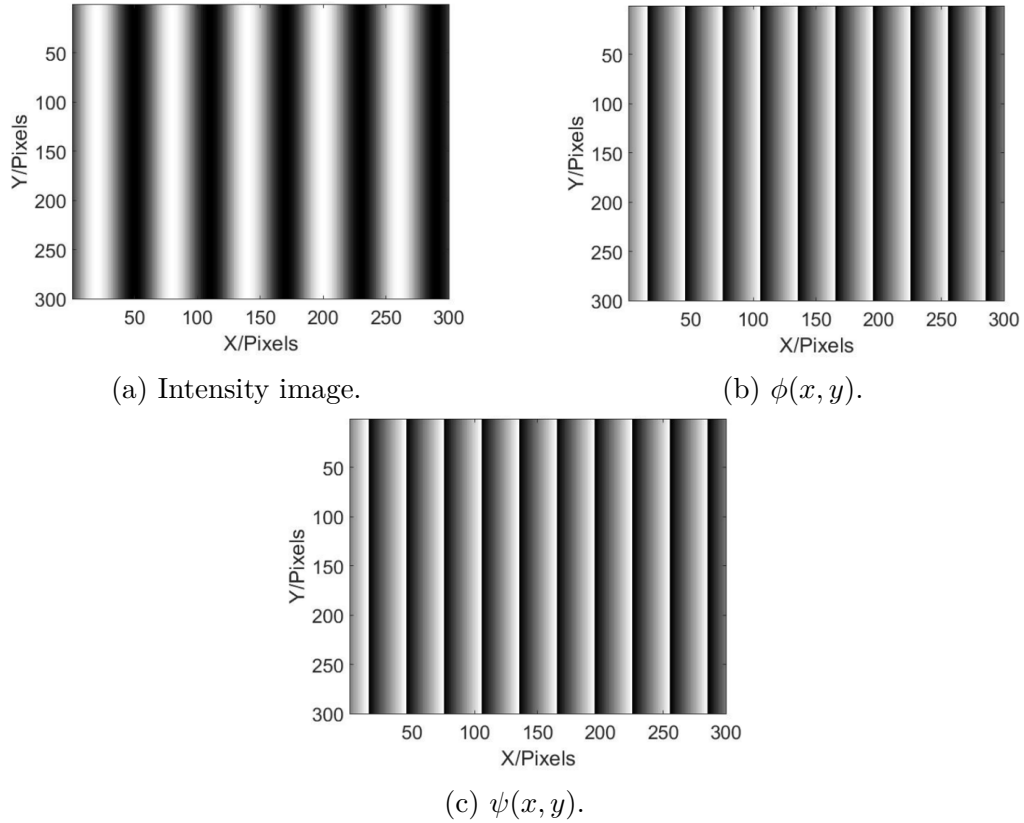


Figure 2.10: 2D phase unwrapping; showing the received image, the wrapped phase, and the unwrapped phase.

Phase unwrapping is a difficult problem and many possible solutions exist which can be roughly separated into two categories; spatial phase unwrapping and temporal phase unwrapping. Spatial phase unwrapping attempts to unwrap the phase from a single phase map whereas temporal unwrapping projects a sequence of patterns used to acquire the absolute fringe order. The various approaches to phase unwrapping are summarised well in Shaheen [32]. Multi-frequency phase unwrapping is a popular temporal phase unwrapping approach wherein a series of phase maps are collected from projected patterns at a range of spatial frequencies. Beginning

with a reference fringe pattern with a wavelength large enough to produce no phase discontinuities the spatial frequency is increased in each subsequent projected pattern. The true phase (ψ) of the n^{th} projected pattern is related to the previous phase by,

$$\psi_n(u, v) = \frac{\lambda_{n-1}}{\lambda_n} \cdot \psi_{n-1}(u, v) \quad (2.22)$$

where λ is the fringe wavelength. Because the reference pattern has a wavelength large enough to cause no ambiguities this implies $\phi_0(u, v) = \psi_0(u, v)$. This allows the fringe number to be calculated for the second phase pattern from,

$$k_1(u, v) = \text{round}\left(\frac{\lambda_0 \cdot (\psi_0 - \phi_1)}{\lambda_1 \cdot 2\pi}\right). \quad (2.23)$$

And thus the phase can be unwrapped using Equation 2.21. This process can be repeated at each spatial frequency until the phase for the shortest spatial wavelength has been unwrapped. The simplest and fastest multi-frequency implementations use only two patterns but for precision applications it is much more common to use $n > 5$ patterns.

Depth can then be calculated from the unwrapped phase through comparison to the phase map of a reference plane (ϕ^r). As can be seen in Figure 2.9, the projected phase on object point P is equal to that which would be projected onto reference plane point C (i.e. $\phi_P = \phi_C^r$). Similarly, the camera detects the phase from object point P in the same pixel in which it detects point D in the reference measurement (i.e. $\phi_D^r \rightarrow \phi_P$) Subtracting the reference phase map from the measured object phase map yields the phase difference at this specific pixel,

$$\Delta\phi_{PC} = \phi_P - \phi_C^r = \phi_C^r - \phi_D^r = \Delta\phi_{CD}^r. \quad (2.24)$$

The triangles $\Delta O_P O_C P$ and ΔCPD are similar, thus the height of point P can be given in terms of the distance \overline{CD} as,

$$Z(u, v) = \frac{\overline{CD} \cdot L_c}{d_{cp} + \overline{CD}}, \quad (2.25)$$

where L_c is the distance from the reference plane to the camera optical center, and d_{cp} is the stereo baseline between the camera and projector optical centres - both values are shown in Figure 2.9. Finally, the distance \overline{CD} can be replaced with the phase difference ϕ_{CD}^r . Assuming d_{cp} is much larger than \overline{CD} ,

$$Z(x, y) = \frac{L_c}{2\pi f_m d_{cp}} \cdot \Delta\phi_{CD}^r \approx c_0 \Delta\phi_{CD}^r, \quad (2.26)$$

where f_m is the spatial frequency of the projected patterns and c_0 is a constant determined through system characterisation. A simple approach to determining c_0 is to simply determine the phase difference of a known step height and to determine the value of c_0 which maps this phase difference to the known height value. For metrology applications, as with photogrammetry, more accurate camera and projector characterisation is required which is outlined in the following section.

2.1.3 Camera characterisation

Any method of coordinate measurement which is based on cameras requires the interior orientation matrix (as was defined in Equation 2.8) and distortion parameters of all cameras in the network to be determined. In the case of DFP, any projectors must also be characterised. A projector is optically identical to a camera, with one sensor and one lens, so can be characterised using identical methods. Errors in the intrinsic and distortion parameters

can lead to poor measurement results in both photogrammetry and DFP so this stage of the measurement pipeline is essential to metrological applications.

Camera characterisation uses the detection of standard targets across a set of images to determine the camera parameter values, Figure 2.11 shows typical characterisation targets used.

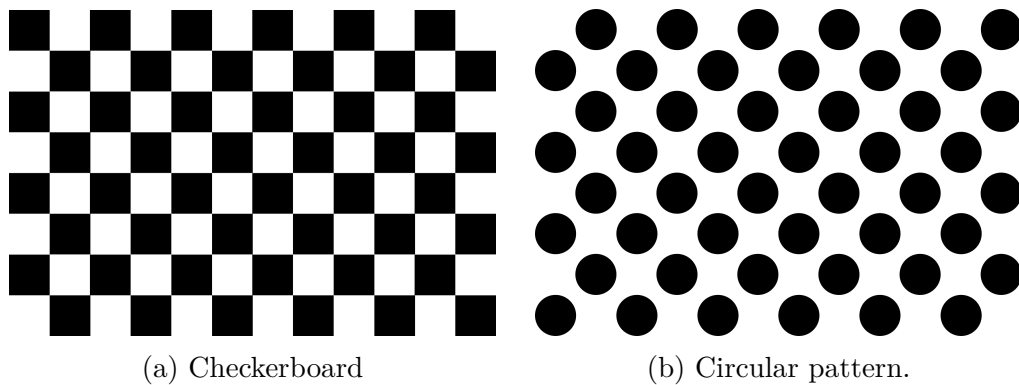


Figure 2.11: Common camera characterisation targets.

In the case of the checkerboard, it is the corners of each square which are localised and used to characterise the camera. In the case of the circular pattern it is the centre of each circle that is localised and used in the characterisation process. The checkerboard design is commonly used in photogrammetry applications where only cameras are used. DFP more commonly uses the circular pattern design as it is easier to localise the circle centres for the characterisation of projectors, this is due to lack of contrast across the dark squares of a checkerboard when projecting fringe patterns.

2.1.3.1 Determination of linear parameters.

The linear camera model given in Equation 2.8 can be rewritten in terms of the projection matrix \mathbf{A} as,

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (2.27)$$

where s is a scale factor which must also be determined. Equation 2.27 can be rearranged to give the collinearity equations,

$$u = \frac{A_{1,1}x + A_{1,2}y + A_{1,3}z + A_{1,4}}{A_{3,1}x + A_{3,2}y + A_{3,3}z + A_{3,4}}, \quad (2.28)$$

$$v = \frac{A_{2,1}x + A_{2,2}y + A_{2,3}z + A_{2,4}}{A_{3,1}x + A_{3,2}y + A_{3,3}z + A_{3,4}}. \quad (2.29)$$

Because the s term cancels out, the world-to-image coordinate transform can be established in terms only of the camera parameters and 3D point coordinates. This relationship can be given in terms of vector matrix A' as,

$$\mathbf{QA}' = 0 \quad (2.30)$$

where,

$$\mathbf{Q} = \begin{bmatrix} x_1 & 0 & \dots & \dots & x_n & 0 \\ y_1 & 0 & \dots & \dots & y_n & 0 \\ z_1 & 0 & \dots & \dots & z_n & 0 \\ 1 & 0 & \dots & \dots & 1 & 0 \\ 0 & x_1 & \dots & \dots & 0 & x_n \\ 0 & y_1 & \dots & \dots & 0 & y_n \\ 0 & z_1 & \dots & \dots & 0 & z_n \\ 1 & 0 & \dots & \dots & 1 & 0 \\ -u_1x_1 & -v_1x_1 & \dots & \dots & -u_nx_n & -v_nx_n \\ -u_1y_1 & -v_1y_1 & \dots & \dots & -u_ny_n & -v_ny_n \\ -u_1z_1 & -v_1z_1 & \dots & \dots & -u_nz_n & -v_nz_n \\ u_1 & v_1 & \dots & \dots & u_n & v_n \end{bmatrix}^T, \quad (2.31)$$

$$\mathbf{A}' = \begin{bmatrix} A_{1,1} \\ A_{1,2} \\ A_{1,3} \\ A_{1,4} \\ A_{2,1} \\ A_{2,2} \\ A_{2,3} \\ A_{2,4} \\ A_{3,1} \\ A_{3,2} \\ A_{3,3} \\ A_{3,4} \end{bmatrix}, \quad (2.32)$$

and n is the number of points. A minimum number of six corresponding features are required to compute all the unknowns in Equation 2.32. However, as can be seen in Figure 2.11 characterisation targets contain many

more target features than six, this leads to an over-determined system of equations which can be solved using a least-squares regression. A typical characterisation contains around 150 target features imaged from no less than 25 locations giving $150 \times 25 \times 2 = 7500$ degrees of freedom in the regression. Solving for the eigenvector of $\mathbf{Q}^T \mathbf{Q}$ provides an efficient method of solving this system of equations for \mathbf{A}' . To provide good quality characterisation it is important to perform the characterisation over a series of images containing the target in range of unique poses covering the entire measurement volume, otherwise the camera model will overfit to minimise reprojection error over a subset of the measurement volume.

2.1.3.2 Determination of non-linear parameters.

If a more complicated camera model is used which includes non-linear distortion modelling the collinearity equations (given in Equation 2.28 and Equation 2.29) must be adapted to include the distortion parameters. The non-linear characterisation problem can be formulated as a minimisation problem as,

$$\min_{\mathbf{x}} \sum_{i=1}^n |\mathbf{x}_i - f(\mathbf{c}, \mathbf{p}_i)|^2. \quad (2.33)$$

where $f(\mathbf{c}, \mathbf{p}_i)$ is the reprojection of 3D point \mathbf{p}_i through camera model \mathbf{c} and \mathbf{x}_i is the corresponding image feature. The minimisation problem formulated in Equation 2.33 is often computed using the non-linear least-squares Levenberg-Marquadt algorithm [33].

Alternatively, as was discussed in Section 2.1.1.4, camera parameters can be optimised as part of bundle adjustment, i.e. simultaneously with the reconstruction of the scene, this is called camera self-characterisation. It can be seen that the bundle adjustment minimisation problem, as given in Equation 2.18 is only a small adjustment to the minimisation formulation

given in Equation 2.33. Using bundle adjustment to determine camera parameters requires no *a-priori* knowledge about the geometry of the scene, this makes the approach hard to validate and is therefore rarely used for metrological applications.

2.2 Machine learning

Machine learning (ML) approaches are well suited to solving highly complex non-linear problems where the relationship between the input and output is poorly understood. There are many such problems in optical metrology implying ML has the potential to be a valuable tool in the field. Particularly the problems of camera characterisation, view planning, phase unwrapping and stereo matching involve complex non-linear, high dimensional problems which ML is well suited to tackle. Further, the analysis of the resultant point clouds is a complex process which has had a wealth of scientific literature published which suggest ML solutions. Although the work in this thesis is only concerned with autonomously generating an optimal point cloud, and not the analysis of this cloud, Section 2.3 reviews the state of the art in all these areas including point cloud analysis for completeness.

In general, ML models can be thought of as a system which is not specifically programmed to solve a problem; it is instead told what problem to solve, given a set of training data, and then learns how best to solve the given problem on its own. More formally [34, 35], for an input space \mathcal{X} and an output space \mathcal{Y} the goal of ML is to learn a mapping function $h : \mathcal{X} \rightarrow \mathcal{Y}$, referred to as a hypothesis, from a dataset of labelled examples. The training dataset has entries $[\mathbf{x}, \mathbf{y}]$ where $\mathbf{x} \in \mathcal{X}$ are referred to as features and $f(\mathbf{x}) = \mathbf{y} \in \mathcal{Y}$ are referred to as labels. Training an

ML model is the process of refining the hypothesis $h(\mathbf{x}) = \hat{y}$, such that \hat{y} best approximates \mathbf{y} and, thus, $h(\mathbf{x})$ approximates the unknown function $f(\mathbf{x})$. Once a hypothesis has been developed, this hypothesis can be used to make predictions on unseen data ($\mathbf{x} \in \mathcal{X}$) that were not present in the initial dataset ($h(\mathbf{x}') = \hat{\mathbf{y}}'$). The accuracy of predictions on unseen data is referred to as the ability of the hypothesis to generalise, and the error between the set of all predictions and the corresponding ground-truth values ($h(\mathbf{x})f(\mathbf{x})\forall\mathbf{x}$) is called the generalisation error. For large or continuous input spaces it is practically impossible to compute the full generalisation error as ML models can only be evaluated over a set of finite input samples, instead the generalisation error is estimated over a finite test dataset with elements $[\mathbf{x}',\mathbf{y}']$. It is the aim of training ML models to minimise the generalisation error. This process is called supervised learning and contrasts with unsupervised learning where the initial dataset contains only features and no labels [36].

There are two main tasks completed by ML algorithms: classification and regression [37]. Classification involves assigning each input datum to a discrete class, whereas regression produces a continuous value (or set of continuous values) for a given input. An example of classification is the detection of the presence of some object part within an image; this could be a binary classification (yes there is an object, no there is not an object) or a multi-class classification (which object, of a set of possible objects, is present in the image). An example of a regression task is the pose estimation of an object within an image, where the ML model is regressing continuous coordinate values which together define a translation vector and rotation matrix relative to the camera coordinates.

2.2.1 Artificial neural networks

An artificial neural network (ANN) attempts to approximate the unknown function $f(\mathbf{x})$ by through a highly dense, interconnected system of non-linear functions. Each individual branch (or neuron) in the system is simple, and through the large inter-connection of these simple components complex functions can be approximated. A basic ANN will look something like Figure 2.12, albeit with a much larger number of nodes per layer.

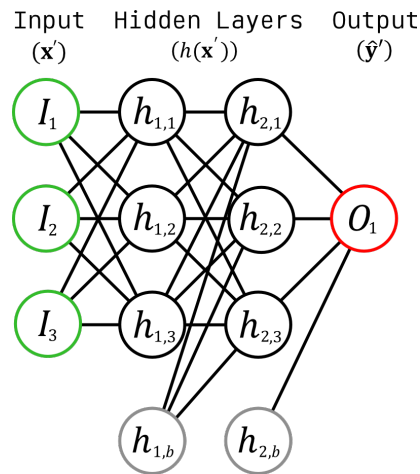


Figure 2.12: Basic ANN architecture. Input nodes shown in green, hidden nodes shown in black (with bias nodes shown in grey) and output node shown in red.

The input nodes take the set of features, either from the training dataset (\mathbf{x}) or some unseen data (\mathbf{x}). In an ANN, the features take the form of a tensor (n -dimensional vector) of numerical values. These data are passed to the hidden layers via a fully connected set of weighted connections (excluding the bias node in each hidden layer, shown in Figure 2.12 as $h_{n,b}$). Each node in the hidden layers passes the weighted sum of these inputs through a non-linear activation function which determines the activation of that node. Each node in the next hidden layer takes the activations of the previous layer as input to the same process, producing complexity through

linear combinations of non-linear activations. This is repeated through all the hidden layers in the given model until the final output nodes' activations are determined by the weighted sum of the activations in the final hidden layer. Figure 2.13 shows a detailed view of one of the neurons from the network shown in Figure 2.12.

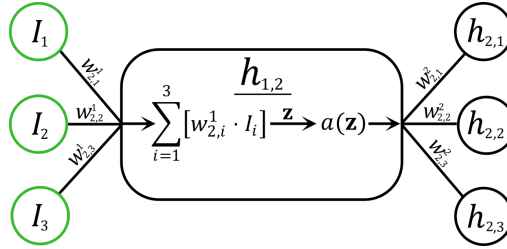


Figure 2.13: Detail of an ANN neuron (specifically $h_{1,2}$ from Figure 2.12). Here $w_{i,j}^l$ is the weight of a connection, l , j and k are indices representing the layer, input node and output node respectively, and $a(\mathbf{z})$ is a non-linear activation function which must be differentiable at all \mathbf{z} .

In order for the model to approximate complex functions, the activation function $a(\mathbf{z})$ must be non-linear. As a composition of linear functions can be described by a single linear function (i.e. $(f_1 \circ f_2)(\mathbf{z}) = g(\mathbf{z})$ where g is a linear function for all f_1, f_2 which are linear), if $a(\mathbf{z})$ were linear this would allow the hidden layers in an ANN to be collapsed into a single hidden layer with a linear activation function, removing the ability of the network to capture complex functions. There are many possible candidate functions for $a(\mathbf{z})$; the most common are the rectified linear unit (ReLU) and the sigmoid (or logistic) function [38]. The ReLU is described by,

$$a(\mathbf{z}) = \begin{cases} \mathbf{z}, & \mathbf{z} > 0 \\ 0, & \mathbf{z} \leq 0 \end{cases} \quad (2.34)$$

In practice, the ReLU function is often approximated by other functions to allow it to be differentiable at zero, otherwise a value of zero or one can

be chosen arbitrarily. A common choice is the SoftPlus function given by,

$$a(\mathbf{z}) = \ln(1 + e^{\mathbf{z}}). \quad (2.35)$$

For negative numbers the SoftPlus function approximates e^z which converges to zero, and for positive numbers approximates $z + e^{-z}$ which converges to \mathbf{z} . The SoftPlus is convenient also as its derivative is the logistic function, which its self is a popular choice of activation function, thus making the SoftPlus twice differentiable at all points. The logistic function is defined by,

$$a(\mathbf{z}) = \frac{e^{\mathbf{z}}}{1 + e^{\mathbf{z}}} = \frac{1}{1 + e^{-\mathbf{z}}} \quad (2.36)$$

By adjustment of the weights that connect together the nodes of an ANN, learning occurs. The adjustment of connection weights is done through a process of gradient descent and back propagation [39]. Once the ANN has been designed (number of hidden layers, nodes per layer, activation function all selected), then the weights of each connection are initiated randomly. The first feature list from the training dataset is input to the network, fed forward through the layers, and then an output is generated. This output is then compared to the given label for that feature set through a loss function. The most common loss function for regression is the mean squared error (MSE), while for classification, binary cross-entropy is commonly used and there are many other options [40] such as mean absolute error (MAE), mean absolute percentage error (MAPE) and Huber loss [41] for regression and categorical cross-entropy for classification. In this thesis, categorical cross-entropy is used for categorisation tasks and is given by,

$$Loss_{cat} = - \sum_{i=1}^N \mathbf{y}_i \cdot \log(\hat{\mathbf{y}}), \quad (2.37)$$

where N is the number of samples the function is evaluated over (ie. the

batch size). The LogCosh loss function is used in this thesis for regression tasks and is given by,

$$Loss_{reg} = \sum_{i=1}^N \log(\cosh(|\hat{\mathbf{y}}_i - \mathbf{y}_i|)). \quad (2.38)$$

The LogCosh loss function is selected as it behaves linearly at high prediction errors ($|\hat{\mathbf{y}}_i - \mathbf{y}_i|$) and quadratically at low prediction errors. These properties are desirable as outliers do not dominate the value of the loss function and the gradient of the function approaches zero as the prediction error shrinks leading to smaller steps taken by the optimiser. The gradient of the selected loss function is calculated with respect to the node weights, which can then be adjusted in the direction which is expected to reduce the loss. This process is repeated until convergence is achieved. The weights are updated according to

$$\Delta w_{j,k}^l = \eta \cdot \frac{\partial E}{\partial a_{l,j}} \cdot \frac{\partial a_{l,j}}{\partial w_{j,k}^l} \quad (2.39)$$

where w is the weight of a connection, l, j, k are indices representing the layer, input node and output node, respectively, E is the loss function, a is the activation function and η is a dimensionless coefficient called the learning rate, typically set around $\eta = 0.001$. The learning rate is used to control the step size taken during back propagation; many optimisation schemes (including the popular Adam optimiser [42]) use a variable learning rate which shrinks as training continues—forcing the optimisation to take smaller steps as convergence to the true value is approached. The Adam optimiser is used exclusively for training models in this thesis.

2.2.2 Convolutional neural networks

Convolutional neural networks (CNNs) differs from an ANN by making use of convolutional layers [43] and have been shown to be particularly effective on image processing tasks, though can be generalised to act upon many data structures. These convolutional layers perform a process of sliding window kernel convolution, a process whereby a grid of values (or kernel) is convolved over the input to produce a set of output feature maps. This convolution is applied over all dimensions (channels) in the input. Depending on the size of the kernel and the step size taken across the pixel grid during convolution (called the stride), kernel convolution can also be used for spatial down-sampling of the input. It is common to use multiple layers of kernel convolution to produce more specific feature maps. This process is illustrated in Figure 2.14.

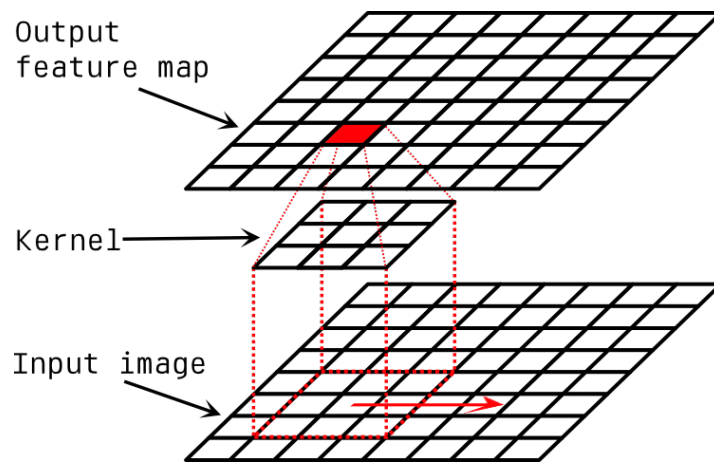


Figure 2.14: Kernel convolution on an n -dimensional tensor.

As with the weights in the fully connected layers, the values within the kernel are learned through back propagation. A typical CNN will have a series of convolutional layers; each layer has a number of different kernel filters operating in parallel. The outputs of these filters are commonly then fed into a fully connected section (as in an ANN) before producing the final prediction, though a fully convolutional neural network (FCNN)

foregoes the fully connected layers and consists entirely of convolutional layers [44]. Convolutional layers are often packaged together with batch-normalisation layers, which normalise the incoming data from the previous hidden layer, called pooling layers. Max-pooling takes a sliding window of inputs and outputs the maximum value, average-pooling does the same but outputs the mean value (pooling is often used for spatial down-sampling) and activation layers apply the previously discussed activation functions, such as ReLU, to the feature map. The work in this thesis makes heavy use of CNNs due to their efficient operation in image processing tasks, kernel convolution naturally lends itself to operation on two dimensional image data. The advanced architectures used in this thesis and in the state of the art are summarised in the following section.

2.2.2.1 Advanced CNN architectures

Using the blocks described in the previous section, numerous architectures have been developed. Briefly summarised below are some architectures of particular relevance to the work in this thesis, although many more have been developed.

2.2.2.1.1 ResNet

The residual neural network (ResNet) was an early but impactful variation on a standard CNN [45]. The main contribution of the residual neural network (ResNet) is the skip connection, which is shown in red in Figure 2.15. The purpose of the skip connection is twofold, first to allow an alternate path for gradient flow during back propagation, mitigating against the effect of vanishing gradients in deep networks caused by repeated mul-

tiplication. The second benefit of a skip connection is allowing more direct influence from early feature maps during inference.

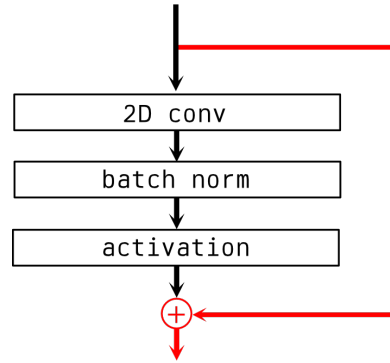


Figure 2.15: Residual block from a residual neural network (ResNet) based architecture - skip connection highlighted in red.

In Chapter 8, a model for detecting the pose of a part within a measurement system is built which is based on the ResNet architecture. The ResNet was selected as a base for the pose estimation model due to improved preservation of spatial information provided by the skip connections.

2.2.2.1.2 EfficientNets

The original EfficientNet publication (Tan and Le [46]) claims “state-of-the-art performance while being $8.4\times$ smaller and $6.1\times$ faster during inference than the best existing [CNN]”. The building block of an EfficientNet is the MBConv layer which is based on the MobileNet family of models. MBConv blocks can be summarised as inverted residual linear bottleneck blocks with depthwise separable convolution and squeeze-excite blocks, Figure 2.16 shows the layers in an MBConv block.

First, the number of channels in the input is increased through a pointwise convolution. A depthwise-separable convolution is applied which con-

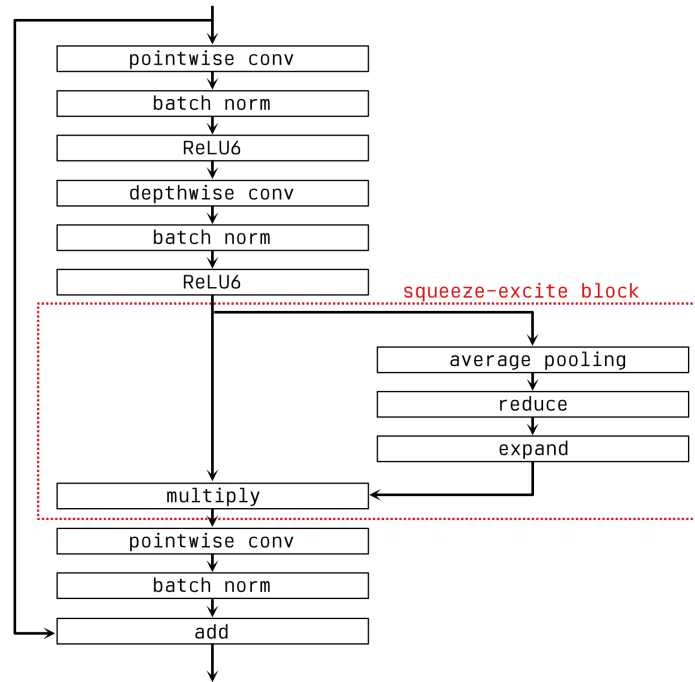


Figure 2.16: EfficientNet MBConv block.

sists of a combination of a depthwise convolution followed by a pointwise convolution. Depthwise separable convolution requires far fewer parameters, and thus fewer computations during inference, than a simple 2D convolutional layer. A squeeze-excite block is inserted in the middle of the depthwise separable convolution which essentially learns a weighting to apply to each channel of the feature map before the pointwise convolution is applied. The squeeze-excite block was first presented by Hu et al. [47] and shown to be beneficial. Finally, the output of the convolution is combined with the output of the previous block in a skip connection as was defined in the previous section. Chapter 4 makes use of an EfficientNet modified for robust camera characterisation, the EfficientNet was selected in this case due to its high performance for a relatively low number of parameters as discussed above.

2.2.2.1.3 Generative adversarial networks

A generative adversarial network (GAN) is a system of two sub-networks trained in a zero-sum-game (first proposed in 2014 by Goodfellow et al. [48]). Given some set of input data, the task is to generate some new data that cannot be distinguished from the original dataset, while capturing the variation present within the original data. To achieve this task, a sub-network, called the generator ($G(\mathbf{z}) \rightarrow \mathbf{i}$) takes an input vector \mathbf{z} randomly sampled from some high dimensional space. This seed value is passed forward through the generator model to produce data \mathbf{i} of the same type and shape as the input (for example, an image). Initially, the generator output is pseudo-random over the input. The second sub-network, called the discriminator ($D(\mathbf{i} \rightarrow p)$), uses data \mathbf{i} sampled from $G(\mathbf{z})$, or taken from the initial dataset, and produces a prediction \mathbf{p} as to whether \mathbf{i} is 'real' (from the dataset) or 'fake' (from the generator). A generic convolutional GAN architecture is shown in Figure 2.17.

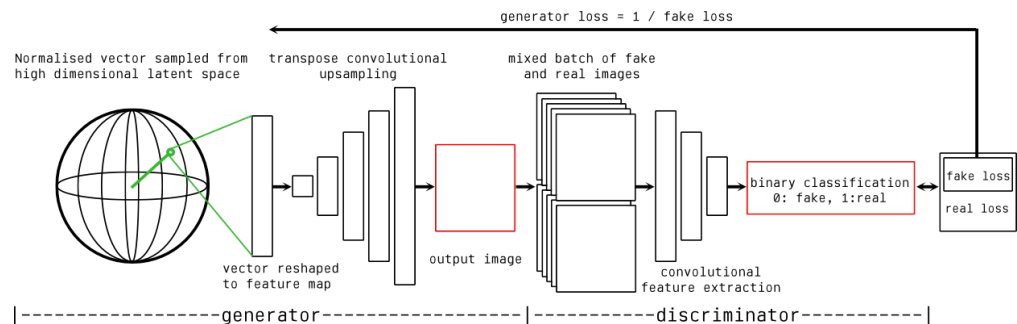


Figure 2.17: A generic generative adversarial network (GAN) showing the generator model, the discriminator model and the zero-sum optimisation through which they are trained. A variety of loss functions are available for calculating the real and fake losses but it is common to use the binary cross-entropy.

The models are trained in a zero-sum-game such that the generator loss function is low when the discriminator loss function is high, i.e., when the generator successfully tricks the discriminator into believing some generated data is real. Once trained, the generator can be deployed to produce

large quantities of new data representative, but distinct, from the data captured in the training set. The output from the generator varies smoothly over the input space. In the original publication [48], both G and D were differentiable functions represented by multi-layer perceptrons; however, it is now more common to use convolutional layers particularly for image processing. Popular convolutional generative adversarial network (GAN) architectures include the progressively growing generative adversarial network (GAN) [49], styleGAN [50], and cycleGAN [51]. Chapter 7 utilises a modified GAN which is trained to produce large datasets of surface texture data which can be used for many tasks useful to metrology, such as for training further ML models such as in Chapter 8.

2.2.2.1.4 Transformer networks

Transformer networks are the current state-of-the-art in processing sequential data such as for natural language processing, control and computer vision tasks [52]. At a high level, the inputs to the model are transformed into a string of tokens (a process called tokenisation), the model is trained to produce the next token in the sequence. A recent example of a transformer model is Gato which is designed by DeepMind to be a generalist agent [53], meaning the model can perform many different tasks in different domains without being retrained. Gato has been seen as a step towards artificial general intelligence, it can caption images, respond to text prompts, play a slough of Atari games, control a robotic arm and more, all with the same model weights. From the context of the input tokens given, the model infers what kind of output tokens to produce. An optional fixed prompt can be provided describing the scene, a sequence of observation tokens are then parsed by the model and a sequence of action tokens produced. The

model has access to all previous input and output tokens in a rolling buffer of 1024 tokens, multi-view attention [54] is used by the model to decide what previous information is relevant to the current inference. Figure 2.18 shows how this model can be used for closed loop control of a robot arm.

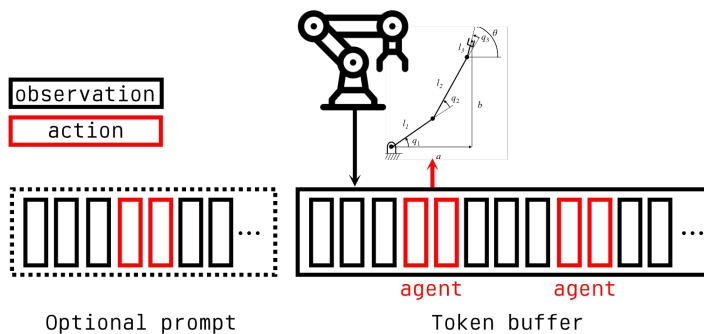


Figure 2.18: Gato for control. A sequence of tokenised observations, separators and previous actions are consumed to predict the next action token in an autoregressive manner. The action is applied, new observations taken, and the process repeats.

Transformer models were not used in this work as they are generally extremely large and require huge computational resources to train, they are included here as the current best performing models in a range of tasks for the sake of completeness.

2.2.3 Support vector machines

While neural networks are popular, there are competing and complimentary approaches to machine learning (ML) that also see widespread application. A SVM is perhaps the most common alternative approach. Simply, SVMs map training data to a higher dimensional space where the categories are separated by a clear linear gap. Predictions can then be made on new data by mapping the unseen examples to the same space and determining which side of the gap they are mapped to. Neural networks are parametric approaches where, once the model is trained, the training dataset may

be discarded, and predictions are made solely on the basis of the learned weights; in contrast, SVMs are memory-based. In this context, memory-based implies that the predictions are made on the basis of the data within the training set. SVMs are a type of kernel method which takes a set of data in an input space that are not linearly separable and maps them into a higher-dimensional space. These kernels are different to the kernels used in sliding window convolution and are simply mapping functions of the input space into a new higher-dimensional space given by,

$$k(x, x') = \phi(x)^T \cdot \phi(x') \quad (2.40)$$

where k is the kernel and ϕ is a feature map. SVMs employ the ‘kernel trick’ [55] which allows the kernel to be computed as a simple function in the input space without explicit knowledge of ϕ . As an example, $k(x, x') = x \cdot x + |x|^2 \cdot |x'|^2$ is a function that is equivalent to applying the feature map $\phi = (x, y, x^2 + y^2)$, this can be easily verified by substitution into Equation 2.40. While this is only a three-dimensional example it is clear that the transformation can be completed without ever calculating an explicit representation of ϕ . This allows efficient computation of transforms into extremely high, sometimes infinite, dimensional spaces. Gaussian kernels are one of the most common kernel types and are an example of kernels that map the input space to an infinitely dimensional space (this can be seen through Taylor expansion of the Gaussian function, which leads to an infinite sum of inner products). When the data have been mapped to a space in which they are separable, a max-margin criterion is used to calculate the optimum hyper-plane to define the class boundary [56]. Figure 2.19 shows two examples of how the same data can be separated.

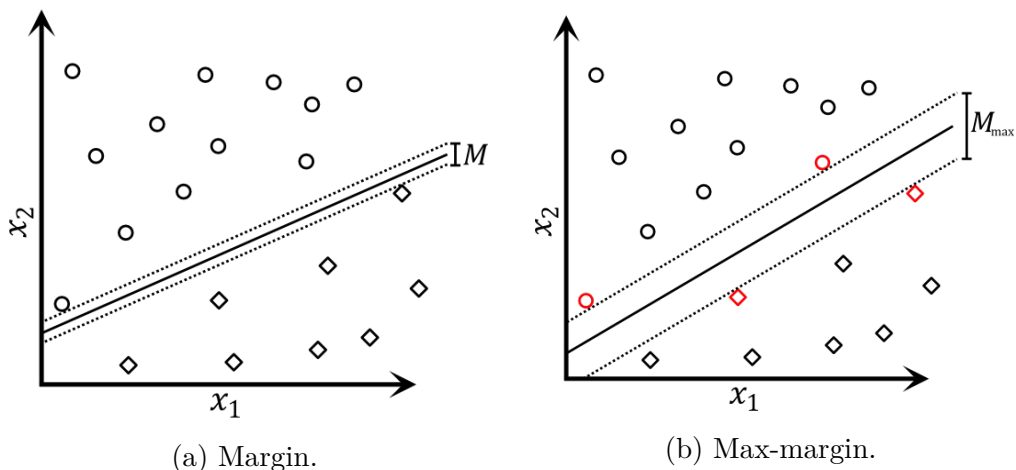


Figure 2.19: The same data separated by different hyper-planes (shown as a solid line). Support vectors highlighted in red in (b).

While Figure 2.19a effectively separates the data and Figure 2.19b uses the max-margin criteria to ensure the closest data point to the hyper-plane from either class is a maximum distance (M_{max}). The max-margin criteria decrease the probability of misclassification when the SVM is applied to unseen data. The highlighted points (shown in Figure 2.19b circled in green) are designated support vectors and are used to fully define the hyper-plane, therefore, when the SVM is deployed all other training data can be discarded and the support vectors alone are used to determine which class the new data belong to. The SVM can also be reformulated to perform regression, by instead finding and selecting support vectors to minimally describe the best fitting hyperplane - this is called support vector regression (SVR). SVMs are not used in this thesis as CNNs are better suited to the image processing based tasks relevant to this project. However, they have been used in many of the works presented in Section 2.3 so are presented here for clarity.

2.2.4 Genetic algorithms

A distinct category of algorithms to the ML models discussed thus far are so called genetic algorithms (GAs) [57]. GAs are inspired by the mechanism of natural selection, where the best candidates from a population of possible solutions are selected for further crossover and mutation to obtain new successors [58, 59]. The process of generating new populations from descendants is repeated until the new population of successors converge. In a GA, some model (which may be represented by some ML model such as an ANN, or by any other parameterised model) is designed to represent the possible solution space to a given problem. A population of models is then generated whose parameters are initialised randomly across some distribution covering the possible parameter space. The initialised parameters are stored in a minimal representation which can be considered analogous to biological genes, often a binary string. Performance of each model is evaluated on a dataset with respect to some objective function, it is the aim of the genetic algorithm to minimise the given objective function. The key operations in a GA as mentioned above are selection, crossover and mutation. Selection is the process of selecting the best performing models in the current population as evaluated by the given objective function, analogous to natural selection. Crossover is analogous to biological mating, where two parent models combine their parameters (genes) to produce an offspring model with parameters set by random contribution from each parent model. Mutation, then, is the application of random perturbations to the offspring parameters as set by the mutation rate. GA mutation represents the random mutations observed in biological genetics and is included to assist in avoiding convergence on local minima and to keep child populations diverse. Figure 2.20 shows the process of two parent models producing a child for a simple model consisting of two parameters each

represented by four-bit binary strings.

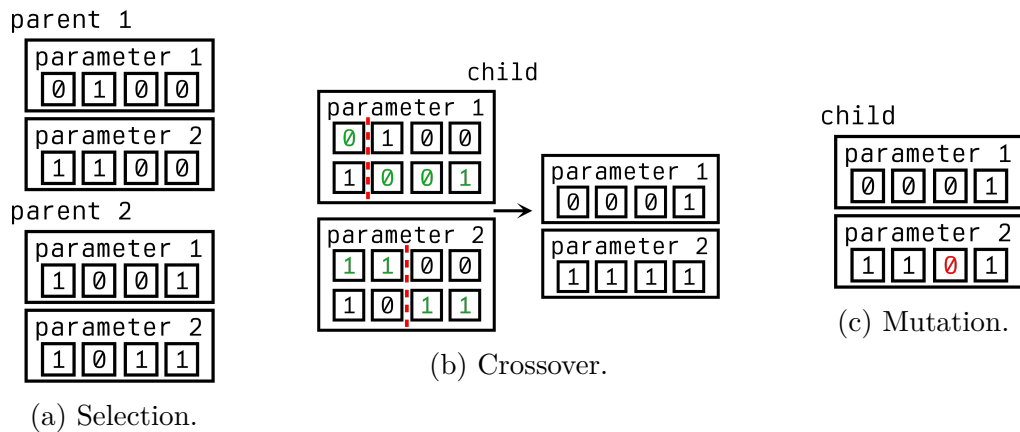


Figure 2.20: The process of two selected parent models producing a child for a simple model consisting of two parameters each represented by four binary "genes". The red dotted line in (b) represents the random crossover point determining how each parent contributes to the resultant offspring. The red value in (c) represents a mutated gene, the mutation rate is normally set such that mutation occurs at a relatively low rate.

Mutation rate, crossover point, population size, conversion criterion and objective function are all hyper-parameters of the GA which should be tuned for the given application.

Since their introduction, GAs and their variants have been used in many areas and shown to be effective for non-linear, complex global optimisation problems (for examples, see the reviews [60,61]). GAs are especially useful in poorly understood scenarios where there is no direct relationship between the input arguments and output target values, and for problems with large search spaces. Due to the complex search space when dealing with multiple cameras, GAs are promising candidates for camera position optimisation [62] and as such are exploited for this purpose in Chapter 5.

2.3 State of the art in machine learning for optical coordinate metrology

With the emergence of high-powered central processing units, the immense parallel computing power of the compute unified device architecture (CUDA) accelerated graphics cards and more access to fully labelled training datasets through the advent of big data [63–65], many of the approaches developed through research can now be implemented commercially. Particular effort is being channelled into the areas of computer vision and image analysis [66, 67]. While machine vision tasks have different requirements and often much less strict constraints when compared to measurement applications, there are many similarities to traditional form and coordinate metrology approaches— stereo vision analysis and scene segmentation in particular are active areas of research due to the advent of self-driving vehicles [64]. Machine learning techniques have already found success in different areas of metrology [68–70], particularly for the measurement of very small parts such as semiconductors [71, 72].

What follows is a state of the art review into currently developed applications of the ML techniques discussed previously to optical coordinate metrology.

2.3.1 Machine learning for stereo matching

A key stage in photogrammetric reconstruction, as discussed in Section 2.1.1.3, is feature detection and feature matching between pairs of images, referred to here as ‘stereo matching’. While the approaches discussed in Section 2.1.1.3 can be highly effective, there has been recent effort to use ML to augment the stereo matching process [73–82] or to replace the tra-

ditional approaches entirely [1, 83–87]. A common approach is to use ML to inform the computation of the stereo matching cost between detected features or local image patches. The multi-view stereo matching problem can be reformulated as a multi-class classification problem, where each class represents all possible views of a given feature [88]; this is shown to reduce the rate of matching errors. Competing approaches instead formulate the problem as a binary classification problem, where pairs of features are fed as inputs and the outputs represent the predicted probability that two features are matched. Early research in this area [81] takes as input two image patches which have been suggested as stereo matches by traditional methods. The two image patches are fed into a single CNN which has two output nodes predicting the probability that the suggested match is a correct match. An example of this type of CNN is shown in Figure 2.21.

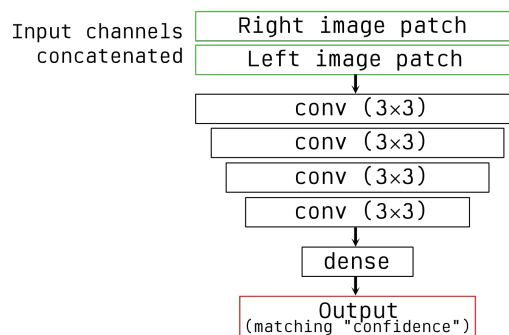


Figure 2.21: CNN to predict the ‘confidence’ that two stereo image patches are correctly matched. The convolutional layers (conv) consist of kernel convolution with the shown kernel size, batch normalisation and ReLU activation.

The CNN prediction is then directly used to adjust the semi-global stereo cost during reconstruction; this approach has been shown to appreciably improve depth estimation. More recently, it has become common in stereo vision applications to utilise twin networks [79, 82, 89] that feed the paired input images into two separate copies of the same architecture, which are then compared to produce an output—this is illustrated in Fig-

ure 2.22.

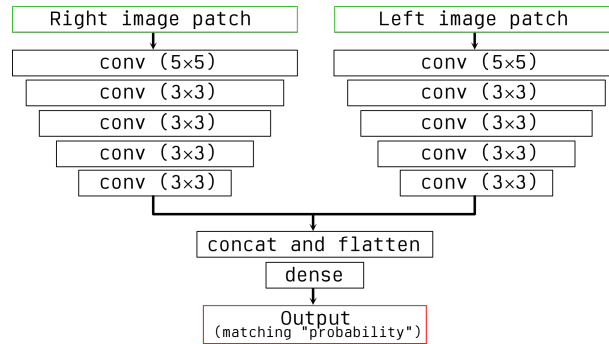


Figure 2.22: Example of a twin CNN for learned stereo matching. In contrast to Figure 2.21, here the stereo pair are input to parallel copies of the same network with shared weights.

A twin network approach was implemented by Feng *et al.* [79], who used the predicted matching probability alone as the stereo matching cost. They showed that the CNN matching approach could outperform traditional approaches in some settings but did not produce accurate results for low-level texture and occluded surfaces. Therefore, they augment the purely ML approach with traditional algorithms, such as semi-global matching, interpolation, sub-pixel enhancement, median filtering and bilateral filtering.

2.3.1.1 Learned stereo machines

Rather than augmenting traditional methods, some ML approaches seek to replace traditional algorithms entirely in a so called learned stereo machine (LSM). Replacing the entire data pipeline with a single ML model is referred to as using end-to-end learning. Learning stereography in this way is not a new idea — an early work [83] described how a neural network could be trained to discover depth from a pair of random dot stereograms of a curved surface—at the time this approach was limited to very simple images and surfaces, but with modern computing power can be expanded

to more general applications. Work more recently [86] has begun to extend LSMs to create high-resolution depth maps from rectified image pairs, generating sub-pixel accuracy without the need for post-processing. Kar *et al.* [1] extend this idea from stereo pairs to multiview systems. They propose a multi-view LSM which uses a recurrent neural network (RNN) based approach. A RNN is a neural network which operates on temporal data, preserving some memory of past activations (see [90]). In this case, the RNN is trained to give two outputs, a 3D voxel occupancy grid and per-view depth maps created through reprojection of the voxel data. A set of images of an object are passed to a feature detection CNN, these features are then un-projected into individual 3D spaces, an RNN is used to match the features and fuse the individual spaces into a single 3D world-space and a 3D CNN is then applied to this world-space to produce the final 3D prediction. This network architecture is shown in Figure 2.24.

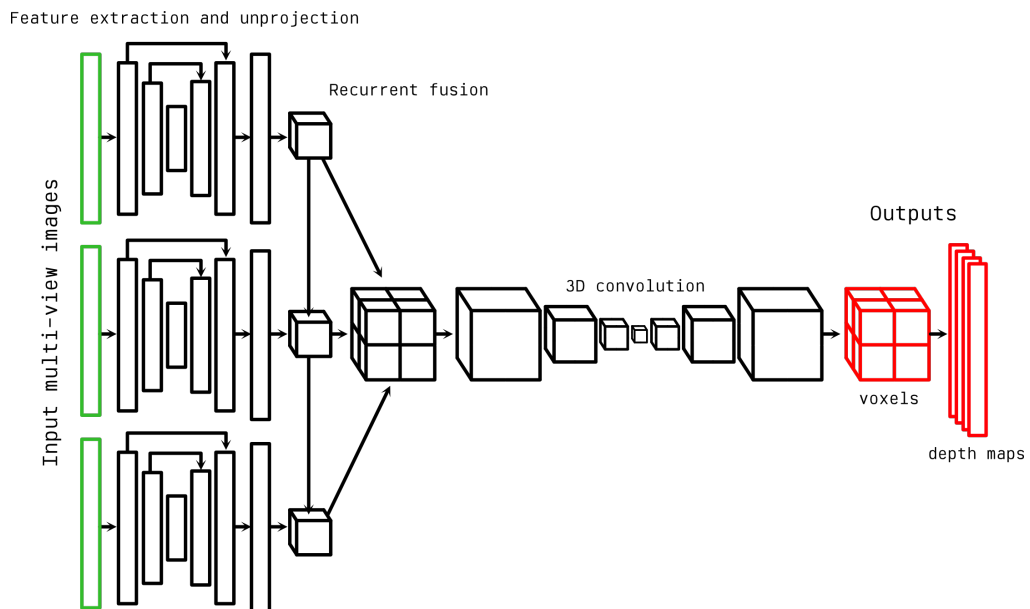


Figure 2.23: Overview of a LSM, based on Kar *et al.* [1]

Using the approach shown in Figure 2.24, reconstructions can be produced to a higher quality with fewer images than compared to traditional

methods, however, the current implementation has a coarse voxel grid that makes it more suited to machine vision tasks than surface measurement. A more recent approach [84] uses a structured SVM to produce disparity maps from a stereo image pair. They show that an SVM based approach may produce significantly better performance than other learning-based methods when applied to the Middlebury-2005 dataset of stereo images [91].

2.3.2 Machine learning for phase unwrapping

ML can be used to address the problem of phase unwrapping in fringe projection optical coordinate measuring systems. Early work in this area used simple ANNs to perform phase unwrapping [92], where they showed the potential for ML to recover fringe order at high speed with little knowledge of the camera characterisation or the details of the measurement device. However, this early work was prone to errors due to noise and variations in the surface characteristics between the training data and the measurement data. Recent work expands this ANN approach [93] by encoding different fringe patterns in the red, green and blue channels of a colour projector, which allows the camera to detect three individually phase-shifted patterns in a single-shot measurement. Corresponding pixels from the three wrapped phase maps are fed into the ANN, shown in Figure 2.24, which can produce phase-unwrapped depth maps at 25.6 times a second, with a relative accuracy of 0.012%.

Rather than feed individual pixels through an ANN, there are many studies which use CNNs to address phase unwrapping [65, 75, 94–98]. Additionally, as coherence scanning interferometry (CSI) [99, 100]) measure-

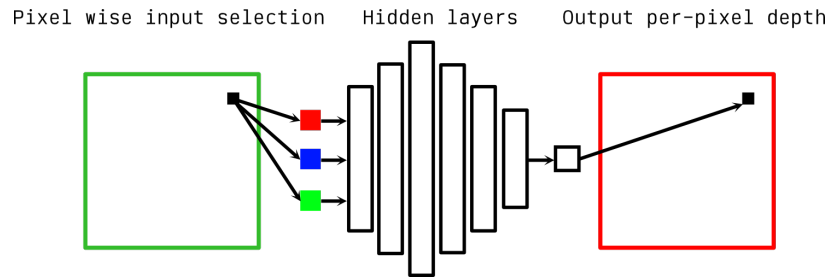


Figure 2.24: ANN for per-pixel phase unwrapping (Nguyen et al 2019), corresponding pixels from the three wrapped phase maps are passed through the network which produces a single output depth prediction.

ments contain the same phase unwrapping problem (albeit on a different scale), some techniques used in the context of CSI can also be applied to fringe projection [101–103].

Recent research uses an FCNN for phase unwrapping and some works [101] treat the problem as a multi-class classification problem, where each class represents a fringe order, while alternative approaches [98] directly regress depth values. The FCNN approach has the advantage that it can take images of different sizes without modification. In the multi-class approach [101], an additional FCNN is implanted to preprocess noisy wrapped phase maps to remove or reduce the noise. Alternatively, an SVM with a radial basis function kernel can be used to classify each pixel in the wrapped phase map by its fringe order [104], also formulating the problem as a multi-class classification task. As this approach is per pixel, it avoids error propagation in spatial unwrapping and only a single projected fringe pattern is required, compared to the many projected fringe patterns used in traditional temporal phase unwrapping. It has been shown that using an SVM for phase unwrapping can produce depth maps of a comparable quality to traditional methods in a much shorter time. As with the CNN based methods, there is a competing SVR approach which formulates the problem as direct regression of depth, rather than a classification problem [103]. Using SVR to directly predict depth values can produce measurement errors of

under 1%.

2.3.3 Machine learning for view planning

Both photogrammetry and multi-view fringe projection require images to be captured from a range of positions to reconstruct a complete part. The decision of where to capture images from and how many to capture is often an arbitrary process. An alternative approach is to leverage ML to produce an optimised measurement plan which considered the geometry of the current part being measured. One approach is to perform this optimisation in real time using a next best view (NBV) method [105,106], which iteratively finds the next camera position based on previously collected data. An approach presented by Arce et al. [107] employed structure from motion to create an initial point cloud, from which the next position was iteratively generated using an unsupervised model. This approach was specifically designed for situations where the CAD model is unavailable, an unlikely scenario in production engineering. If the CAD model of the object is known, the NBV positions can be precomputed. Mendoza et al. [108] took a supervised approach to NBV by using a traditional view-planning method based on ray tracing; the latter was used to calculate labels and generate a dataset of 15,000 training point clouds. A 3D CNN was then used to predict the NBV position directly. Comparing their machine learning based approach to traditional methods, Mendoza et al. showed that machine learning methods appeared to be consistently faster, often by many orders of magnitude. Furthermore, they showed that machine learning approaches were particularly effective at finding early camera locations but performed worse when calculating later positions; consequently, they suggested a fused approach employing machine learning to initially find a small number of

camera positions and then using a traditional algorithm to compute the following positions.

2.3.4 Machine learning for camera characterisation

Although there has been some research to recreate depth from uncharacterised cameras [109], characterisation of cameras remains a key part of camera based coordinate measurements as was discussed in Section 2.1.3. Early works used a genetic algorithm to globally optimise the camera parameters, but this method was shown to have little benefit over traditional approaches [110]. More recently, some researchers have attempted to replace the entire characterisation process with an end-to-end machine learned model [111]. Mohamed et al. [112] explicitly obtain the camera projection matrix through a support vector machine and show this approach to be more robust to noise and more computationally efficient than traditional techniques. He et al. [113] use a K-singular value decomposition sparse dictionary learning approach to perform a non-linear optimisation of the camera parameters, they claim that, once trained, this approach can enable single image characterisation. Other studies, instead, implement a hybrid pipeline which fuses machine learning techniques with the traditional characterisation pipeline proposed by Zhang [17]. Characterisation target detection and localisation specifically is a good candidate for improvement through machine learning as traditional methods can be highly influenced by factors such as noise [114, 115]. It has been shown that CNNs can be more effective than conventional algorithms at locating characterisation targets and features within photographic images [114]. For example, the machine learning for adaptive characterisation template detection (MATE) model proposed by Donné et al. [116] is a CNN trained

to be robust to noisy inputs and high lens distortion. A model developed by Chen et al. [117] has been designed specifically to be robust to views in which some portion of the characterisation target cannot be seen. A new approach to characterising the non-linear distortion parameters is to use an SVM [118], this approach was shown to accurately capture distortion parameters with a height accuracy of 3 μ m. Due to the requirement for multiple images at different angles, traditional characterisation methods can be highly affected by low-accuracy components, such as rotation stages. There is potential to use deep CNNs to characterise a camera's intrinsic parameters from a single image [119], rather than the many images required for conventional approaches. An alternate approach presented by Li and Liu [120] uses a micro-mirror device and a laser to stimulate single camera pixels and a deep CNN was then trained on this single pixel illumination data to characterise the camera. Although this process may not save much time over the traditional methods, it can be automated to remove the dependence on the user. The single pixel illumination ML approach has high accuracy while requiring fewer computations than traditional characterisation algorithms, producing an MSE of 0.0072 mm, which is an improvement of 60% over the traditional method.

Approaches for characterising the extrinsic parameters include the use of a CNN for real time estimation [121]. A CNN was trained to predict azimuthal and elevation angles from shadows cast by a single point source on a specific artefact. The CNN comprised five convolutional layers and a final fully connected layer. It was shown that there was an average $\pm 10^\circ$ error in the azimuthal and elevation angles estimated which resulted in a ± 1 mm misalignment in the measured point cloud when compared to manually re-characterised extrinsic parameters.

2.3.4.1 Implications for measurement uncertainty

As the optical metrology industry moves towards a traceable calibration pipeline for measurement [122], it becomes necessary to be able to quantify uncertainties related to any predictive ML models in the measurement pipeline [123]. Due to the complex nature of ML models, it is difficult to apply methodologies directly from the original Guide to the Expression of Uncertainty in Measurement [124]; rather a Monte-Carlo simulation approach must be taken [125, 126]. Early work in this area showed how uncertainty can be propagated through simple linear regression models [127]. Work by Cheung and Braun [128] extended uncertainty analysis to more general model types and suggested that any analysis involving ML models should consider uncertainty contributions from:

- Model output: uncertainty relating to the difference between the model prediction and the ground-truth value.
- Calibration data: uncertainty in the data which make up the model training dataset.
- Input measurement: uncertainty in the input data to a model.
- Output measurement: output uncertainties outside the calibration of a dataset.

The study by Cheung and Braun demonstrates that increasing the size of the training dataset can reduce the uncertainty contribution from the calibration data but has no effect on the other contributions whose uncertainties are either from the derivation of the model, or inherent to the data themselves.

An example where ML has started to be applied is in the characterisation

of a galvanometric laser scanner [129]. In this study, the laser is controlled through a galvanometric positioned mirror, the mirror deflects the beam to scan over the surface being measured and the laser dot is then triangulated via a camera to a known position relative to the mirror. In characterising this system, Wissel *et al.* showed that ML approaches can outperform model-based approaches and perform similarly to look-up table characterisation, providing coordinate root MSEs as low as 0.029 mm and plane root MSEs of 0.433 mm for a calibration plane size of (100×100) mm. They also show that using ‘off-the shelf’ ANN architectures can lead to large generalisation errors and that reducing the problem using a specifically designed SVM can significantly reduce this error.

2.3.5 Machine learning for point cloud analysis

Point cloud analysis is an important stage of any surface measurement pipeline as it is the stage at which information is extracted from the measurement data. Although the pipeline presented in this thesis is concerned only with generating the measurement point cloud and not performing further analysis upon it, ML approaches to data analysis are included in this review for the sake of completeness. Many ML-based approaches require the measured data to be regularised into a voxel grid because a voxel representation has a known size and known order. In contrast, point clouds do not have a regular ordered grid of points and the total number of points can vary from measurement to measurement. However, depending on the voxel grid resolution, transforming data into voxel form can either incur loss of detail or make the data size much larger. In order for an ML model to effectively operate directly on point clouds, it must fulfil the following criteria [2]:

- For a point cloud of N 3D points, the model must be invariant under the $N!$ permutations of the order of the point cloud.
- The model must be able to capture local structure and interaction between neighbouring points.
- The model must be invariant under certain geometric transforms of the entire point cloud. For example, rotating the entire point cloud should not change the analysis result.

A simple approach to deal with inputs of an unknown size would be to use an RNN which iterates over each element in the input, however, it has been shown that the order of the RNN operating on the input directly effects the output, thus this approach does not satisfy the first requirement for operation on point clouds [130]. A better example using point clouds is given by PointNet [2], which is a neural network approach for point cloud segmentation discussed in the following section.

2.3.5.1 Point cloud segmentation

Segmentation of a point cloud is a widely researched area (see the reviews by Garcia-Garcia *et al.* [64], Grilli *et al.* [131] and Nguyen and Le [132]) and, while most segmentation research is not explicitly conducted for use in measurement applications, there are many situations where it could apply, for example separating the measured object from the background, locating manufacturing defects and measuring multiple objects at the same time. Often the points are segmented into distinct semantic classes (such as background and object) in a process referred to as semantic segmentation. Semantic segmentation is distinct from instance segmentation, where

semantic segmentation would simply label a point as either object or background, while instance segmentation also assigns a specific instance with each label, as shown in Figure 2.25.

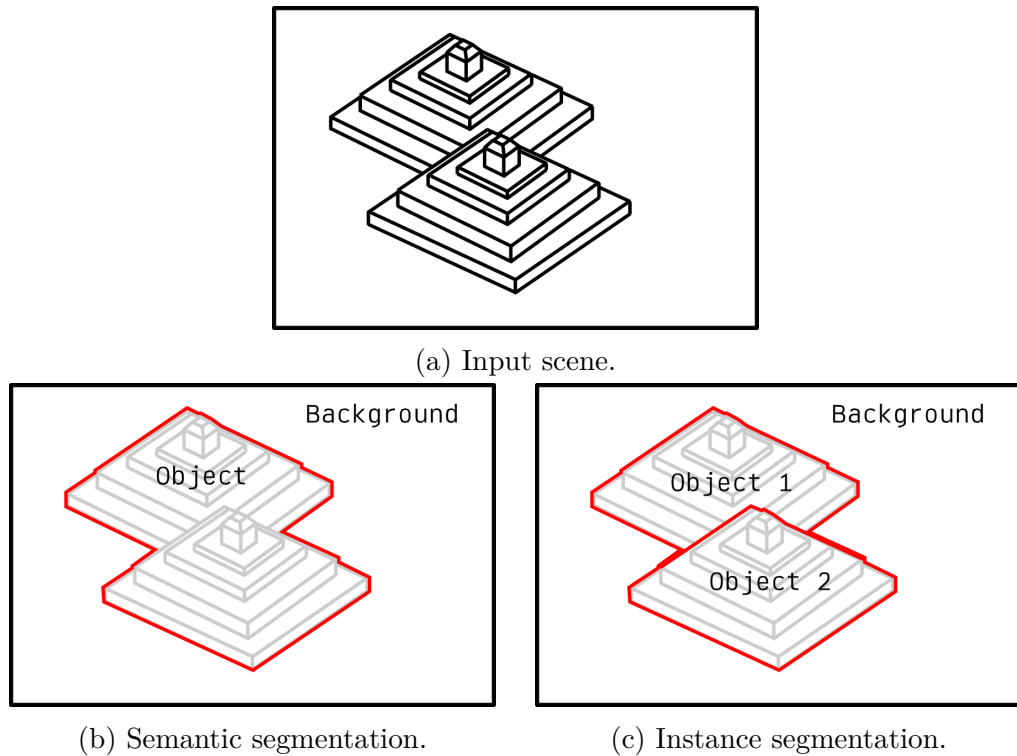


Figure 2.25: Types of scene segmentation. Can be applied to both many data types including images and point clouds.

An approach to segmentation is binary or multi-class classification on a per point basis. If the application can afford the sacrifices associated with using voxel representations, then 3D CNNs are well suited to this task [133, 134]. For example, a point cloud can be transformed into a voxel representation by dividing the space into a grid and determining whether a voxel is occupied (contains at least one point) or is not occupied [133]. Figure 2.26 shows the network used to perform the multi-class classification problem. In their example, there were eight possible classes representing seven given categories and one class representing the case where the voxel grid belongs to no category (i.e. when it is not occupied).

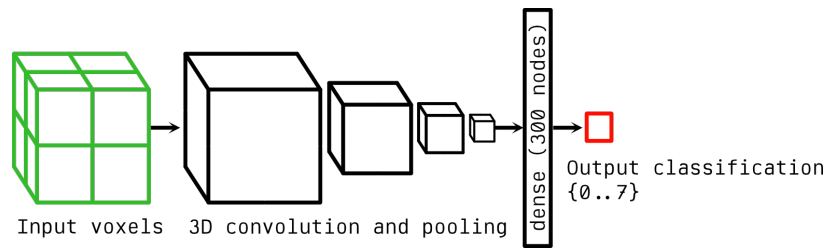


Figure 2.26: Multi-class voxel grid segmentation using a 3D CNN.

As discussed, segmentation can also be applied directly to point clouds [2, 64, 135]. PointNet [2] is essentially a series of ANNs connected together and notably has no convolutional layers. The architecture consists of two branches: the first classifies the entire cloud into one of k possible semantic classes. The global semantic classification is fed backwards into the second branch which classifies each point into one of m possible sub-classes. By concatenating the global classification result with the local feature maps, the local classification can be informed by both the local features and the global semantic class, satisfying the requirement that a model operating on point clouds can observe local structure. To satisfy the requirement for invariance over the order of the point cloud, a general function acting on the point cloud $F(\{x_1 \dots x_n\})$ is approximated by,

$$F(\{x_1 \dots x_n\}) \approx g(h(x_1) \dots h(x_n)), \quad (2.41)$$

where $f : 2^{nR} \rightarrow R$, $h : R^n \rightarrow R^k$ and $g : \{R_1^k \dots R_n^k\} \rightarrow R$, where g is a symmetric function. An ANN used to approximate h and g is represented by a single variable function applied after pooling the values returned by h . By collecting multiple h , the properties of multiple functions f can be captured, and because g is symmetric the application of this approximation is invariant to input permutation. Figure 2.27 shows the architecture of PointNet where these features can be seen.

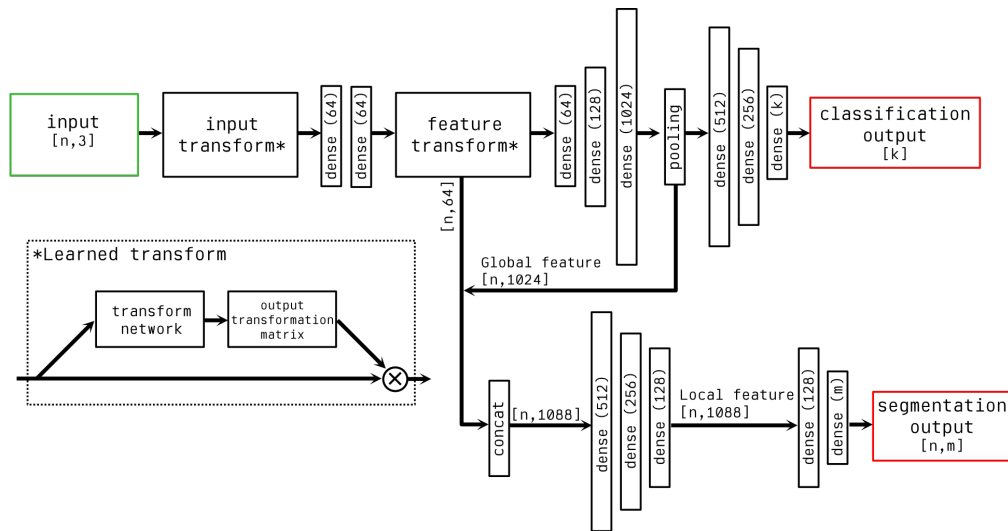


Figure 2.27: PointNet architecture [2]. The classification network takes n input points, applies the learned input and feature transforms, pools the results and provides the global classification. The segmentation network takes the global result and local feature maps to perform per-point classification.

To satisfy the final requirement that the model be invariant to geometric transformations on the entire point cloud, there are two small networks included in the model which both predict an affine transformation matrix. The first is applied to the entire input and the second is applied to the feature space (labelled input transform and feature transform, respectively, in Figure 2.27).

PointNet has become an extremely popular approach due to its robust handling of pointsets, many of the upcoming examples use PointNet at some point in their model pipeline. To improve the performance of PointNet, specifically to capture local structure in the input point cloud, Qi *et al.* [136] introduced a modification of PointNet called PointNet++. PointNet++ applies PointNet recursively on nested partitions of the input point cloud at a range of scales. This allows the learning of features to be robust to sampling density in the point cloud and has shown to be an improvement on the standard PointNet implementation on a set of benchmarks.

Segmentation is particularly useful for the detection and classification of surface defects [66, 137] and, if made fast enough, can be deployed for *in situ* monitoring [138]. This is an active area of research and also provides the motivation for much of the efficiency increases sought in both phase unwrapping [4] and stereo matching [139].

2.3.5.2 Point cloud registration

The registration of CAD data to a measured cloud or of two clouds together has many applications. A recent review by Zhang et al [140] shows that numerous ML models have been proposed to replace many of the traditional approaches to point cloud registration. These models can be considered in two categories, models which aim to improve a single part of the registration pipeline (such as feature extraction), and models which seek to replace the entire pipeline with an end-to-end learning approach. An example of an approach which seeks to integrate into the traditional pipeline is the LORAX algorithm. The LORAX algorithm [141] uses a deep ANN for dimensionality reduction, to attempt to simplify the registration problem. Essentially, the algorithm detects features in the point clouds, compares them for semantic similarity, matches features between the clouds (hence, coarsely aligning the two clouds) and finally refines the alignment with the iterative closest point (ICP) algorithm. Comparing features can be computationally expensive if the dimensionality of the feature space is large, therefore, an ANN is used to create a compact representation of each feature and these representations are compared for similarity. The ANN is set up as an auto-encoder, which is shown in Figure 2.28.

The encoder ANN takes the input and outputs a compact vector. The

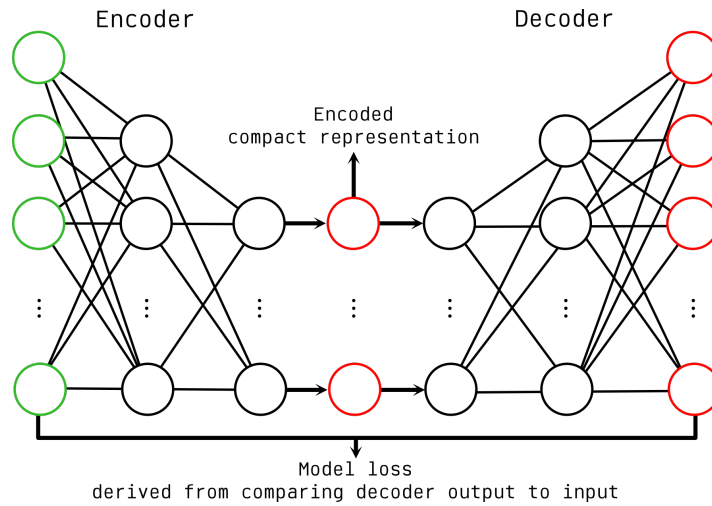


Figure 2.28: Auto-encoder architecture. Training is conducted such that the output can be decoded from the compact representation with minimum difference when compared to the input, therefore, ensuring the compact representation captures the input as fully as possible.

decoder takes this vector and attempts to recreate the input. By comparing the decoded input with the actual input, a model loss can be derived which determines how well the compact representation captures the input. This allows unsupervised training to be conducted, resulting in a network that can take large features and produce a vector which represents that feature in a much more compact manner, allowing for more efficient matching. Related research, called deep closest point [142], has also shown that using learned features to provide initial alignment between two point clouds can greatly facilitate accurate alignment via ICP, and can make ICP far more robust to failure when dealing with point clouds with poor initial alignment. As stated earlier, other approaches, such as DeepVCP [143], attempt to fully replace approaches such as random sample consensus (RANSAC) and ICP with an end-to-end learned model. An example of a fully ML based approach to registration for form measurement is research by Gojicic *et al.* [3], which focuses on combining multi-view point clouds into a completed single point cloud.

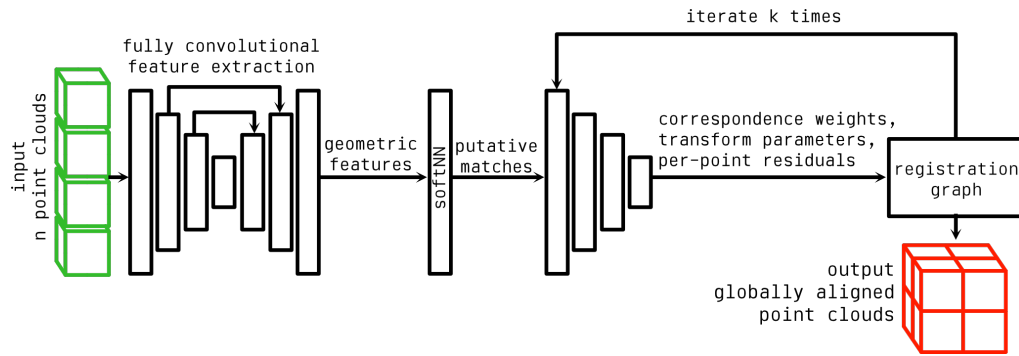


Figure 2.29: ML approach for multi-view point cloud registration [3]. Features are extracted from the input point clouds and then iteratively matched to find the best alignment.

Figure 2.29 shows the approach to multi-view point cloud registration which is outlined here—for each input point cloud, a neural network extracts a set of features. These features are fed into a further set of network blocks that compute stochastic correspondences between each combination of pairs of point clouds. Using these correspondences, a further block (labelled Reg. init. in Figure 2.29) computes initial transformation parameters and residuals, which are refined by the next block (Reg. iter.) from which the registration graph is built. The final two blocks are iterated four times to produce the complete registration. This approach is evaluated on a set of benchmark datasets and shown to outperform competing approaches on the same datasets by 25% in terms of mean rotational error. Furthermore, the approach is shown to be thirteen times faster than RANSAC when registering a set of 60 point clouds and copes well with unseen scenes and objects.

In the conclusion of their review, Zhang *et al.* [140] make the following points: ML models show clear dominance when used as a module within a traditional pipeline, such as for feature extraction, and end-to-end learned systems are growing in popularity and have the potential to become more effective than traditional methods. They also note that some failings of current ML models include the existence of a clear gap between performance on

synthetic datasets and performance on real world data. A further area for improvement is that many systems use feature extraction networks which were designed for other applications, such as PointNet and PointNet++, there is likely to be a performance gain by designing a bespoke approach for registration.

2.3.5.3 Point cloud completion

Often a measurement will result in an incomplete point cloud due to occluded surfaces, either due to insufficient views or surface complexity. It can, therefore, be useful to attempt to predict the missing data, not for extending the measurement data, but for increasing the performance of registration algorithms such as ICP. It is common to use an auto-encoder type network and RNNs as part of the pipeline for point cloud completion [4, 144–146]. Figure 2.30 shows an example approach for point cloud completion [4].

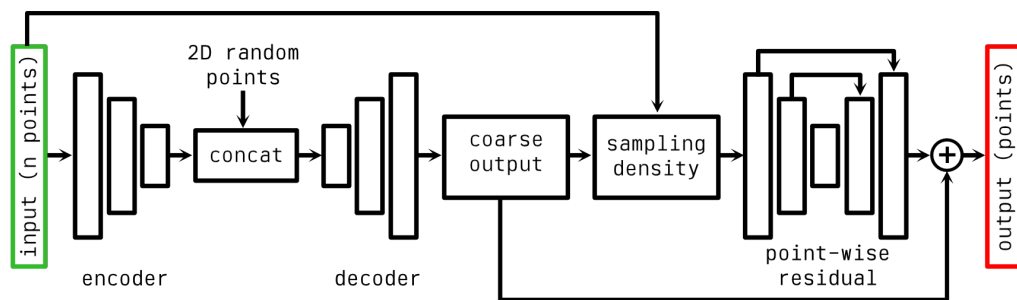


Figure 2.30: Point cloud completion network Liu *et al.* [4]. This approach to point cloud completion uses an encoder to create a compact representation of the input, this is then combined with a random 2D set of points to predict a mapping of those points onto unknown 3D surfaces, a set of these predicted surface are combined to create a coarse completion estimation. This estimation is then further densified through fedforward information from the input and ResNet layers.

The approach shown in Figure 2.30 yielded successful results, and the

point sampling approach used ensures that any known structures from the incomplete point cloud are preserved. It is not clear if this method will generalise well onto unseen objects of different semantic classes to those contained within the training dataset and the supervised learning approach also requires a large labelled training dataset. A similar approach uses PointNet, which was described in Figure 2.27, as the layers in its encoder [144], this has similar trade-offs as using PointNet for point cloud registration as was described in the previous section.

Another approach to point cloud completion is to use a GAN [147, 148]. The discriminator is trained on a combined dataset of real complete and generated point clouds, and is trained to predict whether a given input is real or generated. The GAN sequences these two networks and the discriminator output is used as the loss of the generator. Thus the generator is trained to produce point clouds that the discriminator cannot distinguish from real data. GANs can be used directly on input point clouds, do not require labelled training data and allow more general application of the network to make predictions on unseen data from unknown semantic classes [148]. An extra learning step can be inserted into the GAN pipeline (called RL-GAN-Net [147]) which learns how controlling the input vector affects the output point cloud. This can, therefore, be trained to provide an input which is likely to produce an output cloud which is most similar to the missing data. RL-GAN-Net can generate completed point clouds in approximately 1 ms, allowing it to be deployed in real time situations and produce completed point clouds from input data with up to 70% missing points.

2.3.6 Full automation of the measurement pipeline

What preceded this section was a review of optimisations of individual sections of the measurement procedure. As was stated in Chapter ?? the aim of this thesis is not just to optimise individual parts of a measurement, but to thread these algorithms together in a way which enables fully autonomous data capture. Some attempts to achieve these goals have been made by previous researchers [149–153] but all fall short of fully realising a system which is automated, optimised and general across the geometry of parts measurable by the system.

Abd-Raheem *et al.* presented a system which automates data capture and processing through use of a system consisting of a camera and a rotation stage [149]. Data is captured equally spaced around a part from a fixed location every thirty degrees of rotation leading to a dataset of 24 images. While this approach is general over all objects which can be contained in the FOV of the imaging system it has several drawbacks. Firstly the characterisation of the system is not addressed at all. Secondly, the camera position is fixed in a scan at a position which may not be optimal for all parts. The imaging strategy is also not adjusted for complex parts which may require a greater number of images from a range of different locations to achieve high quality, complete reconstructions.

A system produced by Martins *et al.* [150] allows for automated surface measurement of parts through optical range sensors and a view plan developed from the object's CAD data. Their view planning approach optimises for surface coverage and scanning costs and defines collision free, efficient scanning paths. This approach, however, uses optical depth sensors aligned in the vertical direction which can move in the horizontal plane and vertically while the part is fixed. The three free degrees of freedom means that the reconstructions generated represent height maps relative to the

vertical direction, missing surface information on occluded surfaces. This approach may be valid for some parts where only one functional surface must be measured, but is far from generalisable over all 3D objects without the need for multiple point clouds to be stitched together, a costly and often manual process.

Fan *et al.* present a DFP system which computes an imaging strategy during data acquisition from an initial scene exploration [151]. Multiple objects can be placed within the measurement volume, the imaging system will find a rough geometry of the scene, compute an optimal set of sensor locations to perform a detailed measurement, then perform the measurement. This approach has some advantages, the ability to measure multiple objects at once and generality across object geometries. However, due to the view planning occurring during the data acquisition and the need for initial scene exploration, the approach is relatively slow, with the paper reporting average per-object measurement times of between eight minutes and twelve minutes. Furthermore, the system has only three free degrees of freedom so the number of available imaging positions is limited.

Other approaches to automated measurements are designed for specialised measurement scenarios such as the measurement of turbine blades [153], cities [154], carbon fibre morphology [155] and the detection of specialised photogrammetric targets [152]. Further methods require the generation of fixtures to guarantee the position of measured parts within the measurement volume [156]. Specialised approaches to automation are not considered here due to the desire for generality across 3D geometries,

As can be seen from the above review, there is no such implementation in the current literature which addresses both automation while maintaining generality, surface coverage, fast operation and per-part optimisation. And, to the author's knowledge, no complete pipeline for automation and optimisation of the entire measurement pipeline has been presented beyond

that presented by this thesis in Figure 1.3.

2.4 Summary

In this chapter all background theory required to understand the two measurement methods which this thesis concerns itself with has been given.

The choice of these measurement methods has been justified.

Further relevant background theory pertaining to ML methods has been summarised. Particular focus was given to advanced model architectures which will be employed in the following chapters.

Finally, a review of the current state of the art in applications of ML to optical coordinate metrology was given. From this review it was clear that while ML remains a developing field, there are many areas where ML models can be usefully applied to optical form and coordinate metrology. Considering this review of the state of the art, the areas of camera characterisation, measurement automation and optimisation, point-cloud registration and data generation will all be addressed by work in this thesis directly. While each of these areas has seen some research interest it is very clear they are subjects which are far from solved problems.

Further, it is clear from Section 2.3.6 that there currently exists no satisfactory, optimal solution to full automation of coordinate measurement of generic three dimensional objects. Particularly the combination of automated *and* optimised data capture is lacking. Many systems lack the degrees-of-freedom required to achieve high surface coverage. There is also a tendency to design systems specialised to measuring particular classes of objects as known properties of these objects can be exploited to aid with automation. There is also a lack of literature pertaining to an overall schema for how individual parts of the pipeline should be optimally threaded to-

gether to complete a full measurement from start to finish. Therefore, this review validates the pipeline in Figure 1.3, and its constituent algorithms, as a valuable contribution of this thesis to the field.

Chapter 3

Methods

This chapter summarises the experimental and commercial measurement setups used to test and validate the approaches proposed in Chapters 4 to 8. Also summarised are the reconstruction methods used to produce 3D point clouds, system characterisation methods, computational methods, and finally the collection of test artefacts used to evaluate performance of the proposed approaches.

3.1 Measurement systems

This section presents two photogrammetry systems which were used to test the methods proposed in Chapters 5-8, a DFP system used to test the camera characterisation methods presented in Chapter 4, an optical surface texture instrument used to collect training data for Chapter 7, a commercial CMS used to compare the proposed approaches against a current commercial solution, and a tactile CMM to allow comparison to a calibrated measurement.

As discussed in the introduction, the application is to measurements conducted at "close-range". This typically refers to measuring objects which are sized in the order of centimeters with desired point spacing on the order of tens of micrometers. As such, the optical CMSs presented in this section all have measurement volumes on the order of 30 square centimeters and are equipped with high resolution sensors.

3.1.1 Photogrammetry

3.1.1.1 MMT system

The initial photogrammetry data in this thesis were collected using the system shown in Figure 3.1, referred to as the Manufacturing Metrology Team (MMT) system.

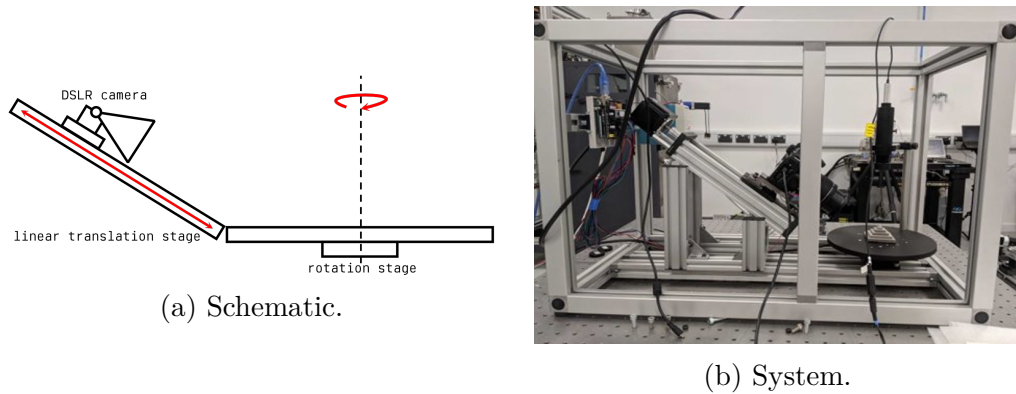


Figure 3.1: The MMT system, used to collect initial photogrammetry data.

As can be seen, the system has two DoFs, which can be controlled computationally. As the camera's optical axis and the linear stage are aligned with the centre of the rotation stage, the motion stages can control a radius and rotation relative to the centre of the rotation stage. The camera used is a Nikon D3500 camera with (4496×3000) pixel resolution, equipped with an AF-P DX NIKKOR 18 – 55 mm f/3.5 – 5.6G lens.

The control software for the system was written in MATLAB with more details given in Sims-Waterhouse [157]. Unless otherwise stated, a measurement on this system is reconstructed from a series of 60 images equally spaced on a ring around the object being measured. An additional image is taken by moving the camera a set distance along the linear motion stage, this distance is then used to scale the resultant point-cloud.

3.1.1.2 Taraz system

A large amount of photogrammetry data was also collected using the Taraz Metrology P2 system which is shown in Figure 3.2, henceforth referred to as the Taraz system.

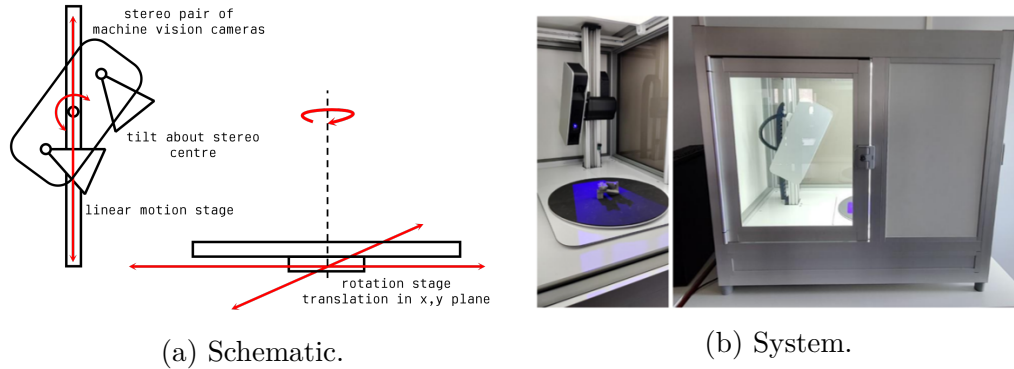


Figure 3.2: The Taraz system, a Taraz Metrology P2 system.

The Taraz system was selected as it overcomes some limitations of the MMT system. First, it is a five DoF system which enables many more imaging positions which is critical for executing the view plan as developed in Chapter 5. High resolution machine vision cameras are used in a stereo pair, the baseline distance between these cameras can then be used to apply scale to any measurements taken with this setup. To ensure the scale is applied as accurately as possible, the baseline distance was characterised using the process highlighted in Appendix C. It was determined that the baseline distance is $264.00 \text{ mm} \pm 0.40 \text{ mm}$ (full results tables are presented in Appendix C.1). This approach to determining scale is much more stable and less likely to introduce scale errors than the approach used by the MMT system which relies heavily on the positional accuracy of the linear motion stage. Finally, higher quality motion stages were used with much higher positional accuracy, greatly improving the positioning of the measurement head. As can be seen in Figure 3.2b the measurement head also contained a projector for performing fringe projection measurements, this capability

was not used in this thesis.

The Taraz system utilises a pair of Basler acA5472-17um cameras with Kowa LM12C 12.55 mm f/1.4 – 16 lenses.

3.1.2 Fringe projection

3.1.2.1 DFP system

A fringe projection system is required to validate the system characterisation approach developed in Chapter 4. The system used, called the DFP system, is shown in Figure 3.3.

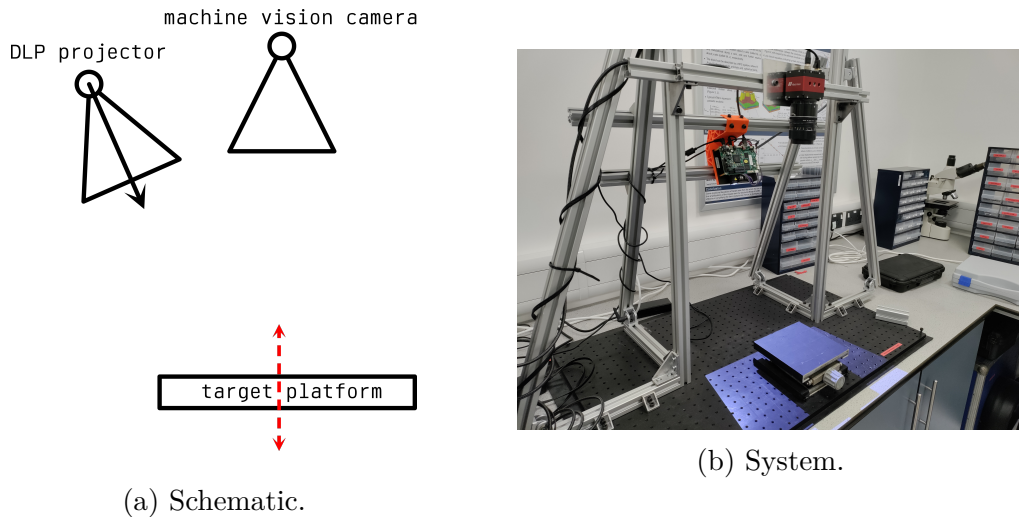


Figure 3.3: The DFP system, used to validated system characterisation methods. Dotted red line indicates that the height of the target platform can be adjusted prior to, but not controlled during, a measurement.

This system was used as it is the simplest embodiment of a DFP system, with a single camera and projector. An adjustable platform was used to allow images to be taken with characterisation targets in a range of locations across the focal ranges of the camera and projector. The imaging system is comprised of a Prosilica GT1520 machine vision camera with a Soligor 35 mm f/2.8 lens. The projector is a Texas Instruments 4500 lightcrafter

with a (912×1140) pixel resolution and a focal length of 20 mm.

3.1.2.2 GOM system

To compare the proposed methods to current optical CMSs, comparisons are made to measurement results obtained by popular commercial solutions. Specifically, a GOM ATOS Core 300 DFP system is used which can be seen in Figure 3.4 and is henceforth referred to as the GOM system.

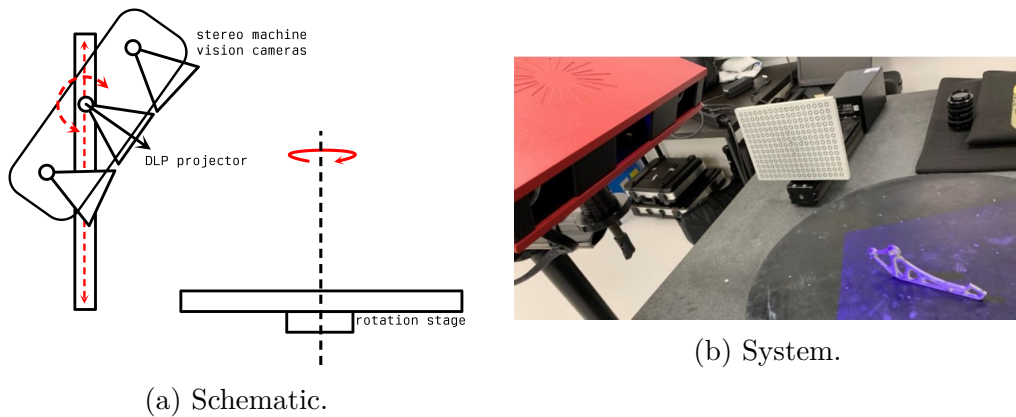


Figure 3.4: The GOM system, a GOM ATOS core 300. Used for comparison to commercial optical CMS. Red line shows controllable motion, dotted red line shows adjustable position prior to measurement.

There are obvious similarities to the Taraz system in terms of physical design embodiment making comparisons between these two systems particularly interesting. GOM provides acceptance testing/performance verification of this product according to the VDI/VDE 2634 part 3 standard [158], the results of this acceptance testing are presented in Appendix A.

3.1.3 Texture measurement

Although the focus of this thesis pertains to form measurement, a small amount of texture measurement is conducted, particularly in Chapter 7.

This texture measurement is important to accurately simulating micro-scale surface detail when creating synthetic images to train models used in photogrammetry, such as the monocular pose estimation approach detailed in Section 8.2.

3.1.4 MMT-LS system

For large scale texture, such as that seen on surfaces treated with industrial coatings, the MMT system which was shown in Figure 3.1 was modified with a laser speckle projector to operate in a texture measurement mode, this is shown in Figure 3.5. Hereon this system is referred to as the MMT laser speckle (MMT-LS) system.

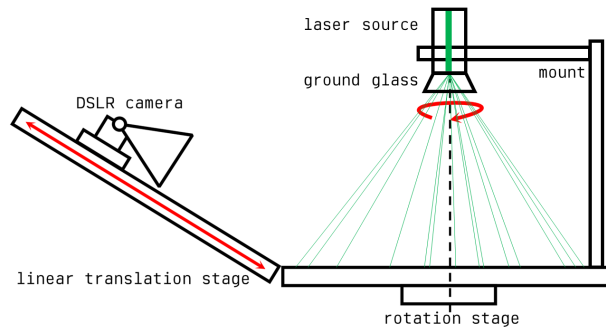


Figure 3.5: The MMT-LS system, including laser speckle projector for large scale surface texture measurement.

As can be seen in Figure 3.5, the laser speckle projector is affixed to the rotation stage to remove any relative motion between the part and the light source. This projector consists simply of a laser source projected through a ground glass lens resulting in a complex pseudo-random texture on the surface. The resultant artificial texture projected onto the surface makes it possible to measure relatively featureless and smooth surfaces. The projector is comprised of a laser diode (532 nm, 4.5 mW), focusing lens (50 mm, biconvex) and glass diffuser (600 grit polished). A green laser was chosen

due to the higher number of green pixels on the Bayer filter of a typical complementary metal-oxide semiconductor (CMOS) sensor [159].

3.1.5 FV system

For smaller scale texture measurement, a focus variation (FV) microscope was utilised. The operating principle of this microscope is to create an image stack over a range of focal plane distances, the depth of each pixel is determined as the focus plane distance for which that pixel has a maximum contrast to the surrounding pixels. Figure 3.6 shows the main components of a focus variation microscope and Figure 3.7 shows how depth information can be inferred from the image stack.

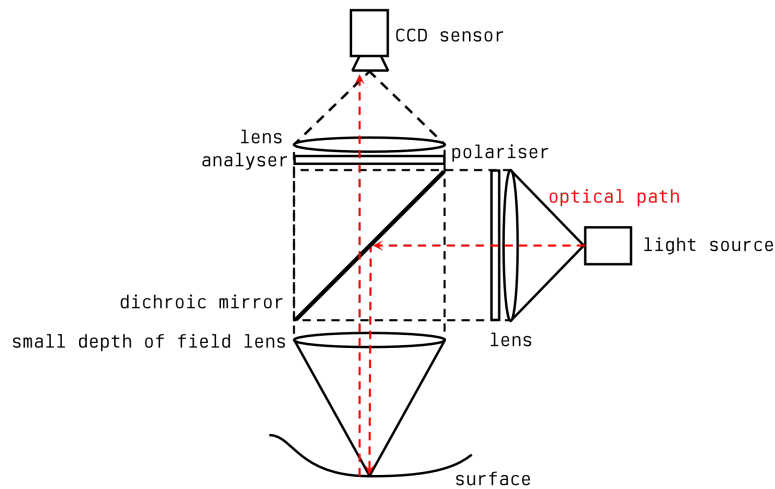


Figure 3.6: Focus variation schematic.

Specifically, an Alicona Infinite Focus G5 was used and is referred to as the FV system for the remainder of this thesis with the following instrument settings: $20\times$ objective lens, numerical aperture 0.4, field of view (0.81×0.81) mm, lateral resolution $3.51 \mu\text{m}$, vertical resolution: 12 nm, ring light illumination, measured area (3×3) mm.

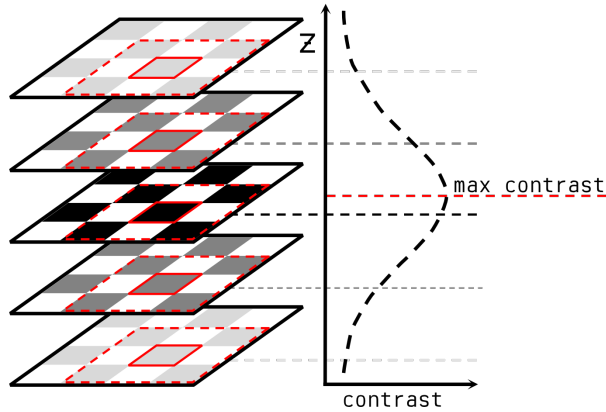


Figure 3.7: Focus variation operating principle.

3.1.6 Tactile measurement

As was noted in Section 1.1, tactile measurements on a CMM can be considered a reliable tool due to their traceable measurement results. To assess the measurement quality of an optical system, it is therefore informative to compare the optical CMS result to a measurement from a CMM. In this case a Mitutoyo Crysta Apex S7106 CMM was used with a 1 mm ruby sphere probe tip, referred to hereon as the CMM. The calibration method, results and certification for the CMM are presented in Appendix B.

3.2 Measurement data analysis

Point cloud analysis was completed using the open source software CloudCompare [160]. CloudCompare was used to perform point cloud registrations, calculation of surface normals, and point cloud comparisons such as calculating point-to-mesh and point-to-point distances of clouds registered to each other or to CAD data. Point cloud registration was completed using the popular iterative closest point (ICP) algorithm [161]. Figure 3.8 shows the basic flow of the algorithm.

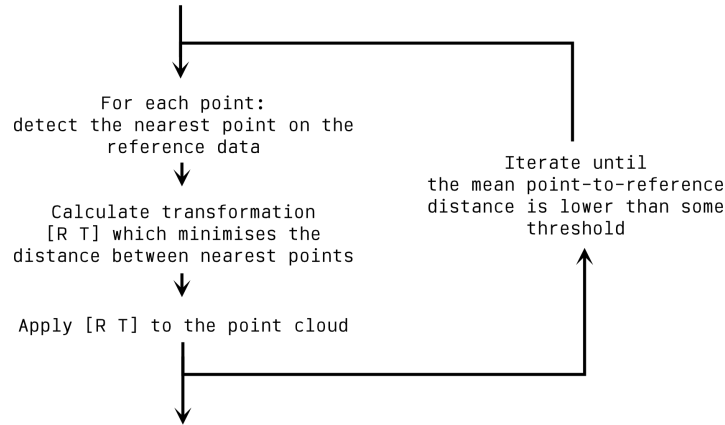


Figure 3.8: ICP algorithm.

ICP requires an initial coarse alignment between the two entities to produce robust results. This is given by the user manually selecting three pairs of roughly corresponding points on the two entities being aligned. Once a measurement result is registered to either CAD data or another measurement result, comparisons between the two registered entities are simple to compute.

Surface texture analysis, performed in Chapter 7, was performed using the software MountainMaps [162]. This software allows easy filtering of data and has built-in ability to compute surface parameters such as those given in ISO 225178 [163].

3.3 Computational methods

All large scale compute tasks such as training deep neural networks, rendering large quantities of training data etc. were completed on the University of Nottingham Augusta high performance cluster (HPC), hereon referred to as the HPC. The HPC [164] is equipped with high core count central processing unit (CPU) nodes, useful for running many smaller models in parallel; and a set of graphics processing unit (GPU) nodes, useful for large

scale models which benefit greatly from hardware acceleration.

3.3.1 Data acquisition

The CMM, FV and GOM systems are all commercial products and as such are packaged with their own data acquisition software. Both the MMT system and the DFP system are controlled via MATLAB code developed by colleagues within MMT. The Taraz system is controlled via Python code developed at Taraz Metrology Ltd. In the case of both the MMT system and Taraz system, imaging positions are provided as list of $[x, y, z]$ locations given relative to the centre of the measurement volume. These global coordinates must be transformed into G-code [165] machine positions to enable automated data acquisition. Appendix D presents how the machine coordinate system is aligned with the global coordinate system to enable this automated data collection.

3.3.2 Photogrammetric reconstruction

Two methods are used for the reconstruction of 3D data from a dataset of images, both methods are agnostic to which system was used to collect the data. First is the open multi-view geometry (OpenMVG) structure from motion (SFM) library [166]. OpenMVG was used as it has convenient Python bindings allowing pipelines to be easily built and integrated with stages beyond reconstruction, such as characterisation and data acquisition. However the output of OpenMVG is a sparse scene rather than a densified point cloud. A separate library such as open multi-view stereo (OpenMVS) [167], which consumes a OpenMVG scene to create a dense reconstruction, must be used if a dense reconstruction is required.

Agisoft Metashape [30] is used to produce dense reconstructions. Metashape is a commercial software for performing reconstruction and represents the current state-of-the-art, making it a useful tool for validating the proposed approaches in this thesis.

3.3.3 Machine learning methods

Each model used is described in its corresponding chapter. All ML models described in this thesis were built, trained, and deployed using TensorFlow [168] and Keras [169]. Keras is a high level application programming interface (API) built on top of TensorFlow for machine learning. TensorFlow itself has a relatively lower level front-end API for performing tensor operations and other useful computations such as automatic differentiation. TensorFlow also has a back-end which acts similar to a compiler providing optimisation and allowing TensorFlow models to make use of acceleration through compute unified device architecture (CUDA) and accelerated linear algebra (XLA).

The only exception to this is the genetic algorithm (GA) described in Chapter 5 which was implemented using the GA toolbox in MATLAB [170].

3.3.4 Rendering methods

All rendering tasks described in later chapters were performed using Blender [171], an open source 3D modelling software packaged with a powerful rendering engine called Cycles. In this thesis Blender was built as a python library allowing automation of many of the processes described in later chapters. Blender can implement the non-linear camera model as was described in Section 2.1.1.1 allowing accurate simulation of properly charac-

terised systems.

3.4 Artefacts

A range of artefacts were required to test the measurement solutions proposed. These artefacts were designed to be produced with AM processes due to the current industrial interest in these manufacturing methods. Additionally described is the characterisation target used for testing the proposed camera characterisation method in Chapter 4.

3.4.1 Characterisation target

The characterisation target used is a dot grid comprised of 184 black circular features on a white plate. This target is shown in Figure 3.9.

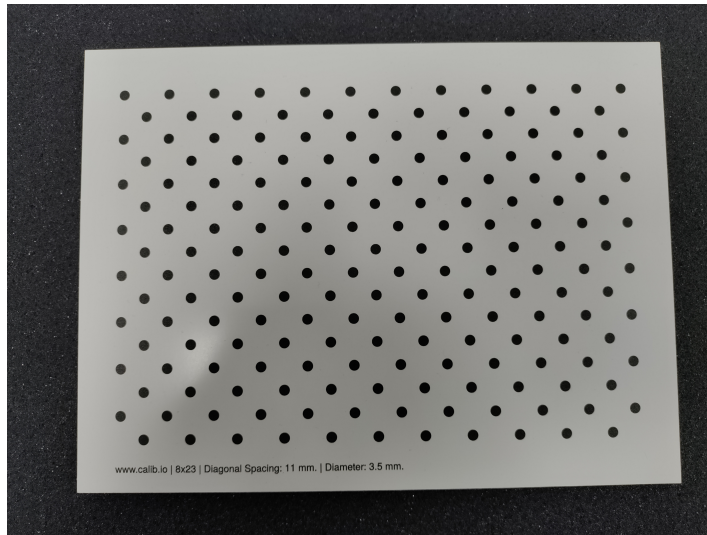


Figure 3.9: Characterisation target used to validate the proposed characterisation approach.

The feature size is 3.5 mm with 11 mm diagonal spacing. The feature locations were adjusted using a measurement of the characterisation target

using an optical CMS with a maximum permissible error given by $MPE_L = (2.5 + \frac{L}{1000})\mu m$.

3.4.2 Measurement artefacts

A set of measurement artefacts were designed and fabricated to be used in validating the methods presented within this thesis. Each was manufactured with an AM process for two reasons, first many of the industrial applications of optical coordinate measurement are to AM parts, second AM parts contain many surface features which are conducive to producing good quality reconstructions in photogrammetry.

3.4.2.1 Simple artefacts

Figure 3.10 shows a set of 'simple' artefacts which were designed to assess coordinate measurements. Having artefacts with relatively simple features was important to allow comparison to CMM.

Each artefact shown in Figure 3.10 has a $50mm \times 50mm$ square base and are referred to as the pyramid, pillars, sphere and recess artefact respectively. These artefacts were fabricated in grey polymer using fusion deposition modelling (FDM), white polymer using laser powder bed fusion (PBF) and Ti-6Al-4V (Ti64) using electron beam powder bed fusion (EB-PBF).

3.4.2.2 Tomas artefact

Figure 3.11 shows the Tomas artefacts (named for its designer) which was designed to include many features to assess the accuracy of form recon-

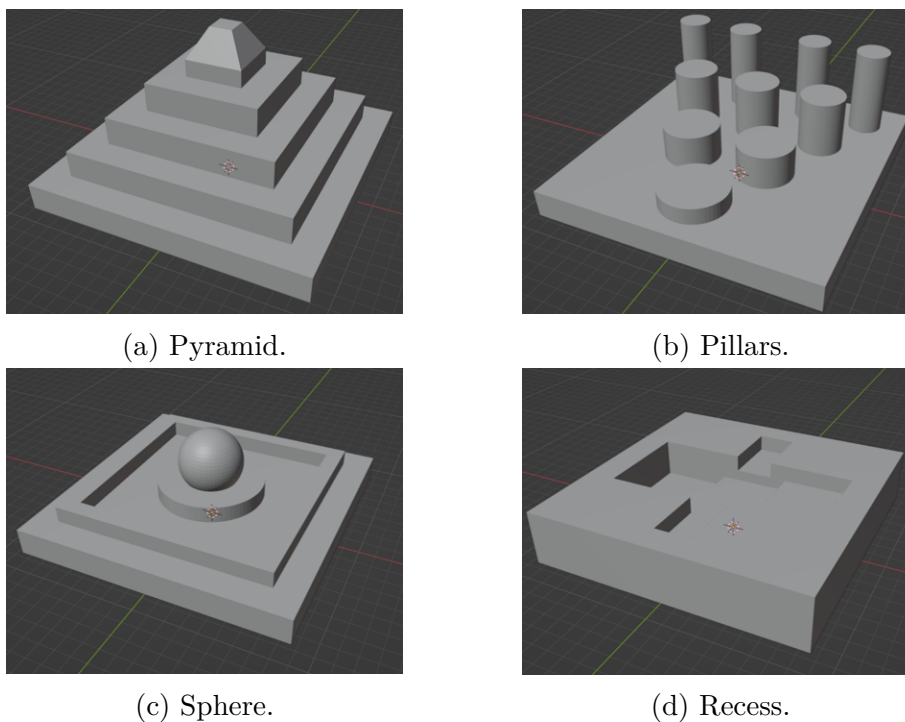


Figure 3.10: CAD data for the four simple artefacts.

struction, e.g. plane-plane distances, sphere-sphere distances, cylindricity, flatness, hole diameter etc.

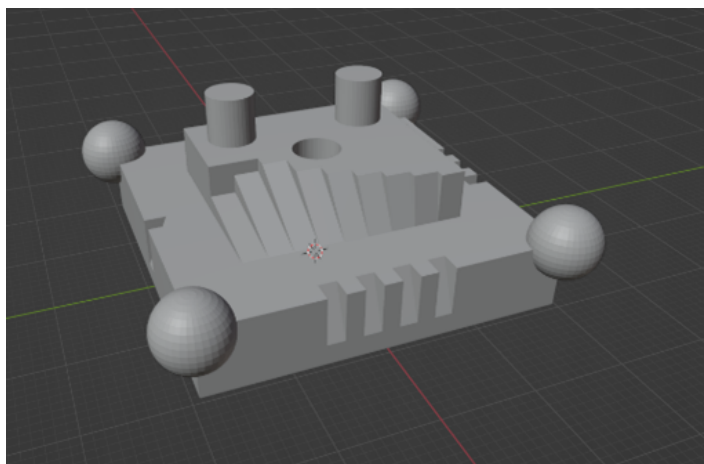


Figure 3.11: CAD for Tomas artefact.

The Tomas artefact was manufactured with EB-PBF from Ti64 and also had a square base of (50×50) mm.

3.4.2.3 Bracelet artefact

The bracelet artefact, shown in Figure 3.12 was designed to investigate the effect of face angle relative to the powder bed on surface texture in PBF processes.

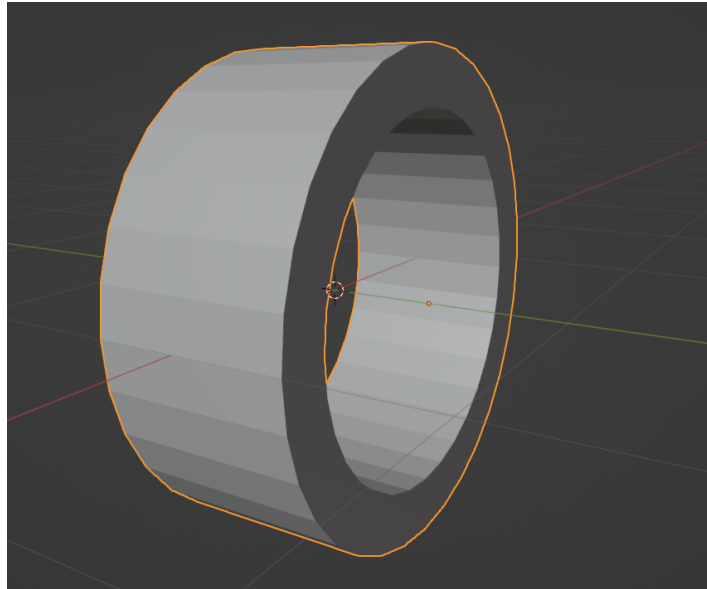


Figure 3.12: CAD for bracelet.

The bracelet was designed with an outer diameter of 91 mm with flat faces at 10° increments. Again, this was manufactured from Ti64 using EB-PBF.

Chapter 4

Improving camera and projector characterisation

The work in this chapter was completed in collaboration with George Gayton who gathered the datasets used herein. Findings from this study were presented at a meeting of the American Society for Precision Engineering at Oak Ridge, TN and published as a journal article in:

Eastwood J, Gayton G, Leach R K, Piano S 2022 Improving the localisation of features for the calibration of cameras using EfficientNets *Opt. Express* **31** 7966-7982.

Figure 4.1 indicates how the camera characterisation method presented in this chapter fits within the overall pipeline. As can be seen, the camera parameters calculated during characterisation are used as input to many processes later in the pipeline making characterisation an important process in the overall measurement procedure.

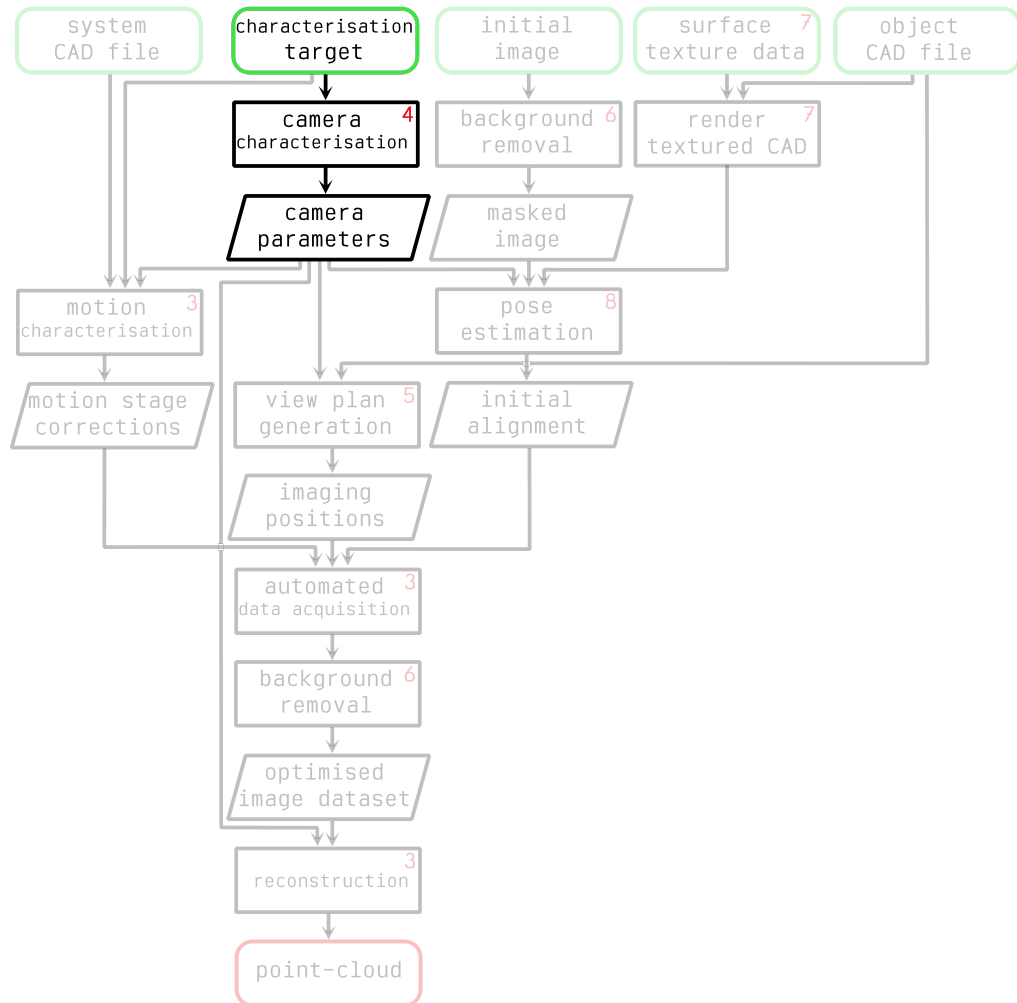


Figure 4.1: Camera characterisation shown within the overall proposed measurement pipeline.

The aim of this chapter is to improve robustness to adverse imaging conditions caused by inexperienced operation. Improving performance in adverse imaging conditions also enables a greater range of the imaging FOV to be covered during characterisation allowing greater performance of the

measurement system by creating a more representative camera model.

As was discussed in Section 2.3.4, some literature has investigated both end-to-end learned solutions as well as hybrid solutions where ML models are used within an otherwise traditional characterisation pipeline. In this chapter a new hybrid approach is proposed where ML is used only to refine the characterisation target feature locations within each image in the characterisation dataset. It is shown that the proposed approach can reduce the mean reprojection error as measured by the residual magnitude by 50%, even in adverse imaging conditions which causes competing traditional refinement methods to fail completely.

4.1 Introduction to camera characterisation

In this chapter, a hybrid ML approach (referred to hereon as the ML method) to the problem of camera characterisation is adopted. In this approach an initial estimate of the feature locations is provided by traditional methods, then refined through a learned model, before these locations are used in a characterisation procedure as was presented in Section 2.1.3. The characterisation target, which was introduced in Section 3.4.1, is comprised of black dot features on a white background and can be seen in Figure 4.2, this is chosen as it provides a large amount of phase information when characterising fringe projection systems. Although this target is chosen for improved performance on DFP systems it is still generally applicable to characterising any camera-based system. Our proposed approach first takes an initial estimate of the location of each dot feature as given by OpenCV `cv::findChessboardCorners`. A set of new images is then created from a (101×101) pixel bounding box around each feature, such that each sub-image contains a single dot with the OpenCV centre location of that feature at the centre

pixel of the new cropped image. Figure 4.2 demonstrates this process.

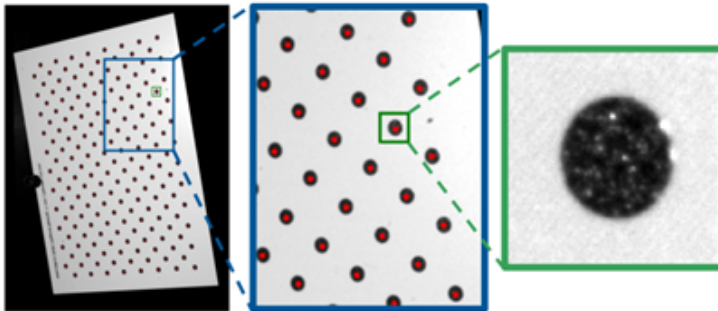


Figure 4.2: An example image of the characterisation target used in this paper. In blue, a zoomed section of the target is shown with OpenCV feature locations shown in red. In green, an example of the cropped sub-images formed around each detected feature is shown.

Each sub-image is passed to a ML model based on the EfficientNet architecture [46] which was introduced in Section 2.2.2.1. The model is trained to predict a sub-pixel correction to the OpenCV centre location. The EfficientNet is trained on synthetic data in which the ground truth centre is known implicitly, the generation of this training data is presented in Section 4.2. Once trained, the EfficientNet model is inserted into the characterisation pipeline, the proposed characterisation pipeline can then be evaluated against real data. First, the ML method is compared against using purely OpenCV (OCV method) and shown to provide large reductions in the re-projection error across a range of imaging conditions. Secondly, the ML method is compared to an alternate refinement approach using traditional image processing based on the line-spread function (LSF method), which is described in Section 4.3. It is shown that the ML method performs comparably to the LSF method in ideal conditions, but the ML method is much more robust to adverse imaging conditions such as noise and the presence of speckles caused by specular reflection. This improved robustness allows the characterisation image set to contain a wider range of views across the measurement volume when the hybrid pipeline is used and, as such, allows

improved characterisation results over the LSF method.

4.2 Dataset creation

As was shown in Figure 4.2, when the model is deployed it will operate on sub-images of a single feature, rather than the full characterisation image. Therefore a labelled training dataset of these sub-images is required, this dataset is built by generating a large set of synthetic ellipse images. Each virtual ellipse feature used in the training data is created using a set of parameters, given by:

1. Ellipse position X
2. Ellipse position Y
3. Ellipse semi-major axis A
4. Ellipse semi-major axis B
5. Ellipse rotation θ
6. Internal pixel distribution
7. External pixel distribution
8. Blurring kernel width
9. Specular size
10. Specular extent

The ellipse parameters (X, Y, A, B, θ) explicitly define the feature shape itself, these parameters are visualised in Figure 4.3.

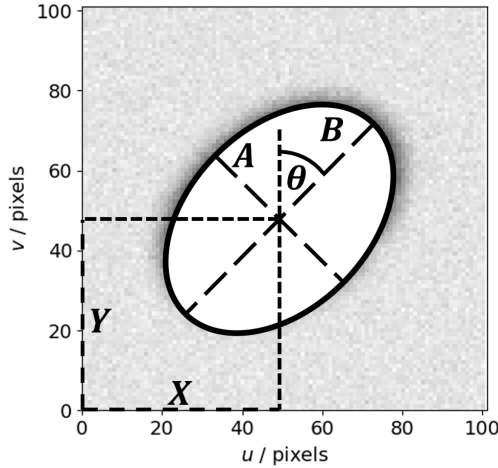


Figure 4.3: Ellipse geometry parameters.

The internal distribution defines the distribution of intensity values inside the feature, while the external distribution defines the distribution of intensity values outside the feature. Both the internal and external pixel distribution are taken to be log-normal distributions [172]. The blurring kernel defines the blur of the image of the feature. Finally, the specular size and specular extent determine the internal pixel values that do not typically conform to the internal pixel distribution because of non-Lambertian reflections within the ellipse. Specular size represents the size of each specular feature in pixels, while specular extent represents the percentage of internal pixels which do not constitute specular artefacts. Figure 4.4 shows the effect of the specular parameters on an example simulated image.

A range of images of real target features and measurements of the ellipse parameters were conducted and the distributions of these parameter values estimated via kernel density estimation (KDE). These probability density functions (PDFs) were then randomly sampled to generate each image in the simulated dataset. The PDFs used to generate some key parameters (ellipse centre and specular parameters) were set manually to exceed the values determined by KDE such that the model was trained to handle outliers. Table 4.1 summarises how each parameter distribution was set.

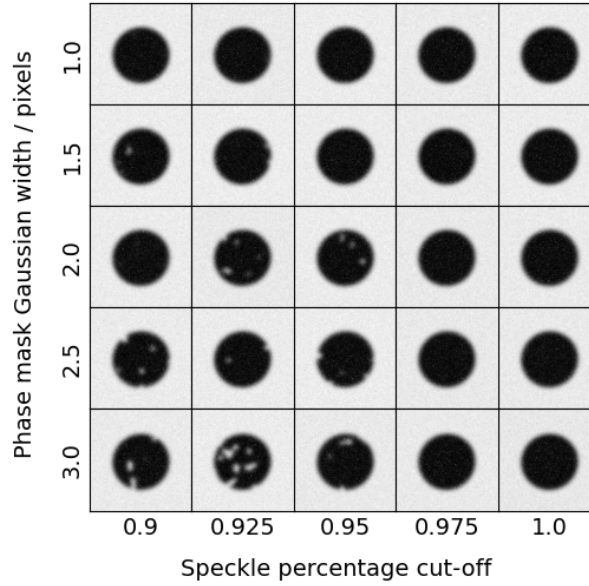


Figure 4.4: Effect of changing the specular parameters on randomly sampled ellipses with all other parameters set to be constant.

Parameter	Distribution
Ellipse Centre (X, Y)	Gaussian, $\mu = 50, \sigma = 0.1$
Ellipse axes (A, B)	Determined via KDE on the real image dataset.
Ellipse rotation (θ)	Determined via KDE on the real image dataset
Internal/External pixel distributions	Determined via KDE on the real image dataset
Blurring kernel width	Determined via KDE on the real image dataset
Specular size	A uniform distribution in the range 1 – 3
Specular extent	A uniform distribution between 0.9 – 1.0

Table 4.1: Parameter distributions used when creating the simulated characterisation dataset.

The creation of an ellipse is shown in Figure 4.5. First, in 4.5a, the parameters (X, Y, A, B, θ) are used to generate a rasterised ellipse comprised of pixel values between 0 and 1. The ellipse is then renormalised to the correct contrast and offset. Then, in 4.5b, sub-optimal reflections are added to the ellipse as a series of random white pixel blobs and in 4.5c, the ellipse is blurred. Finally, in 4.5d, ellipse-specific noise is added to the internal and external portions of the ellipse, with any pixels exceeding the maximum 10-bit value (1023) of the camera being reset to 1023.

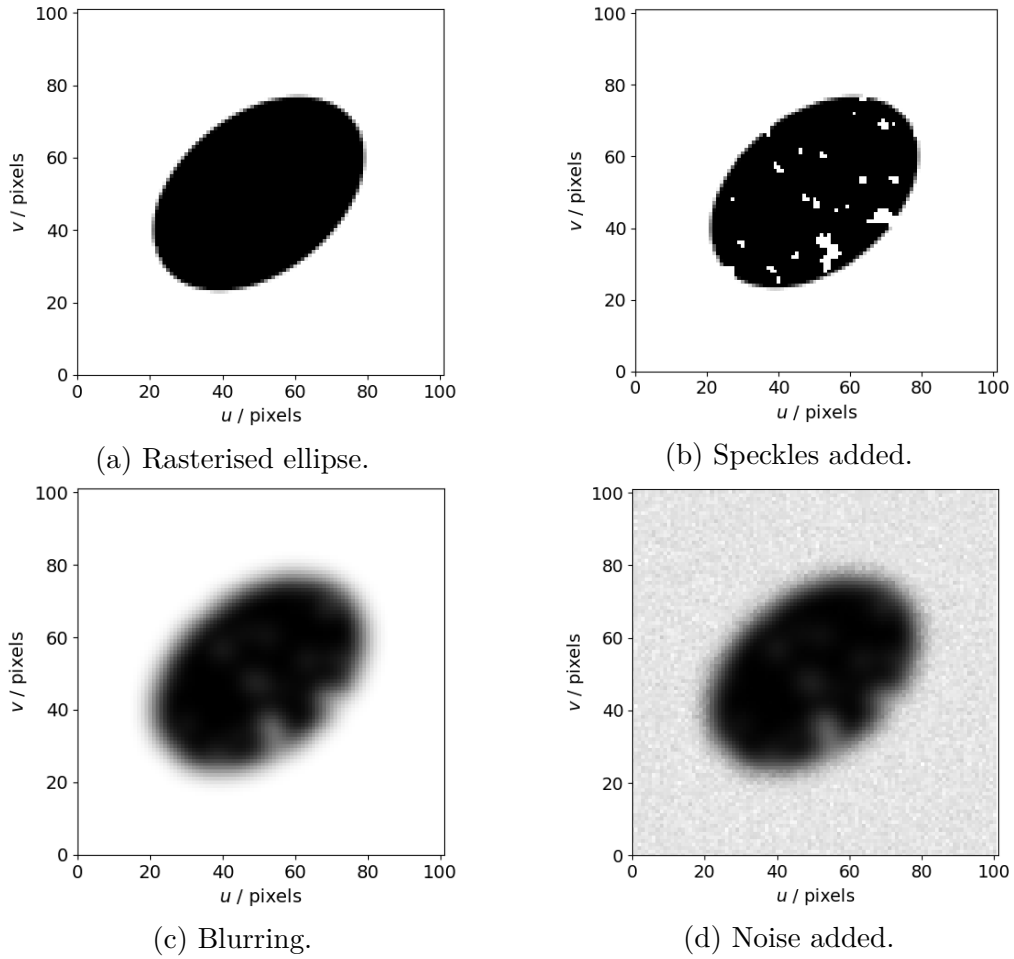


Figure 4.5: Feature sub-image simulation method.

Figure 4.6 shows a comparison between some real and some synthetic sub-images of features. It can be seen that the synthetic images are qualitatively similar to the real data. It is verified that the synthetic data must be a good representation of the real data in Section 4.5.2 when the EfficientNet, trained on the synthetic data, is shown to perform well on real images and produces improved characterisation results.

Using the approach described above, a training set of 10000 synthetic characterisation features was created and a further 1000 were saved for testing. Manually measuring this number of samples would have been infeasible simply due to the large number and variation required between samples. Furthermore, manually measuring and verifying the ground truth

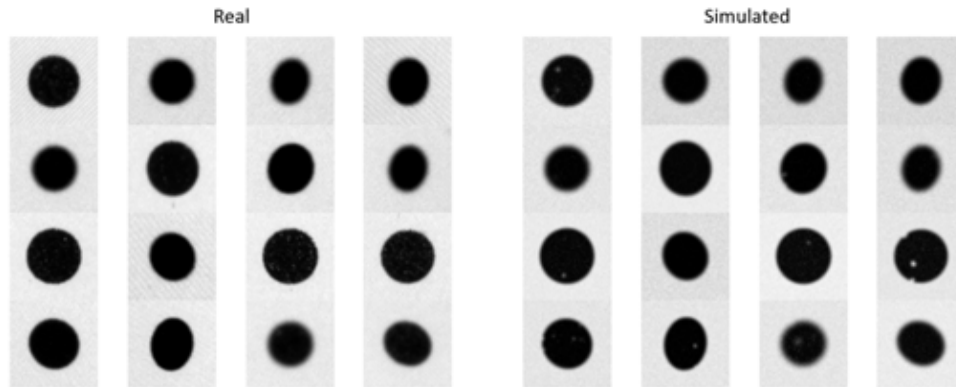


Figure 4.6: Comparison of real and simulated feature sub-images.

centre of this many real features would have made the endeavour practically impossible and it is not clear what the best approach to generate this ground truth data would be. For these reasons simulation was the only viable solution to the creation of a large, varied and labelled dataset.

4.3 Line-spread function approach

A common method to find ellipse centres is to fit an ellipse to edge points estimated from the largest gradients in the image. For robustness, this can be done along interpolated 1D lines from an estimated centre, where each line is called a line-spread function. This is therefore called the LSF method. First, a gradient image of the region containing the ellipse, shown in Figure 4.7a, is found by convolving the region with a Sobel kernel [23]. A series of line-spread functions are taken of the gradient image that expand radially from the estimated centre of the ellipse - it is assumed that the initial ellipse centre estimation is within ± 1 pixel. The line-spread function is interpolated from the gradient image, using a bilinear interpolation, shown in Figure 4.7b. In Figure 4.7b, a Gaussian function is fit to all line-spread functions to estimate the centre of the peak that corresponds to the ellipse boundary. Erroneous peak estimations are filtered out using

a random sample consensus (RANSAC) algorithm and the result is shown in Figure 4.7c.

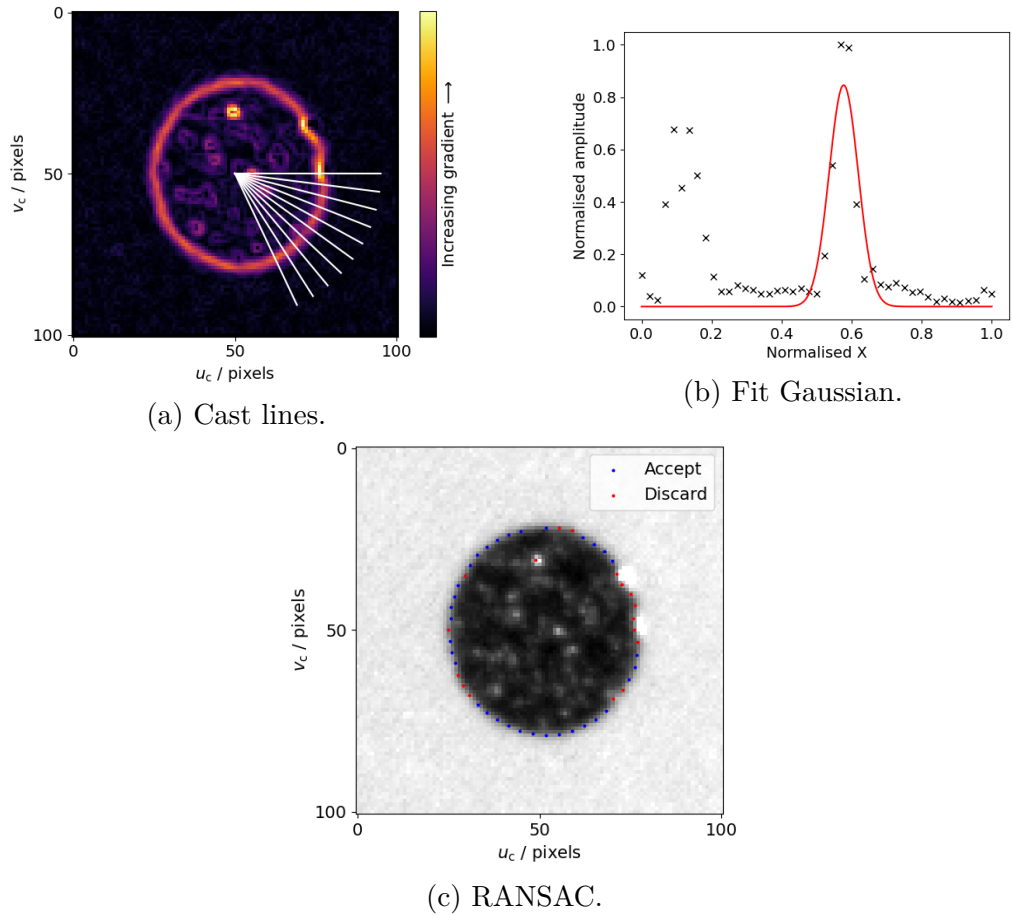


Figure 4.7: LSF approach to ellipse centre localisation refinement.

In Figure 4.7c there are some over-exposed regions of the image – the boundary estimations here do not correspond well with the real ellipse boundary. These erroneous boundary estimation points can have a significant effect on the ellipse fitting result and, in cases when there are many over-exposed regions lying on the ellipse boundary, can cause the ellipse fitting to fail entirely. The EfficientNet based approach given in Section 4 is designed to be much more robust to both noise and specular regions. Once the dot centres have been localised and refined, an extended version of the characterisation procedure presented by Zhang [17] is used. The characterisation results in a final camera model parameterised by intrinsic

parameters $[f_x, f_y, u_0, v_0, s]$, where (f_x, f_y) are the focal lengths, (u_0, v_0) are the principle point offsets and s is the skewness; and distortion parameters $[k_1, k_2, k_3, p_1, p_2, u_{dc}, v_{dc}]$ where (k_1, k_2, k_3) are the radial distortion coefficients, (p_1, p_2) are the tangential distortion coefficients, and (u_{dc}, v_{dc}) are the distortion centre coordinates.

4.4 Machine learning approach

The ML architecture used in this chapter is based on the EfficientNet family of models which was presented in Section 2.2.2.1. This architecture was chosen due to EfficientNet based models performing very well (ranked top three at the time of writing) on the benchmark ImageNet dataset [173] while having vastly fewer trainable parameters than other architectures [174].

Based on this architecture, two families of networks were presented by Tan and Le called EfficientNets [174] and EfficientNetsV2 [46] respectively. This family of models is created by stacking varying numbers of MBConv blocks together. In the case of EfficientNetV2, the early layers of the model eschew depthwise convolution for traditional convolution as it was shown to be more computationally efficient despite the increase in parameters compared to using depthwise separable convolution in the entire model. From these two model families (of EfficientNets, and EfficientNetV2s), nine models were selected for evaluation against the ellipse dataset as shown in Table 4.2.

These models, which were originally designed for classification, were modified with two linear output nodes used to regress the sub-pixel correction, which is applied to the OpenCV estimation. No transfer learning was employed and each model was initialised with randomised parameter

Model	Size (MB)	Parameters (millions)
B0	29	5.3
B5	118	30.6
B6	166	43.3
B7	256	66.7
V2B0	29	7.2
V2S	88	21.6
V2M	220	54.4
V2L	479	119

Table 4.2: EfficientNet models evaluated.

values. Each model was optimised using the Adam optimiser [42] and a LogCosh loss function was used for improved robustness against outliers. The dataset was split into training and validation sets with 5% of the data selected for validation. Training was conducted for 100 hours or 1000 Epochs, whichever ever occurred first. After training the model weights were restored to the epoch of the lowest mean absolute error as evaluated on the validation dataset. Training was conducted in parallel on the HPC CPU nodes with each task assigned 16 CPU cores and 128GB of RAM. Training time varied on model complexity with the smallest B0 model taking an average of 632 ms per 64 image batch, and the largest V2L model taking 6 s per batch. Once trained, the model can be used to refine the centre predictions given by OpenCV and these centre locations are then used in the same characterisation procedure that was outlined at the end of Section 4.3.

4.5 Characterisation results

4.5.1 Model training results

Table 4.3 shows the performance of each model evaluated against the validation set once the best performing parameter values have been restored.

Model	Metric			
	MAE (px)	MAPE (%)	Improvement (px)	Test Loss
B0	0.024	6.20	21.84	9.33E-04
B5	0.018	5.20	21.92	6.15E-04
B6	0.019	5.30	21.92	7.05E-04
B7	0.027	6.20	21.81	1.30E-03
V2B0	0.026	6.30	21.83	1.10E-03
V2S	0.021	6.40	21.92	8.11E-04
V2M	0.021	6.90	21.95	7.63E-04
V2L	0.040	8.40	21.51	2.66E-03

Table 4.3: Validation results for each model, best performing model shown in bold.

As is clear from Table 2, the EfficientNetB5 architecture was the best performing model in this test with a mean absolute error of 0.018 pixels which translates to a mean percentage error of 5.2%. As both the smaller B4 model and the larger B6 model had higher test mean absolute error (MAE), B5 was taken to be the optimal EfficientNet model size for this application. Figure 4.8 shows how the model metrics evolved over the training period for the EfficientNetB5 model.

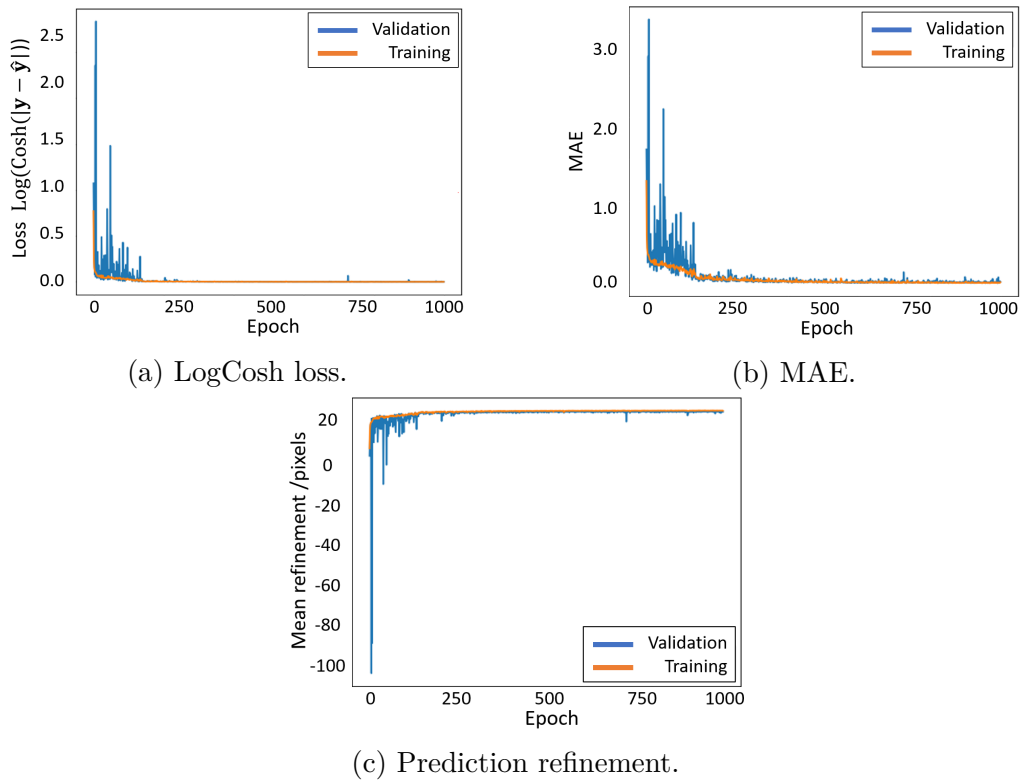


Figure 4.8: EfficientNetB5 metric evolution during the training period.

As can be seen in Figure 4.8a, the training result converged relatively quickly and stably. The minimum validation mean absolute error was 0.0183 pixels and occurred at epoch 992, therefore the model weights were restored to this point before the model was deployed into the characterisation pipeline. The full architecture of the model as implemented for this application is given in Appendix E.

4.5.2 Results on real characterisation data

The performance of both the LSF and ML method are compared by using their corresponding dot locations to characterise the camera in the DFP system, as was presented in Section 3.1.2.1. The two dot localisation methods are also compared against the OCV method – using the function `cv2::findCirclesGrid` from OpenCV 4.5.5 only [175]. The difference between the feature location as predicted by the given method (LSF, OCV or ML) and the same feature location when reprojected through the camera model back to the imaging plane is called the residual and this residual is minimised during the calibration using the Levenberg-Marquardt algorithm [33] as was discussed in Section 2.1.3. The final residual value is derived from the combination of errors in the dot grid artefact, errors in the dot locations and possible differences in local vs global minima [176] in the characterisation. Assuming the errors caused by the manufacture of the dot grid artefact to be constant, and assuming the Levenberg-Marquardt algorithm has converged to the global minimum, the final residual value can be considered a direct evaluation of the accuracy of the feature localisation method. The residual of the i^{th} point in the characterisation is given by $(\Delta x_i, \Delta y_i)$, where the characterisation uses N points, the feature localisation accuracy of each method are compared using the mean residual

magnitude (R),

$$R = \frac{1}{N} \sum_{i=1}^N \sqrt{\Delta x_i^2 + \Delta y_i^2}. \quad (4.1)$$

The LSF, ML and OCV methods are all be tested using two distinct characterisation datasets: a cooperative dataset and an uncooperative dataset. In the cooperative dataset, the characterisation data has been taken by minimising specular reflection components – completed by providing feedback to the operator during characterisation when there was excessive saturation of pixels. Pixel saturation was identified using an image of the dot grid under a projected image comprised of only pixel value 255. Pixels in the camera image that were at maximum pixel value were classified as saturated. In the uncooperative dataset, there is no limit on the position and orientation of the dot grid, and so some positions will be outside the nominal operating ranges of the LSF method. The cooperative dataset contains images of the calibration target from 18 positions while the uncooperative dataset contains images from 22 positions. The characterisation target contains 184 circular features leading to dataset sizes of 3312 ellipse images for the cooperative datasets and 4048 ellipse images in the uncooperative dataset. Figure 4.9 and Figure 4.10 show a number of examples from the cooperative and uncooperative datasets respectively.

Figure 4.11 shows the internal pixel distributions in each dataset as quantified by the distance from a “normal” value as determined using Otsu’s method [5].

As can be seen in Figure 4.11 the uncooperative dataset contains many more outlying pixels which represents an increased rate of specular artefacts. These specular artefacts are mainly caused when imaging at high angles relative to the imaging plane which were excluded from the cooperative dataset due to the limiting of imaging positions which produce

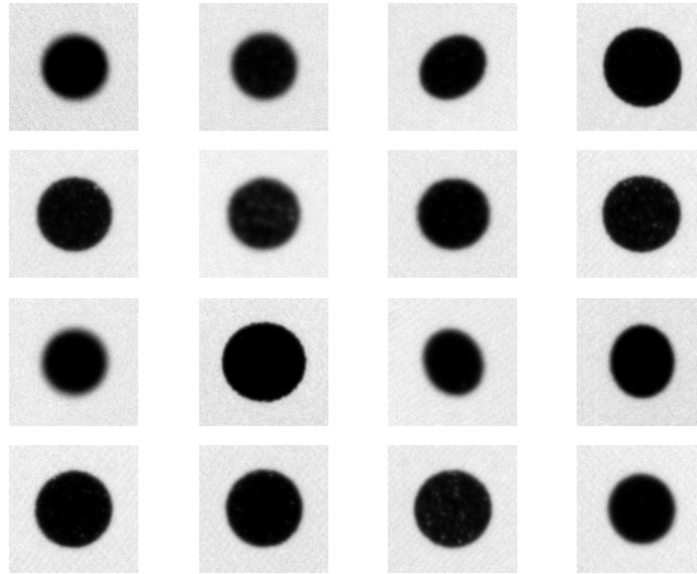


Figure 4.9: Example characterisation target images included in the cooperative dataset.

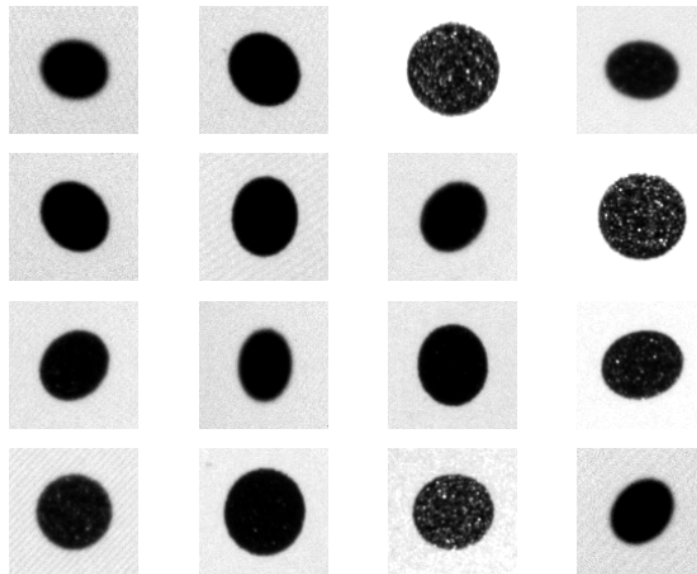


Figure 4.10: Example characterisation target images included in the uncooperative dataset.

saturated pixels as described previously.

Figure 4.12 shows the residual values for each method and dataset and the mean magnitude residuals are given in Table 4.4.

Using the values provided in Table 4.4, the benefit of using the two re-

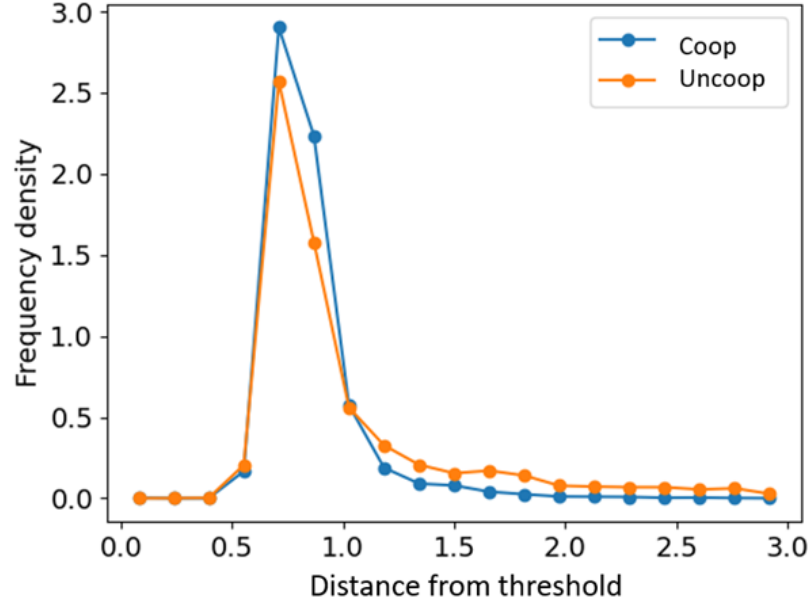


Figure 4.11: Pixel distributions internal to each ellipse from the cooperative and uncooperative datasets as a distance from a threshold determined by Otsu's method [5].

	Cooperative dataset (pixels)	Uncooperative dataset (pixels)
OCV	0.37	0.38
LSF	0.19	0.51
ML	0.18	0.19

Table 4.4: Mean residual magnitude.

finement methods can be quantified. In the case of the cooperative dataset it is clear that both the LSF method and ML method provide a considerable reduction in reprojection error. The percentage reduction in the mean residual magnitude is 49% in the case of the LSF method and 51% in the case of the ML method. However, when the dataset is uncooperative the LSF method in fact degrades the performance of the characterisation and the mean magnitude residual increases by 34%. In contrast, the ML method can still provide a reduction in the mean residual magnitude of 50%. Table 4.5 summarises the effect on parameter estimation of each method on each dataset.

It is hard to draw any direct conclusions from the estimated parameters

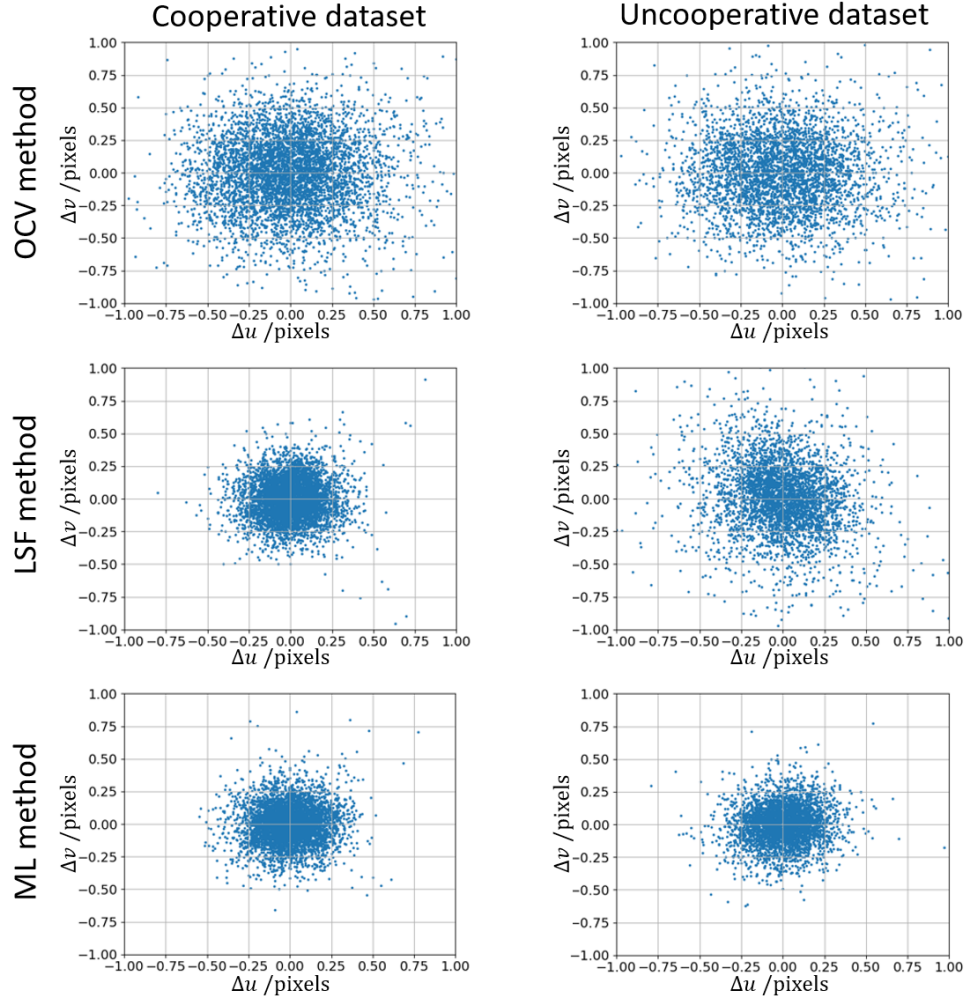


Figure 4.12: Distribution of residual errors in the reprojection of features for each characterisation method for each data set

Parameters		LSF		ML		OCV	
		Coop	Uncoop	Coop	Uncoop	Coop	Uncoop
Intrinsic parameters / pixels	f_x	8521	8547	8520	8533	8516	8545
	f_y	8521	8553	8521	8534	8516	8552
	s	-0.22	-0.12	-0.23	-0.91	-0.58	-1.17
	\mathbf{u}_0	2697	2627	2697	2656	2699	2626
	\mathbf{v}_0	2553	2568	2553	2552	2554	2567
Distortion coefficients / AU	\mathbf{k}_1	0.1199	0.1135	0.1374	-0.0388	0.1144	0.1182
	\mathbf{k}_2	0.0864	0.0579	0.0555	0.2145	0.0616	0.1244
	\mathbf{k}_3	0.0991	0.0910	0.1422	-0.6019	0.1292	0.1302
	\mathbf{p}_1	0.0838	0.1015	0.0698	-0.0013	0.0667	0.1265
	\mathbf{p}_2	0.1317	0.1142	0.0990	0.0001	0.1403	0.0790
	\mathbf{u}_{dc}	0.1351	0.0955	0.0642	-0.0145	0.1453	0.1402
	\mathbf{v}_{dc}	0.0665	0.0641	0.1130	0.0155	0.1468	0.0824

Table 4.5: Estimated parameters from each dataset.

shown in Table 4.5. due to the lack of any ground truth camera parameters and as such they are included here only for completeness. However, the greater performance of the ML estimated parameters as shown in Table

4.4 implies that the ML estimated parameters are likely to be closer to the true values than the LSF and OCV estimations.

4.6 Discussion of characterisation results

In this chapter, the sub-images were sampled from the captured calibration images at a scale of 101×101 pixels. This size was chosen as all imaging positions useful for the characterisation task produced features which fit within this size. If a different choice of characterisation target or camera were made then this size may need to be adjusted.

As was summarised in Table 4.4, the ML method can improve the feature localisation in both cooperative and uncooperative datasets. In comparison, the LSF method can reduce the localisation accuracy as evaluated by the mean residual magnitude by 34%. The decrease in localisation accuracy is, in part, due to the fact that the LSF method was unable to make reasonable estimations of all the features in the uncooperative dataset. Figure 4.13 shows some example failure cases of the LSF method.

The complete failure of the LSF method to fit an ellipse to the boundary can be seen in Figure 4.13a and Figure 4.13b: there has been no good estimation of the boundary points and RANSAC filtering becomes ineffective. However, the LSF method does not always fail in these conditions – Figure 4.13c and Figure 4.13d show reasonable approximations under similar conditions. This shows the LSF method to be unreliable under measurement conditions similar to those exhibited in Figure 4.13.

In comparison, as was shown, the ML method produced a characterisation result of similar quality to that of the cooperative dataset showing that the desired improvements to robustness have been achieved. This improvement

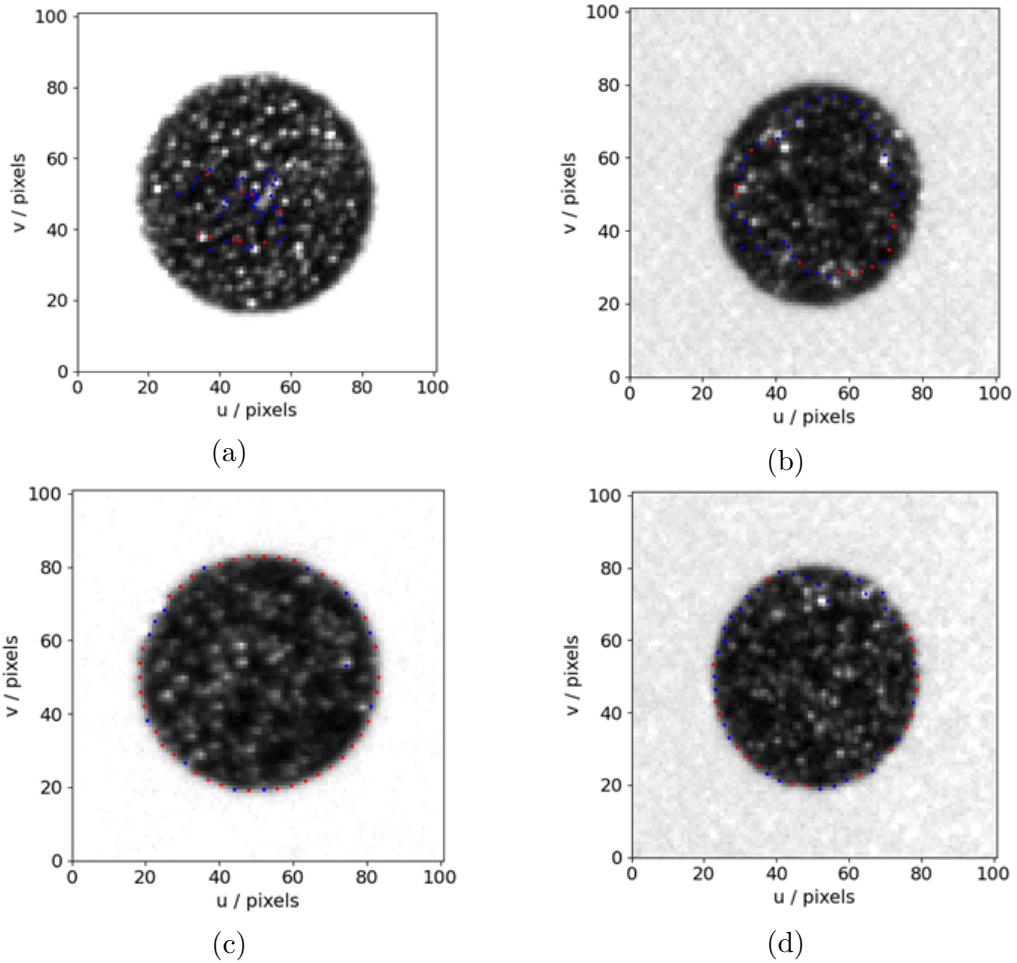


Figure 4.13: Showing some features from the uncooperative dataset, where blue dots have been discarded by the RANSAC algorithm and red dots have been kept as estimated boundary points. (a) and (b) show cases where ellipse fitting failed to produce a good outcome and (c) and (d) show cases where ellipse fitting was successful despite some outliers.

in robustness is visually evident in Figure 4.12, where the residual distribution of the ML method is shown to be similar under both characterisation conditions. It can also be seen in Figure 4.12 that the ML method clearly outperforms the pure OpenCV localisation in both cases with a reduction in the mean residual of approximately 50%.

The LSF method can be a good approach and provides high quality characterisation results but is highly dependent on a cooperative characterisation dataset. This is not always possible particularly in industrial setting where instruments need to be characterised in-situ by operators who may

not be experts. The requirement for a cooperative dataset also limits the number of feasible views which can be acquired as part of the characterisation dataset, high-quality characterisations require a high number of views across the measurement volumes at a range of angles relative to the imaging plane. This is evident when considering the impact on parameter estimation given in Table 4.5.

Improving robustness allows a greater range of views to be captured and can therefore improve the characterisation result which will, in turn, improve any measurement results captured by the system. It may be possible to improve the LSF method to handle a greater range of measurement conditions by fine-tuning hyper-parameters and using alternative filtering methods – the complexity would outweigh the benefit.

As was discussed in Section 2, steps were taken to make the training data representative of the real data, particularly by basing parameter distributions on those observed in the real datasets. These parameter distributions were extracted before splitting the real data into cooperative and uncooperative sets to ensure the full range of likely parameters were considered. The improvements gained when using the ML localisation refinement on real data, as shown in Figure 4.12, imply that the training data did indeed cover enough useful samples to successfully train the EfficientNet model to conduct the given task well enough to improve the estimation of camera parameters. Further statistical analysis into just how well the training data represents the real data could be conducted, and if the data could be made more representative it is likely that greater performance could be gained, although the magnitude of these further gains may be marginal.

4.7 Characterisation conclusions

Two methods for refining the localisation of characterisation targets were presented, one based on the line-spread function of the local image gradient (LSF method), and one based on an EfficientNet CNN (ML method). The two methods were compared to unrefined feature localisation (OCV method) by using two characterisation scenarios – a cooperative scenario with minimal over-exposures to produce clean ellipses for feature estimate, and an uncooperative scenario with high levels of specular reflection and over-exposure. It is shown that both refinement approaches lead to a reduction in the mean residual reprojection error magnitude over the OCV method of approximately 50%, with the ML method outperforming the LSF method by 2%. However, in the uncooperative scenario, use of the LSF method increases the mean residual magnitude by 34%. In contrast, the ML method maintains the 50% reduction in mean residual magnitude. This shows the EfficientNet has learned to provide localisation refinements which are robust to the adverse conditions present in the uncooperative characterisation image dataset. This improved robustness allows the characterisation dataset to include a larger range of imaging positions across the measurement volume, leading to improved parameter estimation and therefore higher quality measurement outcomes.

The contributions to science given by the work in this chapter can be summarised as: a novel application of a modified state of the art ML model to assist in camera characterisation which is shown to provide improved camera modelling over industry standard characterisation and improved robustness over state of the art characterisation refinement based on traditional image processing methods.

This approach represents a solution to the problem of intelligent camera characterisation which is more robust than the previous state of the art and

leads to higher quality camera models. This algorithm can now be inserted into the proposed fully automated and optimised measurement pipeline. The next chapter details how the now characterised camera parameters can be used to pre-plan optimised imaging locations based on the known geometry of the part to be measured.

Chapter 5

Automated and optimised view planning from CAD

This work was completed in collaboration with Hui Zhang who gathered the datasets and conducted the data processing presented herein; alongside Danny Sims-Waterhouse and Mohammed Isa who consulted on the journal article which was published as:

Zhang H, Eastwood J, Isa M A, Sims-Waterhouse D, Piano S, Leach R K 2020 Optimisation of camera positions for optical coordinate measurement based on visible point analysis *Precis. Eng.* **67** 178-188.

Figure 5.1 shows where the generation of the imaging strategy, referred to here as view planning, fits within the overall measurement pipeline.

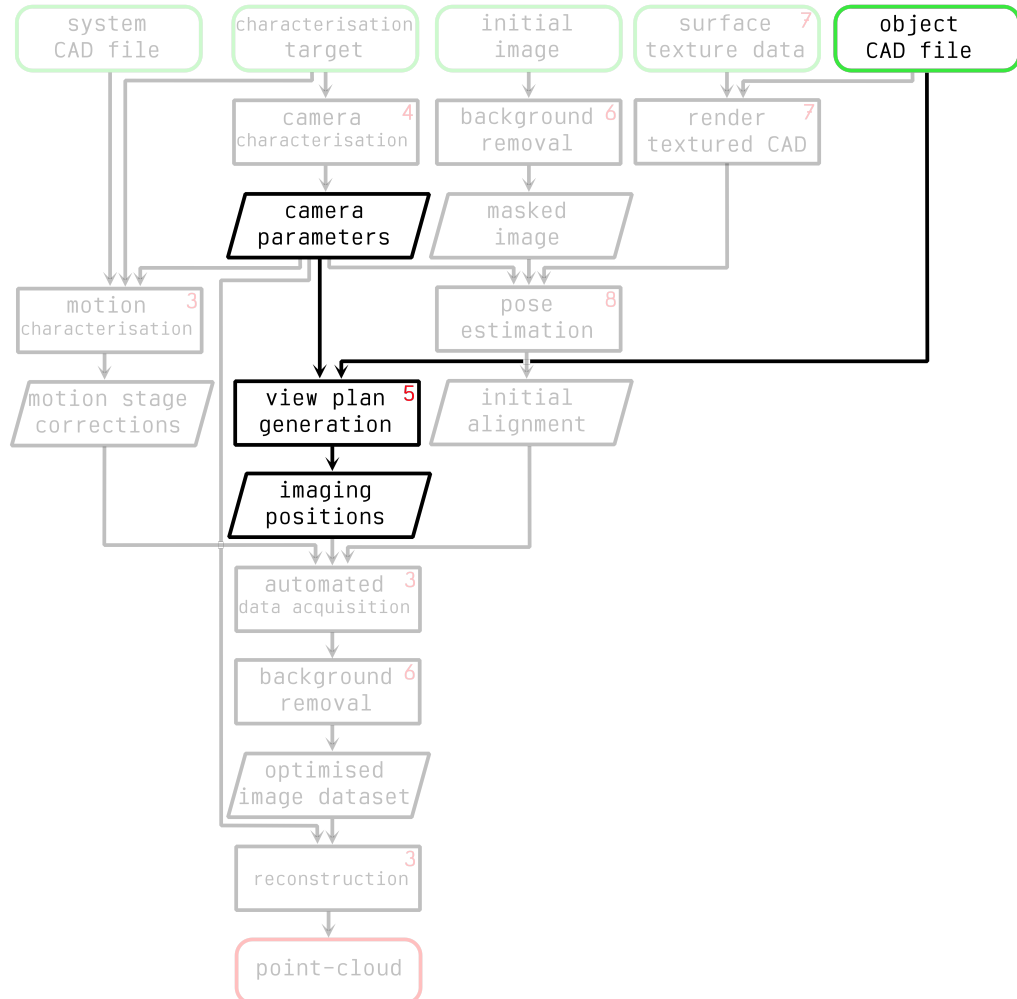


Figure 5.1: View planning shown within the overall proposed measurement pipeline.

In this chapter, a method for pre-optimising the view plan based on the geometry of a given part is described, along with an improved method for analysing visible points on a CAD surface from a given viewing location. In this case an optimal view plan minimises the number of imaging locations while maintaining reconstruction accuracy.

5.1 Introduction to view planning

Although optical coordinate measurement has taken a market share in industrial inspection, the approach lacks an established and automated method for inspection planning; something that is common for tactile CMMs [177, 178]. Camera positioning is one of the most significant issues that makes the use of optical CMSs restricted to experienced operators [179]. Optimal camera positioning is critical because the selected positions affect not only the image acquisition time and post processing of the data, but also the coverage of the object surfaces and the accuracy of the measurement. Published solutions to camera positioning in optical CMSs [179–183] are often application specific and the number of cameras is given in advance.

In this chapter, a novel technique is proposed for determining optimal camera positions based on a visible point analysis approach. A genetic algorithm (GA) is adopted to find the optimal combination of camera positions that results in high surface coverage of an object and maintains reconstruction quality while minimising the total number of cameras required.

5.2 Proposed view planning approach

For a given manufactured part, it is a complex task to directly identify how many images are necessary to fully cover the surface of the part, and the positions from which those images should be taken. A large number of images takes more time to acquire and is more computationally expensive to analyse. Furthermore, if the cameras are at unsuitable positions, accurate reconstruction of the object will not be possible. The optimisation of camera positions is greatly affected by the total number of camera positions

required. Hence, for accurate and fast 3D measurement by a multi-view optical CMS, it is necessary to first determine the number and positions of the images required to form an efficient network of camera viewpoints.

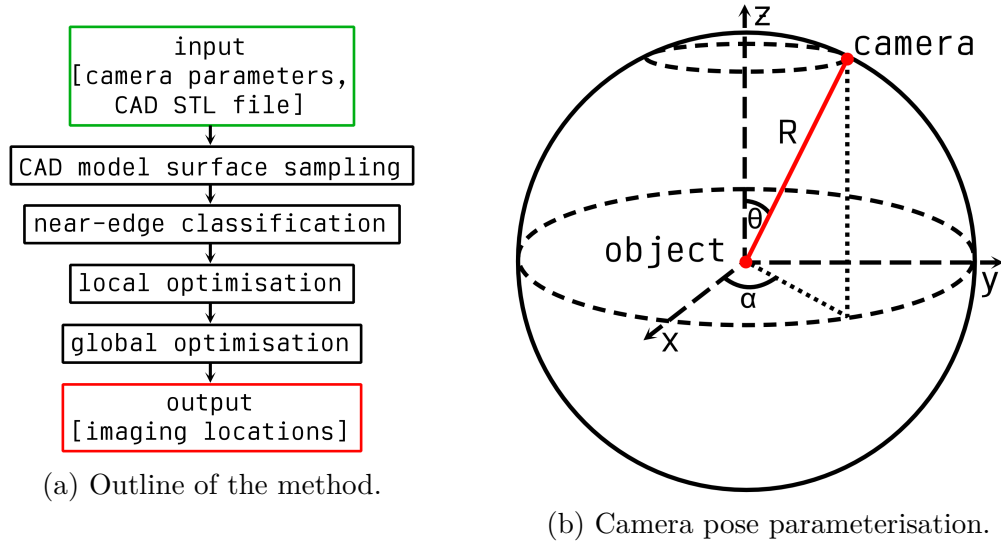


Figure 5.2: Outline of the view planning method and camera pose parameterisation .

The proposed method for camera positioning is illustrated in Figure 5.2a. Points on the surfaces of the CAD data for a given artefact are sampled to approximately 10000 points, the ‘sample points’ function in the open source software CloudCompare [160] was used to achieve this. A technique for analysing which of these discretised surface points are visible from a given camera position has been developed; this technique is used in the optimisation procedure as follows. An initial local optimisation determines the single camera position which provides the highest surface coverage. Following this, the locally optimised position is used as a seed location for a global optimisation of n camera positions. The global optimisation uses a GA (GAs were introduced in Section 2.2.4) to maximise an objective function which considers the surface coverage, image overlap and inter-camera angles. Next, the process of global optimisation is repeated with increasing

values of n until a threshold objective function value is achieved. These optimisation procedures are described in detail in Section 5.2.2. The method is validated using the MMT system which was presented in Figure 3.1 and is validated on the four simple artefacts which were shown in Figure 3.10. To determine the potential camera poses, the working area is parameterised as shown in Figure 5.2b. Two variable parameters are used to represent the camera pose in 3D: the azimuth angle ($0^\circ \leq \alpha < 360^\circ$) in the x, y plane, and the elevation angle ($0^\circ \leq \theta < 180^\circ$) with respect to the z axis. The principal axes of the cameras are set to be convergent at the object centre. A third parameter, the radius from the object centre R , must also be set. R is fixed such that the cameras are all placed sufficiently far from the object centre such that increasing the value of R has a negligible impact on the number of visible points. R is restricted to this point so that the angles produced by the optimisation do not depend on the field of view or resolution of the specific camera used. In the case of the example artefacts, the minimum radius at which the maximum number of surface points can be seen was found at $R = 500$ mm, therefore, R is fixed to this value for the rest of this chapter.

5.2.1 Visible points analysis

There are several techniques to find which points on an object's surface are visible, given the camera's viewpoint relative to that object. These techniques can be classified as surface triangulation-based techniques, voxel-based techniques and point-based techniques [184]. Surface point based approaches are highly efficient but have been shown to have poor performance at areas of high local curvature, such as sharp edges [185]. Therefore,

first any surface points near sharp edges are classified; this allows the use of a more accurate but less efficient triangulation approach on the near-edge points, while using the more efficient point-based technique on the remaining surface points.

5.2.1.1 Triangulation-based technique

In the triangulation-based technique, the surfaces of the object are discretised into a set of tessellated triangles. The camera-to-object surface distance is computed by calculating the minimum distance from the camera centre O_c to the point P_i on the surface. For triangle vertices V_0, V_1, V_2 , the ray-triangle intersection formulation is given by [186],

$$\begin{bmatrix} P_{ix} & V_{0x} - V_{1x} & V_{0x} - V_{2x} \\ P_{iy} & V_{0y} - V_{1y} & V_{0y} - V_{2y} \\ P_{iz} & V_{0z} - V_{1z} & V_{0z} - V_{2z} \end{bmatrix} \begin{bmatrix} D_i \\ u \\ v \end{bmatrix} = \begin{bmatrix} V_{0x} - O_{cx} \\ V_{0y} - O_{cy} \\ V_{0z} - O_{cz} \end{bmatrix} \quad (5.1)$$

where D_i is the camera-to-object distance and (u, v) are the barycentric coordinates of the intersection point. Within a given triangle, the point with the smallest distance D_i is classified as visible while all other points are classified as not-visible. The triangulation intersection approach is effective but results in high computational costs, as the intersections need to be evaluated on all the points over all the triangles. The triangulation-based approach is, therefore, not efficient as the order of growth of the algorithm is $O(N_p \cdot N_\Delta)$, where N_p is the number of points and N_Δ is the number of triangles.

5.2.1.2 Point-based technique

A point-based technique, referred to as hidden point removal (HPR) [187, 188], is widely used in the areas of computer vision and graphics. HPR is composed of two steps: point inversion and convex hull computation. Point inversion reflects all points inside a bounding sphere to the outside of that sphere. The coordinate system which defines this inversion has its origin at the camera origin. The inversion can be defined mathematically as,

$$\hat{p}_i = F(p_i) = p_i + 2 \cdot (R_s - \|P_i\|) \cdot \frac{p_i}{\|p_i\|} \quad (5.2)$$

where R_s is the sphere radius [187] and \hat{p}_i is the inverted coordinate corresponding to p_i . Points which are visible from a camera position, when transformed, now lie on the convex hull of the inverted point cloud. The convex hull calculation constructs a non-ambiguous representation of the convex hull of the inverted point cloud, thus allowing the visible points to be categorised. The order of growth of the convex hull calculation is $O(N_p \cdot \log(N_p))$ which is a much slower rate of growth than that of the triangulation approach and could be further improved through parallel computing.

Due to its higher efficiency when compared to the triangulation approach, the HPR technique is suitable for denser point clouds. However, disadvantages of the HPR technique are that it is sensitive to noise in the point cloud [188] and misclassification errors are expected to occur around regions of high local curvature [185]. To reduce the misclassification around the edges, an enhanced visible points analysis technique is proposed which combines the triangulation and HPR techniques and is described in the following section.

5.2.1.3 Enhanced visible point analysis technique

Since HPR misclassifies points around high-curvature areas, such as sharp edges, it is preferable that the points are first classified into two sets: the set of near-edge points P_e or the set of points remote from an edge P_o , which can be expressed by,

$$P_e = \{p_i \mid D(p_i) < D_{th}\}, p_i \in P, \quad (5.3)$$

$$P_o = \{p_i \mid p_i \notin P_e\}, p_i \in P, \quad (5.4)$$

where P is the set of all surface points, D_{th} is a distance threshold and $D(p_i)$ is the minimum distance from the point p_i to an edge. In standard tessellation language (STL) models, the CAD model is represented by a set of triangular faces. Edges can, therefore, be classified along triangular boundaries where the neighbouring triangular faces have large differences in the directions of their surface normals. Once the edges are located, all the surface points can be filtered by their Euclidean distance to the nearest edge and, therefore, categorised into either P_e or P_o according to Equation 5.3. Points are then evaluated for visibility using either HPR if they are in P_e or by the triangulation-based intersection technique if they are in P_o . This enhanced visible point analysis pipeline is shown in Figure 5.3.

As the distance threshold D_{th} increases, the proportion of points classified as near-edge increases. To decide at what value to set the distance threshold, D_{th} is varied from 0.01 mm to 5 mm. Figure 5.4 shows how changing D_{th} changes the resulting classification of surface points.

It can be seen that there is some variation between the curves gener-

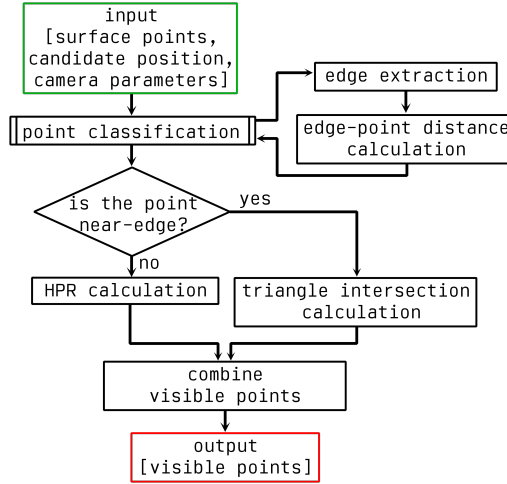


Figure 5.3: Enhanced visible points analysis technique.

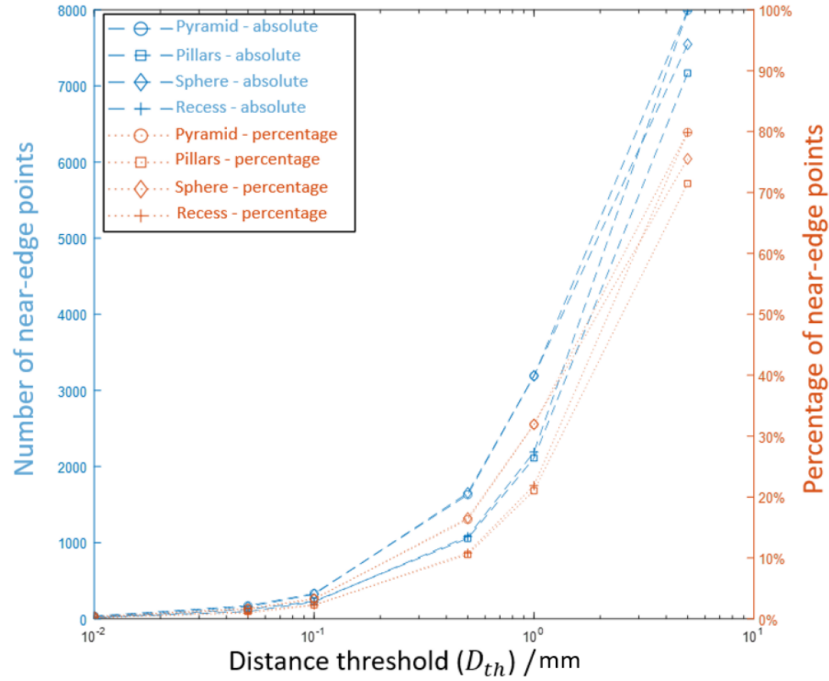


Figure 5.4: Near-edge point classification based on different threshold values for four objects.

ated for the four objects. This variation between the four objects is due to the differing amounts and distributions of edges present in each object. However, the overall contours of the curves for each object is similar, with a clear change in gradient at 0.1 mm. As a result of this clear change in gradient, a distance threshold value of $D_{th} = 0.1i$ mm is chosen. In the case of the four test artefacts, when using $D_{th} = 0.1$ mm, around 5% of the

surface points are considered near-edge points, i.e. classified into the set P_e . In the case of a purely freeform object with no sharp edges, this approach would not be required. However, as the relative edge density of the part increases, selecting the threshold value through a convergence criteria on the gradient of the slope as shown in Figure 5.4 will provide a suitable value of D_{th} .

Using this combined, enhanced technique of visible point analysis proffers an improved divide-and-conquer solution to determining the set of surface points which are visible. Compared to pure HPR analysis, the enhanced technique results in a reduction of misclassified points from 3% of the total points to 1% of the total points. Misclassification by pure HPR could be due to both visible points being incorrectly classified as not-visible, and not-visible points being misclassified as visible. Figure 5.5 shows the points which are misclassified when using pure HPR but are correctly classified when using enhanced visible point analysis as proposed.

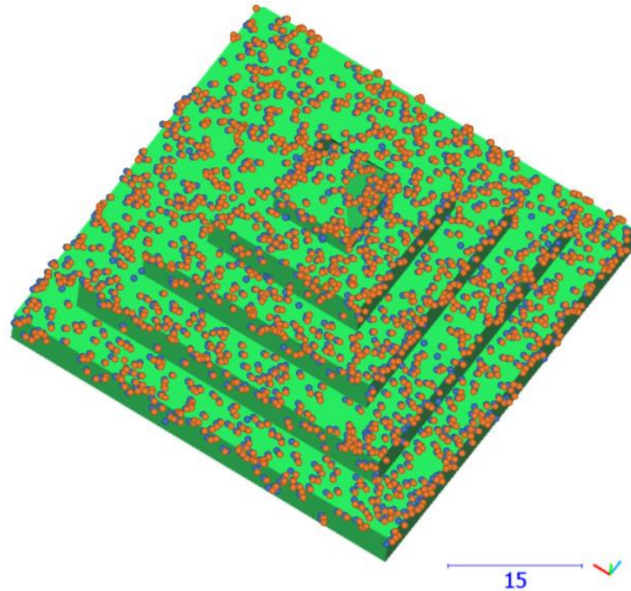


Figure 5.5: Misclassified points when using HPR which are correctly classified when using the proposed enhanced visible point analysis. Visible points classified as invisible are shown in blue, and invisible points classified as visible are shown in orange. Scale is in millimetres.

Case	Triangulation		HPR		Enhanced HPR		
	Time /s	Visible points	Time /s	Visible points	Time /s	Visible points	Reduction in misclassified points /%
1	2.93	3151	0.09	3052	0.18	3054	2.02
2	2.93	4579	0.10	4605	0.20	4590	57.69
3	2.97	4593	0.10	4619	0.20	4614	19.23
4	3.09	4600	0.10	4642	0.19	4634	19.05
5	3.03	4645	0.10	4577	0.18	4574	-4.41
6	3.19	4575	0.09	4615	0.22	4608	17.50
7	3.07	3751	0.06	3758	0.16	3757	14.29
8	2.95	3540	0.06	3548	0.17	3546	25.00
9	2.97	4687	0.09	4706	0.19	4696	52.63
10	3.01	4705	0.10	4716	0.20	4711	45.45

Table 5.1: Performance comparison of three visible point analysis methods: triangulation-based, HPR and enhanced HPR. Including reduction in misclassified points when using enhanced HPR.

Table 5.1 compares the performance of the three possible approaches. It is clear that HPR is the most efficient and the triangulation-based method the least efficient. This makes sense as the order of growth of the triangulation based approach is quadratic which is a much faster growth than for HPR, as was discussed in Section 5.2.1.2. It can further be seen in Table 5.1 that the enhanced approach, while taking more time than pure HPR, is an order of magnitude faster than the triangulation approach. The enhanced HPR offers between 2% to 57% reduction in misclassified points over HPR; this creates a reasonable trade off between algorithmic efficiency and performance.

5.2.2 Optimisation scheme

Utilising the visible point analysis technique described above, an optimisation scheme for determining optimal camera positions has been developed. Former work in this area has assumed a fixed number of camera views [60] whereas in the proposed scheme the number of views can also be varied and optimised. Firstly, an initial camera position is found through a local optimisation process. Additional cameras are then added to a global op-

timisation process until an objective function threshold is achieved. The basic outline of this procedure is shown in Figure 5.6.

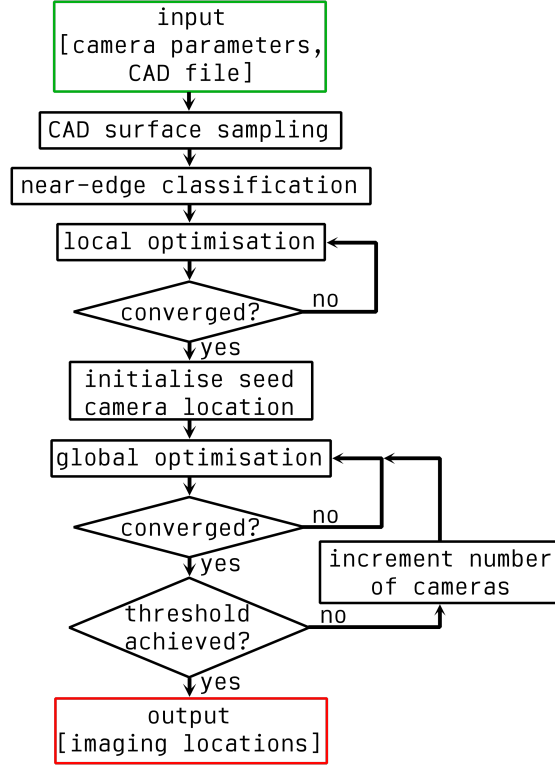


Figure 5.6: View optimisation scheme.

5.2.2.1 Local optimisation

In a first step, the optimum position of a single camera based on surface coverage alone is determined. The locally optimised position will then be used as a seed location from which to perform the global optimisation. To determine this camera location the following objective function is maximised,

$$F_{binary} = \sum_{k=1}^N [\text{vis}(p_k)], \quad (5.5)$$

where N is the total number of surface points, p_k is the k^{th} surface point, and $\text{vis}(p_k)$ returns one if the point is visible and zero otherwise (using the previously described analysis). While a GA could be employed here, a

simple search algorithm can be used as the search space for a single camera is well constrained. The results of the local optimisation process for the four simple artefacts are shown in Figure 5.7.

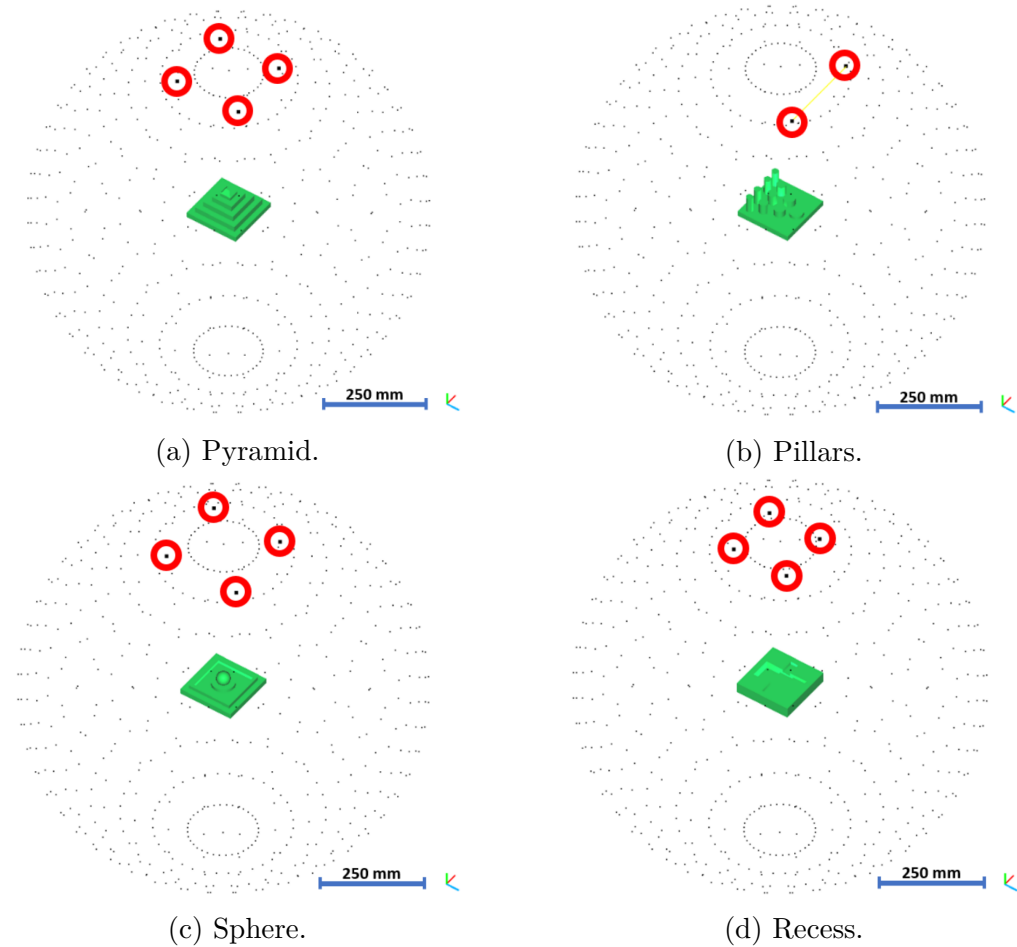


Figure 5.7: Camera positions from which the maximum number of surface points are visible for each artefact.

As can be seen in Figure 5.7b, the pillars artefact has two equally optimal camera positions due to its two-fold rotational symmetry, while the remaining artefacts have four equally optimal positions due to their four-fold symmetry. In the case where multiple positions are equally optimal, one of these positions can be chosen arbitrarily. Excluding the pillars artefact, the optimum camera positions in each case are aligned with the four corners of that artefact. Furthermore, it can be inferred that the optimum

camera elevation angle depends to the relative height of the object, at 12° for the shallow recess artefact and 18° for the more prominent pyramid object.

5.2.2.2 Global optimisation

After the seed camera location is found through local optimisation, a global optimisation procedure is conducted from this location. The global optimisation process aims to optimise for two criteria: that each surface point is seen by a minimum of four cameras and an inter-camera convergence angle of 90° for all cameras, at all surface points. Attempting to view each surface point from four camera locations maximises surface coverage while promoting overlap between images. Additionally, promoting a camera convergence angle of 90° has been shown in previous work to provide the highest reconstruction accuracies [179]. In contrast to the local optimisation procedure, now multiple camera images are considered at once. The global objective function is given by,

$$F_{global} = \frac{1}{4N} \left(\omega \sum_{i=1}^n \sum_{k=1}^N [\cos(\gamma_{ik})] + \frac{\omega - 1}{2(n - 1)} \sum_{i=1}^n \sum_{j=i+1}^n \sum_{k=1}^N [\sin(\beta_{ijk})] \right), \quad (5.6)$$

where ω is a weighting coefficient, n is the total number of camera images, γ_{ik} the angle at the intersection between a ray cast from camera c_i and the surface normal at surface point p_k and β_{ijk} is the triangulation angle between the ray-lines projected from the pair of cameras c_i and c_j which intersect at surface point p_k . In the case where there are more than four cameras in the optimisation, the value of $\sum_{k=1}^N [\cos(\gamma_{ik})]$ may exceed four – it is, however, capped at this value. Capping this value ensures that it is more optimal for every point to be seen by a few cameras, than for a single

point to be seen by many cameras. Capping this value at four implies the optimal score is given when every surface point is viewed from at least four camera positions. The first half of the objective function (which considers γ_{ik}) is similar to F_{binary} but has been adapted to loop over all surface points for all camera positions. It also now considers not just if a point is visible but the cosine of γ_{ik} for all visible surface points from a given view. This gives a higher weighting to views which are orthogonal to surface faces, which is desirable for high quality reconstructions.

The global optimisation procedure is conducted as follows. The objective function is maximised for four cameras by a GA, these four cameras are initialised using the seed position found in the local optimisation procedure. When this optimisation is complete, if the objective function has not reached a 95% threshold value, then an additional two cameras are inserted and the optimisation is reapplied. A convergence threshold of 95% was selected as it provides similar reconstruction results to a higher threshold value at a much smaller computational cost. As the number of cameras n in the optimisation increases, the number of inter-camera angles β_{ijk} scales with $\frac{1}{2}(n^2 - n)$. To prevent the inter-camera component dominating F_{global} , the value of the weighting coefficient is, therefore, set by,

$$\frac{\omega}{1 - \omega} = \frac{n - 1}{2}. \quad (5.7)$$

In this implementation, as was noted in Section 3.3.3, the default MATLAB GA [170] was used with the following modifications: a population size of 500, a cross-over rate of 80% and a 5% elite population classification rate. To ease the computational load, the algorithm uses flexible parameters that allow for a small population with broad tolerances at low numbers of camera positions and larger populations with narrower tolerances at high numbers of camera positions. Figure 5.8 shows the optimisation results for

the four artefacts.

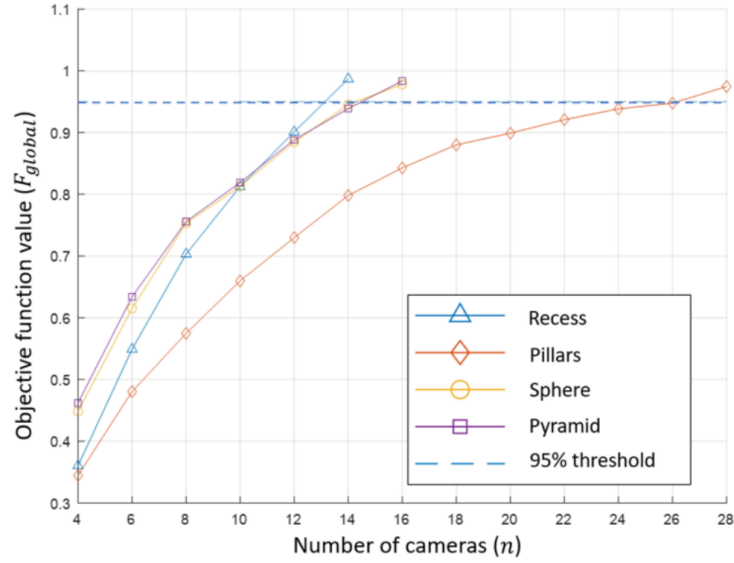


Figure 5.8: Global optimisation for the four test artefacts showing the number of camera views required to pass the objective function threshold.

Convergence to the threshold criteria is achieved with a different number of cameras for each object, the pyramid and sphere require fourteen images, the recess only twelve images and the pillars require twenty-two images. The differing minimum number of camera viewpoints required corresponds to the relative complexity of each artefact and the number of occlusions due to that artefact's features. The time to run the GA varies with the number of cameras in the simulation, the computer hardware, and the specific implementation details of the algorithm. In this case using a Lenovo PC (Lenovo PC Think Center M910s i3-7100 3.9 GHz, 8G RAM, 1T HDD) for twelve camera positions, the GA took around two hours for optimisation, while for twenty camera positions, the GA took around seven hours for optimisation. These times are likely to be significantly reduced through a parallel implementation and faster hardware.

5.3 View planning results

Using the MMT system shown in Section 3.1.1.1, images at optimised positions were captured, then reconstructed using Agisoft Metashape [30] to create dense point clouds. The point clouds were registered to their reference models using ICP [160]. Lastly, the deviations of the points from the reference models were analysed to assess the quality of the reconstructions. To acquire reference models of the artefacts, the GOM system introduced in Section 3.1.1.2 was used as an industrial comparison and the CMM introduced in Section 3.1.6 was employed to create ground truth measurements. These measured reference models of the manufactured artefacts, rather than CAD models, are used for comparison because the manufacturing process of the artefacts can contribute significant shape changes relative to the intended design model. As such, when comparing against a CAD model it would be impossible to tell if a deviation was due to measurement error or manufacturing error. The measurements using the GOM system are acquired from eight different positions with field of view $300\text{ mm} \times 200\text{ mm}$, probing size error 0.006 mm , and sphere spacing error 0.020 mm (as quoted by the manufacturer [189]). For the CMM results in Section 5.3.3, the contact probe calibration results are given in Appendix B.

5.3.1 Photogrammetry using optimised camera positions

The optimisation process proposed in Section 5.2.2 was implemented. Two measurements were taken of each artefact, one set with twelve optimised camera images and one set with eighteen images. These numbers were chosen as they lie on the objective function threshold as shown in Figure

5.8. Images captured at the optimised positions were used to reconstruct textured dense point clouds of the artefacts through the photogrammetric pipeline of Metashape. Figure 5.9a shows the dense and textured point cloud of the pyramid artefact using twelve images. A qualitative improvement in the point cloud can be observed when the number of images was increased to eighteen, as shown in Figure 5.9b. These reconstruction results align well with the high value of the objective function for the artefacts when using eighteen camera positions, as shown in Figure 5.8.

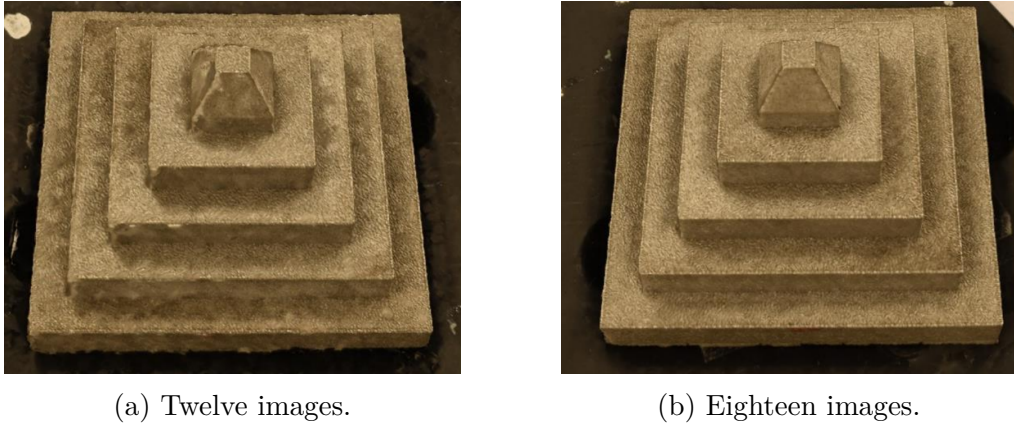


Figure 5.9: Dense colourised reconstructions for the pyramid artefact using the proposed optimised camera positions.

Further to the qualitative analysis, the difference in reconstruction accuracies for the twelve and eighteen image reconstructions are given in Figure 5.10. Reconstructed points using twelve and eighteen optimised images were compared with the reference triangular-mesh model obtained from the commercial GOM system.

To remove possible outliers, only point to mesh (PTM) distances within four standard deviations of the mean are shown. For the reconstruction using twelve images, shown in Figure 5.10a, significant discrepancies are observed over the corners, some upper surfaces and the vertical walls of the pyramid. However, when using eighteen optimised images, as shown

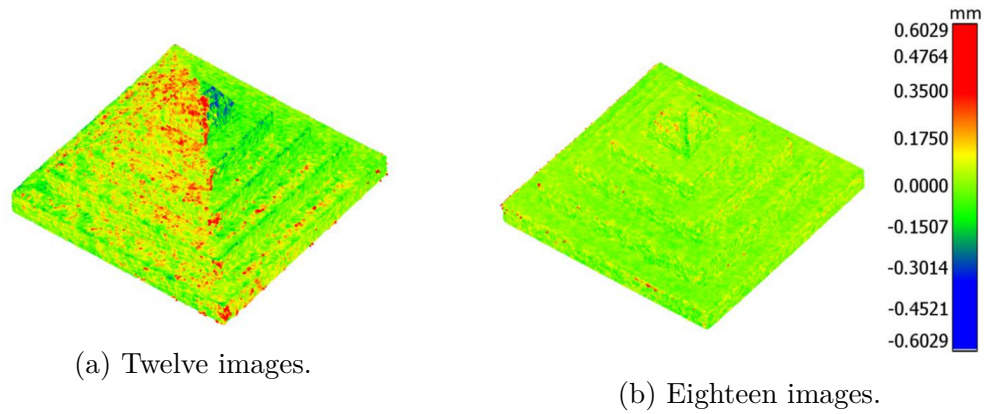


Figure 5.10: Deviations in measurement results from the reference measurement given by the GOM system.

in Figure 5.10b, the discrepancies are much diminished around the corners and are barely observed on most flat surfaces of the object. In addition, the root mean squared (RMS) value of the PTM distances is 0.101 mm for twelve camera positions compared to 0.052 mm for eighteen camera positions. The loss in quality when using twelve optimised images rather than eighteen images as suggests that the optimisation threshold set in Figure 5.8 as at an appropriate level as using fewer camera positions than suggested does, in fact, lead to a decrease in photogrammetric reconstruction quality.

5.3.2 Comparison of equally spaced and optimised camera positions

To assess the effectiveness of the proposed camera positioning technique, reconstructions using the optimised camera positions were compared with reconstructions using an equal number of camera images, positioned evenly around the artefact. The use of camera positions equally spaced on a circle surrounding an object is a common practice in small-scale photogrammetry [190–193]. To enable this comparison, throughout Sections 5.3.2 and

5.3.3 the elevation angle is fixed at 35° and only the azimuth angles are varied.

Reference measurements obtained by the GOM system are used to evaluate the deviation of the point clouds for the pyramid and pillar artefacts. The standard deviations of the PTM distances of the reconstructions are shown in Figure 5.11 over a range of ten to thirty total camera positions. The evaluation of the deviations is repeated on five sets of measurement data, the variations in the standard deviation of the repeated measurements are shown by error bars. Generally, the standard deviations of the pillar artefact are higher than the pyramid artefact, likely because of the greater self-occlusion caused by pillars. When the number of camera positions is less than twenty, the proposed technique performs with clearly lower deviation than the equally distributed camera positions. Additionally, the error bars are, on average, wider for the equally spaced camera positions, indicating improved stability with the proposed technique. When the number of camera positions is more than twenty-two, the two techniques perform comparably. The similarity in performance above twenty-two camera locations is because a high number of camera positions allows most regions on the artefact's surface to be sufficiently covered without optimisation.

As can be seen in Figure 5.11b, the reconstruction with sixteen optimised image positions performs similarly to a reconstruction using twenty-two un-optimised image positions. Table 5.2 shows a comparison in the performance of the reconstruction algorithm for the reconstructions shown in Figure 5.11b.

As can be seen, reconstruction using sixteen optimised image positions takes much less time to generate both depth maps and the dense point

5.3. VIEW PLANNING RESULTS

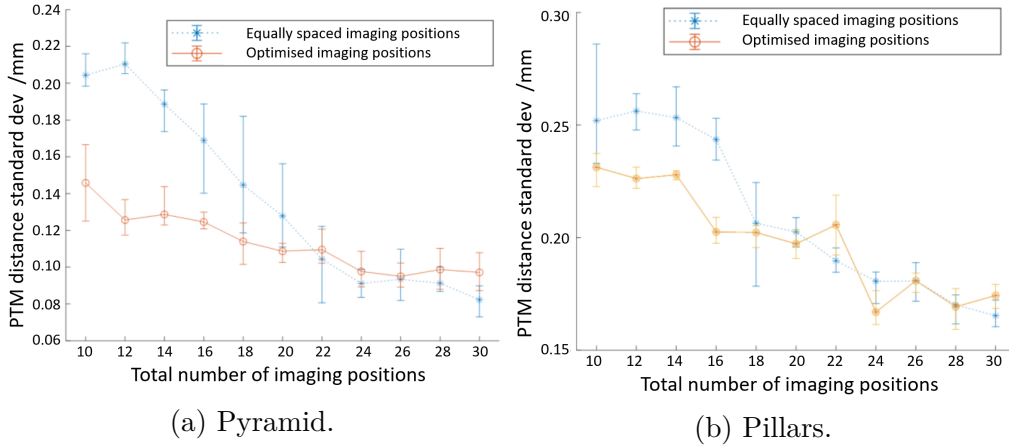


Figure 5.11: Comparison of the standard deviation in PTM distances for both optimised and equally spaced camera imaging positions.

Images /N	Optimised locations			Equally spaced		
	Time to generate depth maps /s	Time to generate point cloud /s	Points /N	Time to generate depth maps /s	Time to generate point cloud /s	Points /N
10	11	26	1 376 044	11	20	1 186 167
12	15	33	1 385 261	32	28	1 374 713
14	26	44	1 357 144	73	38	1 376 383
16	27	62	1 412 859	51	48	1 305 737
18	22	83	1 423 958	57	66	1 275 405
20	29	106	1 357 347	44	84	1 249 084
22	36	146	1 358 505	46	99	1 210 973
24	39	167	1 368 261	53	209	1 435 513
26	53	224	1 347 636	63	108	1 207 208
28	56	283	1 346 136	67	192	1 153 688
30	95	439	1 498 454	65	265	1 450 326

Table 5.2: Comparison of reconstruction performance for equally spaced and optimised camera locations. Shown in bold are the values for sixteen optimised image positions and twenty-two un-optimised image positions which were shown to perform similarly in Figure 5.11b

cloud, while producing 200000 more points, than using twenty-two un-optimised image positions. The proposed method takes consistently less time to produce depth maps than the un-optimised approach and produces point clouds with consistently many more points.

To visually compare the analysis of the measured point clouds, the deviations of point clouds of the pyramid and pillar artefacts obtained from the two sets of camera positions are juxtaposed in Figure 5.12. Two sets of fourteen camera positions, one equally spaced and the other optimised, are used for reconstruction in this case.

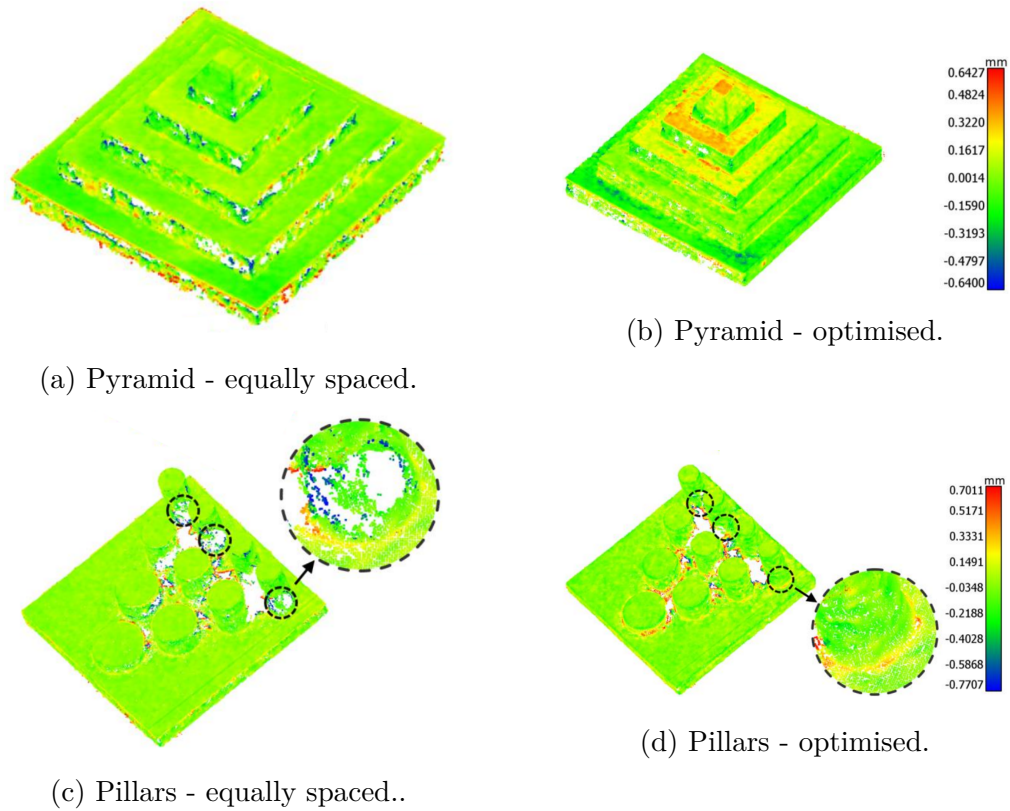


Figure 5.12: Comparison of the PTM deviations of the pyramid and pillar point clouds from GOM results.

It can be seen from Figure 5.12 that the deviations are lower in the optimised cases, especially in the vertical faces. Furthermore, the coverage of the surfaces is far more complete when using the optimised positions; this is seen optimised on the vertical walls of the pyramid and in the inset images at the base of the pillars. Ultimately, using the optimised camera positions results in more accurate, complete and stable reconstruction when compared to using the same number of equally spaced images; and furthermore, requires fewer total images to achieve accurate reconstruction.

5.3.3 Comparison with CMM data

The 3D point clouds of the pyramid artefact are further compared with measurements carried out using the CMM. Comparison to the CMM measurement is carried out for reconstructions resulting from both the optimised and the equally spaced camera positions. In Figure 5.13, a point cloud generated by the CMM is compared with meshes obtained from Metashape photogrammetric reconstruction using eighteen camera positions. A probe tip diameter of 1 mm was used in scanning mode to measure contours on the surface of the pyramid artefact. Points were sampled at 10 μm along each contour and the spacing between the contours was 200 μm . The gaps seen in the two measurements are due to regions that were omitted by the CMM path program to avoid potential collision of the part with the stem of the stylus.

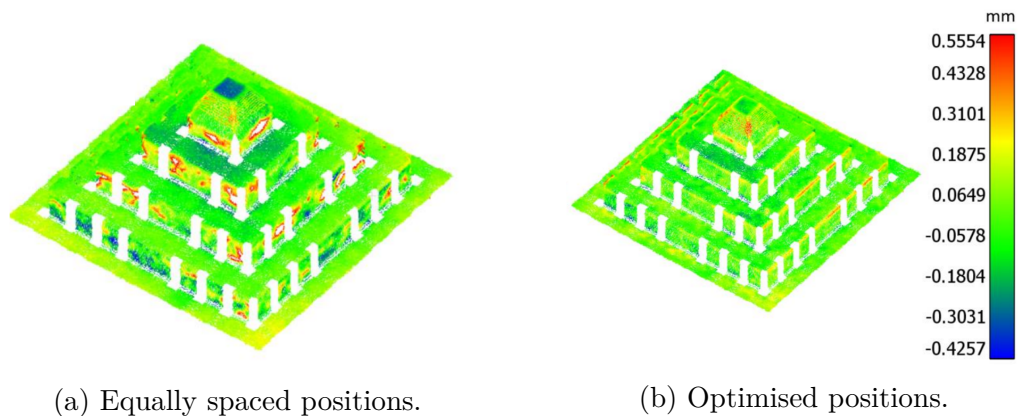


Figure 5.13: PTM distances for a CMM comparison of the reconstructions of the pyramid artefact.

The distributions of the PTM distances are observed to be consistent with the PTM distances from the GOM system. Points omitted on the vertical walls of the pyramid in Figure 5.13a have distances that exceed the range on the colour scale. In addition, the RMS PTM distances reduce from 0.145 mm to 0.095 mm when using the optimised camera positions

rather than the equally spaced positions.

In general, using both contact and non-contact reference measurement techniques, the point clouds reconstructed from images captured at the optimised camera positions are shown to be more accurate and complete. This work shows that using an initial CAD model of an object, the combinations of camera positions can be optimised to improve optical 3D coordinate measurements.

5.4 View planning conclusions

A technique for the optimisation of camera positions for optical coordinate measurement is presented in this chapter. Camera positions used in optical coordinate measurements are determined based on visible point analysis and global optimisation. From an object's computer aided design model, the surfaces are discretised into points. An enhanced visible point analysis technique is derived, and used to determine which of these surface points are visible from a given camera position. The enhanced visible point analysis technique adopts a combination of the use of a hidden point removal algorithm for the majority of the surface points, and a triangulation-based intersection algorithm for the near-edge points. The enhanced approach is used to decrease the misclassification of visible points. The optimisation technique determines not only the optimal camera positions for a given number of total camera positions, but also the minimum number of total camera positions required to meet a threshold criterion. Iterating the optimisation for increasing numbers of camera positions allows the minimum required number of camera positions to be determined for a given object, allowing more efficient computation during reconstruction.

A proposed objective function which considers the visible points, as well as

image overlap and intercamera angles, is presented. A genetic algorithm is employed for global optimisation of the camera positions with respect to this objective function.

Comparisons of results acquired using the proposed technique with results from equally spaced camera positions are conducted. The quality of these reconstructions is analysed by comparison with an industrial optical fringe projection instrument and a tactile coordinate measurement machine. It is shown that using the optimised positions improves the coverage of an object's surface and produces point clouds with lower point-to-mesh distances when compared to the reference measurements. Furthermore, it is demonstrated that a measurement using a lower number of optimised camera positions performs as well as, or in some cases better than, a measurement using a higher number of un-optimised camera positions. By enabling the use of fewer images while maintaining reconstruction quality, measurement time and data processing time can both be reduced using the optimised camera positions. Although the proposed technique is shown to be beneficial, there are still some issues that require further investigation; among them, improving the time for conducting the optimisation and investigating the effects of non-uniform lighting on visibility.

The contributions to science given by the work in this chapter can be summarised as: a new method for evaluating visible points on a surface from a given view which improves on the state of the art by operating faster than triangle intersection methods while misclassifying fewer points than HPR. Also, a novel approach to view planning which improves on the state of the art by being general across object geometries and creating higher quality reconstructions at fewer imaging positions than current industry practice. In the next chapter an alternate approach to optimising the measurement processing time and measurement result is proposed. In Chapter 7 and Chapter 8 a method is developed for establishing the initial location of

the part in the measurement volume enabling the use of these optimised positions in an autoamated data acquisition without any prior knowledge of the part location, eg. from specialised fixturing.

Chapter 6

Automated background removal

Findings from this work were presented at a meeting of the European Society for Precision Engineering and Nanotechnology at CERN, Switzerland and published as a journal article in:

Eastwood J, Leach R K, Piano S 2022 Autonomous image background removal for accurate and efficient close-range photogrammetry *Measurement Science and Technology* **34** 035404..

As can be seen in Figure 6.1, removal of background pixels from images is required at two stages of the proposed pipeline: first, to help establish the initial relative pose of the object and the camera system; second, to improve the efficiency and measurement result of the final reconstruction.

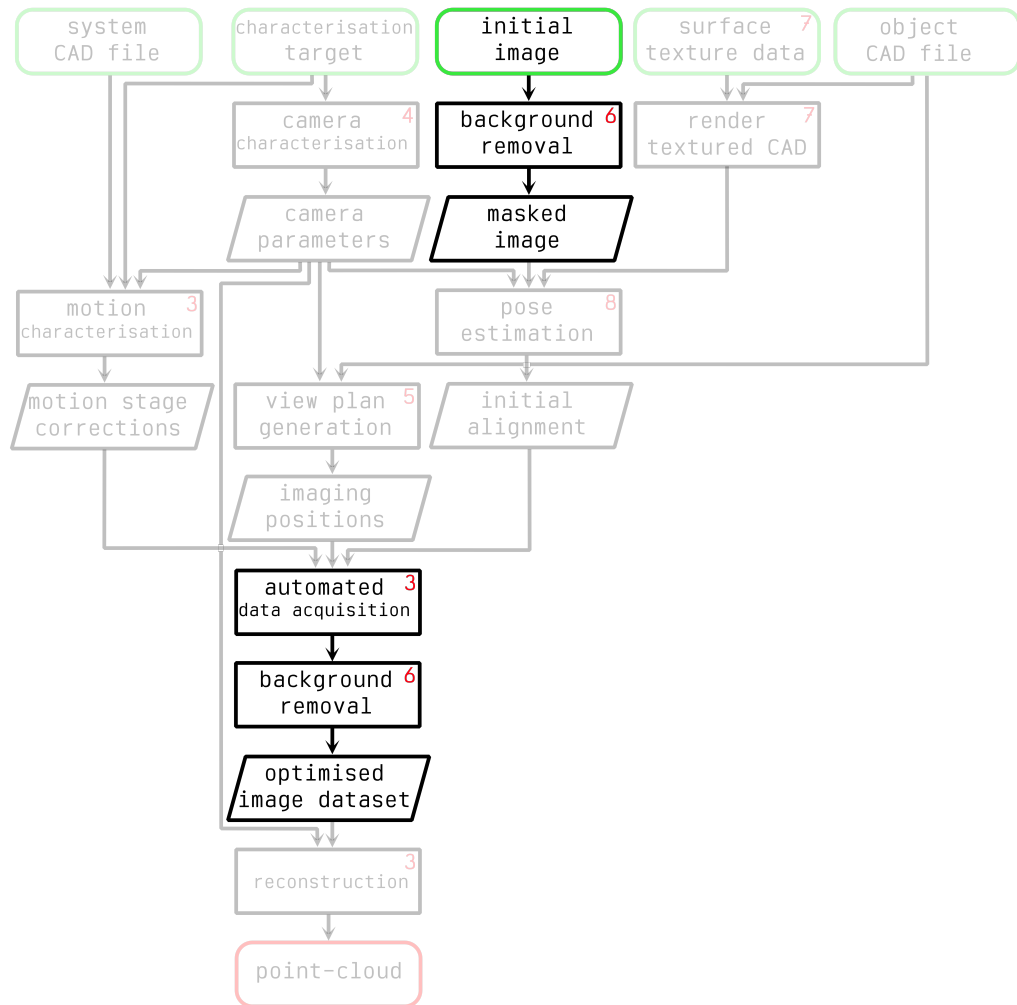


Figure 6.1: Background removal shown within the overall proposed measurement pipeline.

In this chapter, a method for autonomously removing the background from an image is given. Existing methods for background removal can produce inconsistent results, by exploiting known properties of a specific system the method proposed here perform much more reliably while remaining autonomous. The method relies on the assumption that the background

data contains no closed contours, this is true in the case of both photogrammetry systems presented in Section 3.1.1 and the proposed automated system in future work (Section 9.1 will be designed with this assumption in mind. Consequently, the effect of using images with the background removed directly in the reconstruction pipeline is investigated and shown to be highly beneficial in terms of both processing time and measurement result.

6.1 Introduction to background removal

In manufacturing metrology applications, we are only concerned with measuring points on a given part, so any background points reconstructed are not useful to the measurement task and must be removed. In this chapter, a method for improving the efficiency of photogrammetric reconstructions by removing superfluous background pixels from the captured images used in reconstruction is proposed. It is shown not only that this improves the speed of reconstruction by up to 41 % and reduces the number of background points by up to 98 %, but also improves the measurement result's agreement with measurement data taken on the CMM.

6.1.1 Previous work

The previous chapter proposed a view-planning optimisation approach to minimise the number of imaging positions while maintaining reconstruction quality. However, reducing the number of images in the measurement data will eventually impact the measurement result and so can only be taken so far. For example, computer vision tasks often only use two images to reconstruct a scene at high speeds, but the accuracy requirements of

computer vision applications are often much lower than those of metrology applications [194, 195]. Therefore, it is also desirable to increase the per-image processing efficiency.

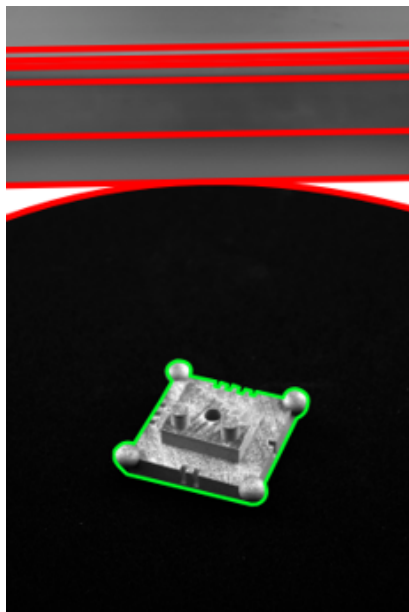
Removing the background from images has the potential to improve computational time as it reduces the number of features present in the image which will be extracted and then matched. Most current approaches to background removal rely on manual masking of images by the user [196, 197]. If the background is static relative to the camera, such as in measurement systems which use a rotation stage, the stationary pixels can be exploited to remove the background in an automated way (see [198]). Furthermore, as static background feature matches can cause the reconstruction algorithms to fail, the removal of these features has the additional benefit of making reconstruction more stable. Because of these benefits, both of the reconstruction software introduced in Section 3.3.2 can accept masks as part of their reconstruction algorithms. In the case of OpenMVG [166], the library can use binary masks to determine which features are included in the reconstruction. However, generating these masks is left entirely up to the user. In the case of Metashape [30], the program can generate image masks but requires the user to manually outline the object in a sub-set of the images used for reconstruction.

Here is presented a method for automated masking of the object from the background of the measurement system. It is shown that the algorithm performs well across a range of object geometries and materials. Passing these masked images to the photogrammetric reconstruction algorithms directly is shown to decrease processing time, memory usage, and number of reconstructed background points. Further, it is shown that when background masking is applied the number of object surface points reconstructed increases and that the measurement result agrees more closely with a measurement from the CMM over repeated measurement of two

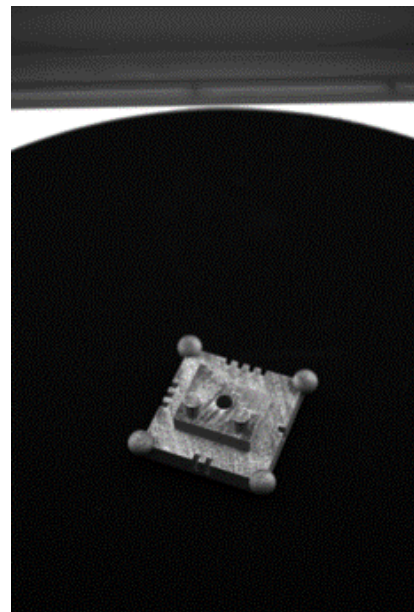
example artefacts.

6.2 Background removal technique

For the background removal algorithm to be general across any object which can be placed in the measurement volume, one major assumption about the measurement system is made; the proposed approach assumes that the background of the scene never contains any closed contours regardless of the measurement head position. While designing a system, this assumption is a relatively simple design constraint to work within. Figure 6.2 shows how the Taraz system meets this requirement.



(a) Image.



(b) Image contours highlighted.

Figure 6.2: Example image and image contours, open contours shown in red, maximum closed contour shown in green.

As can be seen in Figure 6.2, all the background contours are open and thus the largest closed contour in the image must represent the boundary of the object. As such, the problem of background masking can be reduced

to the extraction of the largest closed contour in the image.

6.2.1 Algorithm detail

Python bindings for the OpenCV image processing library [175] was used to perform all image processing operations and file input/output (IO) in the implementation presented here. The steps used to robustly extract the largest closed contour from an image can be split into three stages; preprocessing, edge extraction, and contour selection. The details of each stage of the background removal pipeline are summarised in the diagram shown in Figure 6.3.

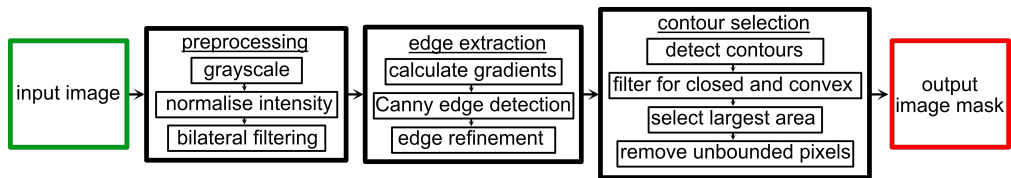


Figure 6.3: The proposed background removal algorithmic pipeline.

First, the image is converted to grayscale as the required contours can be extracted from image intensity information alone. Next, the average pixel intensity is calculated across the entire image, individual pixel values are then scaled linearly so that the average intensity across the image is equal to 52. This is to correct any changes due to material differences between different objects, 52 was used as it was the average intensity recorded across a range of artefact measurements.

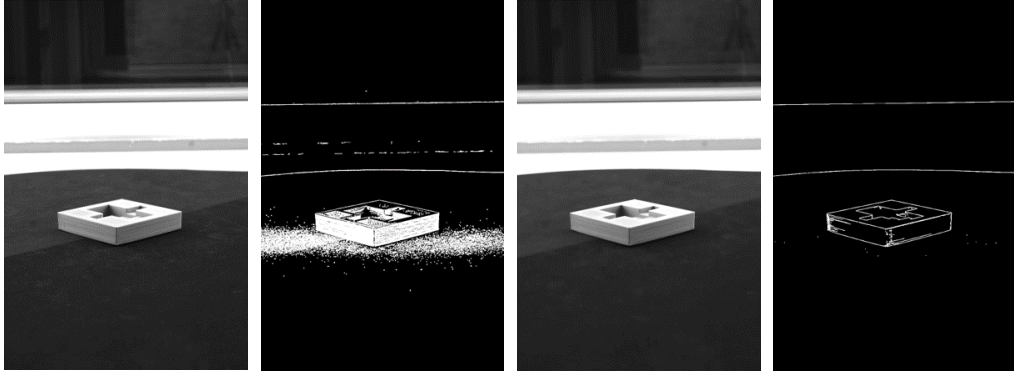
During edge detection, high spatial frequency information such as the rough surface texture of an AM part can negatively impact edge detection and contour extraction from the image. To prevent this effect, a denoising scheme is applied to the image. Recent publications have suggested multiple approaches for image noise reduction. Popular approaches include wavelet transforms [199–201], non-local methods [202, 203], and ML tech-

niques [204]. Due to the for simplicity, computational efficiency, generalisation and robustness an edge preserving smoothing filter was selected as the best approach. To this end, a bilateral filter is applied to the image [205]., the bilateral filter was chosen as it can smooth out high spatial frequency information while preserving edges. A bilateral filter is composed of two Gaussian convolutions, one spatial and one intensity filter (referred to as the range filter). The spatial filter f acts as a standard Gaussian blur parameterised by the bilateral kernel size k and the spatial standard deviation σ_s . The range filter g acts over the space of pixel intensities and is parameterised by k and the intensity standard deviation σ_r . The result of a bilateral filter on image \mathbf{I} is calculated by,

$$\mathbf{I}' = \mathbf{I} * (f(k, \sigma_s) \times g(k, \sigma_r)), \quad (6.1)$$

where $*$ is the convolution operator. This results in, for small values of σ_r , pixels which are spatially close to the current pixel but remote in intensity contributing little to the final smoothing. Therefore, pixels which lie on opposite sides of a boundary do not contribute highly to the smoothing operation compared to pixels on the same side of this boundary. As $\sigma_r \rightarrow 255$ for 8-bit images, the bilateral filter acts just like a Gaussian blur. In this case, through experimentation, the filter values were set to $k = 25$ pixels, $\sigma_r = 25$, $\sigma_s = 150$. This results in large Gaussian blurring on faces but strong edge preservation. Figure 6.4 shows the effect of this filter on an example image and the impact on the performance of Canny edge detection [206].

Once the image has been filtered, Canny edge detection is applied [206]. In brief, image gradients are extracted, areas of high gradient are taken to



(a) Raw image. (b) Detected edges. (c) Filtered image. (d) Detected edges.

Figure 6.4: Impact of bilateral filtering on Canny edge detection.

be edges, these edges are then thinned using non-maximum suppression in the direction of the gradient at that location, finally edges are further refined using a hysteresis pruning algorithm. Image gradients can be found efficiently by decomposing the Sobel operator [23] into four 1D convolutions given by,

$$\mathbf{G}_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \left(\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} * \mathbf{I} \right), \quad (6.2)$$

$$\mathbf{G}_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * \left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \mathbf{I} \right), \quad (6.3)$$

where \mathbf{G}_x and \mathbf{G}_y represent the horizontal and vertical components of the gradient respectively. From these components, the gradient magnitude \mathbf{G} and direction Θ can be calculated from,

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}, \quad (6.4)$$

and,

$$\Theta = \text{atan2}(\mathbf{G}_y, \mathbf{G}_x). \quad (6.5)$$

Each pixel is set to the value of the local image gradient at that image coordinate. Then, every pixel is compared to its two neighbours in the direction of the local image gradient. If the pixel is not a maximum compared to these neighbours, it is set to zero. This process can be iterated until only thin edges remain. Finally, hysteresis pruning is applied to the remaining gradient values to produce the final detected edges. Two threshold gradient values are set, one high and one low. If the image gradient at a given pixel is larger than the high threshold, it is considered an edge pixel and is left untouched. If the image gradient at a given pixel is lower than the low threshold, it is not considered an edge pixel and is set to zero. If an image pixel lies between the two thresholds, it is considered an edge pixel only if at least one of its eight neighbours is also considered an edge pixel. Setting the low and high thresholds is normally done by the user – in this case, because of the desire for automation as well as the need for the algorithm to work on any object - a slightly modified version of the Canny edge detector is used called AutoCanny [207]. Here the high and low thresholds are set based on the median image intensity $\tilde{\mathbf{I}}$ as,

$$t_{high} = \min([255, (1 + 0.33) \cdot \tilde{\mathbf{I}}]), \quad (6.6)$$

$$t_{low} = \max([0, (1 + 0.33) \cdot \tilde{\mathbf{I}}]). \quad (6.7)$$

AutoCanny was found to perform well over a set of artefacts of many shapes and materials. The detected edges are dilated with a 25 square pixel kernel and then blurred. This helps connect any discontinuities in the extracted edges, which can then be eroded to re-thin the now connected edges. The final edge image is passed to `cv2::findContours` which implements a pixel following algorithm to extract continuous contours from the detected edges. Finally, the contours are sorted by the area they inscribe, and the contour of maximum area is selected. This boundary is then used to mask the

background from the image by setting each pixel outside its area to zero, and each pixel within its area equal to its value in the original colour image. Figure 6.5 summarises each stage of the background removal pipeline and Figure 6.6 shows the results of applying this method on a range of artefacts.

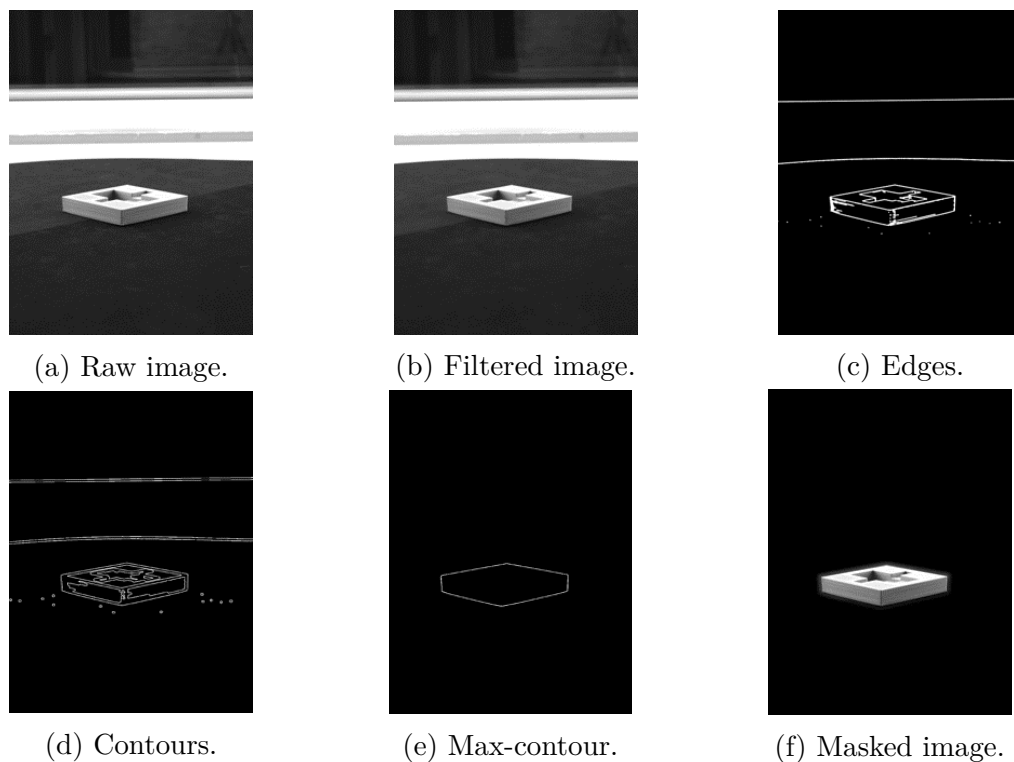


Figure 6.5: Background removal pipeline.

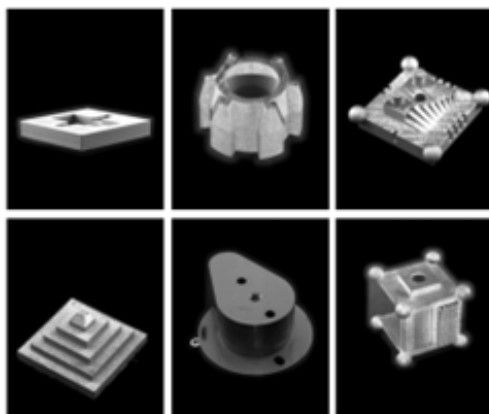


Figure 6.6: Example results of background removal across a range of artefacts.

6.2.2 Experimental procedure

In each experimental test, images were collected using the Taraz system introduced in Section 3.1.1.2. Every scan was comprised of 60 pairs of stereo images captured in two equally spaced rings of 30 positions. Figure 6.7 shows an example reconstructed scene showing the 120 individual imaging positions.

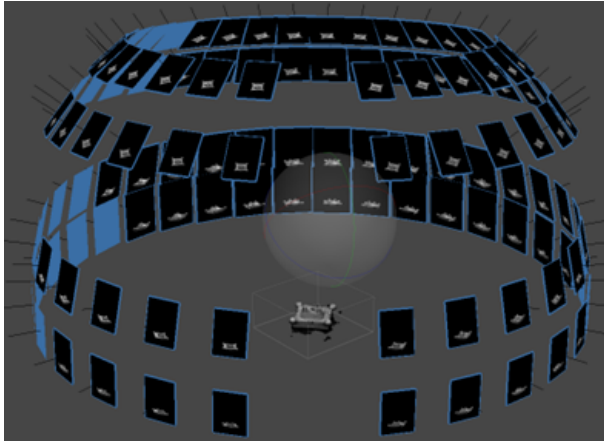


Figure 6.7: Imaging positions used for every scan.

Using the imaging strategy shown in Figure 6.7, the Taraz system was used to measure the pyramid and Tomas artefacts presented in Section 3.4.2. The pyramid artefact shown in Figure 3.10a was measured once, while the measurement of the Tomas artefact, shown in Figure 3.11, was repeated three times to assess the variance and repeatability of the method. During each reconstruction time and memory utilisation were recorded. Finally, each reconstruction was compared to a set of CMM measurements.

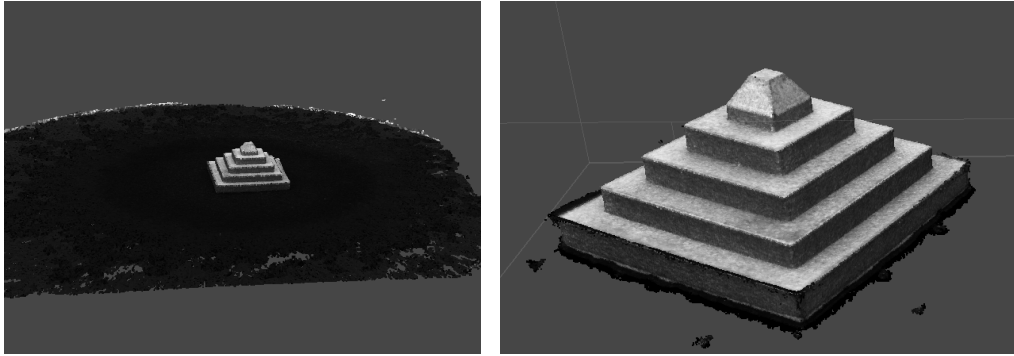
6.3 Background removal results

Reconstruction was performed using Metashape using ‘high’ camera alignment settings and ‘medium’ dense reconstruction settings [30]. Every mea-

sured point cloud was then scaled using the stereo baseline distance between the optical centres of the cameras in the measurement head. Any background points were then manually removed to assess the ratio of object to background points in the scene

6.3.1 Impact on reconstruction efficiency and point density

Figure 6.8 shows the reconstructed dense point clouds of the pyramid artefact both with and without image masking applied.



(a) Without masks.

(b) With masks.

Figure 6.8: Comparison of dense reconstruction of the pyramid artefact.

It is clear that a large number of background points, shown in black were produced in Figure 6.8a but that this number was vastly reduced when masking was applied in Figure 6.8b. Table 6.1 summarises the impact of applying these masks on the reconstruction performance.

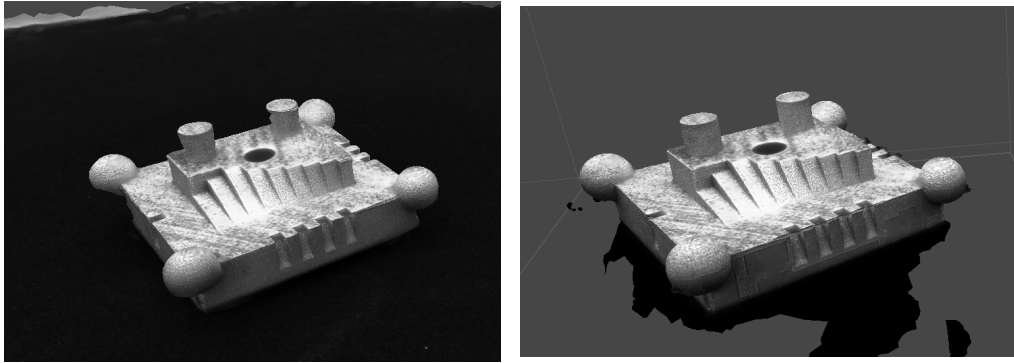
	Without background masking	With background masking	Difference
Densification time /s	1 398	903	-495
Object points	215 467	1 140 950	+925 483
Background points	4 302 679	71 500	-4 231 179

Table 6.1: Impact of applying background masks on dense reconstruction.

As can be seen in Table 6.1, applying the background masks reduced overall processing time by approximately eight minutes. The reason for

this speed up is clear as when the masks are applied around 4 million fewer overall points are reconstructed. Of these 4 million missing points, the vast majority are from the background which would be removed in further data analysis steps anyway. The number of points on the surface of the pyramid object itself has increased by almost a million points.

Figure 6.9 shows the results of applying masking to the reconstruction of the Tomas artefact.



(a) Without masks.

(b) With masks.

Figure 6.9: Comparison of dense reconstruction of the tomas artefact.

Again, Figure 6.9a shows a large number of superfluous background points were reconstructed compared to when masking was applied in Figure 6.9b. Tables 6.2 to 6.4 show a detailed breakdown of the processing time, memory usage, and points reconstructed, averaged over the three repeated measurements of the Tomas artefact.

	Time /s				
	Image processing	Feature matching	Camera alignment	Densification	Total
No Mask	0.0	157.7	81.0	2 459.0	2 697.7
Mask	180.2	223.7	52.7	1 143.3	1 599.7
Difference	180.2	66.0	-28.3	-1 315.7	-1 098.0

Table 6.2: Comparison of time expended at each reconstruction step, averaged across three reconstructions of the Tomas artefact.

The results on the Tomas artefact agree with what was shown for the pyramid. That the overall processing time is reduced, the number of background points reconstructed are reduced, the memory usage is reduced, but

	Memory usage /GB				
	Image processing	Feature matching	Camera alignment	Densification	Total
No Mask	0.000	0.365	0.067	2.999	3.430
Mask	0.471	0.536	0.099	0.923	2.028
Difference	0.471	0.171	0.032	-2.075	-1.402

Table 6.3: Comparison of memory usage at each reconstruction step, averaged across three reconstructions of the Tomas artefact.

	Points			
	Sparse points	Dense points	Object points	Background points
No Mask	58 373	3 600 644	282 721	3 317 923
Mask	71 436	377 862	312 560	65 301
Difference	13 062	-3 222 783	29 839	-3 252 622

Table 6.4: Comparison of points reconstructed, averaged across three reconstructions of the Tomas artefact.

the number of points reconstructed on the object surface is increased.

6.3.2 Comparison to CMM

The reconstructed dense point clouds were then triangulated into a mesh. When the background points had been removed ICP was employed to register the meshes to data taken from the CMM. The PTM distances could then be calculated between the CMM and photogrammetry data. The results for the pyramid are shown in Figure 6.10.

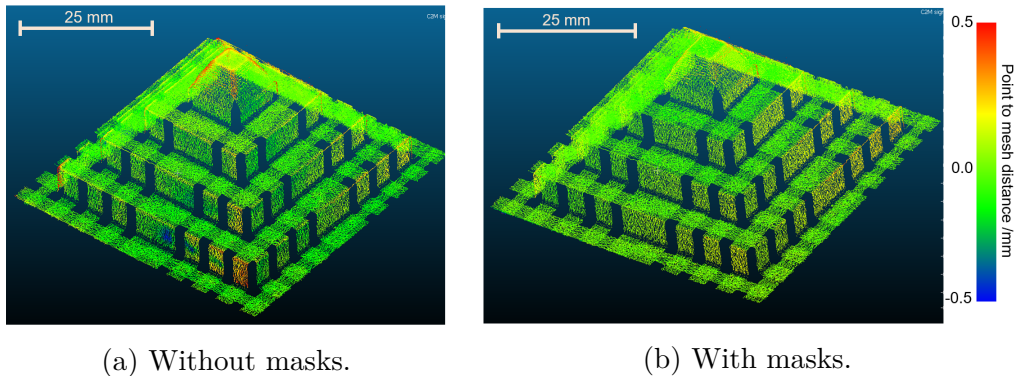


Figure 6.10: Comparison of PTM distances for the pyramid artefact.

As can be seen in Figure 6.10a, the unmasked reconstruction contains a

higher number of outlying points compared to Figure 6.10b, shown in red. Figure 6.11 shows a comparison between the histograms of PTM distances across the two measurement comparisons using 400 bins.

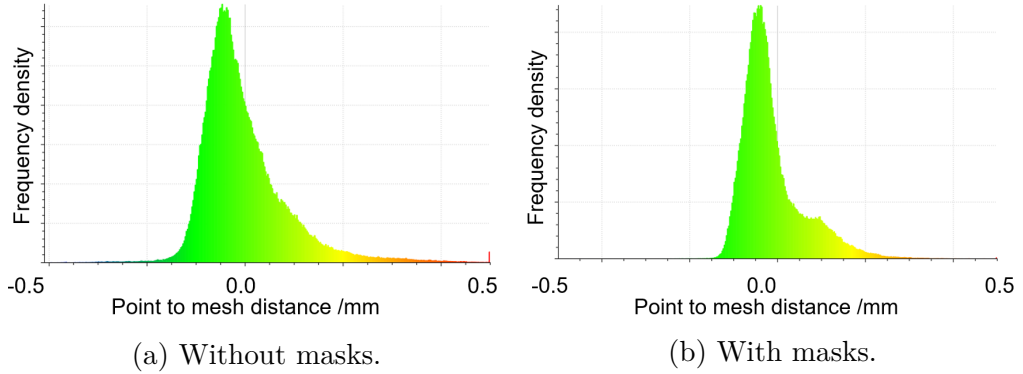


Figure 6.11: Comparison of the distribution of PTM distances for the pyramid artefact.

As can be seen in Figure 10, when background masking is applied the PTM distance spread is reduced. Fitting a Gaussian to the distribution in Figure 6.11a yields a standard deviation of $85\ \mu\text{m}$, while fitting a Gaussian to the distribution in Figure 6.11b yields a standard deviation of $70\ \mu\text{m}$. In addition to the lower deviation in the PTM distances, there are also many fewer outliers when masking is applied, this can be seen in the spike on the far left of the distribution in Figure 6.11a which represents PTM distances larger than $500\ \mu\text{m}$.

Figure 6.12 shows one of the three repeat measurements of the Tomas artefact. Figure 6.13 shows the combined histograms over all three repeat measurements for both masked and unmasked reconstructions.

In Figure 6.13a the combined standard deviation of the PTM distances over three repeat measurements was $93\ \mu\text{m}$. When background masking was applied to the measurement data in Figure 6.13b the standard deviation reduced to $70\ \mu\text{m}$ with masking, representing a decrease of $23\ \mu\text{m}$. Additionally, the number of outlying points with PTM distances greater than $500\ \mu\text{m}$ also, was again, reduced by applying background masking. Fig-

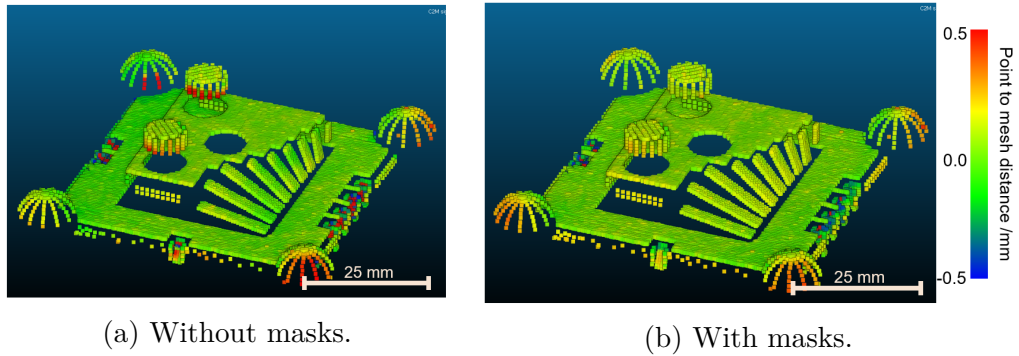


Figure 6.12: Comparison of PTM distances for the Tomas artefact.

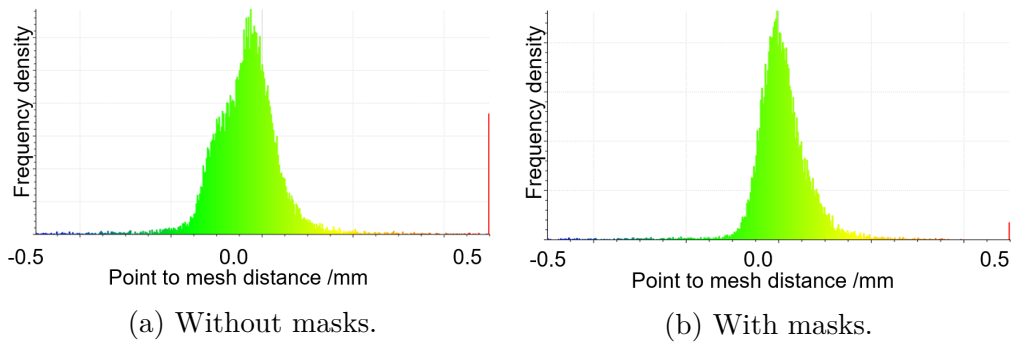


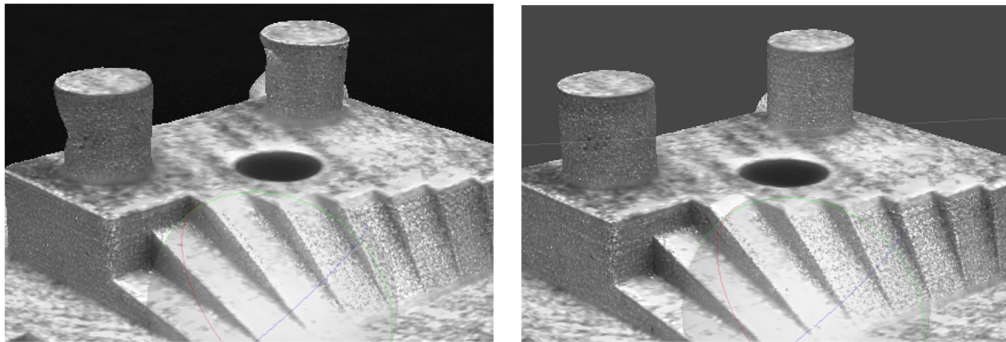
Figure 6.13: Comparison of the distribution of PTM distances for the Tomas artefact.

Figure 6.12a shows that many of the outlying points are concentrated around the more complex features such as the cylinders, spheres and recesses which were reconstructed more faithfully in Figure 6.12b as shown by the reduced number of red points.

6.4 Background removal discussion

As can be seen in Figures 6.10 to 6.12 the agreement with CMM is improved when background masking is applied. This is likely mainly due to the removal of the static background from the image. As was discussed in Section 6.1.1 the static portion of the background, present due to the use of a rotation stage, creates a set of points which are static relative to the camera while the rest of the points have undergone some relative motion.

This means that when the bundle adjustment algorithm attempts to globally optimise the camera positions and point locations, the triangulation of the object points can be degraded. Figure 6.14 shows a comparison of the cylindrical features on the Tomas artefact, it can be seen that the reconstruction quality of these features improves when background masking is used. Because these cylinders are prominent features and intersect with static portion of the background in many views, this reinforces the idea that it is the removal of the static background that leads to improved measurement results.



(a) Without masks.

(b) With masks.

Figure 6.14: Comparison of the dense reconstruction of cylindrical features.

A further possible contributing factor could be the patch-based densification as was discussed in Section 2.1.1.5. The dense reconstruction algorithms operate by growing and refining rectangular patches of points. In the case of the masked data these patches can only be produced on the object surface, whereas in the unmasked case many are created in the background data.

The histograms in Figures 6.11 and 6.13 show the distribution of all the PTM distance data has some skew. This is likely due to a small error in the scale applied to the photogrammetric portion of the data. This scale is based on the stereo baseline, the distance between the optical centres

of the stereo cameras. This is difficult to measure directly so was established through reconstructing a ball bar of known size. However, the tighter spread of the data when background masking was applied, as shown by the lower standard deviations, is still strong evidence of greater agreement with CMM despite this potential scaling error.

Although the background removal approach is quite robust to a range of objects, as was shown in Figure 6.6, it is not perfect and there are occasional viewing angles which cause the masking process to erroneously remove some of the object data. Figure 7.13 shows an example of this from the data used to reconstruct the pyramid artefact.

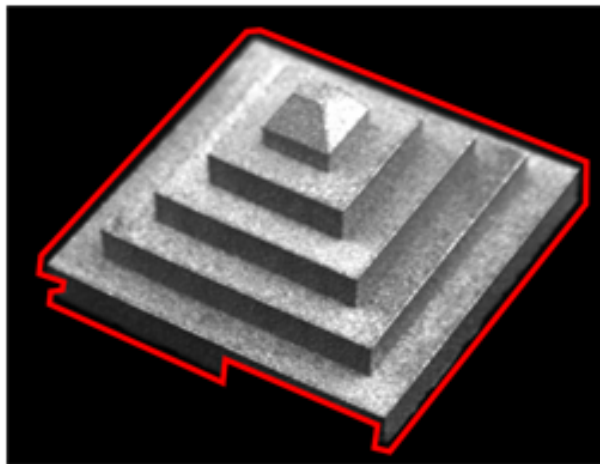


Figure 6.15: Example failure case of the background removal, masking contour shown in red.

The erroneous masking shown in Figure 7.13 occurs when the bottom edge of the object is in shadow due to the lighting conditions present within the measurement system. The shadow effectively blurs the boundary of the object and causes the edge detection part of the pipeline to fail. However, because this only occurs from very few viewing angles, enough of the surrounding views detect and triangulate points in the area of missing data that the result on the final measurement result is minimal. These erroneous cases could be prevented entirely by changing the lighting conditions in the

measurement volume to be as diffuse and even as possible.

It is worth noting that the exact time taken to mask each image (approximately 1.5 s per image) is likely largely dependent on the implementation of the presented algorithm. It is likely that an optimised and compiled version of the algorithm could operate much faster than the Python implementation used here, especially with the many file IO operations required. There is also an obvious hardware dependence, in this case all image processing and reconstruction operations were performed on the same PC with an Intel Xeon W 2123 CPU, 32G GB of RAM.

Recent research has explored the use of ML methods for both edge extraction (see review [208]), and end-to-end background removal (see review [209]). However, many of these methods as they are currently implemented are either inaccurate such as extracting only bounding boxes [210] or are developed for specific applications and as such would not generalise well across any possible measurement artefact [211]. Furthermore, as this method has been shown to be effective with only traditional methods it avoids the computational overhead required to train a ML model. Where ML and related methods are unavoidable to complete the tasks required in other chapters, avoiding “black box” style neural networks here also makes these results simple to interpret and understand.

6.5 Future work on background removal

Some tests were conducted on using the background removal strategy proposed here with the optimised imaging positions proposed in Chapter 5. However, at such few numbers of images in the scan it became evident that the background features were key to accurately reconstructing the scheme and not a hinderance in this case. As such, some future research in com-

binning the two approaches and adjusting the global objective function to account for this would be valuable.

6.6 Background removal conclusions

In this chapter, an image processing technique for the removal of background pixels from images taken within a photogrammetric measurement system has been proposed. This pipeline is dependent on there being no closed contours in the background portion any images taken in the scan and uses this assumption to reduce the background masking problem to the extraction of the closed contour of largest area within the image. This work contributes to the state of the art by showing that exploiting known properties of the system allows background removal to be performed reliably on a large selection of geometries while avoiding the overhead of training large ML models leading to computational savings and improved measurement outcomes.

To test the impact of using masked images directly within photogrammetry measurements, two test artefacts were reconstructed both with and without background masking applied to the input images. It was shown in both cases that applying imaging masking reduced reconstruction times and memory usage, increased the number and density of surface points reconstructed, and dramatically reduced the number of superfluous background points reconstructed.

The impact on the measurement result was investigated by comparing to measurement data gathered through repeat tactile measurement using a CMM. It was found that applying background masking reduced the number of outlying points reconstructed and reduces the standard deviation in the PTM distances when the photogrammetry and CMM data are regis-

tered together. This improvement in measurement agreement with CMM is likely due to the static background degrading the triangulation quality of the points when background masking is not applied.

The contributions to science given by the work in this chapter can be summarised as: a method for autonomously removing background pixels from images taken by a given imaging system. This improves on the state of the art by exploiting known properties of the imaging system to improve robustness while avoiding the overhead of training a large model. Further, it is shown that the use of this algorithm improves standard reconstructions by reducing processing time, increasing reconstruction quality, and reducing the reconstruction of unwanted points.

In Chapter 8 a method for using the image masks produced by this method is proposed to solve the relative pose between the camera system and the part, allowing for integration of the view plan proposed in Chapter 5 within the overall pipeline shown in Figure 1.3.

Chapter 7

Generating surface texture data

This work was completed in collaboration with Lewis Newton who gathered the datasets used herein, was presented to the scientific technical committee for surfaces of the International Academy of Production Engineering (CIRP) and was published as a journal article in:

Eastwood J, Newton L, Leach R K, Piano S 2021 Generation and categorisation of surface texture data using a modified progressively growing adversarial networks *Precis. Eng* **74** 1-11.

To use the view planning strategy presented in Chapter 5 without the need for special fixturing, the initial spatial relationship between the object to be measured and the camera system must be established. Before a pose-estimation approach can be developed, as can be seen in Figure 7.1, a method for applying surface texture data to create realistic renderings of a given part from its CAD data is required.

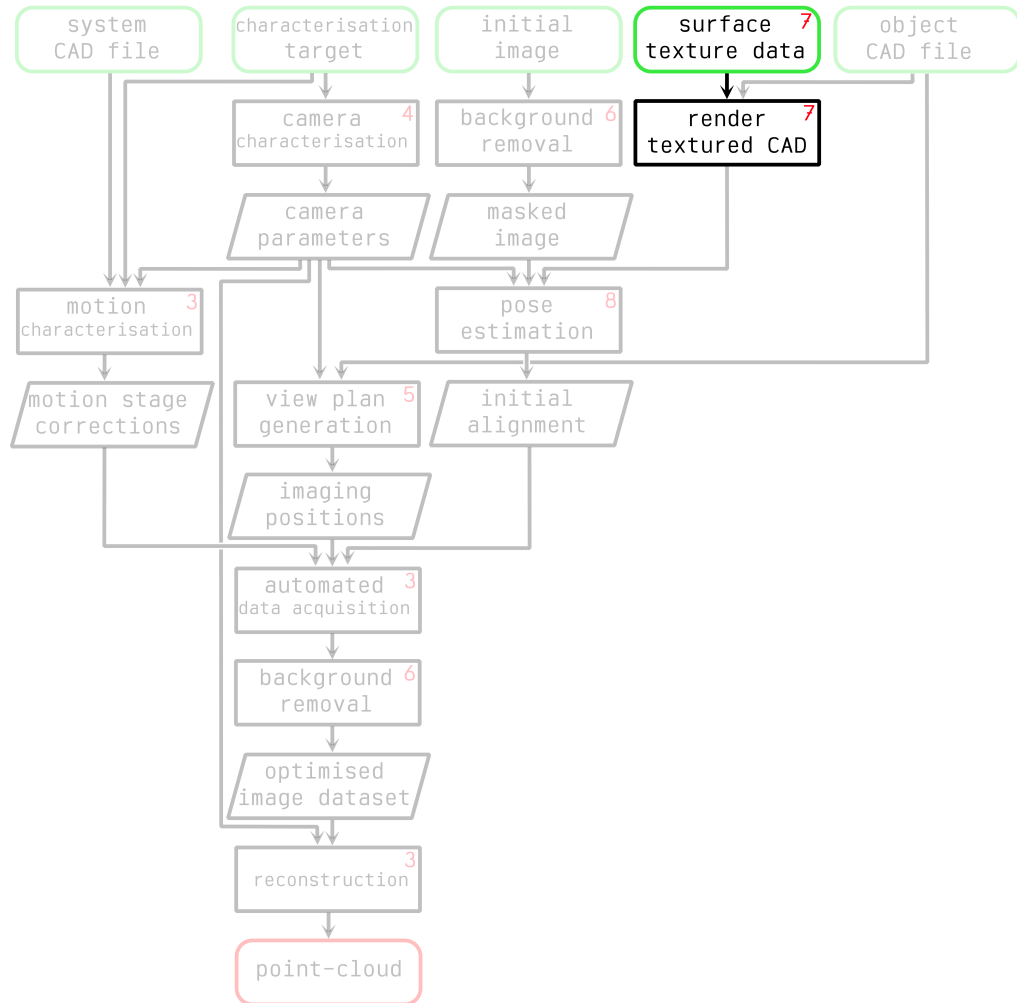


Figure 7.1: Surface texture generation shown within the overall proposed measurement pipeline.

In this chapter, a method for generating large quantities of surface data which is shown to be representative of real data. It is shown that the presented model can learn to represent a range of surface types and a method of categorising this data autonomously is also presented. Finally,

a material shader is developed which transforms the output of the surface generator model into a seamless and infinitely repeating texture which can be applied to CAD to produce realistic data which is then used to train a pose estimation technique presented in Chapter 8. This can then be used to generate large datasets of realistic simulated images for the training of ML models such as the monocular model presented in the following chapter.

7.1 Introduction to texture generation

The ability to generate synthetic surface texture data which convincingly represents the result of a real measurement has many applications [212,213]. For example, often large quantities of data are required to train statistical models that would be difficult if not infeasible to collect manually [212]. Furthermore, representative synthetic textures are useful for other applications such as for use within virtual instruments [213], or for accurate image rendering.

Previous approaches for simulating surface data have been computationally intensive at runtime, limited to the representation of a single manufacturing process, or requiring an analytic representation of the surface [214–219]. Earlier work, presented in Section 8.2.1.1, used a synthetic surface texture of an AM part to produce photorealistic renders. These renders were then used to train a CNN for object pose estimation. The synthetic texture was simulated by analysing the real surface data of a part made with the same AM process, extracting the dominant spatial frequencies and amplitudes, and layering various pseudo-random noise functions at these frequencies and amplitudes. While this approach produces a good estimation of the surface parameters, it does not capture properties related to the surface features, such as feature shape and surface anisotropy. Software developed by

Todhunter et al [214] defined the surface to be simulated as a sum of cosine waves; the surface complexity can be increased further with the addition of pseudo-random noise in the form of multi-scale Fourier space Gaussian blur. This Fourier approach has some advantages as the generated surface parameters can be known explicitly, but the ‘realism’ of the generated surface is user-dependant. Another study used an analytic representation to generate surface form combined with smaller scale noise to simulate texture, resulting in a full synthetic topography [215]. The synthetic surface data was then used in the creation of synthetic interferometry data by phase-wrapping the simulated surface.

An alternate approach to realise synthetic surface data is to produce a full physics-based simulation of the manufacturing process of interest using numerical methods [216–219]. For example, Zhou *et al.* [219] focused specifically on the melt-pool of an arc-welding AM process. Using a combination of a volume of fluid model and continuum surface force model to simulate both heat and mass transfer in the powder bed, they were able to predict the final surface profiles, which compare favourably with experimental data. This physics-based approach is wholly reliant on the accuracy of the physical simulation and is computationally expensive. Physical simulation models have the further disadvantage of being specific to a single manufacturing process; if surfaces are required to be simulated across a range of processes and materials, a large amount of development time would be required to develop new physical models.

The method proposed in this chapter overcomes many of the shortcomings of these previous approaches. Firstly, the model can simulate any process and measurement method that can be represented as a depth map. Furthermore, the model can be trained to represent a range of surface types simultaneously (i.e. without the need for retraining) so long as the desired variation is represented within the training data. Moreover, it is shown

that the generated surfaces are representative of real data without the need for manual analysis of the desired surface features. These benefits are achieved through adapting an approach initially developed for generating high resolution synthetic images: a progressively growing generative adversarial network (PG-GAN) [49]. By encoding training measurements as high-resolution images with height data represented by the pixel values, a dataset is created to train the PG-GAN. Once the model is trained, it will generate images with the same encoding, which can then be decoded back into height data. By front loading the computational expense to training time, once deployed, the model can quickly generate large quantities of new surface data. Further, it is shown that a single model can simulate a variety of surface types simultaneously and then extend the PG-GAN model to automatically categorise the generated surfaces into predefined surface types. The performance of the proposed method is validated on two very different datasets: a collection of industrial coatings measured using the MMT-LS system and an AM part measured using the FV system which were introduced in Section 3.1.4 and Section 3.1.5 respectively. The surface generation model is then extended to perform a categorisation of the generated surface types creating a model that can produce surfaces with predictable properties. Finally, a quantitative comparison of the categorised generated surfaces with their real counterparts is undertaken and shows that the model provides a sound representation of the surface types.

7.2 Description of the surface generation model

A GAN is a system of two sub-networks trained in a zero-sum-game (first proposed in 2014 by Goodfellow et al. [48]) and was introduced in Section

2.2.2.1. A PG-GAN is an extension of the traditional GAN architecture that was originally proposed by NVIDIA [49]. A PG-GAN improves variability and stability when operating on high resolution images by beginning with a highly down-sampled version of the training data, in this case (4×4) pixels. After a predefined number of training periods (epochs), an additional transpose convolution layer is appended to the generator model and a conventional convolution layer is prepended to the discriminator, doubling the resolution of the generated image. The resolution doubling is repeated until the final resolution is achieved (in this case (512×512) pixels). The additional layers are faded-in to the model smoothly over a period of epochs to avoid any jerk to the network and encourage stability; this is shown in Figure 7.2.

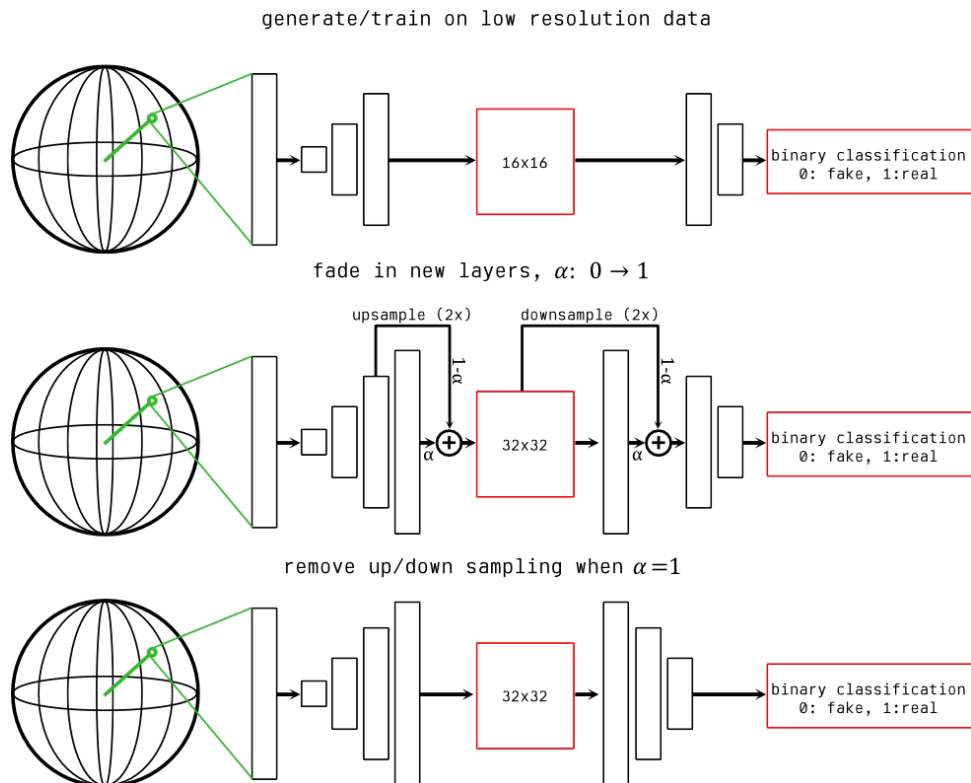


Figure 7.2: The process of doubling the resolution smoothly in a PGGAN using the α parameter

This smooth fading is achieved by adding a $2 \times$ up-sampling layer to the

generator and a $2\times$ down-sampling layer to the discriminator. The output of the new convolutional layers is combined in a weighted sum with the up-sampling/down-sampling output, where the relative weighting of each contribution is controlled by a parameter α . Over a predefined number of epochs (where this number is a hyper-parameter of the PGGAN model) the weighting parameter linearly increases until the up-sampling/down-sampling layers no longer contribute to the model and can be removed. The input vectors to the generator are sampled from a ‘latent space’; in this case the space is the unit hypersphere S^{99} which is defined by,

$$S^{99} = \{ \mathbf{x} \in R^{100} \mid \|\mathbf{x}\| = 1 \} \quad (7.1)$$

In the case of the PG-GAN, the discriminator, rather than classifying the input as either real or fake, assigns a continuous ‘realness’ value to the input. Using a continuous realness value rather than discrete classification supplies a smoother gradient and leads to more stable training [220]. In turn, these realness prediction values are fed into a loss function based on the Wasserstein distance, a measure of the minimum amount of work required to turn one distribution into another [221]; in the case of a GAN, this is the distribution of the critic predictions compared to the real distribution of real/fake images. A Wasserstein based loss prevents vanishing and exploding gradients when compared to cross-entropy approaches (which is the popular alternative).

Once the PG-GAN is trained and the discriminator is discarded, the generator can be used to generate new images that have been shown to be indistinguishable from the real dataset discriminator. For this application, the generator is extended by piping the output of the generator into the input of a CNN to classify the type of surface produced. This process

allows one to make meaningful comparisons of statistical surface texture parameters to ensure that the synthetic surfaces are representative of the full space of real measured surfaces. Full details of the CNN extension are given in Section 7.4.1.

7.3 Creating training datasets

Two datasets were developed to train, validate and test the model: industrial coatings and AM surfaces. To show that the approach is applicable to a range of measurement techniques and surface types, the datasets described below use different measurement techniques on different surface types. In both cases, the same procedure was used to convert the measured data into the final set of (1024×1024) pixel images. The measured data were first converted into depth maps before a polynomial form removal was applied. The height data were then encoded as a set of grayscale images. A process of dataset augmentation was used to expand the datasets. This process involved rotating, mirroring and cropping the images to a size of (512×512) pixels.

7.3.1 Industrial coating dataset

A set of sample surfaces created from a variety of industrial coatings were produced which were then measured using the MMT-LS system as was introduced in Section 3.1.4. The procedure for data treatment described above was applied; in this case each encoded image represents a (20×20) mm area and depth values are encoded as a linear mapping to grayscale

values in the range (0 to 1) from depth values in the range (-50 to 50) μm . A sample of the final treated data is shown in Figure 7.3. The industrial coating surfaces were considered a suitable case study because there are various combinations of process parameters that can create a large range of resultant surfaces, however, there is a fundamental limitation on how many surfaces could be economically produced – making the ability to simulate the potential “design-space” of all possible surfaces a valuable endeavour.

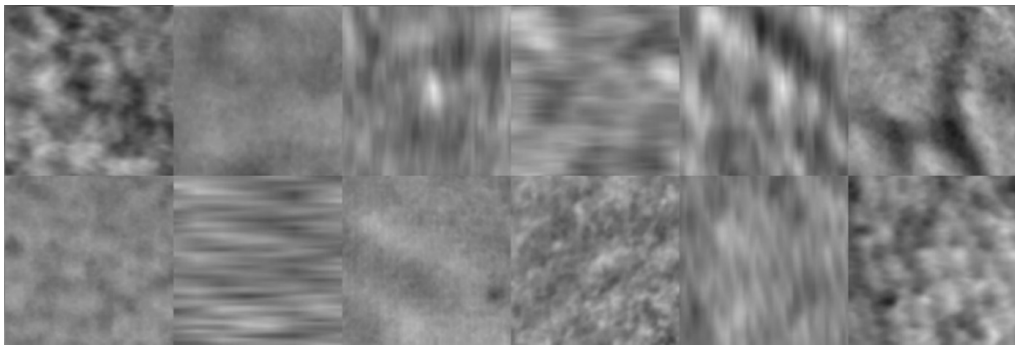


Figure 7.3: Twelve example encoded images taken from the industrial coatings dataset, showing the range of different surface types present in the training data.

7.3.2 AM surface dataset

Another dataset was constructed from FV measurements of a metal AM part [222]. The bracelet artefact consists of a series of thirty-six plane faces at 10° increments with minimal supports produced by EB-PBF [223]. This artefact was chosen as it will give a range of surface types dependent on the relative orientation of the face to the powder bed. The CAD of this part is shown in Figure 7.4a.

Figure 7.4c shows the part manufactured from Ti64 using an Arcam A2X EBPBF process. The data processing steps outlined previously were

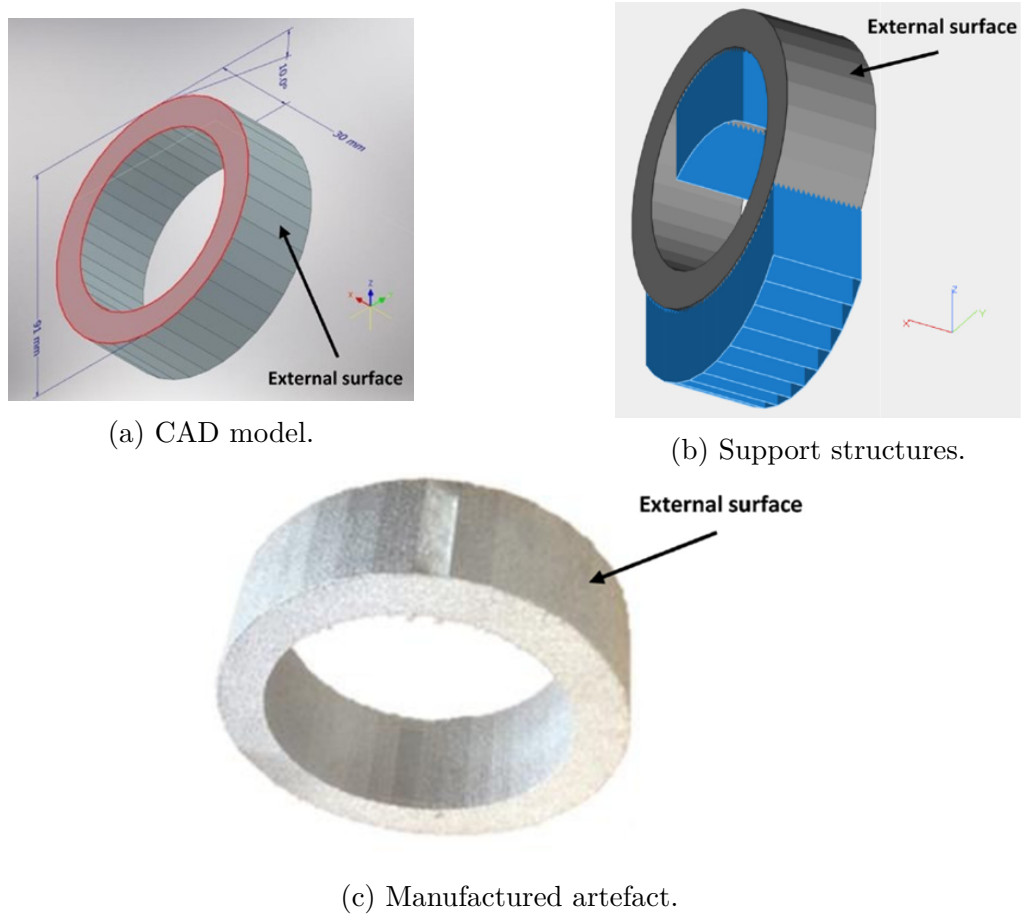


Figure 7.4: Bracelet artefact used to create the AM dataset.

applied, this time mapping depth values of (-70 to 70) μm to grayscale (0 to 1) and the image size representing (1 \times 1) mm. Figure 7.5 shows a sample of the measured data with the post processing steps applied.

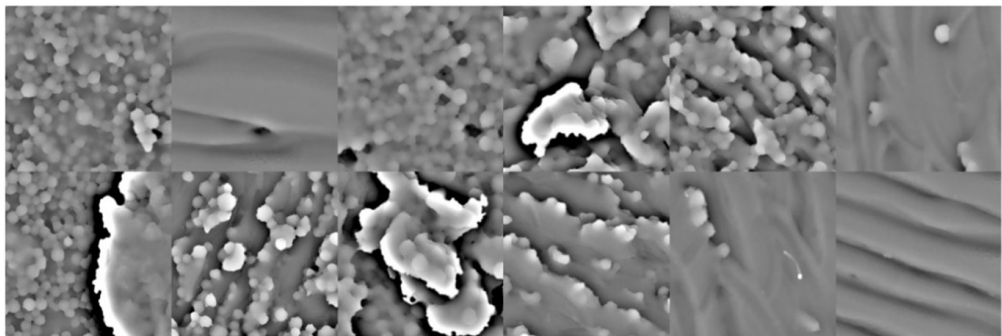


Figure 7.5: Twelve example encoded images taken from the AM dataset showing the large variation present between surfaces included the dataset.

It is clear that there is a large variation in the types of surfaces mea-

sured from the part. This variation is dependent on the relative angle of the measured face to the powder bed [222]. For example, the large-scale features that can be seen in Figure 7.5 are the remnants of the support structures shown in Figure 7.4b, which were required for the printing process and then removed post-process. These support structures only occur on the down-skin surfaces. Additionally, the smooth, straight weld tracks only occur on the top face, which is parallel to the powder bed. As the surface angle increases relative to the build plane, the presence of particles agglomerated to the surfaces increases, eventually occluding the weld tracks entirely. Figure 7.6 shows these surface types, their location on the artefact, and an example from the final dataset of each type. Section 7.4.2 discusses these surface types in more detail and how the generator can be extended to produce surfaces of a known type.

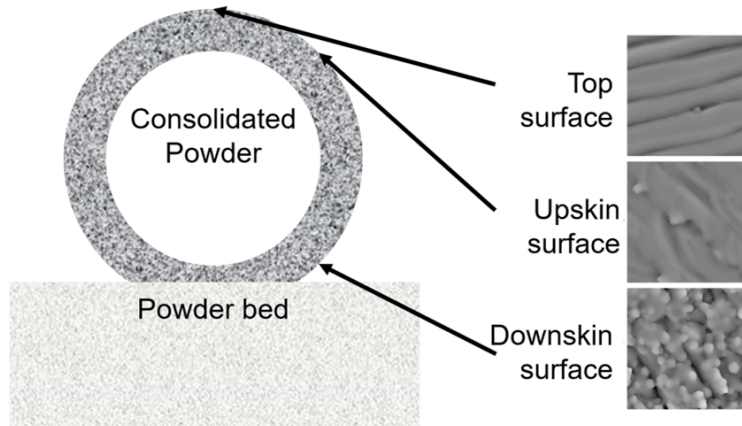


Figure 7.6: Surface types and their locations on the bracelet artefact with examples taken from the AM surface dataset.

7.4 Surface generation results

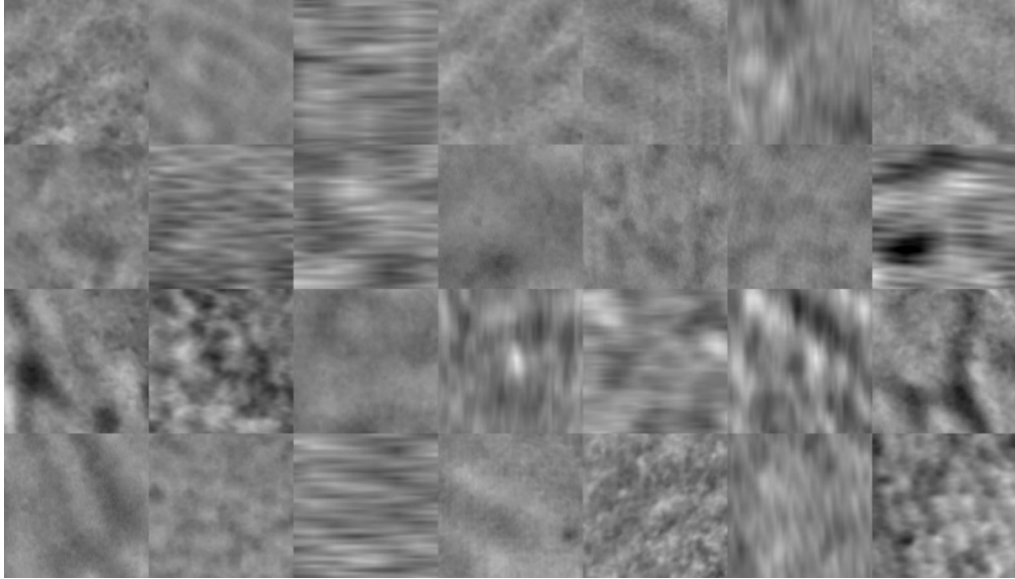
For both datasets, the PG-GAN took five days to complete training on a HPC GPU node [164]. The length of the training time is in agreement with

the original PG-GAN paper for producing images of a similar resolution and once trained the model can produce new surface texture data in less than a second. Once training had concluded, the trained generator model was deployed to create 1000 images of both the coated surfaces and the AM surfaces. Figure 7.7 shows a comparison of the real and synthetic images for the industrially coated dataset.

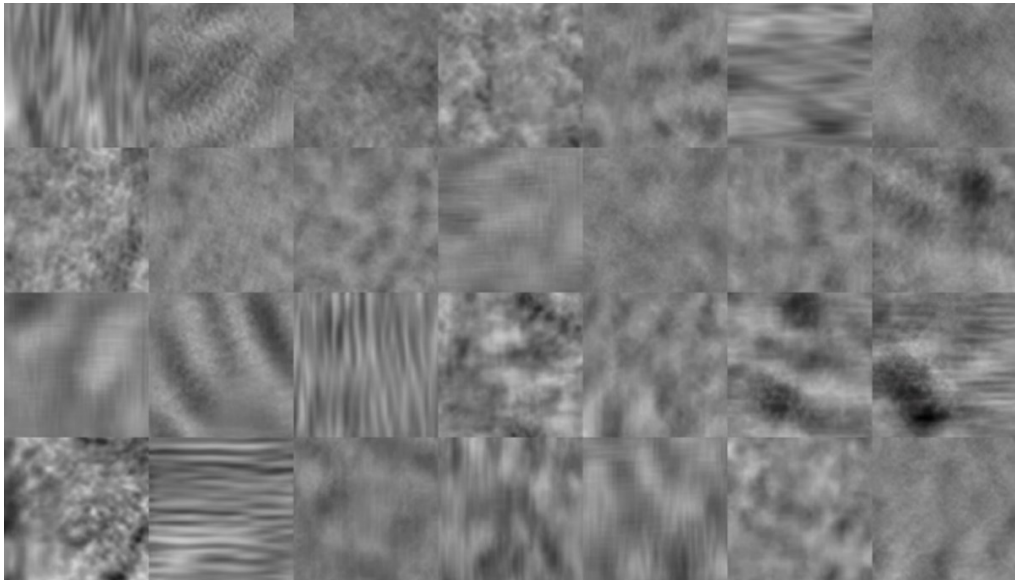
The examples shown in Figure 7.7a were selected to show the range of possible surface data contained within the training dataset. It can be seen in Figure 7.7b that this variation is captured by the generator with considerably different features visibly present across the output data. Figure 7.8 shows a similar comparison for the AM dataset.

The model outputs still need to be decoded from grayscale images into true height data. To do this, the reverse of the encoding process described previously is applied to the 1000 generated images. Figure 7.9 shows example decoded surfaces compared with real surfaces of the same type from the training data.

As can be seen in the scale of Figure 7.9, the heights generated and scales of features generated match closely with those in the training data. Figure 7.9d shows the model has learned to represent defects in the weld tracks which indicates this method could be useful for training defect detection models - a common issue within the field (see review by Meng et al. [71]). Simulated surfaces that possess features found in the real surface case could also allow for more training data for the development of improved feature-based characterisation approaches.



(a) Real encoded depth data from the industrial coating dataset.

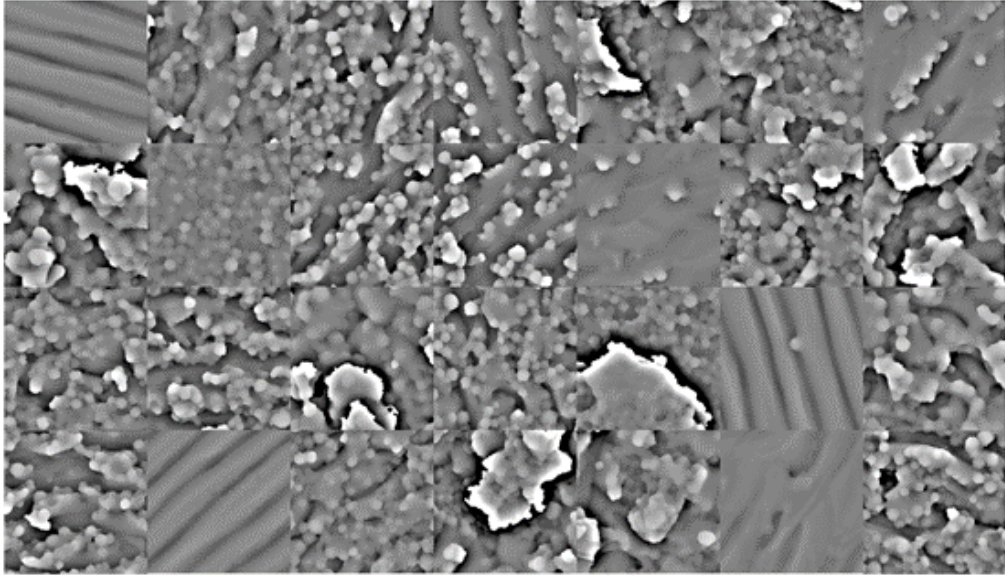


(b) Synthetic encoded depth data created by the trained PG-GAN generator.

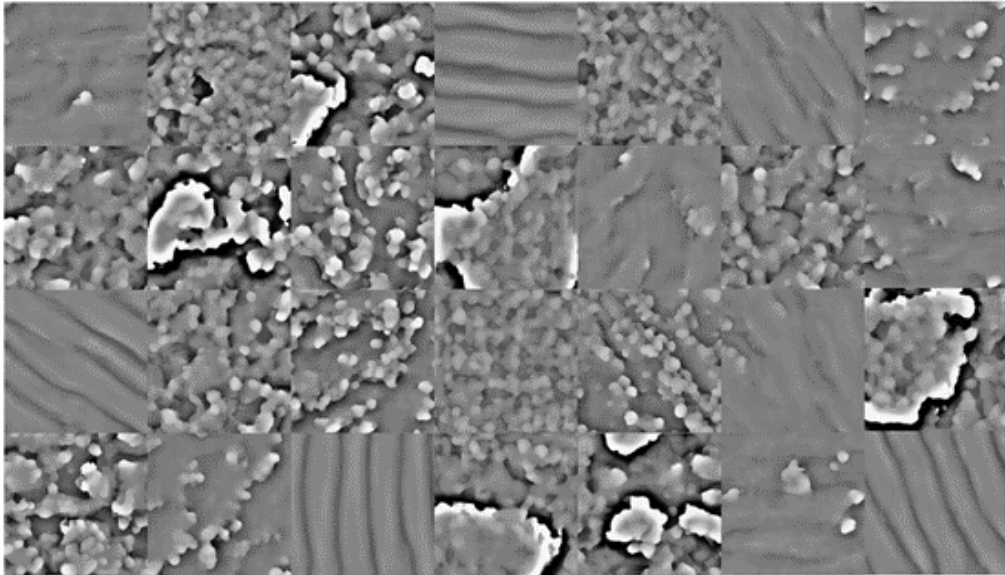
Figure 7.7: A comparison between a random sample of twenty-eight coating surface images, both real and generated.

7.4.1 Surface categorisation results

The trained model generates surfaces by randomly sampling a point from S^{99} and passing the input coordinate through the generator, producing a randomly sampled surface from the possible output-space. As discussed in Section 7.4, in the case of the AM dataset specifically it is clear that there are distinct types of surfaces encapsulated by the dataset, which were



(a) Real encoded depth data from the AM dataset.



(b) Synthetic encoded depth data created by the trained PGGAN generator.

Figure 7.8: A comparison between a random sample of twenty-eight AM surface images, both real and generated.

shown in Figure 7.6. Top surfaces are characterised by distinct weld tracks and the absence of agglomerated particles, top surfaces are produced when the face is parallel with the powder bed. In up-skin surfaces, the weld tracks can still be seen but, as the angle relative to the powder bed approaches 90° , particle agglomeration begins to dominate. Finally, down-skin surfaces are fully dominated by agglomerated particles due to the increased inter-

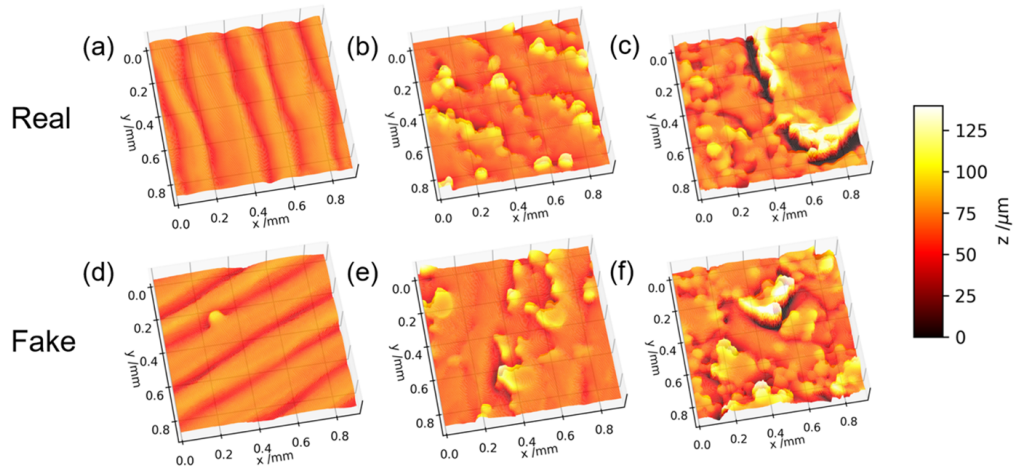


Figure 7.9: Surface topography height maps of real measurement data from the AM surface dataset compared to decoded fake data from the generator.

action with the powder bed and the remnants of support structures (from those shown in Figure 7.4b) can sometimes be seen.

Given these categories, the model would be more useful if it could be used to generate surfaces of a known type rather than a random surface. To this end, the generator model is extended by piping the output directly into the input of a secondary CNN that predicts the generated surface type. The categorisation CNN architecture is shown in Figure 7.10.

Moreover, the same AM surface training set can be used for the PG-GAN to train the categorisation model. During the measurement, the angle of the face being measured was recorded and stored in the meta-data associated with the measurement data. It is a straightforward process, therefore, to propagate this metadata through the data augmentation process to supply ground-truth labels. Measurement data originating from the 0° face were labelled as top, data from faces in the 10° to 90° interval were labeled as up-skin, and 100° to 180° as down-skin. There is some ambiguity as to whether the 90° face should be categorised as up-skin or down-skin and, because the physical characteristics transition smoothly between these two

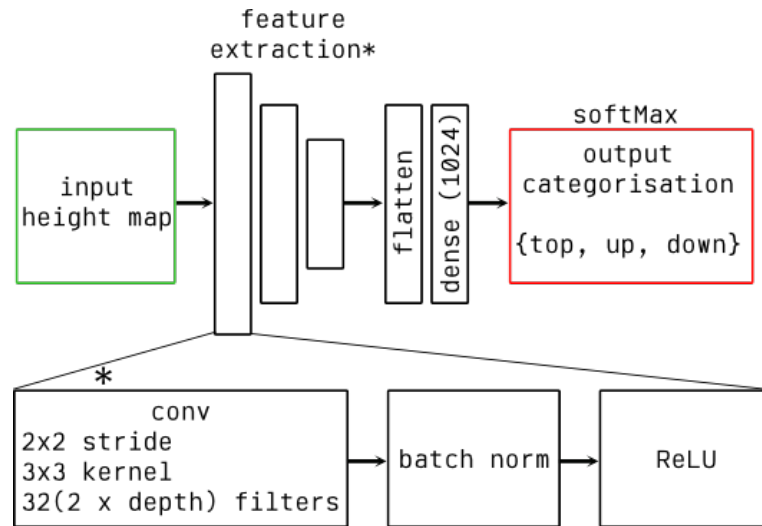


Figure 7.10: Categorisation CNN, which takes an input image, extracts features through a sliding window kernel convolution, flattens the feature maps, feeds through a fully connected layer and produces a predicted class label (T: top,U: up-skin, D: down-skin).

categories, there is likely to be some misclassification of surfaces near the boundary. For other datasets it may be optimal to set the boundaries at different angles, due to the effect of gravity during processing for example, however it was found that the classification in this case was most accurate when using the boundaries detailed above.

A softmax function (a normalised exponential function which can be thought of as a generalisation of the logistic function to n-dimensions [224]) was used as the activation function in the output layer of the model. The Adam optimiser [42] with a learning rate of 0.0001 with a categorical cross-entropy loss function were used in the training of the CNN. Due to the relative simplicity of the categorisation model when compared to the PG-GAN model, a HPC compute node was used for training, which was completed within twelve hours. Figure 7.11 shows a plot of the training history. The loss values shown are the values of the cross-entropy over that image batch; the accuracy is calculated by simply taking the argmax (the index of the output tensor containing the maximum value) of the values of the output nodes.

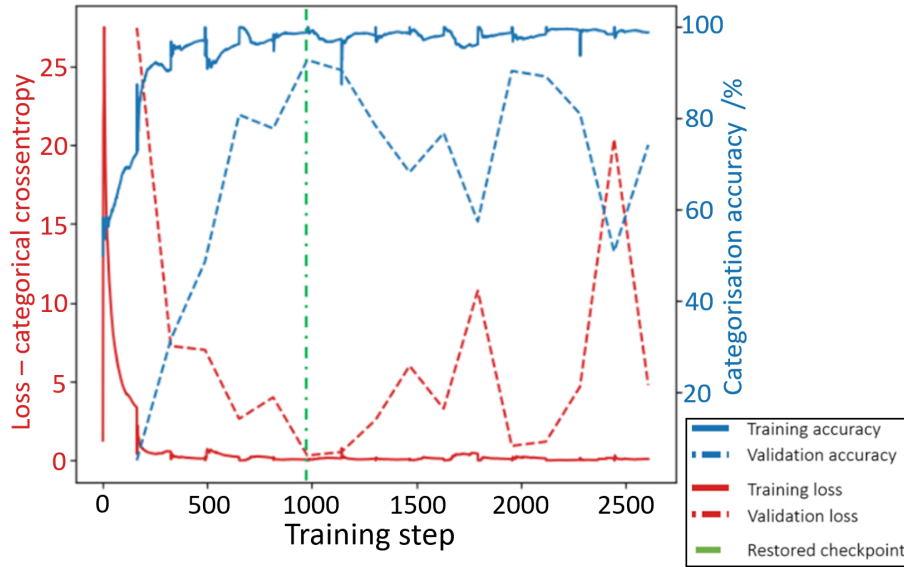


Figure 7.11: Categorisation CNN training history. To prevent overfitting, ten percent of the dataset was used for cross validation and model weights restored to the maximum validation accuracy.

As can be seen in Figure 7.11, the validation loss reached a minimum at around 1000 training steps (batches of sixty-four images) after which overfitting began to occur. To prevent overfitting, two mitigation strategies were employed. The first strategy was to use an early stopping criterion which monitored validation accuracy and ceased the training procedure if no improvement was observed within ten epochs. The second strategy was to use a model checkpointing system which, once training is finished, restores the model weights to the point at which validation accuracy was a maximum. In this case, the maximum validation accuracy achieved was 96%.

When deployed, rather than simply taking the argmax of the output nodes to perform the categorisation, a ‘certainty threshold’ was set at 80%. That is to say, the input was only assigned to the predicted class if the value of the corresponding output node was larger than 80%; if this condition was not met, the image was instead categorised as ‘uncertain’.

The categorisation operation can be performed on the 1000 generated sur-

faces (a sub-sample of which is shown in Figure 7.8b). Of these 1000 images, 4.6% were classified as uncertain. This level of uncertainty is an indication that the generated data does in fact accurately capture the input space as the rate of uncertainty correlates with the misclassification rate of the model during validation. When inspecting which surfaces are misclassified, the majority bear similarities to the surfaces around the 90° angle relative to the powder bed, as was expected. An example of this is shown in Figure 7.12.

Forty-three of the forty-six ‘uncertain’ images fell into the category shown

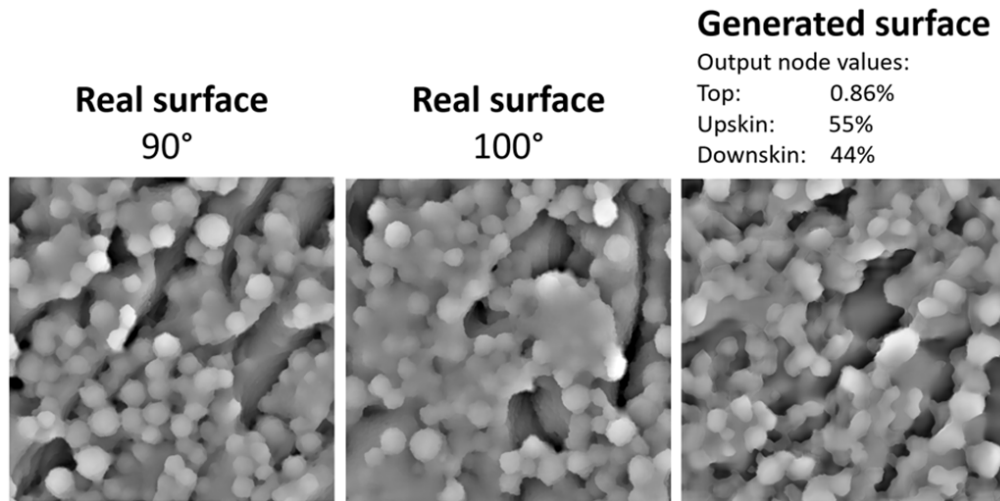


Figure 7.12: Comparison of encoded depth data for an example of a common misclassified surface with real surfaces around the borderline of the up-skin/down-skin categories

in Figure 7.12; the remaining three images had a different failure mode. These generated surfaces are classed as uncertain because they are not representative of the surfaces contained in the training data. Specifically in this case, they are all surfaces that have the distinctive weld track features of the top surfaces but also the large amount of particle agglomeration of the other surface types. Figure 7.13 shows an example of this surface.

As the unrepresentative images, such as the example shown in Figure 13, occur at such a low rate (0.3% in this test) and the model has been trained to sort them from the generated images which are representative of the real

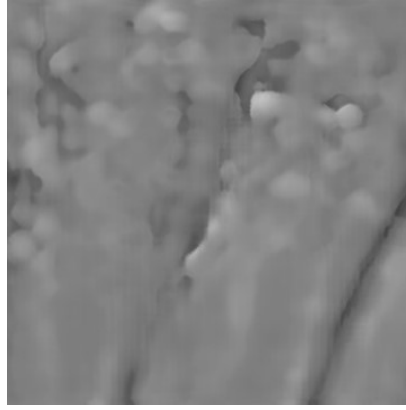


Figure 7.13: An unrepresentative image showing encoded depth data produced by the generator showing clear weld tracks and noise from agglomerated particles.

data, these surfaces are discarded and only the successfully categorised surfaces are used in analysis.

7.4.2 Quantitative comparison to real surface data

A benefit of the extended model is that surface statistics can be compared between the surface categories independently rather than averaged statistics for the complete dataset. As these surfaces have such different features, this method will provide a much more robust analysis than without this extension. First, parameters based on the surface height distribution relative to the mean plane are considered: Sq is defined as the root-mean-square height deviation and Sz is simply the maximum height [163,225]. The distributions of these two parameters are shown in Figure 7.14 for real and simulated surfaces across each surface category.

As can be seen, the distributions produced by the generator show good agreement with the training data. While the comparison of height param-

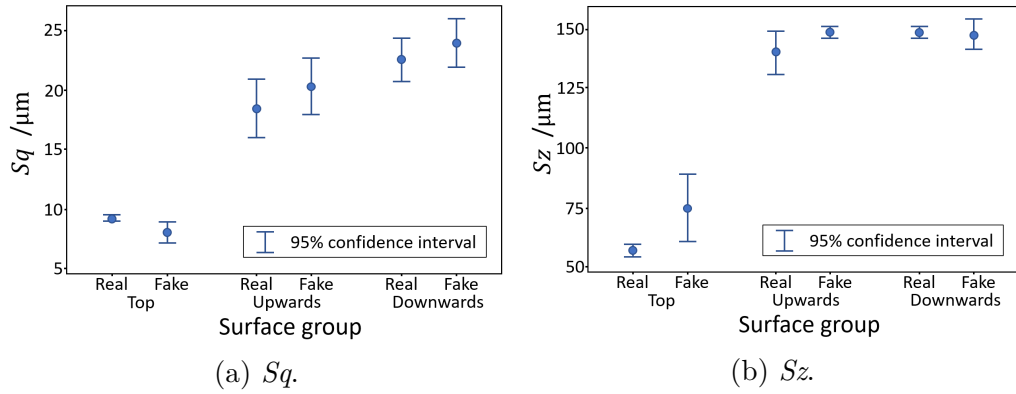


Figure 7.14: Comparison of the mean and 95% confidence intervals of ISO height parameters for real and generated AM surfaces.

eters is useful to begin to show good representation, it is not the full picture, for example, two very different surfaces could have similar Sz values. Considering spatial parameters in addition to amplitude parameters can provide a more complete comparison. Spatial parameters describe properties related to the distributions of the shape and size of the features that make up the surface texture. In this case, three parameters are evaluated: Sal , amplitude of the dominant spatial wavelength and dominant spatial wavelength. Sal is the fastest decay autocorrelation length, which is a measure of the distance from given point on the surface to a point which has minimal correlation with the starting point [163, 225]. The distributions of these three parameters are given in Figure 7.15.

In both Figures 7.14 and 7.15 the synthetic and real surface parameter distributions overlap in most cases or are different by small absolute amounts. For example, Figure 7.15a shows the mean autocorrelation length of top surfaces differs by only two microns despite the distributions not overlapping. It makes sense that the top surfaces are less well represented than the other surface categories as they make up a relatively smaller proportion of the dataset (only 5% of the input data were top surfaces com-

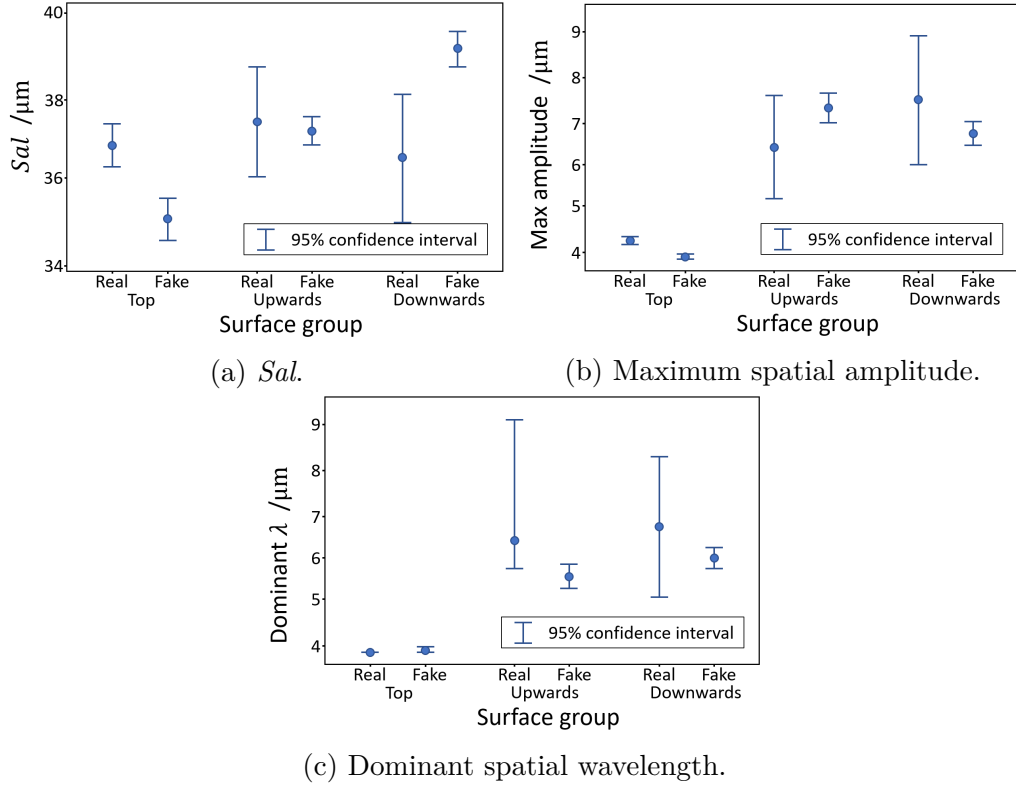


Figure 7.15: Comparison of the mean and 95% confidence intervals of ISO spatial parameters for real and generated AM surfaces.

pared to 48.2% for the remaining categories), this is simply because there were fewer top surfaces to measure on the ring artefact. This is evidence that the synthetic surfaces are not only qualitatively similar to the real data but quantitatively similar, and that any differences are small. This similarity shows that the output space of the model is at least partially representative of the input surfaces. It is interesting to note that, particularly for the spatial parameters, the distributions of the generated surfaces are much tighter than the distributions present in the training data. This tighter spread is likely due to the generator learning to represent a subset of the input space - this is discussed in Section 7.5.

7.5 Discussion of surface texture generation

The PG-GAN model was selected over other generative methods for two main reasons. Firstly, it has been developed specifically to encourage stability and variation in the generator outputs at high resolution. This means that the model is likely to learn to represent a larger portion of the input space than competing methods at the resolution of data within the example datasets. Secondly, recent variations of the PG-GAN have been developed for more specific applications, such as style transfer (see StyleGAN [226] and Cycle-GAN [51]), whereas the original PG-GAN implementation can be applied generically to any input image dataset.

As was noted at the end of Section 7.4.2, the distributions of areal surface texture parameters show good agreement between the real and generated surface but do not match exactly. Firstly, the variance in the generated data is, in general, smaller than the variance among the training data. It is a known shortcoming of GANs that commonly only a subset of the possible variation is represented by the trained model [227]. This is intuitive, as it is simpler for the model to learn to represent a subset of the input space to a high enough quality to trick the discriminator than to learn to represent the entire space. As stated previously, many of the features of the PG-GAN are specifically designed to increase variation in the output (this is discussed at length in the original PG-GAN paper [49]) but, at least for this application, there is still some work to do in this area.

A feature of using a ML approach is that the model will learn to represent patterns that are present within the training data. This means that any measurement errors present in the training data will be replicated in the synthetic surfaces. Replicating measurement errors could be disadvantageous to some applications where it is desirable to produce simulations of true surface topographies, however, it is an advantage if the application

calls for simulated measurement data from a real instrument. This method has been shown to be effective at generating both photogrammetry and FV measurement data, however, it is general to any technique where the data can be represented by a height map. Photogrammetric surface measurement operates at larger scales than most optical texture instruments and focus variation has relatively low resolution when compared to a technique such as coherence scanning interferometry, and these characteristics are replicated in the generated data. If the proposed approach was applied to another measurement technique, one would expect limitations inherent to that technique to be reproduced by the trained model.

Encoding the training data as a grayscale image does not effect the spatial resolution of the training data as each measured surface point is represented by a unique pixel value and the data is stored in a lossless format to ensure no compression artefacts are introduced. However, one shortcoming of the method is that the vertical resolution of the model is limited to 255 discrete pixel values. In the case of the AM dataset this introduces an uncertainty of $\pm 0.25 \mu\text{m}$ when encoding the input data. This is close to the $0.1 \mu\text{m}$ spacing between stacked focus variation images so unlikely to have a large effect on the data quality. Replacing the input encoded image with a floating point array would effectively eliminate these errors.

7.6 Example application - producing renders of AM parts

An application of the surface generator of particular use for this thesis is to produce realistic surface texture which can be applied to CAD data

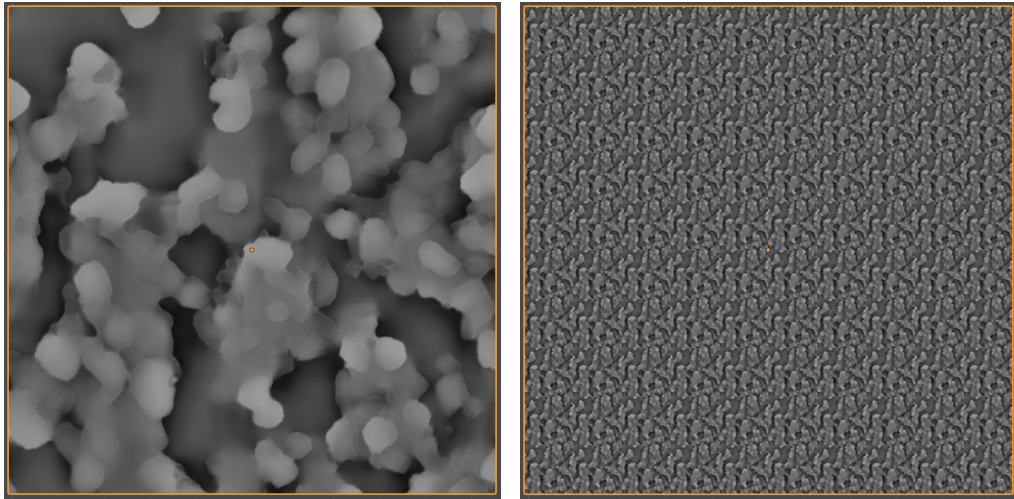
for the visualisation of AM parts, enabling photorealistic rendering of AM parts before they are produced. However, because the model generates only 1 mm square height maps as an output it is necessary to convert these height-maps into an infinitely and seamlessly repeating texture. The approach shown here was developed by Poliigon [228], a 3D art asset repository, which they refer to as UberMapping. The UberMapping approach uses a mosaic texture with random rotation to turn a non-repeating image texture into an infinitely repeating image texture which is visually seamless. The effect of applying this UberMapping node to an example texture produced by the generator is shown in Figure 7.16.

This UberMapping node is incorporated into a material shader for the Blender Cycles rendering engine [171], this shader can then be applied to any CAD data, allowing Cycles to accurately visualise how the part will appear once produced.

7.6.1 AM material shader

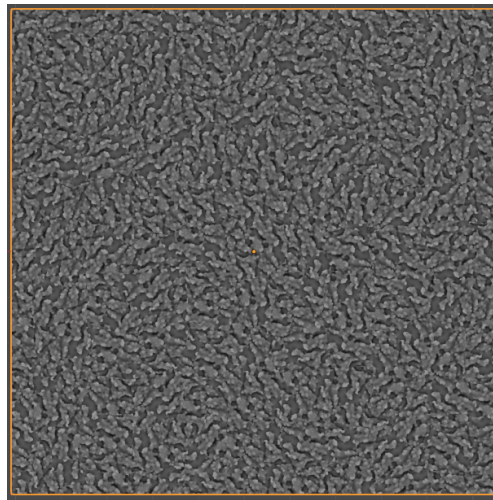
The infinitely repeating textures generated by the trained generator model and UberMapping node can be integrated into a complete material shader in Blender [171]. Figure 7.17 shows an outline of the shader model developed within Blender’s material shader node editor, the full shader model is also included in Appendix F. Each of the major components of the shader are described below.

The shader is comprised of four main blocks which can be seen in Figure 7.17. The top block mixes a set of metallic principled bi-directional scatter-



(a) Height map.

(b) Height map tiled.



(c) Height map with UberMapping.

Figure 7.16: Visualisation of how the Poliigon UberMapping node can be used to create an infinite and visually seamless image texture. (b) and (c) shown at 10:1 scale compared to (a).

ing distribution function (BSDF) shaders to produce the colour variation seen on Ti64 surfaces. The second block creates the weld lines often generated at sharp corners of a PBF part. The third block uses the face surface normals of the CAD file to create the layers visible on vertical faces of AM parts. The final block generates the underlying texture from the generated height maps, the surface normal is used to decide which surface category to take from the generator model (top, upskin or downskin). The UberMapped surface textures are then blended together using a further

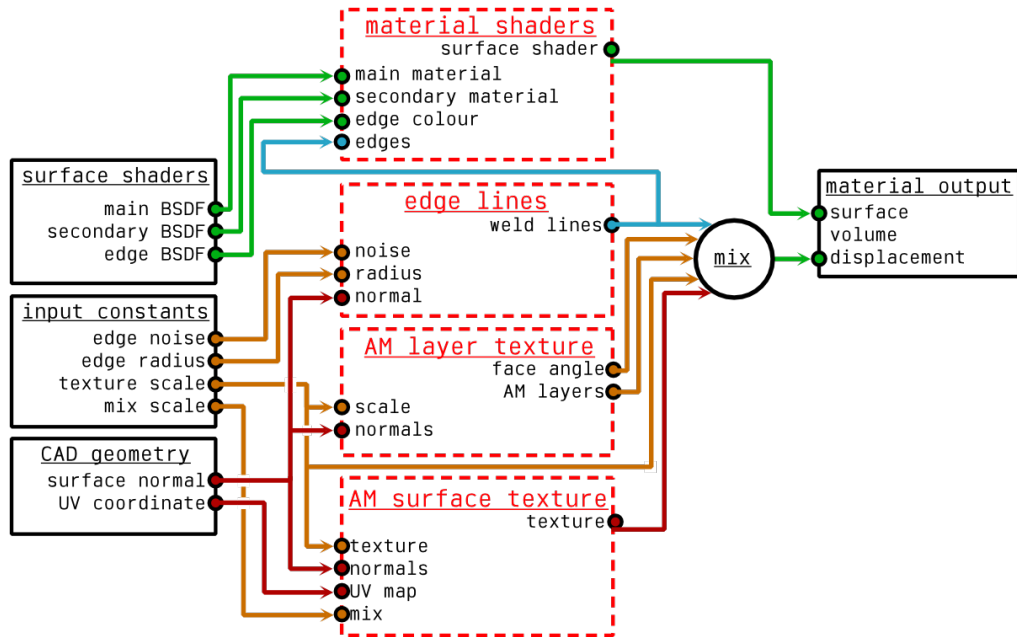


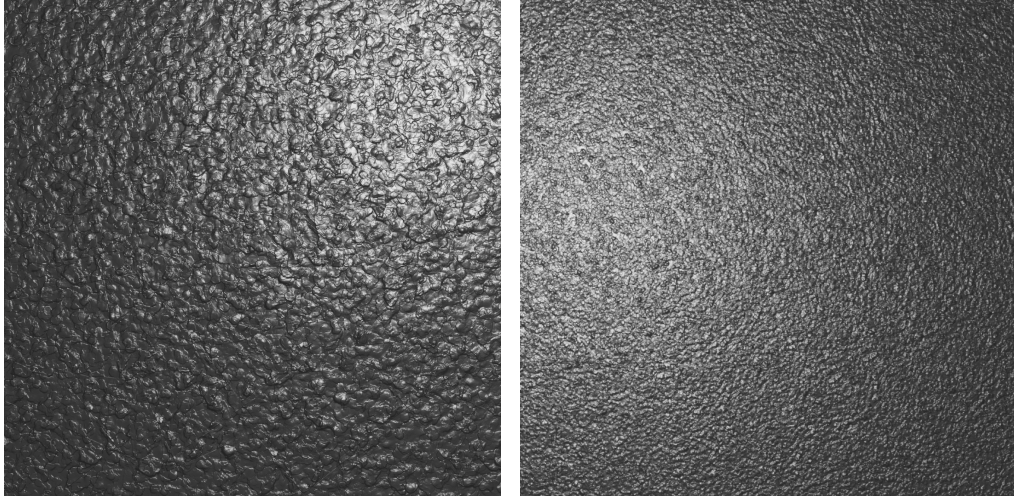
Figure 7.17: Material shader overview, full shader model included in Appendix F.

Polligon node, physically based rendering (PBR) mixing to create a smooth transition between textures from the generator.

Figure 7.18 shows the final material shader applied to a flat surface at a variety of scales, Figure 7.19 shows the shader applied to an example part.

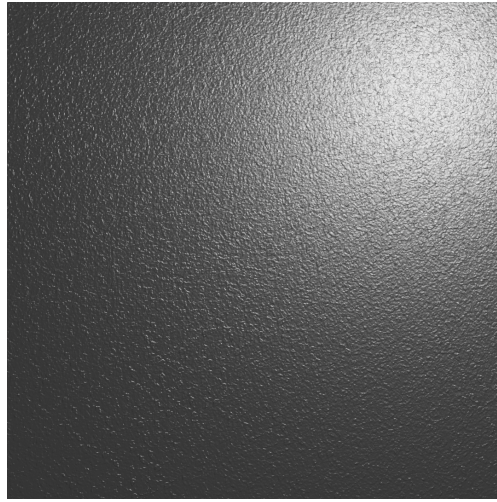
7.7 Surface texture generation conclusions

A novel approach to the generation of synthetic surface data has been proposed through exploiting an approach initially designed for the synthesis of high-resolution images. It has been shown that by encoding the surface height data into the grayscale channel of an image, a PG-GAN model can be trained to produce new data that represents a training set of images. By applying a process of dataset augmentation, the model is made robust to some transformations, such as rotation, and allow the initial measured dataset to be relatively small (fewer than 100 measurements). A further



(a) Large scale.

(b) Medium scale.



(c) Small scale.

Figure 7.18: Material shader applied to flat plane at a range of scales.

CNN can be used to categorise the surfaces produced into categories that are known in the initial dataset. This categorisation allows the model to produce surfaces of a desired type, rather than a random sample from the space of all possible surfaces that the model can represent. Furthermore, this categorisation allows for specific comparisons between the distribution of areal surface texture parameters over the categories of surfaces, rather than the full datasets.

Two case study datasets, one derived from photogrammetric measurements of industrially coated surfaces and the other from focus variation measurements of metal AM surfaces. It is shown, in both cases, that the generated

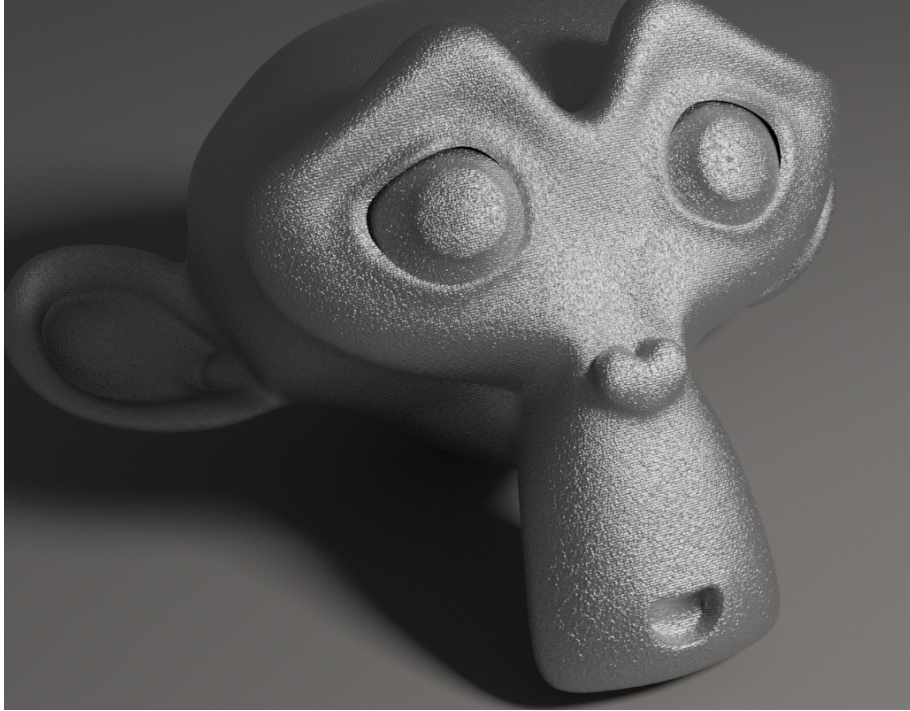


Figure 7.19: Material shader applied to complex 3D model.

surfaces are visually similar to those in the original dataset. In the case of the AM surfaces, it is shown that the approach successfully classifies 96% of the unseen data. The 4% of the data classified as ‘uncertain’ was due to fuzzy boundaries between the up-skin and down-skin categories and a small number of generated images that poorly represented the training data (0.3%). Finally, a quantitative analysis of both amplitude and spatial areal surface texture parameters was conducted. The distributions of these parameters for the synthetic data shows relatively good agreement with the distributions of the real data. There is an indication, due to the tighter distributions in the synthetic data, that only a subset of the possible real surfaces have been represented by the generator model. This lack of variation is a known shortcoming of the GAN and although the PG-GAN takes steps to increase variation in the generator, in the case of the AM surface data at least, this is an open issue.

As the surfaces used have been shown to be quantifiably representative of those within the training data, large quantities of synthetic surface data

can be produced extremely quickly to go on to be used in a variety of possible applications including, but not limited to, training statistical models, virtual instruments, and accurate surface simulation and rendering.

Finally, an example use of the surface generator was shown. A material shader was developed to take output surfaces from the generator and mix them into a visually seamless infinitely scaling texture which can be applied to CAD models in renders. This shader can be integrated into the overall measurement pipeline to generate training data for the monocular pose estimation technique presented in Chapter 8.

The contributions to science given by the work in this chapter can be summarised as: a novel approach to the generation of simulated surface texture data which improves on the state of the art by being less computationally intensive than physics based simulation but more representative than a pure mathematical representation of a surface, and is able to produce near-unlimited new data.

It was intended to use this model to generate texture in the training set of the monocular pose estimation model in Chapter 8) and in the future hardware integration of this pipeline this model would indeed be used for this purpose. However, a simpler model was used instead based on statistical surface parameters as the monocular model was completed before the PG-GAN had been developed and trained.

7.7.1 Future work on surface generation

A simple further next step in analysis of this work would be to consider hybrid parameters such as Sdr (a measure of total developed area of all tessellations) and Sdq (mean quadratic slope) which could provide further insight into the synthetic surfaces.

Taking the model further will include the use of principal component analysis (PCA) on the early activation layers of the generator model to map the latent space. An implementation of PCA on similar models has been presented recently (called GANSpace [229]) and has been shown to allow the development of semantic control over the generator output. For the application to surface texture, this could allow the generation of surfaces with prescribed properties. An area of particular interest is to generate interpretable controls for creating synthetic surfaces representative of those which would be produced through a specific combination of process parameters. Additionally, further work refining the model architecture to be more performant specifically on datasets of the form presented here could yield generator models with greater stability and variation.

Chapter 8

Pose estimation

Early work towards the findings presented in this chapter has been presented in a series of conference papers:

Eastwood J, Sims-Waterhouse D, Piano S, Weir R, Leach R K 2019 Autonomous close-range photogrammetry using machine learning *International Symposium on Measurement Technology and Intelligent Instruments*.

Eastwood J, Zhang H, Isa M A, Sims-Waterhouse D, Leach R K, Piano S 2020 Smart photogrammetry for three-dimensional shape measurement *Proc. SPIE* **11352** 43-52.

Eastwood J, Sims-Waterhouse D, Piano S, Leach R K 2020 Pose estimation from a monocular image for automated photogrammetry *20th International euspen Conference*.

The final novel development required to complete the software pipeline, as was shown in Figure 1.3, is the ability to establish the initial location and rotation of the object within the measurement volume relative to the camera system. Doing so in an autonomous way removes the need for accurate manual placement of the object or specialised fixturing, allowign the part to placed anywhere within the measurement volume. Furthermore, this initial pose can be used as an initial alignment estimation to fully automate CAD registration via ICP during data analysis. Figure 8.1 shows how this pose estimation fits within the pipeline.

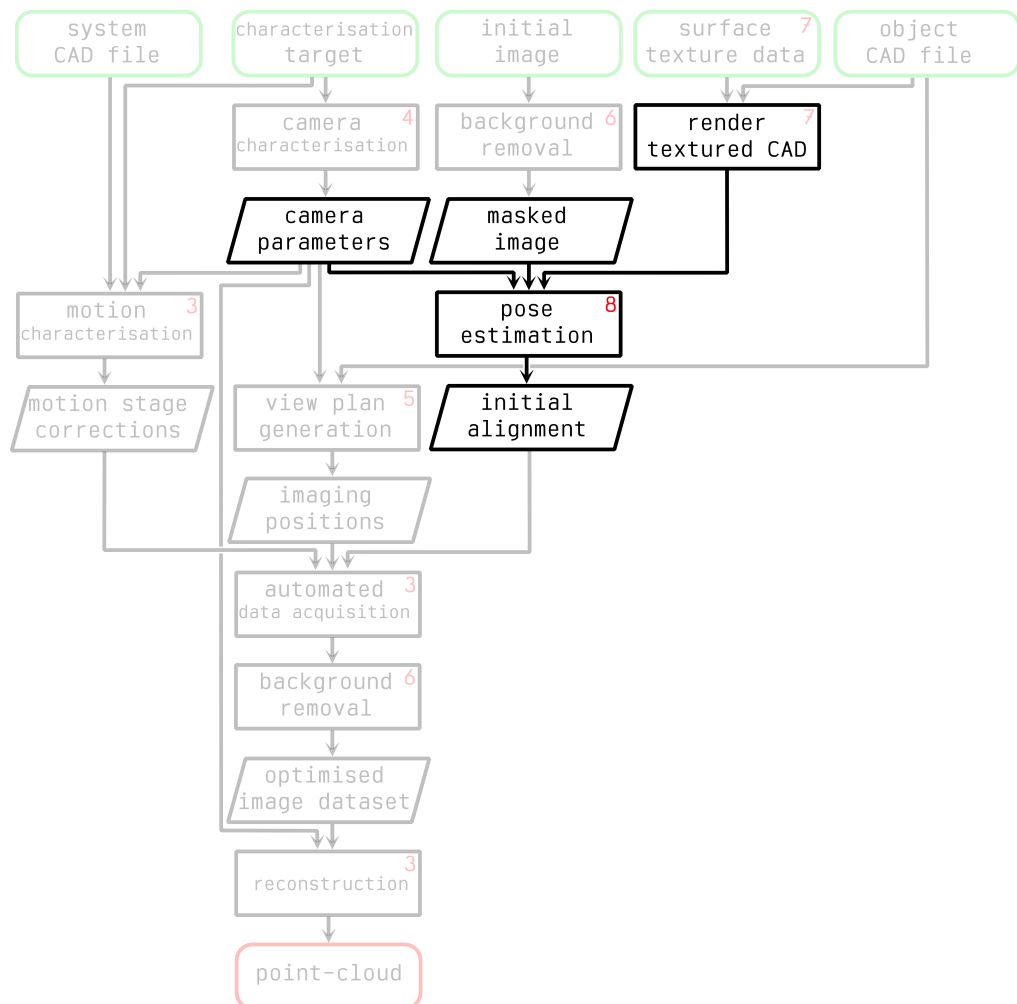


Figure 8.1: Pose estimation shown within the overall proposed measurement pipeline.

In this chapter, two possible solutions to the problem of pose estimation are presented. The first relies on the textured CAD data as can be produced by the work in Chapter 7, the second relies on the background masks produced by the work in Chapter 6. Both approaches are evaluated and their respective pros and cons discussed. In summary, the first method can make a prediction from a single image but requires pretraining on a given part. The second method is more general, does not require pretraining but requires *a-priori* knowledge of the relative orientation between a minimum of two images. Therefore, the first method is more suitable to automated measurement of a single part, for example as a verification step in an assembly line, whereas the second method can be used for more ad-hoc measurement needs.

8.1 Introduction to pose estimation

The estimation of the initial pose of an object within the measurement volume is critical to enabling the automation of optical coordinate measurement. The view-planning approach presented in Chapter 5 provides a list of imaging positions relative to the CAD file's coordinate system. Therefore, to directly use these imaging locations the object must be perfectly aligned with the coordinate system. Achieving this accurate positioning of the part would likely require custom fixturing to be developed on a per part basis and it would fall to the measurement operator to ensure this alignment is conducted precisely [230]. An alternate approach is to allow the CAD and measurement volume coordinate systems to be arbitrarily misaligned, calculate a prediction of this misalignment and correct the view plan accordingly. This approach of adjusting the view plan to account for any offset in the part's position within the measurement volume then allows

the optimised measurement to be conducted autonomously with no input required from the operator.

Previous methods of object pose estimation commonly only define the bounding box of a given object [231, 232]. For some geometries, the aspect ratio of the bounding box uniquely defines a pose. However, in the majority of cases this is not the case and there is not a singular solution meaning some further processing would be required to define the full six DoF pose [233]. Current ML approaches are often incredibly domain specific, for example there is a wealth of work covering human pose estimation [234, 235] and human hand pose estimation [236]. ML approaches to general object detection often require some depth information to have already been calculated in the form of depth images which requires an additional computationally expensive data processing step [237, 238]. Recent developments allow for single image pose estimation, but are limited by requiring the object to be from a known class [239]. It is also common to rely on fiducials [240] to perform photogrammetric triangulation. For the application of automated measurement it is highly desirable to avoid the need for fiducials because, as with fixturing, this is a source of user reliance. This chapter presents two methods for estimating the relative pose of an object. The first, referred to as the monocular method, takes a single image of the measurement volume and a CNN trained to recognise a set of artifacts detects which artefact is currently being measured and its relative pose in the scene. The second, referred to as the stereo method, takes a set of two or more input images with known relative pose between the imaging locations. The background removal algorithm presented in Chapter 6 is used to create a binary mask from each position, the centroids of these masks are then triangulated to give an initial rough alignment. A set of predicted masks are rendered from each imaging position and a global minimisation is conducted, refining the pose to minimise the difference between

the real and predicted binary masks.

Both of the pose estimation methods are tested against synthetic data but also are shown to generalise well onto real photogrammetric input data. Each method has its own advantages and disadvantages over the other method. The monocular method works from a single image, makes predictions at high speeds, and can be trained to operate on a range of objects but requires further training if a new object is required to be measured and the data generation and training times are relatively high. The stereo method does not require any pre training or dataset, it can be used on any part that fits in the field of view of the measurement system, but is slower to make predictions and requires a characterised multi-view system. A more in depth discussion of how the two methods compare is given in Section 8.6.

8.2 Monocular method

To estimate pose information from a single image, a custom CNN architecture is developed. A multi-task learning (MTL) approach is adopted wherein a model is trained to perform two or more tasks in parallel, in this case object categorisation and pose regression, where knowledge gained from performing the secondary task (object categorisation) improves performance on the primary task (pose estimation) due to a shared architecture. To train the model, a simulated version of the MMT system is developed from the system's CAD data. This simulation is used to create a large amount of labelled synthetic images on which to train the model, this process is described fully in Section 8.2.1. In this case the model is trained to recognise the four simple artefacts introduced in Section 3.4.2.1. To make the rendered images as photorealistic as possible, a material shader is de-

veloped which is applied to the object CAD when rendering each training sample, which is outlined in Section 8.2.1.1. Finally, the model is tested on real photogrammetric measurements of each artefact. By registering the CAD data to the measurement data, the predicted pose can be compared to the value given by ICP which is assumed to be close to the ground truth.

8.2.1 Generation of training data

To train a CNN with minimum generalisation error requires a large, labelled and representative dataset. It would be highly impractical, if not impossible, to generate manually, so instead a synthetic dataset was created from a simulation of the MMT system, an extracted material texture and the CAD data of the object. Figure 8.2 shows a comparison between the real and simulated photogrammetric instruments – the simulation was created within Blender and all synthetic images were generated using the Cycles render engine [171].

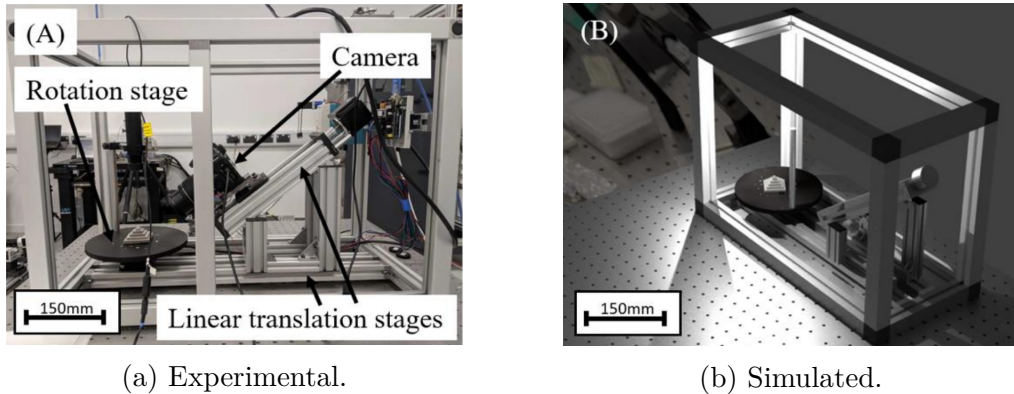


Figure 8.2: Real and simulated versions of the MMT system.

As stated, it is important for the dataset to be as representative as possible of the set of all valid input images. This requires a large spread of object and camera positions, and as photo-realistic render as possible. The former requirement is satisfied by generating 10,000 images, each with

a random possible camera and object configuration; the size of this dataset should ensure a good coverage of possible configurations. Photorealism is achieved by using a render engine, representative lighting conditions, known characterised intrinsic camera parameters, and a realistic material shader applied to the CAD data.

8.2.1.1 Material shader

Because the results in this section were gathered before the development of the work shown in Chapter 7, an intermediate shader was developed which was similarly drawn from the real physical properties of the surface. This intermediate shader was developed using the node editor within Blender using the glossy bi-directional scattering distribution function (BSDF) node, the full shader is shown below in Figure 8.3.

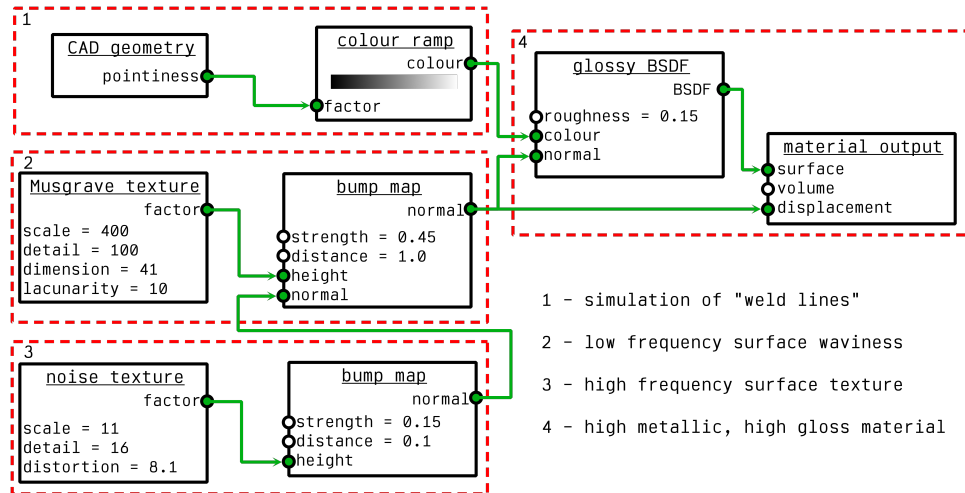
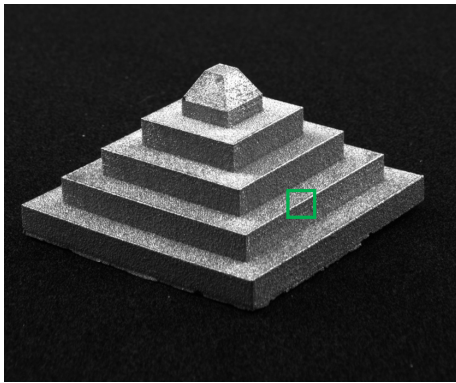


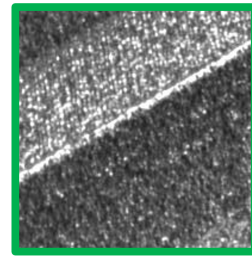
Figure 8.3: Intermediate material shader.

The parameters for the shader were determined by taking a photogrammetric scan of an object produced from the same material and process as the object of interest (in this case Ti-64 and metal PBF respectively). The Fourier power spectrum [241] of this surface was then used to find the

most prominent spatial frequencies. A bump map was then generated using this spatial frequency distribution, capturing both the surface waviness and high-frequency texture due to individual particles. This was combined with a colour ramp at edges, simulating weld-lines present at sharp edges on PBF parts, Figure 8.4 shows the formation of one such of these weld-line features present on the Pyramid artefact.



(a) Image of Ti64 artefact.



(b) Weld-line detail.

Figure 8.4: Example of weld-line artefact visible at sharp corners on PBF parts.

Using this shader, the CAD models of each artefact can be rendered within the simulated measurement setup shown in Figure 8.2b. Figure 8.5 shows an example render of the Pyramid artefact and how it compares to a real image.

As can be seen in Figure 8.5 the rendered images are qualitatively very similar to real images captured on the MMT system. Although the intermediate shader considers some physically measured attributes, and qualitatively looks representative of the real texture, the generator based material shader presented in Section 7.6 has been shown quantitatively to accurately represent real surfaces and as such will be used in any future work.

As this dataset is rendered through simulation, the ground-truth labels are

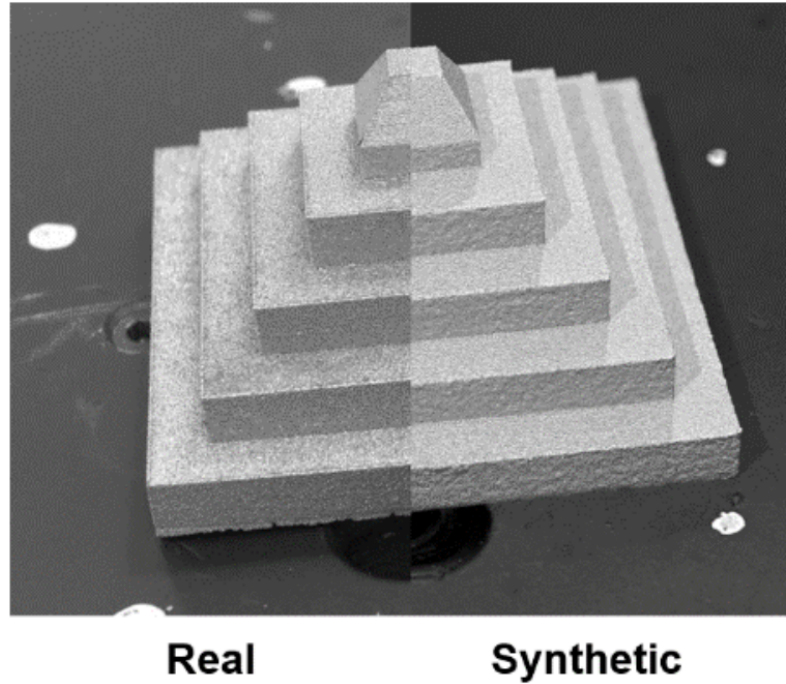


Figure 8.5: Comparison of real and synthetic image of the pyramid measurement artefact. The dashed connection allows information flow during the forward pass only, not during back propagation.

known implicitly and are saved for use during training. A total of 5000 images of each artefact were generated at random combinations of object rotations and translations on the stage, stage rotations and camera locations. These data were then converted to grayscale and down sampled to a size of (409×300) pixels for dimensionality reduction. The pixel values and labels were then scaled to the interval $\{0, 1\}$ to promote numerical stability.

8.2.2 Model

A custom CNN architecture was developed to perform two operations - first, to categorise which of a set of known CAD files is the one currently being imaged; and second, to regress the six DoF relative pose of that part. Although the categorisation task is not necessarily required to perform the

pose estimation, it has been shown that MTL can improve performance on the primary task and was found to be beneficial in this case in some initial prototyping. The monocular pose estimation architecture developed is shown in Figure 8.6.

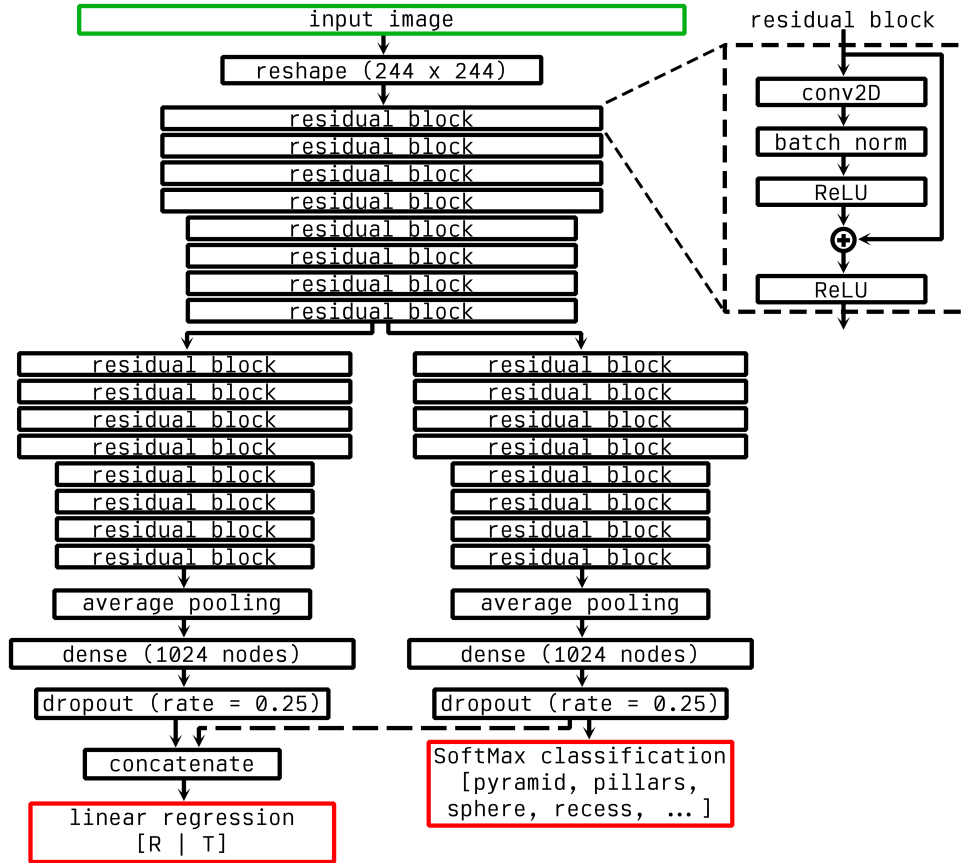


Figure 8.6: Monocular pose estimation CNN. The dashed connection allows information flow during inference only, not during back propagation.

The architecture makes use of residual blocks as was introduced in Section 2.2.2.1 and shown in Figure 2.15. From a shared backbone for high level feature extraction useful to both tasks, the model splits into a twin architecture with separate weights. One arm leading to a cross-entropy classification of the current artefact, the second arm leading to a LogCosh regression of the pose. The pose is parameterised by the translation vector \mathbf{T} which is regressed directly, and the rotation matrix \mathbf{R} which is regressed as an axis-angle representation. The axis-angle representation of rotation is given by a three parameter vector \mathbf{a} such that the corresponding unit

vector $\hat{\mathbf{a}} = \frac{\mathbf{a}}{|\mathbf{a}|}$ represents the axis of rotation and the vector magnitude $|\mathbf{a}|$ represents the angle of rotation about that axis. The two loss function values for each arm are combined into a global loss which is then back-propagated to update the model weights. Additionally, the final layer from the categorisation branch is concatenated with the final layer of the regression branch. This allows the regression task to be informed by the result of the categorisation task. During back propagation gradient is not permitted to flow through this concatenation so that the categorisation arm is not influenced by the results of the pose estimation arm. The Adam optimiser [42] was used with an initial learning rate of 0.0001. An early stopping criterion was used to cease training and restore the model weights to the best performing state if the validation loss did not increase for ten epochs.

8.3 Stereo method

If the measurement system has a binocular camera system, the stereo baseline between the cameras can be exploited to provide additional information. In this proposed pose estimation method, first an initial pair of images is captured, then the method presented in Chapter 6 is used to create binary masks of the object. The centroids of the two masks can be triangulated using the stereo baseline and characterised camera parameters providing an initial location estimation. Using the CAD of the object, a new binary mask can be rendered from the initial prediction. By defining a loss function between the estimated binary masks and the real binary masks an optimisation can be performed to minimise the difference between the real and estimated masks by updating the predicted pose. What results is an estimation of the pose of the object which provides a minimum difference

between the real and estimated object masks.

8.3.1 Calculating initial alignment

Figure 8.7 shows the overall procedure for establishing the initial alignment.

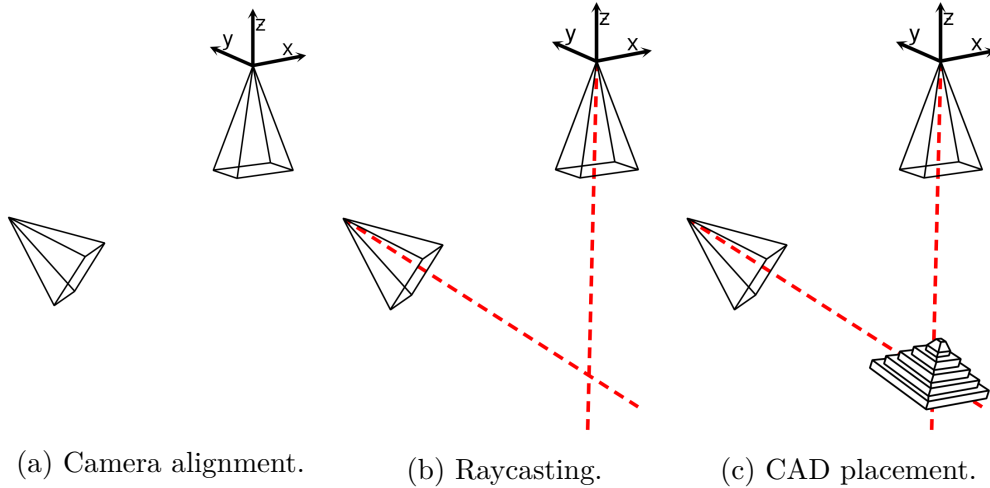


Figure 8.7: Initial alignment of CAD data using stereo binary image masks.

First, the extrinsic matrix $[\mathbf{R}|\mathbf{T}]$ of camera 1 relative to camera 2 must be determined. This is achieved using the baseline characterisation procedure outlined in Appendix C. Using the characterised extrinsic matrix, both cameras can be placed in the same coordinate system as shown in Figure 8.7a. Rearranging the camera model given in Equation 2.9, a ray can be cast from the camera principle point through a pixel location using,

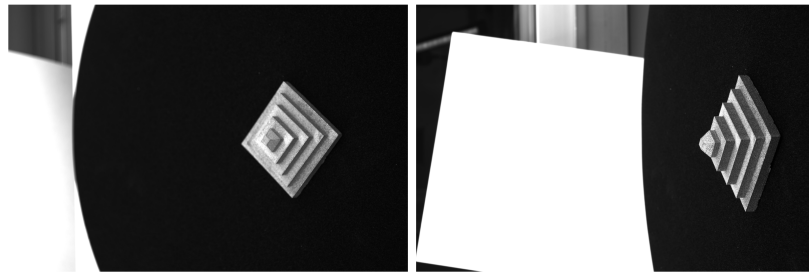
$$[X, Y, Z, 1]^T = (\mathbf{K} \cdot [\mathbf{R}|\mathbf{T}])^+ \cdot [x, y, 1]^T, \quad (8.1)$$

where \mathbf{A}^+ is the pseudo-inverse of the matrix \mathbf{A} given by,

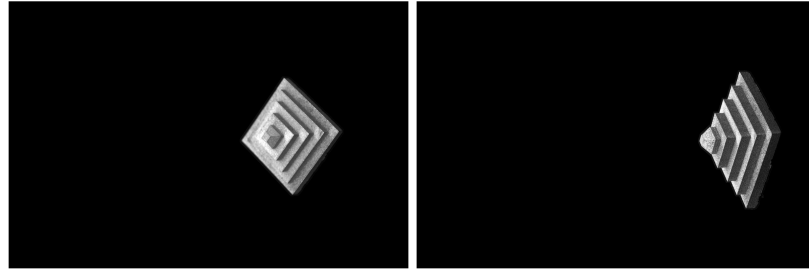
$$\mathbf{A}^+ = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T. \quad (8.2)$$

Using the method proposed in Chapter 6, an image captured from each image can be converted into a binary mask of the object, where pixels on

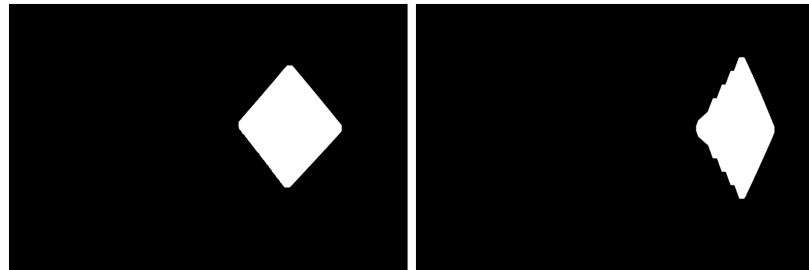
the object's surface are labelled one and all other pixels labelled zero. The centroids of the binary mask pair are then calculated by finding the average position of all the non-zero pixels. An example of these binary masks is given in Figure 8.8 with the calculated centroid locations shown in Figure 8.8d.



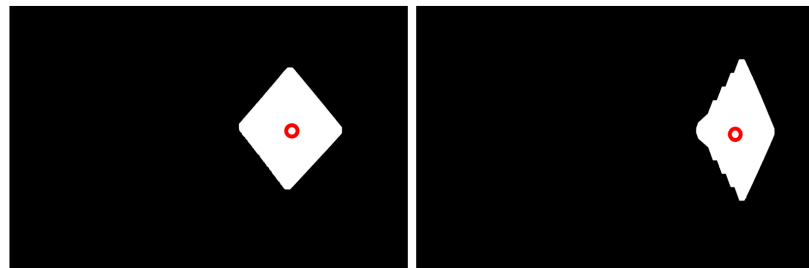
(a) Stereo image.



(b) Background removal.



(c) Binary mask.



(d) Centroid shown in red.

Figure 8.8: Stereo image pair taken by the Taraz system.

Using Equation 8.1 rays can be cast from the camera principle point

through a given pixel coordinate. A ray is defined by a direction unit vector and a origin point. A ray is cast from each camera origin through each centroid location as shown in Figure 8.7b. The point at which the two cast rays meet is taken as an initial estimate of the location of the object relative to the imaging system. Dependent on the part geometry, it is unlikely that the two cast rays will actually intersect each other. Instead, the point at which the two rays are closest to intersecting is calculated, using the function given in Appendix G.2, by finding the mid point of the line which is perpendicular to both rays. The CAD data representing the part being measured is then placed at this location with a random rotation as is shown in Figure 8.7c, this pose is taken as the initial alignment. Figure 8.9 shows an example alignment using the stereo image pair which was shown in Figure 8.8a.

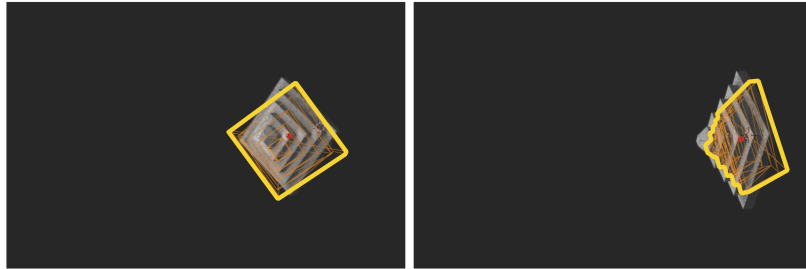


Figure 8.9: Example initial alignment result. Predicted alignment shown in yellow wireframe overlaid over each image.

It can be seen that the translation of the CAD is close to the true location however the rotation is considerably less accurate. This is logical as so far only translation information has been inferred while the rotation has been randomised.

8.3.2 Pose optimisation

Using the Blender API and the characterised camera parameters, a predicted binary mask is rendered from each camera based on the initial

alignment of the CAD data given by the method presented in the previous section. Assuming the cameras have been characterised correctly, the only variables controlling the predicted binary mask are parameterised by the six DoF pose of the CAD data. The six DoF is given by a translation vector $\mathbf{T}_{CAD} = [x, y, z]^T$ and a rotation given as an axis-angle representation. A loss function representing the pose error can be formulated as the sum of the magnitude of the pixel difference between the real and predicted binary masks,

$$Loss(\mathbf{T}_{CAD}, \mathbf{a}) = \frac{\sum_{u=0}^U \sum_{v=0}^V \|real(u, v) - predicted(u, v, \mathbf{T}_{CAD}, \mathbf{a})\|}{2UV}, \quad (8.3)$$

where (U, V) is the resolution of the binary masks, $real(u, v)$ is the pixel value of the real binary mask at pixel coordinate (u, v) , and $predicted(u, v, \mathbf{T}_{CAD}, \mathbf{a})$ is the pixel value of the predicted binary mask (rendered with pose $\mathbf{T}_{CAD}, \mathbf{a}$) at coordinate (u, v) . Figure 8.10 shows the magnitude pixel difference between the real mask shown in Figure 8.8c and the mask rendered from the initial alignment prediction shown in Figure 8.9.

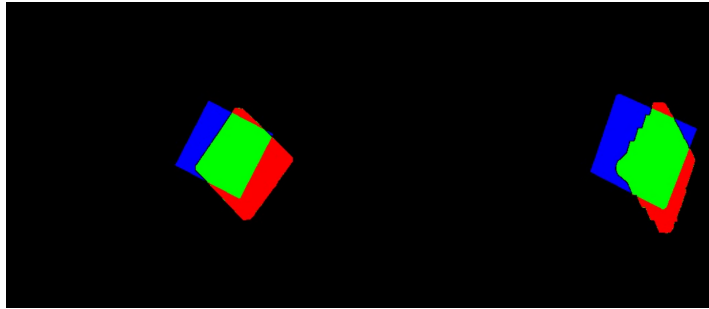


Figure 8.10: Loss function visualised on the initial pose prediction given by the rough alignment procedure. Correctly classified pixels shown in green, misclassified background pixels shown in blue, misclassified object pixels shown in red.

The CAD pose can now be optimised to minimise the loss function given in Equation 8.3. Two approaches were explored, direct search via Powell's method [242] and a gradient descent method via the Broyden–Fletcher–Gold-

farb–Shanno (BFGS) algorithm [243]. Both optimisation schemes are implemented using `sp.optimize.minimize` from the SciPy library [244].

8.3.2.1 Powell’s method

The main advantage of Powell’s method is that it does not require the calculation of the loss function gradient. Here instead, the algorithm begins with a set of N search vectors $\{\mathbf{s}_1, \dots, \mathbf{s}_N\}$ where N is equal to the open DoFs in the loss function (in this case $N = 6$) and each vector \mathbf{s}_n is a normal vector aligned with each parameter. A golden-section bi direction search [245] is then performed to find the loss function minima over each search vector. These minima can be formulated as sums over each search vector as,

$$\{\mathbf{x}_0 + \alpha_1 \cdot \mathbf{s}_1, \mathbf{x}_0 + \sum_{i=1}^2 \alpha_i \cdot \mathbf{s}_i, \dots, \mathbf{x}_0 + \sum_{i=1}^N \alpha_i \cdot \mathbf{s}_i\}, \quad (8.4)$$

where \mathbf{x}_0 is the initial estimate and α_i is the scalar determined in the bi-directional search. From these minima, a new estimation is made as the linear combination of the minima along each search vector,

$$\mathbf{x}_1 = \mathbf{x}_0 + \sum_{i=0}^N \alpha_i \cdot \mathbf{s}_i. \quad (8.5)$$

The search vector with the largest α_i is removed from the list of search vectors and is replaced with the direction of the new estimation from the previous estimation ie. $\mathbf{x}_{n+1} - \mathbf{x}_n$. This process is repeated until the distance $|\mathbf{x}_{n+1} - \mathbf{x}_n|$ converges to a small value, in this case 10^{-6} was found to provide a good comprise between precision and speed.

8.3.2.2 BFGS algorithm

BFGS is a gradient based minimisation approach which approximates the Hessian matrix \mathbf{H} of second-order partial derivatives to precondition the gradient. In brief, from an initial guess \mathbf{x}_0 a direction vector \mathbf{s}_1 is determined from,

$$\mathbf{s}_1 = -\mathbf{B}_0 \cdot \mathbf{J}_{Loss(\mathbf{x}_0)}, \quad (8.6)$$

where \mathbf{B} is an approximation to \mathbf{H}^{-1} and \mathbf{J}_f is the Jacobian matrix for first-order partial derivatives of some function f . As with Powell's method, a line search is performed to find the step size α_1 in the previously calculated direction \mathbf{s}_1 which minimises the loss function. A new estimation \mathbf{x}_1 is then calculated from $\mathbf{x}_1 = \mathbf{x}_0 + \alpha_1 \cdot \mathbf{s}_1$. Finally, the estimation of the inverse Hessian is updated from the gradient change $\Delta\mathbf{J}_1 = \mathbf{J}_{Loss(\mathbf{x}_1)} - \mathbf{J}_{Loss(\mathbf{x}_0)}$ by,

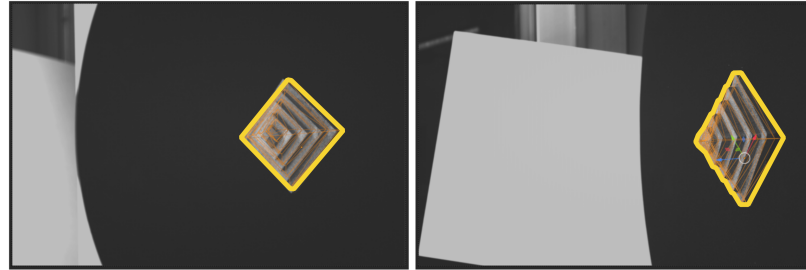
$$\mathbf{B}_1 = \mathbf{B}_0 + \frac{(\mathbf{s}_1^T \Delta\mathbf{J}_1 + \Delta\mathbf{J}_1^T \mathbf{B}_1 \Delta\mathbf{J}_1) \cdot (\mathbf{s}_1 \mathbf{s}_1^T)}{(\mathbf{s}_1^T \Delta\mathbf{J}_1)^2} - \frac{\mathbf{B}_1 \Delta\mathbf{J}_1 \mathbf{s}_1^T + \mathbf{s}_1 \Delta\mathbf{J}_1^T \mathbf{B}_1}{\mathbf{s}_1^T \Delta\mathbf{J}_1}. \quad (8.7)$$

On the first iteration the identity matrix is used as as the first estimate of the inverse Hessian, ie. $\mathbf{B}_0 = \mathbf{I}$ such that the first iteration is equivalent to an unconditioned gradient descent - as the optimisation progresses, iterative application of Equation 8.7 refines this estimation.

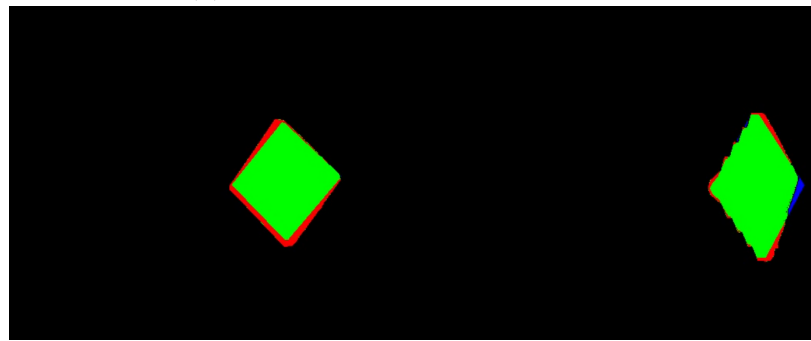
In this application, the loss function gradient cannot be calculated either analytically or through automatic differentiation [246]. This is due to the loss function implementation calling an external function to perform the rendering of the predicted binary mask. Instead, the Jacobian matrix is estimated numerically using a finite forward differencing scheme [247]. This requires an additional six frames to be rendered per gradient calculation but may lead to faster convergence. The step-size taken during finite differencing is reduced over the course of the optimisation as the pose prediction

converges.

Figure 8.11a shows the refined pose estimation overlaid on the input images while Figure 8.11b shows the minimised loss function.



(a) Refined pose overlaid on inputs.



(b) Loss function visualisation.

Figure 8.11: Example result of refined pose estimation.

8.4 Monocular pose estimation results

First the training results are given, including validation results on unseen synthetic images. Then, a 60 image scan of each artefact was taken. After reconstruction, ICP can be used to determine the location of the CAD within the measurement volume. The pose of the registered CAD data can then be used to recreate approximations of the ground truth pose data relative to each input image in the scan. The trained model is then tested on all 60 images of each scan and compared to against the previously calculated ground truth pose.

8.4.1 Monocular model training results

The model and validation losses from each output were recorded after each epoch of training. Fig. 8.12 shows the overall training and validation losses over the training period. It can be seen that the early-stopping criterion was met and the simulation terminated after fifty-five epochs.

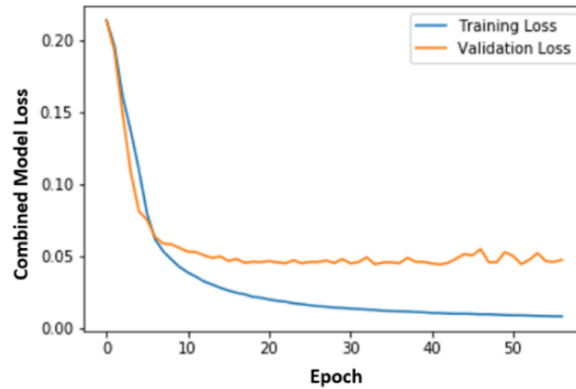


Figure 8.12: Training and validation losses of the initial network at each training epoch.

The relatively large gap shown in Figure 8.12 indicates that there is some generalisation error as performance on unseen data is worse than the performance on the training set. Splitting the combined loss into its individual components, Figure 8.13 shows the individual validation losses for each output. The results for the regression of the rotation of the part are shown here as Euler angles. In the results from the regression task in Figure 8.13a it can be seen that the loss in each (x, y, z) dimension is very low, with values less than 0.01. Comparing the translation losses to the rotational losses, it is clear that the model has more difficulty when predicting rotations. The generalisation error (a measure of prediction error on unseen data) of the model can be estimated by the final difference between the validation loss and the training loss, which in this case is approximately 0.04.

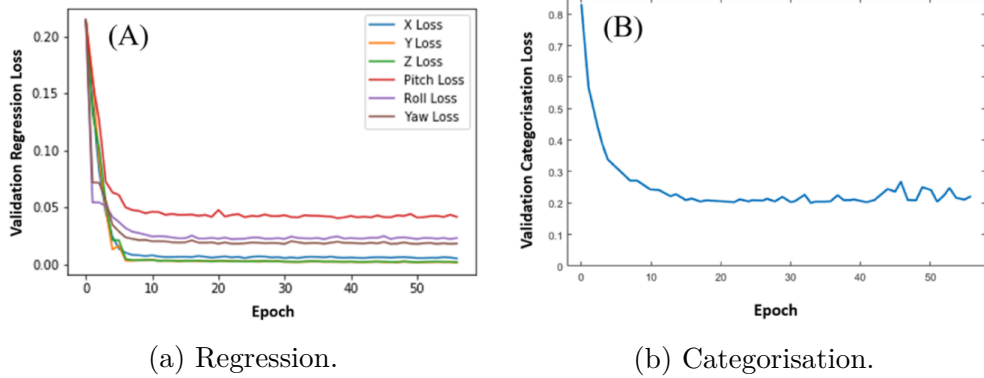


Figure 8.13: Separated loss contributions.

The categorisation loss, as shown in Figure 8.13b, can be used to determine the categorisation accuracy A_{cat} – the percentage of correct categorisations compared to the total number of categorisations given by,

$$A_{cat} = \frac{1}{N} \cdot \sum_{p=0}^N \left[\operatorname{argmax}(Q(i)) == \operatorname{argmax}(P(i)) \right]_p, \quad (8.8)$$

where N is the number of input images and p is an index referring to the current image. In this case, the categorisation accuracy is found to be 97%. Figure 8.14 is a visualisation of some example predictions; the predictions are shown as a wireframe overlaying the original image.

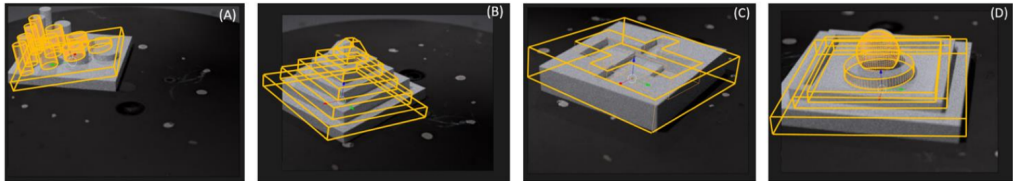


Figure 8.14: Example pose estimations for each artefact.

It can be seen qualitatively in these images that the estimations are relatively close to the true pose, with most of the error made in the estimation of part rotation. This conclusion is backed up quantitatively in Figure 8.13. The trained model can now be deployed to make predictions on real photographic inputs.

8.4.2 Results on real data

As described previously, sixty images were captured around each artefact using the MMT system and used to generate a dense point cloud – the ICP algorithm, with an initial estimation input by the user, was used to produce the ground-truth values with which to compare the predictions from the CNN. Figure 8.15 shows an example prediction made on an image of the Pyramid artefact.

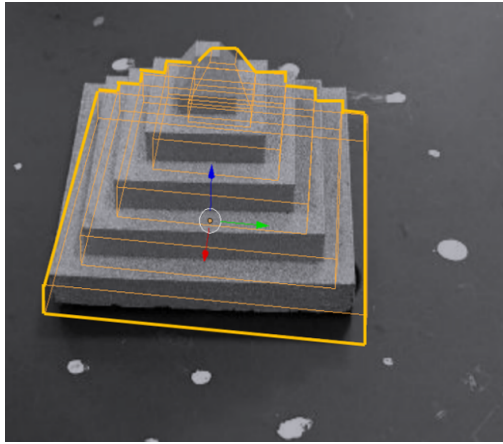


Figure 8.15: Example result on real image.

It can be seen in Figure 8.15 that the predicted pose is qualitatively similar to the quality of the pose predictions shown on the synthetic data in Figure 8.14. This is a good initial indication that the model generalises well onto real data and that the simulation presented in Section 8.2.1 produces data representative of the real system.

The model was used to make a prediction on each of the 60 images of each of the simple artefacts introduced in Section 3.4.2.1. The translational residual can be calculated simply as the mean absolute distance error. Defining a rotational error is not simple, in this case the residual Euclidean distance between the real and predicted axis-angle vectors are used to quantify this error given by,

$$\mathbf{R}_{residual} = \frac{1}{N} \sum_{i=0}^N |\hat{\mathbf{a}}_i - \mathbf{a}_i|, \quad (8.9)$$

where N is the total number of predictions, $\hat{\mathbf{a}}$ is the predicted axis-angle vector, and \mathbf{a} is the true axis-angle vector. The data shown in Table 8.1 shows the mean translation and rotational residual magnitude for each artefact.

	MAE			Residual magnitude	
	x /mm	y /mm	z /mm	T /mm	R /radians
Pyramid	19.78	11.33	0.67	22.80	0.19
Pillars	8.22	8.89	0.10	12.11	0.48
Sphere	16.44	4.22	8.89	19.16	0.45
Recess	7.33	11.56	2.67	13.95	0.62
Mean	12.94	9.00	3.08	17.00	0.44

Table 8.1: Error in position and translation estimate on real images.

As can be seen in Table 8.1, the network was tested on 240 real images of four different artefacts. The mean magnitude residual across all four artefacts was 17 mm in translation and 0.44 in rotation. It should be noted that although the residual magnitude in the rotation has units of radians, it does not directly represent the rotational error and so should not be interpreted this way. It is instead the Euclidean distance between the axis angle vector representations. It is given here just for future comparison with the stereo method.

8.5 Stereo pose estimation results

To test the binocular pose estimation a synthetic test was developed. In this test a CAD file is placed with a random pose relative to the camera and a ground truth mask is rendered with accompanying ground truth pose information. The pose estimation approach is then used on the synthetic mask image and the predicted pose can be compared to the ground truth

pose. This process can be automated, and as such can be tested on a large number of synthetic examples - in this case 250 sample pose estimations were performed. To test against real data, the same approach was taken that was used to test the monocular model. A scan of each artefact was conducted on the Taraz system, the CAD data was registered to the measured pointcloud allowing the pose relative to each stereo pair to be determined. The pose estimation was then tested on each stereo image pair in the scan.

8.5.1 Results on synthetic data

To generate the synthetic test set, the Blender "Suzanne" mesh was used which is shown in Figure 8.16.

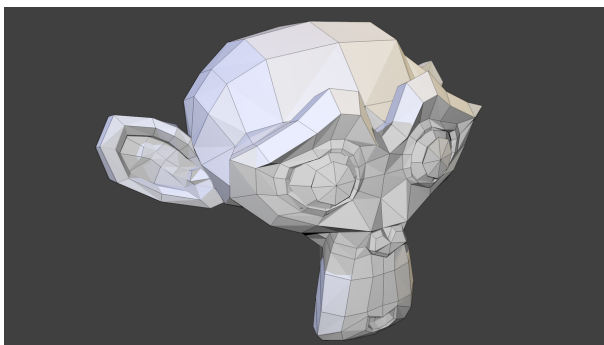


Figure 8.16: Suzanne mesh.

This mesh was chosen as the features are relatively complex while having a relatively low number of vertices to preserve computational load when building the dataset. A set of 250 binary masks were simulated using the same approach used to create the predicted masks during optimisation. The mesh was placed at a random location and rotation within the field of view of each camera and the ground truth pose relative to the camera system was recorded. Finally the pose estimation procedure was run on each synthetic mask pair and the refined pose estimation was compared to

the real pose.

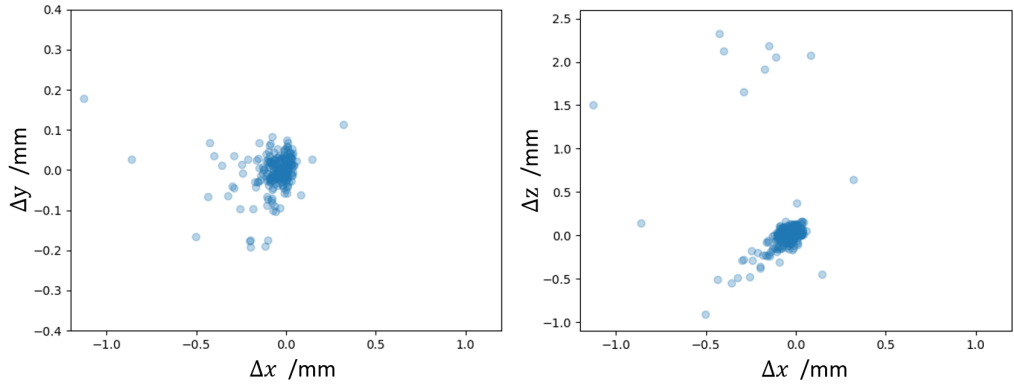
This process was used to evaluate the two optimisation approaches proposed; Powell’s direct search method, and the BFGS gradient descent method. Table 8.2 shows the mean loss and residuals over the synthetic dataset for both methods.

	Powell's method		BFGS	
	Mean	Std dev	Mean	Std dev
Loss /%	1.52	1.47	5.34	1.63
T residual /mm	0.18	0.39	2.75	2.61
R residual /rads	1.90	1.38	1.86	1.87
Time /s	165.38	33.07	75.40	44.94

Table 8.2: Pose optimisation results on the synthetic dataset using both minimisation methods.

As can be seen in Table 8.2, both optimisation methods achieve similar rotational prediction accuracy. However, there is a clear trade-off between translational accuracy and processing time. Powell’s method achieved a mean translational residual magnitude of 0.18 mm compared to 2.75 mm when using BFGS, a reduction of 98 %. This is likely due to the finite forward differencing scheme creating errors in the estimated Jacobian matrix. This improved location accuracy comes at the cost of processing time, with Powell’s method taking an average of 90 s more than BFGS to compute, this is due to the gradient method quickly converging on a minima. For the improved localisation of the object, Powell’s method is selected as the optimisation method going forward. Figure 8.17 shows a visualisation of the distribution of the translational residuals of the estimations given by Powell’s method on the synthetic dataset.

Figure 8.17 shows a very dense grouping around the point of zero residual error, with a small number of outliers. It can be seen that the outlying residuals in the z direction (which is aligned with the primary camera’s



(a) Pose estimation residuals in the x, y plane. (b) Pose estimation residuals in the x, z plane.

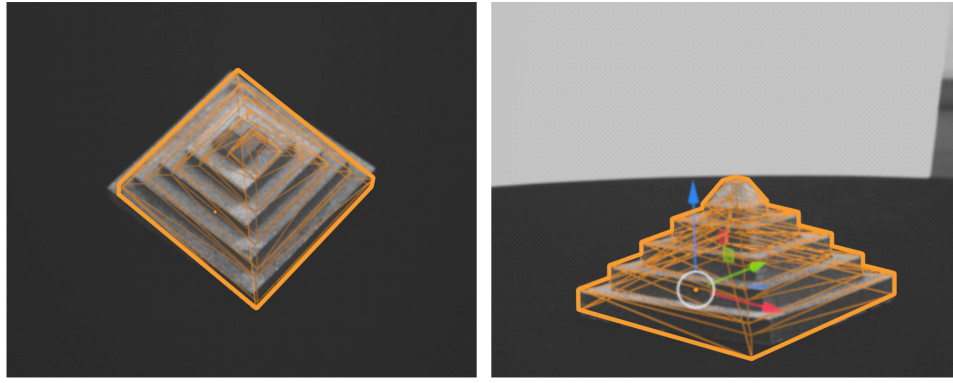
Figure 8.17: Pose predictions for each artefact, each prediction shown is overlaid in yellow wireframe on the input stereo image pair.

axis) can be much larger than is seen in the x, y plane. This is likely because the loss function is much less sensitive to changes in the z axis. Similarly, the smaller bias in the x direction, seen most clearly in Figure 8.17a, may be because the secondary camera's principle axis is close to this direction.

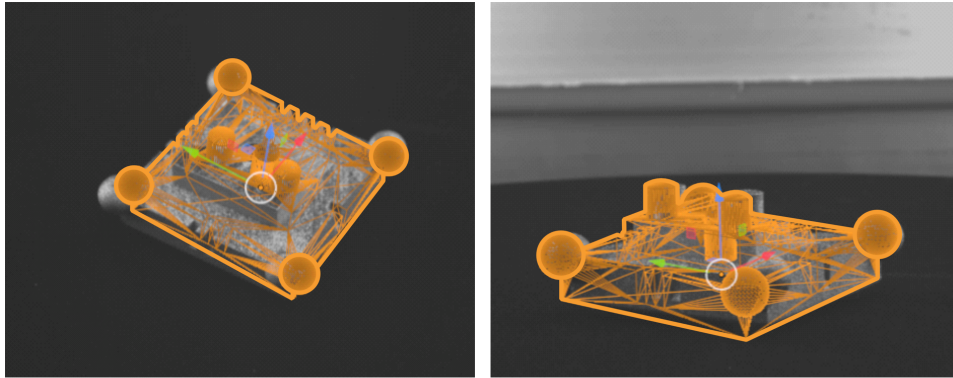
8.5.2 Results on real data

From the testing on synthetic data it was determined that Powell's method provided more consistent pose predictions. As with the monocular method, a photogrammetric scan of each artefact was conducted comprised of 60 imaging positions each, again leading to 240 total images. A pose estimation prediction was made using the stereo method on each pair of images. As with the monocular method, ICP registration to the measured point-cloud was used to find the ground-truth pose of each artefact. Figure 8.18 shows an example pose prediction for each artefact made on real photographic data from each dataset.

As can be seen, these results appear to lie close to the true pose of each



(a) Pyramid.



(b) Tomas.

Figure 8.18: Pose predictions for each artefact, each prediction shown is overlaid in yellow wireframe on the input stereo image pair.

artefact. To quantify how close, the Euclidean distance in the translational prediction and the axis angle rotation representation, alongside the loss function from each view, were calculated relative to the pose as given by ICP. The results for both artefacts are given in Table 8.3.

	RMS errors			Mean Residuals		Mean final loss /%	Mean optimisation time /s
	x /mm	y /mm	z /mm	T /mm	R /radians		
Pyramid	0.27	0.23	0.21	0.38	0.63	1.14	193.59
Tomas	0.38	0.22	0.17	0.44	2.13	1.55	155.00

Table 8.3: Results of the stereo pose estimation method on both artefacts over 120 images of each artefact.

As can be seen in Table 8.3, the stereo method can pose residuals on the real data are relatively close to the results on synthetic data which was shown in Table 8.2. The increased error is likely due to errors in creating the binary mask from the real images, this is discussed further in Section 8.6.

The mean magnitude translation residual across all 120 images was found to be 0.38 mm in the case of the Pyramid artefact and 0.44 mm in the case of the Tomas artefact. Interestingly, the performance degradation in the z and x directions does not appear in the real data. When considering the rotational magnitude residual, it can be seen in Table 8.3 that the pyramid has a much lower residual than the Tomas artefact. From observing the results, it is clear that this is due to the Tomas artefact being prone to becoming stuck in local minima. Figure 8.19 shows an example where the optimisation got stuck in a local minima, and the effect on the rotation estimation this has.

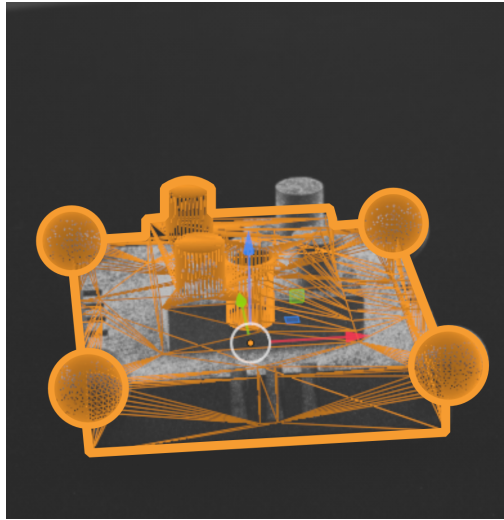


Figure 8.19: Local minima optimisation result on the Tomas artefact. The local z axis rotation can be seen to be 180° incorrect while the translation is still relatively accurate.

8.6 Discussion of both pose estimation approaches

To compare the monocular and stereo methods directly, the results on the Pyramid artefact given in Tables 8.1 and 8.3 respectively are used as both models were evaluated on this artefact. As can be seen, the mean transla-

tional error is much lower when the stereo method is used, reduced from 22.8 mm to 0.38 mm, a reduction of 98 %. This is likely due to the stereo baseline providing much more reliable depth information than the CNN is able to extract from a single image. In contrast, the monocular method is much better at predicting the rotation with the residual reduced by 69 % compared to the stereo method. This is likely because the CNN is better able to extract this information directly from the image, and the depth information provided by the stereo baseline is less critical for this task.

As mentioned in Section 8.1, there are inherent advantages and disadvantages of the monocular and stereo method. The monocular method can be deployed on single view systems, such as the MMT system while the stereo method requires a system with a minimum of two cameras with a static relative pose which can be accurately characterised, such as the stereo camera pair in the Taraz system.

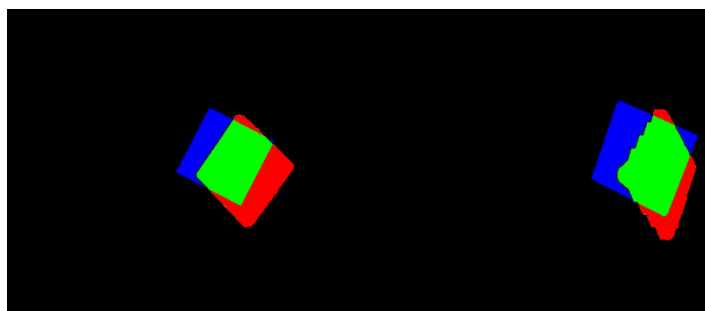
the monocular approach can only make predictions on object for which it has been explicitly trained on the CAD data while the stereo method can be used on any object with CAD data available. In manufacturing metrology, there are many applications where a single object, or small group of objects, must be repeatedly measured, for example for part verification on assembly lines. Additionally, the training time of the model, around 24 hours, is less than the manufacture time of many AM parts. Therefore the training of the model can be done in parallel with the manufacture of the part which can then be verified straight off the printer. For these reasons, the requirement for pretraining does not prevent the monocular method from being a useful approach.

The monocular model, once deployed, can make pose predictions in a fraction of a second. As can be seen in Table 8.3, the stereo method takes much longer - on the order of two - three minutes. It is highly likely that this is mainly an implementation detail. Currently the Blender API contains

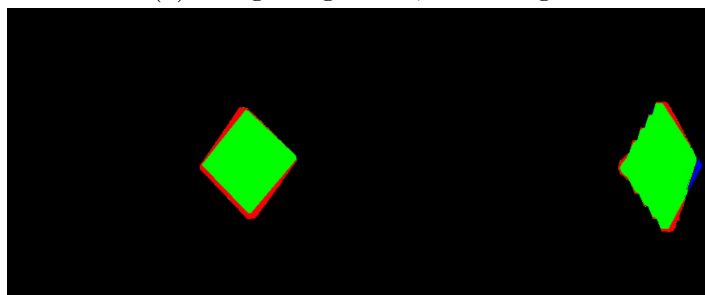
a bug where pixels from the currently rendered frame cannot be directly accessed, this means to access the pixel information for the currently rendered mask it must first be saved and reloaded from the hard-drive. These unnecessary IO operations waste a large amount of time, if deployed commercially a custom ray casting algorithm written as a graphics shader could compute the masks and loss functions much faster [248]. Additionally, if the ray casting algorithm could be written entirely in a framework such as JAX [249], an automatic differentiation approach could be used to efficiently calculate the Jacobian in a much more reliable manner than the finite differencing scheme presented. This could enable use of the BFGS or similar algorithm. This could lead to much faster optimisations again as the gradient descent method was shown to be much faster than Powell's method, but the current approach to gradient estimation was too inaccurate to make this a viable approach.

Figure 8.20 shows a comparison between the stereo loss function evolution of an example from the real dataset, and the synthetic dataset.

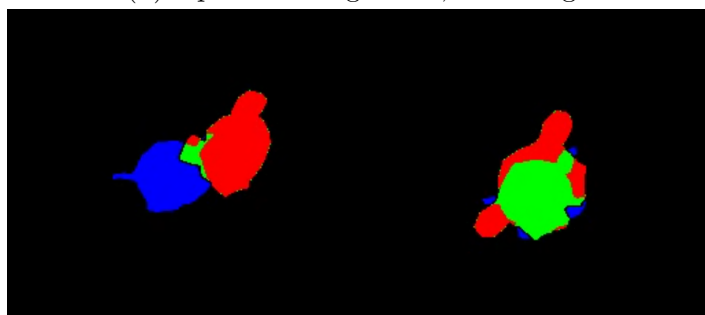
As can be seen in Figure 8.20, the optimisation on the synthetic data can reduce the loss function very close to zero, where there is still a reasonable amount of misclassified pixels even in a well performing sample from the real dataset (in this case 1 % misclassification). The reason for this is twofold; first, the virtual camera system used to render the synthetic samples is identical to the one used to render the prediction masks, while this camera model is based on characterised parameters from the real camera there may be some modelling errors causing lower performance on real image data. Secondly and perhaps most importantly, the synthetic examples are perfect masks of the input data because they are rendered directly from CAD data. In contrast, the real optimisation is conducted against masks made using the method which was presented in Section 6. In general, this method was



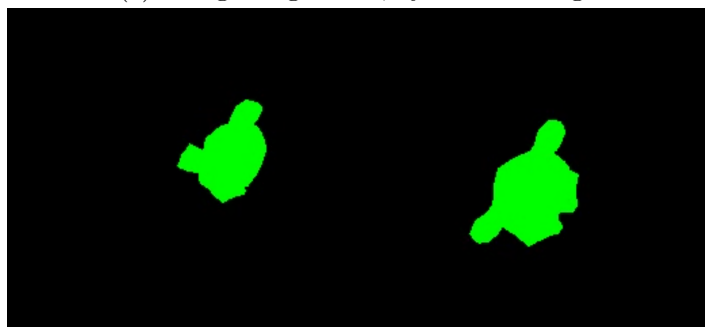
(a) Rough alignment, real image.



(b) Optimised alignment, real image.



(c) Rough alignment, synthetic image.



(d) Optimised alignment, synthetic image.

Figure 8.20: Loss function visualisation for the rough and optimised alignments of high performing samples taken from the real and synthetic datasets.

designed to oversize the masks produced to minimise any missing data from the surface of the object. This oversizing of the real masks gives rise to the imperfect matching which can be seen clearly in Figure 8.20b.

Another error which can be caused by the masking process occurs when the masking is imperfect, as was shown in Figure 7.13. Figure 8.21 shows how imperfect masking effects the pose estimation result in the stereo method.

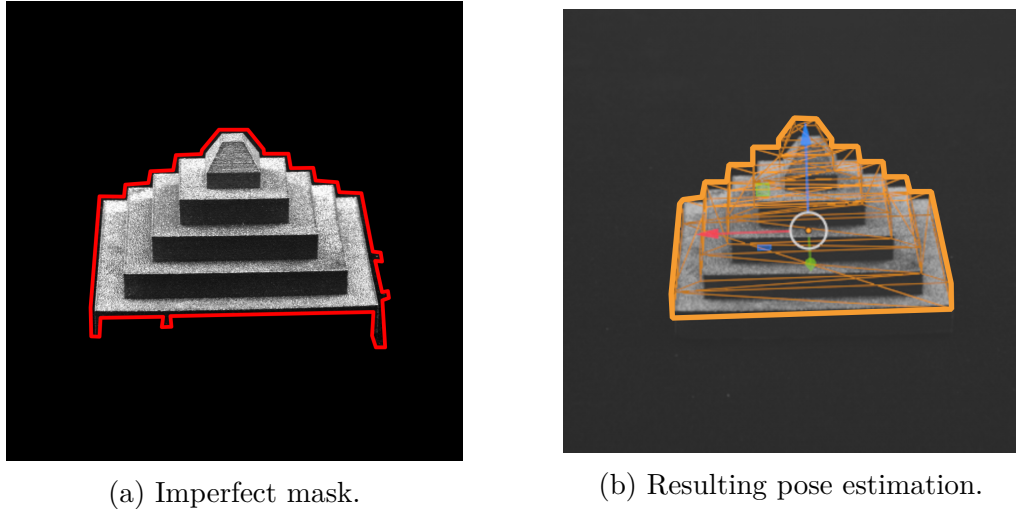


Figure 8.21: Effect of an imperfect mask on the optimised pose prediction.

As can be seen in Figure 8.21a, the bottom face of the Pyramid is not masked correctly as, due to the lighting conditions in the Taraz system, it falls in shadow causing it to blend with the rotation stage. In Figure 8.21b, it can be seen that this leads to the optimisation attempting to fit the pyramid in an erroneous area, causing the error in the local z axis to be high. There are a few ways to address this problem, first improving lighting conditions in the measurement volume to ensure high contrast at all edges of the part. Secondly, adapting the loss function to perform an edge matching operation rather than a pixel matching operation. Alternate loss function were considered during development of this model, however many proved too computationally expensive to be calculated at each optimisation step efficiently. This ease of computation is the biggest advantage of the current loss function, in the future more optimised solutions may be developed and deployed.

The monocular approach bares similarities to neural network based approaches in the literature [250–254]. The proposed approach is differenti-

ated through the novel architecture, which has lower parameter count than many of the models in the literature leading to fast inference and training. The largest differentiating factor, however, is the dataset that the model is trained upon. All the models referenced previously are trained on the same benchmark datasets [255,256]. While this makes comparison of model performance simple between these models (and difficult to compare directly to the models presented here) these datasets are not representative of the data used in manufacturing metrology applications. The development of the novel dataset from simulation and surface texture generation makes the proposed approach well fit to a specific measurement system and manufacture process.

In contrast the stereo method is, to the author's knowledge, a totally unique approach. The greatest benefit this method provides over state of the art ML models is the lack of pre-training required. Any part should be able to be located in the measurement volume so long as its associated CAD data is available. Further, the use of a stereo sensor array leads to increased certainty due to some pose ambiguities being removed by consensus between the two cameras. For even greater certainty a larger array of cameras could be utilised in a multi-view system, the algorithm can be used unchanged with an arbitrary number of cameras so long as the relative pose between the cameras is known and the cost of greater computational expense.

8.7 Pose estimation conclusions

Two methods of estimating the pose from an initial image capture were presented; one monocular method based on a single image, and one based on a stereo pair of images with a known baseline distance between the two cameras. The monocular method trained a custom CNN architecture on a

series of synthetic images rendered from CAD. A material model was developed and a characterised camera model was employed to make the rendered images as representative of real input data as possible. The stereo method relies on generating binary masks of the input stereo pair using the procedure given in Chapter 6. The centers of these masks are triangulated to give an initial pose estimation which is then refined using Powell's method of direct search. Powell's method was selected over a gradient based method due to higher positional accuracies at the cost of longer data processing times, a different choice may be made depending on the specific application and implementation. The loss function which is minimised in the optimisation procedure is derived from raycasting a predicted binary mask from the current pose prediction and calculating the misclassified pixels compared to the input mask.

After verification on synthetic data, both methods were validated on real data, with the stereo method more accurately predicting the translational position of the artefact (22 mm residual compared to 0.39 mm) and the monocular method more accurately extracting the rotation by 69 % as measured by the mean magnitude residual in the rotation prediction.

Each method has a range of advantages and disadvantages over the other which are discussed in detail in Section 8.6. In summary, the monocular method is preferred if fast prediction times, single image input, and repeated measurement of the same set of artefacts are required. The stereo method is preferred if many objects will be measured on the system, and lack of training overhead and accurate positional pose predictions are required.

Future work should involve a thorough investigation of alternate loss functions, improvements to the lighting conditions of the system to promote correct masking of the object, and a generalisation of the stereo method onto multi-view systems with more than two cameras. It is likely that a

wide range of views around the part will help reduce rotational ambiguities. These monocular method contributes to the state of the art due to its novel architecture and novel dataset generation method making it well suited to manufacturing metrology tasks specifically. The stereo method contributes an entirely new approach which eschews the need for any pre-training and gains confidence via consensus across multiple views.

Using the methods presented in this chapter in the overall pipeline, the view plan generated by the method in Chapter 5 can be adjusted autonomously based on the arbitrary pose of the part in the measurement volume. This adjustment is shown for an example measurement of the pyramid artefact in Figure 8.22

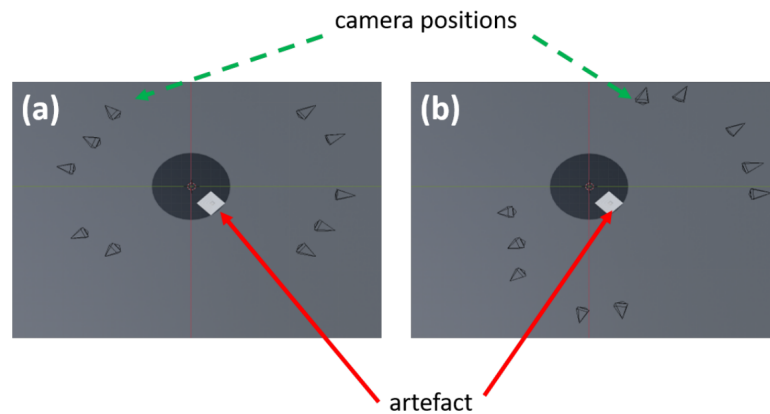


Figure 8.22: Top view showing the optimised camera positions. (a) About the centre of the instrument; (b) corrected for the placement of the artefact using the CNN pose prediction

With this contribution, all the developments required to enable the pipeline as was presented in Figure 1.3 have been made.

Chapter 9

Conclusions and future work

From Chapter 1: **It is the aim of this thesis to develop a software pipeline to enable, for the first time, a fully automated coordinate measurement system which conducts measurements in an optimised manner.** In this thesis, a novel fully automated and optimised pipeline for conducting optical coordinate measurements was indeed proposed and each part of the pipeline has been developed and tested individually including validation against competing commercial and tactile solutions. First presented in Figure 1.3, the outline of this pipeline is repeated here for convenience.

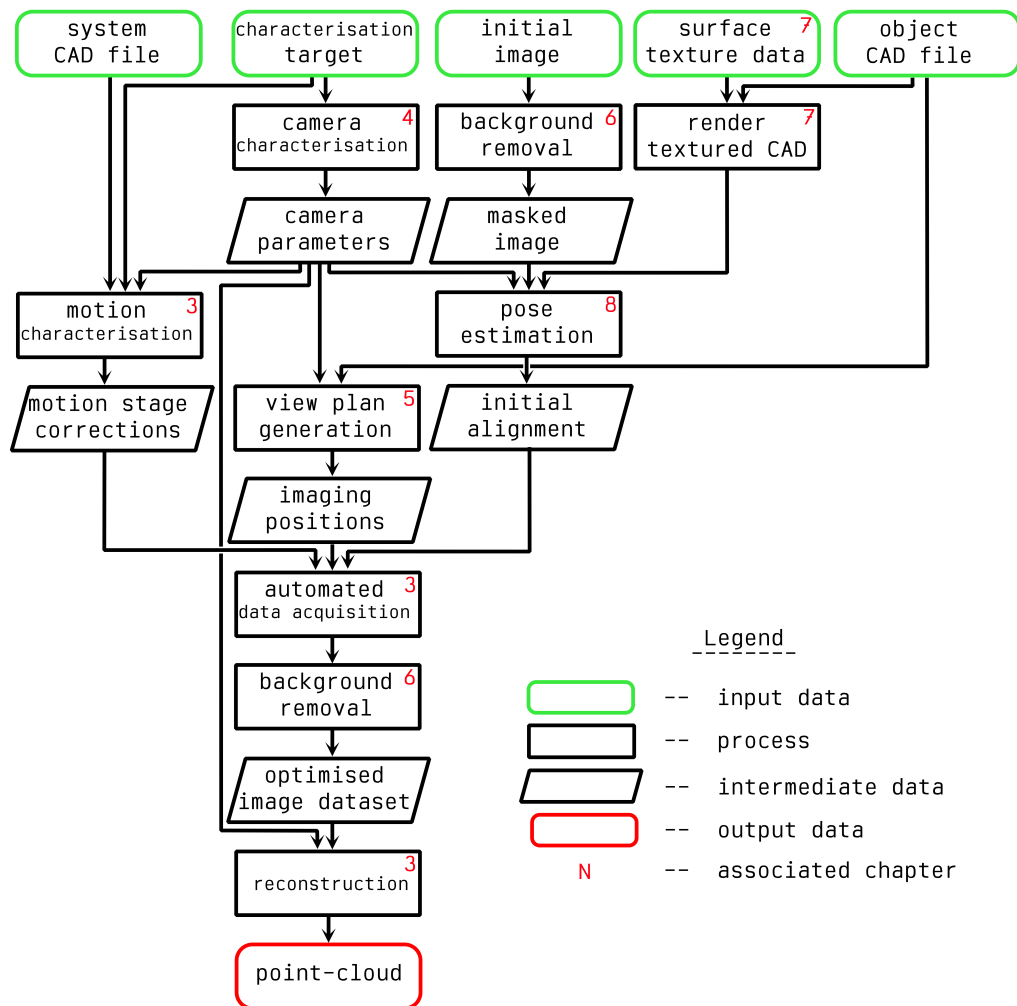


Figure 9.1: The now complete automated and optimised software pipeline.

As can be seen in Figure 9.1, each chapter in this thesis represents a necessary and novel part of this pipeline which, together, provide a fully

automated and optimised solution for conducting optical coordinate measurements. This pipeline is enabled through the exploitation of known properties of both the object being measured and the measurement systems themselves assisted by recent rapid advanced in the field of machine learning and intelligent data processing.

A review of this thesis' novel contributions to the field can be briefly summarised as:

1. A software pipeline enabling automated and optimised optical coordinate measurement for the first time.
2. A new hybrid ML approach to characterisation which is more robust and with lower residual errors than the current state of the art.
3. An improved method for analysing visible surfaces of a given object from a given view combining the benefits of two state of the art approaches by operating at lower cost than triangle intersection with fewer misclassified points than HPR.
4. A procedure for the global optimisation of the imaging strategy showing improved reconstruction results over standard industry practice while operating at much faster speeds due to dramatically decreased dataset size.
5. A generative ML model for the generation and categorisation of synthetic surface texture data for the creation of large realistic datasets. A novel approach to surface simulation which is much less intensive at run time than physics based simulation and more representative than pure mathematical surface representations.
6. A method to autonomously segment background pixels from object pixels within an image. More reliable than current approaches by

exploiting known properties of the system while not requiring the overhead of ML methods. Also shown to lead to improved reconstruction quality with a large reduction in unwanted points in the final cloud.

7. A monocular ML model for the 6 DoF pose estimation of an object. Novelty here lies in both the model architecture and the dataset which was created through photo-realistic rendering with a novel surface texture material model. The trained model is shown to perform well on a set of test artefacts.
8. A stereo raycasting method for the 6 DoF pose estimation of an object. A completely novel approach to the pose estimation problem which achieves high quality part localisation on both synthetic and real datasets of a range of objects.

A brief summary of the content and conclusions of each chapter is given below: **Chapter 2** first presents the required background theory to understand the measurement techniques used in this thesis and the machine learning techniques necessary to provide full automation and optimisation. This background theory was followed by the first comprehensive review of machine learning for optical coordinate metrology, the current state of the art prior to the developments made in this thesis.

Chapter 3 introduced all measurement systems used to gather data for the results presented later in the thesis, alongside summaries of computational methods used through out and a summary of all test artefacts used. **Chapter 4** proposes a hybrid ML approach to camera characterisation which outperforms the popular OpenCV method by approximately 50 % as measured by the mean magnitude residual. It was also shown that the proposed ML method is more robust to adverse imaging conditions than

a competing refinement method based on a line-spread function approach. This should enable a greater range of imaging positions to be included in the characterisation dataset, leading to improved characterisation outcomes. The camera parameters given by this process are used in many other processes in the proposed measurement pipeline, including view planning and pose estimation.

Chapter 5 provided a method for optimising the imaging strategy of an optical coordinate measurement on a per-part basis. First, an improved method for evaluating which surface points are visible from a given viewing position from the object's CAD is proposed. It is shown that the proposed solution strikes a good balance between accuracy and speed of calculation. This visible point analysis method is used in a genetic optimisation to find the minimum number of imaging positions which can produce a high quality measurement result as assessed by a custom global objective function. View optimisation is then conducted on a range of test artefacts and shown to produce high quality scans from a very low number of images as assessed through comparison to other photogrammetry measurements, commercial DFP measurements, and tactile CMM measurements. This view plan is then used later in the pipeline, once adjusted for the pose of the object, to conduct an optimised measurement.

Chapter 6 proposes a solution for autonomous removal of background data from the images comprising a photogrammetric scan. A method for autonomous background masking is presented based on image processing techniques assuming there are no closed contours in the background pixel information. Testing the proposed approach on the data in photogrammetric measurements is shown to have numerous benefits including reduced processing time, improved memory usage, decreased numbers of background points reconstructed, and increased object point density. It is also shown that the measurement result is improved quantitatively through

greater agreement to CMM and improved reconstruction of surface features. This background removal technique is also used to generate binary masks used in pose estimation later in the measurement pipeline.

Chapter 7 presents a method for producing synthetic surface texture data. A progressively growing generative adversarial network (PG-GAN) is trained to produce a wide range of surface types which are shown to be representative, but distinct, from real measurement data. The model is also trained to categorise what type of surface it is producing, top, upskin, or downskin when generating AM data. This model was developed to enable photo-realistic renders of manufactured parts from their CAD data, an approach to using the surface generation model in a material shader is presented. These simulated images are used to produce synthetic photogrammetry data used to train models for pose estimation.

Chapter 8 Finally, two approaches to object pose estimation are presented. One which relies on the simulated data from Chapter 7 to train a CNN to directly regress the 6 degrees-of-freedom (DoF) pose of the object relative to a single camera. The second uses binary masks generated by the algorithm presented in Chapter 6 alongside predicted masks generated by raycasting the CAD data through a characterised camera model to minimise a loss function defined between the real image mask and the predicted mask. Both models are shown to produce good results on real photographic data, with the monocular method producing better rotational prediction and the stereo method producing better positional predictions. Each method is suitable for different applications; in summary, the monocular method is preferred if fast prediction times, single image input, and repeated measurement of the same set of artefacts are required. The stereo method is preferred if many objects will be measured on the system, and lack of training overhead and accurate positional pose predictions are required.

The aims and objectives of this thesis were summarised in Section 1.3.1 as:

1. To create a software pipeline which will enable the creation of a fully automated measurement system.
2. To develop algorithms as part of this pipeline to perform measurements in a way which maximises surface coverage and reconstruction quality while minimising computational expense and time.
3. To allow for arbitrary placement of the measurement object within the measurement volume, ie. no fixturing or fiducial marking required.

The pipeline has been presented. The algorithms have been developed, presented and compared to the current state of the art and common industrial practice. Arbitrary object placement is enabled through use of either of the pose estimation algorithms presented in Chapter 8 in concert with the view planning algorithm proposed in Chapter 5. It is clear that these aims are now achieved.

9.1 Future Work

Future work which could be conducted to continue the research presented within this thesis includes testing the monocular method while using the generator based material shader presented in Section 7.6, testing the background removal method from Chapter 6 under more favourable lighting conditions and integrating the background removal with the view planning algorithm from Chapter 5. Perhaps most of all, to deploy the entire automated pipeline into a physical system.

While hardware design and deployment was not feasible in the scope of this project, the Midlands Centre for Data Driven Metrology (MCDDM) at the University of Nottingham is directly working on taking the ideas proposed in this thesis and integrating them into a hardware demonstration. Along with fully automated multi-sensor photogrammetry and fringe projection as enabled by the methods outlined within this thesis, the demonstrator will also feature integrated surface texture measurement, data fusion, and advanced data analysis. Figure 9.2 shows the proposed CAD design of this system. The system consists of four projectors, four pairs of stereo

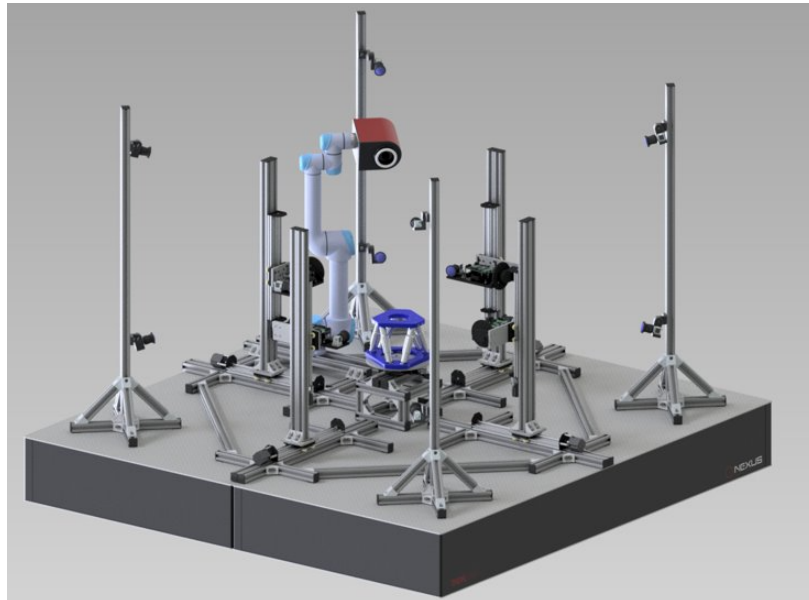
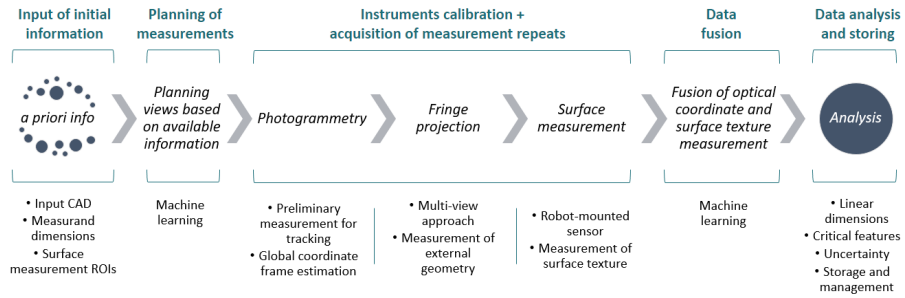
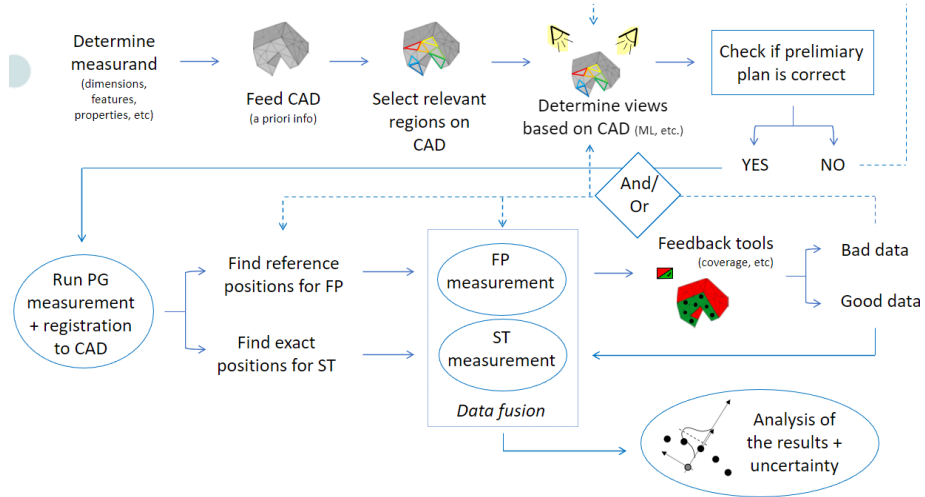


Figure 9.2: MCDDM demonstrator CAD.

machine vision cameras, a hexapod for part positioning, and numerous motion stages for the repositioning of each sensor. In addition, an optical surface texture instrument is placed on a robotic arm. Figure 9.3 shows the proposed data flow in the demonstrator software. Due to the modular design of the software developed in this thesis, the separate methods should be easily transferable into the new system. Some of the methods may require adaptation to best fit the MCDDM demonstrator. For example, the view planning algorithm is optimised for photogrammetry rather than DFP



(a) Proposed measurement pipeline



(b) Detailed data flow

Figure 9.3: MCDDM demonstrator data flow.

and the range of motion of the cameras relative to the part must be accurately considered. Other methods, such as the camera characterisation approach and object pose estimation, should be "plug-and-play". This is a perfect system to investigate using multi-view stereo for the stereo method pose estimation, as was discussed in Section 8.7, as there are four separate imaging locations. This is likely to lead to an improvement on the pose prediction accuracy as pose ambiguities due to occluded features will be reduced.

I am excited to see how my ideas are integrated into this product.

9.2 Summary

This thesis has achieved its aim of developing a software solution to enable autonomous and optimised optical coordinate metrology, a goal driven by the large influence of the operator on the quality of measurement results drawn from current optical CMSs. A selection of novel algorithms were developed and tested individually. From camera characterisation, through measurement planning to final data acquisition and processing; these algorithms are valuable in their own right and are shown to have a range of benefits over the previous state of the art and industry standard practice, as summarised above. Beyond their individual value, they are part of a greater whole. A software pipeline, shown in Figure 9.1, threads together the algorithms presented herein to perform all operations required to perform an optimal measurement of an object via a CNC optical CMS for the first time.

Bibliography

- [1] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Proc. NIPS*, pages 365–376, 2017.
- [2] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, pages 652–660. IEEE, 2017.
- [3] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *Proc. CVPR*, pages 1759–1769. IEEE, 2020.
- [4] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proc. AAAI-20*, volume 34, pages 11596–11603, 2020.
- [5] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.*, 9:62–66, 1979.
- [6] JCGM. *International Vocabulary of Metrology*. BIPM, 4 edition, 2014.
- [7] ISO 10110. *Optics and photonics - preparation of drawings for optical elements and systems - part 8:Surface texture; roughness and waviness*. International Organization for Standardization, Geneva, 2019.
- [8] Richard Leach. *Fundamental principles of engineering nanometrology*. Elsevier, 2014.
- [9] David R McMurtry. Touch probe, 1992. US Patent 5,146,691.
- [10] Richard K Leach. *The measurement of surface texture using stylus instruments*. The National Physical Laboratory: London.

- [11] William E Frazier. Metal additive manufacturing: a review. *J. Mater. Eng. Perform.*, 23:1917–1928, 2014.
- [12] Saint-Clair T Toguem, Baltej S Rupal, Charyar Mehdi-Souzani, AJ Qureshi, and N Anwer. A review of am artifact design methods. In *euspen SIG Adv. Precis. Addit. Manuf.*, pages 132–137, 2018.
- [13] G Gayton. *Improvements to the characterisation of fringe projection*. PhD thesis, University of Nottingham, preprint, Nottingham, UK, 2022.
- [14] Jae Heun Woo, Chu-Shik Kang, Jong-Ahn Kim, Jae Wan Kim, Sunghoon Eom, and Jae Yong Lee. Time-of-flight measurement-based three-dimensional profiler system employing a lightweight fresnel-type risley prism scanner. *Proc. SPIE*, 61(5):054104, 2022.
- [15] Sergi Foix, Guillem Alenya, and Carme Torras. Lock-in time-of-flight (tof) cameras: A survey. *Sens.*, 11(9):1917–1926, 2011.
- [16] Thomas Luhmann, Stuart Robson, Stephen Kyle, and Jan Boehm. *Close-range photogrammetry and 3D imaging*. de Gruyter, 3 edition, 2019.
- [17] Zhengyou Zhang. A flexible new technique for camera calibration. *Proc. IEEE*, 22(11):1330–1334, 2000.
- [18] Duane C Brown. Decentering distortion of lenses. *Photogramm. Eng. Remote Sens.*, 32:444–462, 1966.
- [19] Duane C Brown. Close-range camera calibration. *Photogramm. Eng.*, 37(8):855–866, 1971.
- [20] Hans Peter Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, Stanford University, 1980.
- [21] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS*, volume 6, pages 281–305, 1987.
- [22] David G Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, volume 2, pages 1150–1157, 1999.
- [23] Manoj K Vairalkar and SU Nimbhorkar. Edge detection of images using sobel operator. *Int. J Emerg. Technol. Adv. Eng.*, 2(1):291–293, 2012.

- [24] Richard S Stephens. Probabilistic approach to the hough transform. *Image and vision computing*, 9(1):66–71, 1991.
- [25] Manolis Lourakis and Antonis Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical report, Technical Report 340, Institute of Computer Science, Heraklion, Crete, 2004.
- [26] Clive S Fraser. Digital camera self-calibration. *ISPRS J. Photogramm. Remote Sens.*, 52(4):149–159, 1997.
- [27] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:1362–1376, 2009.
- [28] Shizeng Yao, Hadi AliAkbarpour, Guna Seetharaman, and Kannappan Palaniappan. 3d patch-based multi-view stereo for high-resolution imagery. In *Proc. SPIE*, volume 10645, pages 146–153, 2018.
- [29] Lichun Wang, Ran Chen, and Dehui Kong. An improved patch based multi-view stereo (pmvs) algorithm. In *Proc. CSSS*, pages 9–12, 2014.
- [30] MetaShape. *version 1.6.2 (build 10247)*. Agisoft, St. Petersburg, 2019.
- [31] Song Zhang. *High-speed 3D imaging with digital fringe projection techniques*, chapter Introduction, pages 1–13. CRC Press, 2018.
- [32] Amrozia Shaheen. *Development of a multi-view fringe projection system for coordinate metrology*. PhD thesis, University of Nottingham, 2021.
- [33] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.*, 2:164–168, 1944.
- [34] Liang Sun, Shuiwang Ji, and Jieping Ye. *Multi-label dimensionality reduction*. Boca Raton, FL: CRC Press, 2013.
- [35] Michel Valstar. *Introduction lecture notes Machine Learning COMP3009*. University of Nottingham, delivered 16 October 2018, 2018.
- [36] Sergios Theodoridis, Aggelos Pikrakis, Konstantinos Koutroumbas, and Dionisis Cavouras. *Introduction to pattern recognition: a matlab approach*. New York: Academic Press, 2010.
- [37] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Berlin: Springer, 4 edition, 2006.

- [38] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv:1710.05941*, 2017.
- [39] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.
- [40] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *Schedae Inform.*, 25:49–59, 2017.
- [41] Peter J Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35:73–101, 1964.
- [42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proc. ICLR*, 2015.
- [43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Proc. NIPS*, pages 1097–1105, 2012.
- [44] A Zhang, Z C Lipton, M Li, and A J Smola. Fully convolutional networks (fcn). *Dive into Deep Learning*, 2019.
- [45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778. IEEE, 2016.
- [46] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *Proc. ICML*, pages 10096–10106. PMLR, 2021.
- [47] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proc. CVPR*, pages 7132–7141, 2018.
- [48] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NIPS*, 2014.
- [49] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proc. ICLR*, 2018.
- [50] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proc. CVPR*, pages 8110–8119, 2020.
- [51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. ICCV*, pages 2223–2232. IEEE, 2017.

- [52] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proc. ACL*, pages 38–45, 2020.
- [53] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.
- [54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. NIPS*, 30, 2017.
- [55] Ethem Alpaydin. *Introduction to machine learning*. Cambridge, MA: MIT press, 2020.
- [56] Ben Taskar, Dan Klein, Mike Collins, Daphne Koller, and Christopher D Manning. Max-margin parsing. In *Proc. EMNLP*, pages 1–8, 2004.
- [57] Darrell Whitley. A genetic algorithm tutorial. *Stat. Comput.*, 4:65–85, 1994.
- [58] Rajesh Kumar Singh, VK Panchal, and Bhupesh Kumar Singh. A review on genetic algorithm and its applications. In *Proc. ICGCIoT*, pages 376–380. IEEE, 2018.
- [59] Patrick Kwaku Kudjo, E Ocquaye, and Wolali Ametepe. Review of genetic algorithm and application in software testing. *Int. J. Comput. Appl.*, 160:1–6, 2017.
- [60] Ms Trupti Bhoskar, Mr Omkar K Kulkarni, Mr Ninad K Kulkarni, Ms Sujata L Patekar, GM Kakandikar, and VM Nandedkar. Genetic algorithm and its applications to mechanical engineering: A review. *Mater. Today: Proc.*, 2:2624–2630, 2015.
- [61] M Mahalakshmi, P Kalaivani, and E Kiruba Nesamalar. A review on genetic algorithm and its applications. *Int. J. Comput. Algorithm*, 2.
- [62] Gustavo Olague and Roger Mohr. Optimal camera placement for accurate reconstruction. *Pattern Recognit.*, 35:927–944, 2002.
- [63] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv:1512.03012*, 2015.

- [64] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *Appl. Soft Comput.*, 70:41–65, 2017.
- [65] Wicky Law, Daniel PK Lun, et al. Deep learning based period order detection in structured light three-dimensional scanning. In *Proc. ISCAS*, pages 1–5. IEEE, 2019.
- [66] Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skočaj. Segmentation-based deep-learning approach for surface-defect detection. *J. Intell. Manuf.*, 31:759–776, 2020.
- [67] Tanzeel U Rehman, Md Sultan Mahmud, Young K Chang, Jian Jin, and Jaemyung Shin. Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. *Comput. Electron. Agric.*, 156:585–605, 2019.
- [68] Pdraig Timoney, Roma Luthra, Alex Elia, Haibo Liu, Paul Isbester, Avi Levy, Michael Shifrin, Barak Bringoltz, Eylon Rabinovich, Ariel Broitman, et al. Advanced machine learning eco-system to address hvm optical metrology requirements. In *Proc. SPIE*, volume 11325, page 113251H, 2020.
- [69] Dexin Kong, Daniel Schmidt, Jennifer Church, Chi-Chun Liu, Mary Breton, Cody Murray, Eric Miller, Luciana Meli, John Sporre, Nelson Felix, et al. Measuring local cd uniformity in euv vias with scatterometry and machine learning. In *Proc. SPIE*, volume 11325, page 113251I, 2020.
- [70] Bryan M Barnes and Mark-Alexander Henn. Contrasting conventional and machine learning approaches to optical critical dimension measurements. In *Proc. SPIE*, volume 11325, page 113251E, 2020.
- [71] Lingbin Meng, Brandon McWilliams, William Jarosinski, Hye-Yeong Park, Yeon-Gil Jung, Jehyun Lee, and Jing Zhang. Machine learning in additive manufacturing: a review. *J. Micros.*, 72(6):2363–2377, 2020.
- [72] Mark-Alexander Henn, Hui Zhou, Richard M Silver, and Bryan M Barnes. Applications of machine learning at the limits of form-dependent scattering for defect metrology. In *Proc. SPIE*, volume 10959, page 109590Z, 2019.
- [73] Fu Li, Quanlu Li, Tianjiao Zhang, Yi Niu, and Guangming Shi. Depth acquisition with the combination of structured light and deep learning stereo matching. *Signal Process. Image Commun.*, 75:111–117, 2019.

- [74] Dan Kong and Hai Tao. Stereo matching via learning multiple experts behaviors. In *Proc. BMVC*, pages 97–106, 2006.
- [75] Wei Yin, Chao Zuo, Shijie Feng, Tianyang Tao, and Qian Chen. High-speed 3d shape measurement with the multi-view system using deep learning. In *Proc. SPIE*, volume 11189, page 111890B, 2019.
- [76] Jure Zbontar, Yann LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.*, 17:2287–2318, 2016.
- [77] Xiaoyan Hu and Philippos Mordohai. A quantitative evaluation of confidence measures for stereo vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34:2121–2133, 2012.
- [78] Andy Motten, Luc Claesen, and Yun Pan. Binary confidence evaluation for a stereo vision based depth field processor soc. In *Proc. ACPR*, pages 456–460. IEEE, 2011.
- [79] Huizong Feng, Gaofeng Wu, Mingchi Feng, Ming Cen, and Yibo Liu. Research on global measurement method based on multi-cameras. In *IOP Conf. Ser.: Mater. Sci. Eng.*, volume 382, page 052039. IOP, 2018.
- [80] Ralf Haeusler, Rahul Nair, and Daniel Kondermann. Ensemble learning for confidence measures in stereo vision. In *Proc. CVPR*, pages 305–312. IEEE, 2013.
- [81] Akihito Seki and Marc Pollefeys. Patch based confidence prediction for sense disparity map. In *Proc. BMVC*, volume 23, pages 1–13, 2016.
- [82] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proc. CVPR*, pages 1592–1599. IEEE, 2015.
- [83] Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163, 1992.
- [84] Yunpeng Li and Daniel P Huttenlocher. Learning for stereo vision using the structured support vector machine. In *Proc. CVPR*, pages 1–8. IEEE, 2008.
- [85] Sean Ryan Fanello, Christoph Rhemann, Vladimir Tankovich, Adarsh Kowdle, Sergio Orts Escolano, David Kim, and Shahram Izadi. Hyperdepth: Learning depth from structured light without matching. In *Proc. CVPR*, pages 5441–5450. IEEE, 2016.

- [86] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proc. ICCV*, pages 66–75. IEEE, 2017.
- [87] Sen Xiang, You Yang, Huiping Deng, Jin Wu, and Li Yu. Multi-anchor spatial phase unwrapping for fringe projection profilometry. *Opt. Express*, 27:33488–33503, 2019.
- [88] Vincent Lepetit, Julien Pilet, and Pascal Fua. Point matching as a classification problem for fast and robust object pose estimation. In *Proc. CVPR*, volume 2, page 2. IEEE, 2004.
- [89] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proc. CVPR*, pages 5695–5703. IEEE, 2016.
- [90] Larry Medsker and Lakhmi C Jain. *Recurrent neural networks: design and applications*. CRC press, 1999.
- [91] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *Proc. CVPR*, pages 1–8. IEEE, 2007.
- [92] FJ Cuevas, M Servin, ON Stavroudis, and R Rodriguez-Vera. Multi-layer neural network applied to phase and depth recovery from fringe patterns. *Opt. Commun.*, 181:239–259, 2000.
- [93] Hieu Nguyen, Nicole Dunne, Hui Li, Yuzeng Wang, and Zhaoyang Wang. Real-time 3d shape measurement using 3lcd projection and deep machine learning. *Appl. Opt.*, 58:7100–7109, 2019.
- [94] Shijie Feng, Qian Chen, Guohua Gu, Tianyang Tao, Liang Zhang, Yan Hu, Wei Yin, and Chao Zuo. Fringe pattern analysis based on convolutional neural networks. *Proc. SPIE*, 10991:109910C, 2019.
- [95] Wei Yin, Chao Zuo, Shijie Feng, Tianyang Tao, and Qian Chen. Bi-frequency temporal phase unwrapping using deep learning. In *Proc. SPIE*, volume 10991, page 109910D, 2019.
- [96] Wei Yin, Qian Chen, Shijie Feng, Tianyang Tao, Lei Huang, Maciej Trusiak, Anand Asundi, and Chao Zuo. Temporal phase unwrapping using deep learning. *Nature: Sci.Rep.*, 9:1–12, 2019.

- [97] Chen Yang, Wei Yin, Hao Xu, Jiachao Li, Shijie Feng, Tianyang Tao, Qian Chen, and Chao Zuo. Single-shot 3d shape measurement with spatial frequency multiplexing using deep learning. In *Proc. SPIE*, volume 11189, page 111891P, 2019.
- [98] Sam Van der Jeught and Joris JJ Dirckx. Deep neural networks for single shot structured light profilometry. *Opt. Express*, 27:17091–17101, 2019.
- [99] Peter de Groot. Coherence scanning interferometry. In R K Leach, editor, *Optical measurement of surface topography*, pages 187–208. Berlin: Springer, 2011.
- [100] Rong Su. Coherence scanning interferometry. In R K Leach, editor, *Advances in Optical Surface Texture Metrology*.
- [101] Junchao Zhang, Xiaobo Tian, Jianbo Shao, Haibo Luo, and Rongguang Liang. Phase unwrapping in optical metrology via denoised and convolutional segmentation networks. *Opt. Express*, 27:14903–14912, 2019.
- [102] Daichi Kando and Satoshi Tomioka. Phase extraction from interferogram using machine learning. In *Proc. EI2019*, pages 1–5, 2019.
- [103] Chuqian Zhong, Zhan Gao, Xu Wang, Shuangyun Shao, and Chenjia Gao. Structured light three-dimensional measurement based on machine learning. *Sensors*, 19:3229, 2019.
- [104] Sen Xiang, Huiping Deng, Jin Wu, and Changjian Zhu. Absolute phase unwrapping with svm for fringe-projection profilometry. *IET Image Process.*, 14:2645–2651, 2020.
- [105] Enrique Dunn and Jan-Michael Frahm. Next best view planning for active model improvement. In *Proc. BMVC*, pages 1–11, 2009.
- [106] Riccardo Monica and Jacopo Aleotti. Prediction of depth camera missing measurements using deep learning for next best view planning. In *Proc. ICRA*, pages 8711–8717, 2022.
- [107] Samuel Arce, Cory A Vernon, Joshua Hammond, Valerie Newell, Joseph Janson, Kevin W Franke, and John D Hedengren. Automated 3d reconstruction using optimized view-planning algorithms for iterative development of structure-from-motion models. *Remote Sens.*, 12:2169, 2020.
- [108] Miguel Mendoza, J Irving Vasquez-Gomez, Hind Taud, L Enrique Sucar, and Carolina Reta. Supervised learning of the next-best-view for 3d object reconstruction. *Pattern Recognit. Lett.*, 133:224–231, 2020.

- [109] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:418–433, 2005.
- [110] M Roberts and AJ Naftel. A genetic algorithm approach to camera calibration in 3d machine vision. In *Proc. IEE CGAIPV*, pages 12–1, 1994.
- [111] Li Deng, Gen Lu, Yuying Shao, Minrui Fei, and Huosheng Hu. A novel camera calibration technique based on differential evolution particle swarm optimization algorithm. *Neurocomputing*, 174:456–465, 2016.
- [112] Refaat Mohamed, Abdelrehim Ahmed, Ahmed Eid, and Aly Farag. Support vector machines for camera calibration problem. In *Proc. IEEE ICIP*, pages 1029–1032, 2006.
- [113] Hao He, Haiyan Li, Yunbao Huang, Jingwei Huang, and Pu Li. A novel efficient camera calibration approach based on k-svd sparse dictionary learning. *Meas.*, 159:107798, 2020.
- [114] Syed Navid Raza, Hafiz Raza ur Rehman, Suk Gyu Lee, and Gyu Sang Choi. Artificial intelligence based camera calibration. In *Proc. IWCMC*, pages 1564–1569. IEEE, 2019.
- [115] Ben Chen, Caihua Xiong, and Qi Zhang. Ccdn: Checkerboard corner detection network for robust camera calibration. In *Proc. ICIRA*, pages 324–334, 2018.
- [116] Simon Donné, Jonas De Vylder, Bart Goossens, and Wilfried Philips. Mate: Machine learning for adaptive calibration template detection. *Sens.*, 16(11):1858, 2016.
- [117] Ben Chen, Yuyao Liu, and Caihua Xiong. Automatic checkerboard detection for robust camera calibration. In *Proc. ICME*, pages 1–6, 2021.
- [118] B Ergun, T Kavzoglu, I Colkesen, and C Sahin. Data filtering with support vector machines in geometric camera calibration. *Opt. Express*, 18:1927–1936, 2010.
- [119] Manuel Lopez, Roger Mari, Pau Gargallo, Yubin Kuang, Javier Gonzalez-Jimenez, and Gloria Haro. Deep single image camera calibration with radial distortion. In *Proc. CVPR*, pages 11817–11825. IEEE, 2019.
- [120] Jin Li and Zilong Liu. Camera geometric calibration using dynamic single-pixel illumination with deep learning networks. *IEEE Trans. Circuits Syst. Video Technol.*, 30:2550–2558, 2019.

- [121] Petros Stavroulakis, Shuxiao Chen, Clement Delorme, Patrick Bointon, Georgios Tzimiropoulos, and Richard Leach. Rapid tracking of extrinsic projector parameters in fringe projection using machine learning. *Opt.Lasers Eng.*, 114:7–14, 2019.
- [122] Adam Thompson and Nicholas Southon. Performance verification for optical coordinate metrology. In R K Leach, editor, *Advances in Optical Form and Coordinate Metrology*, pages 801–825. Bristol: IOP Publishing, 2020.
- [123] Marcela Vallejo, Carolina De La Espriella, Juliana Gómez-Santamaría, Andrés Felipe Ramírez-Barrera, and Edilson Delgado-Trejos. Soft metrology based on machine learning: a review. *Meas. Sci. Technol.*, 31:032001, 2019.
- [124] JCGM. *GUM 1995 with Minor Corrections, Evaluation of Measurement Data—Guide to the expression of Uncertainty in Measurement—JCGM 100*. Sèvres: BIPM, 2008.
- [125] JCGM. *Evaluation of Measurement Data—Supplement 1 to the Guide to the Expression of Uncertainty in Measurement—Propagation of Distributions Using a Monte Carlo Method— JCGM 101*. Sèvres: BIPM, 2008.
- [126] Sona Sediva and Marie Havlikova. Comparison of gum and monte carlo method for evaluation measurement uncertainty of indirect measurements. In *Proc. ICCV*, pages 325–329. IEEE, 2013.
- [127] Li Song, Gang Wang, and Michael R Brambley. Uncertainty analysis for a virtual flow meter using an air-handling unit chilled water valve. *HVAC&R Res.*, 19:335–345, 2013.
- [128] Howard Cheung and James E Braun. A general method for calculating the uncertainty of virtual sensors for packaged air conditioners. *Int. J. Refrig.*, 63:225–236, 2016.
- [129] Tobias Wissel, Benjamin Wagner, Patrick Stüber, Achim Schweikard, and Floris Ernst. Data-driven learning for calibrating galvanometric laser scanners. *IEEE Sens. J.*, 15:5709–5717, 2015.
- [130] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *Proc. ICLR*, pages 1–11, 2015.
- [131] Eleonora Grilli, Fabio Menna, and Fabio Remondino. A review of point clouds segmentation and classification algorithms. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, 42:339–344, 2017.

- [132] Anh Nguyen and Bac Le. 3d point cloud segmentation: A survey. In *Proc. RAM*, pages 225–230. IEEE, 2013.
- [133] Jing Huang and Suya You. Point cloud labeling using 3d convolutional neural network. In *Proc. ICPR*, pages 2670–2675. IEEE, 2016.
- [134] Zhongyang Zhao, Yinglei Cheng, Xiaosong Shi, and Xianxiang Qin. Classification method of lidar point cloud based on threedimensional convolutional neural network. In *J. Phys. Conf. Ser.*, volume 1168, page 1168, 2019.
- [135] Mingye Xu, Zhipeng Zhou, and Yu Qiao. Geometry sharing network for 3d point cloud classification and segmentation. In *Proc. AAAI*, volume 34, pages 12500–12507, 2020.
- [136] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Proc. NIPS*, 30:5105–5114.
- [137] Vivian Wen Hui Wong, Max Ferguson, Kincho H Law, Yung-Tsun Tina Lee, and Paul Witherell. Automatic volumetric segmentation of additive manufacturing defects with 3d u-net. *Proc. Spring Symp. AI Manuf.*, 2020.
- [138] Zhenbiao Tan, Qihang Fang, Hui Li, Sheng Liu, Wenkang Zhu, and Dekun Yang. Neural network based image segmentation for spatter extraction during laser-based powder bed fusion processing. *Opt. Laser Technol.*, 130:106347, 2020.
- [139] Zhongwei Li, Xingjian Liu, Shifeng Wen, Piyao He, Kai Zhong, Qingsong Wei, Yusheng Shi, and Sheng Liu. In situ 3d monitoring of geometric signatures in the powder-bed-fusion additive manufacturing process via vision sensing methods. *Sensors*, 18:1180, 2018.
- [140] Zhiyuan Zhang, Yuchao Dai, and Jiadai Sun. Deep learning based point cloud registration: an overview. *Virtual Real. Intell. Hardw.*, 2:222–246, 2020.
- [141] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Proc. CVPR*, pages 4631–4640. IEEE, 2017.
- [142] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proc. ICCV*, pages 3523–3532. IEEE, 2019.

- [143] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcv: An end-to-end deep neural network for point cloud registration. In *Proc. ICCV*, pages 12–21. IEEE, 2019.
- [144] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *Proc. CVPR*, pages 1939–1948. IEEE, 2020.
- [145] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *Proc. MVA*, pages 1–6. IEEE, 2019.
- [146] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3d siamese tracking. In *Proc. CVPR*, pages 1359–1368. IEEE, 2019.
- [147] Muhammad Sarmad, Hyunjoo Jenny Lee, and Young Min Kim. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proc. CVPR*, pages 5898–5907. IEEE, 2019.
- [148] Xuelin Chen, Baoquan Chen, and Niloy J Mitra. Unpaired point cloud completion on real scans using adversarial training. *arXiv:1904.00069*, 2019.
- [149] Alaa Abd-Raheem, Farah AlDeiri, and Musa Alyaman. Design of an automated 3d scanner. In *Proc. ACIT*, pages 1–5, 2018.
- [150] Fernando António Rodrigues Martins, Jaime Gómez García-Bermejo, Eduardo Zalaman Casanova, and José R Perán González. Automated 3d surface scanning based on cad model. *Mechatronics*, 15:837–857, 2005.
- [151] Xinyi Fan, Linguang Zhang, Benedict Brown, and Szymon Rusinkiewicz. Automated view and path planning for scalable multi-object 3d scanning. *Proc. ACM TOG*, 35:1–13, 2016.
- [152] Frank A Van den Heuvel. Automated 3d measurement with the dcs200 digital camera. In *Proc. SPIE*, volume 2252, pages 63–71, 1994.
- [153] Stefan Holtzhausen, S Schreiber, Ch Scho¨ne, R Stelzer, K Heinze, and A Lange. Highly accurate automated 3d measuring and data conditioning for turbine and compressor blades. In *Proc. ASME Turbo Expo*, volume 48876, pages 37–41, 2009.
- [154] Claus Brenner, Norbert Haala, and Dieter Fritsch. Towards fully automated 3d city model generation. *Proc. AEMMOASI III*, 9, 2001.

- [155] Christian Lorbach, Ulrich Hirn, Johannes Kritzinger, and Wolfgang Bauer. Automated 3d measurement of fiber cross section morphology in handsheets. *Nord Pulp Paper Res J.*, 27:264–269, 2012.
- [156] Nino Krznar, Ana Pilipović, and Mladen Šercer. Additive manufacturing of fixture for automated 3d scanning—case study. *Procedia Eng.*, 149:197–202, 2016.
- [157] D Sims-Waterhouse. *Camera-based close-range coordinate metrology*. PhD thesis, University of Nottingham, Nottingham, UK, 2019.
- [158] VDI/VDE 2634. *Optical 3D—Part 3: Measuring Systems—Multiple View Systems based on Area Scanning*. Gesellschaft Mess- und Automatisierungstechnik, Berlin, 2014.
- [159] Bryce E. Bayer. Color imaging array, 1975. US Patent US3971065A.
- [160] GPL Software. *CloudCompare (version 2.11.3)*, 2020.
- [161] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. vis.*, 13:119–152, 1994.
- [162] DigitalSurf. Mountainsmap, 2020.
- [163] ISO 25178. *Geometrical Product Specifications (GPS)—Surface Texture: Areal—Part 2: Terms, Definitions and Surface Texture Parameters*. International Organization for Standardization, Geneva, 2012.
- [164] The University of Nottingham. Uon high performance computing. <https://www.nottingham.ac.uk/dts/researcher/compute-services/high-performance-computing.aspx>, 2022.
- [165] EIA RS-274-D. *Interchangeable Variable Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically Controlled Machines*. Washington D.C.: Electronic Industries Association, 1979.
- [166] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *Proc. RRPR*, pages 60–74, 2016.
- [167] D Cernea. Openmvs: Open multiple view stereovision (version 2.0.1), 2022.
- [168] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard,

- Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Leventberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [169] François Chollet et al. Keras. <https://keras.io>, 2015.
- [170] MATLAB. *version 9.6 (R2019a)*. The MathWorks Inc., Massachusetts, 2019.
- [171] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Amsterdam, 2018.
- [172] Mikhail Konnik and James Welsh. High-level numerical simulations of noise in ccd and cmos photosensors: review and tutorial. *arXiv:1412.4031*, 2014.
- [173] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255. IEEE, 2009.
- [174] Mingxing Tan and Quoc Le. Efficientnet: rethinking model scaling for convolutional neural networks. In *Proc. ICML*, pages 6105–6114, 2019.
- [175] Gary Bradski. The opencv library. *Dr. Dobb’s Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.
- [176] Richard I Hartley, Eric Hayman, Lourdes de Agapito, and Ian Reid. Camera calibration and the search for infinity. In *Proc. ICCV*, volume 1, pages 510–517. IEEE, 1999.
- [177] Xiobing Zha. Research on process planning and program generation technology for key parts of marine diesel engine. Master’s thesis, Jiangsu University of Science and Technology, 2017.
- [178] Scott O Mason and Armin Grün. Automatic sensor placement for accurate dimensional inspection. *Comput. Vis. Image Und.*, 61:454–467, 1995.
- [179] Konstantinos A Tarabanis, Peter K Allen, and Roger Y Tsai. A survey of sensor planning in computer vision. *IEEE Trans. Robot. Autom.*, 11:86–104, 1995.

- [180] Elivelton O Rangel, Daniel G Costa, and Angelo Loula. On redundant coverage maximization in wireless visual sensor networks: Evolutionary algorithms for multi-objective optimization. *Appl. Soft Comput.*, 82:105578, 2019.
- [181] L Barazzetti. Network design in close-range photogrammetry with short baseline images. In *ISPRS Ann.Photogramm. Remote Sens. Spatial Inf. Sci.*, pages 17–23, 2017.
- [182] Ali Hosseininaveh Ahmadabadian, Stuart Robson, Jan Boehm, and Mark Shortis. Stereo-imaging network design for precise and dense 3d reconstruction. *The Photogram. Rec.*, 29:317–336, 2014.
- [183] Aaron Mavrinac and Xiang Chen. Modeling coverage in camera networks: A survey. *Int. J. Comput. Vis.*, 101:205–226, 2013.
- [184] B Alsadik, M Gerke, and G Vosselman. Visibility analysis of point cloud in close range photogrammetry. *ISPRS Ann.Photogramm. Remote Sens. Spatial Inf. Sci.*, 2:9, 2014.
- [185] Ravish Mehra, Pushkar Tripathi, Alla Sheffer, and Niloy J Mitra. Visibility of noisy point cloud data. *Comput. Graph.*, 34:219–230, 2010.
- [186] Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. *J. Graph. Tools*, 2:21–28, 1997.
- [187] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM Trans. Graph.*, volume 26, page 24. 2007.
- [188] Sagi Katz and Ayellet Tal. On the visibility of point clouds. In *Proc. ICCV*, pages 1350–1358. IEEE, 2015.
- [189] GOM. "<https://www.gom.com/>", 2022.
- [190] F Lavecchia, MG Guerra, and LM Galantucci. Performance verification of a photogrammetric scanning system for micro-parts using a three-dimensional artifact: adjustment and calibration. *Int. J. Adv. Manuf. Tech.*, 96:4267–4279, 2018.
- [191] Bala Muralikrishnan, Steve Phillips, and Daniel Sawyer. Laser trackers for large-scale dimensional metrology: A review. *Precis. Eng.*, 44:13–28, 2016.
- [192] Gianluca Percoco, Maria Grazia Guerra, Antonio Jose Sanchez Salmeron, and Luigi Maria Galantucci. Experimental investigation on camera calibration for 3d photogrammetric scanning of micro-features for micrometric resolution. *Int. J. Adv. Manuf. Tech.*, 91:2935–2947, 2017.

- [193] Danny Sims-Waterhouse, Samanta Piano, and Richard Leach. Verification of micro-scale photogrammetry for smooth three-dimensional object measurement. *Meas. Sci. Technol.*, 28:055010, 2017.
- [194] Andrew O’Riordan, Thomas Neue, Gerard Dooly, and Daniel Toal. Stereo vision sensing: Review of existing systems. In *Proc. ICST*, pages 178–184, 2018.
- [195] Rui Fan, Li Wang, Mohammad Junaid Bocus, and Ioannis Pitas. Computer stereo vision for autonomous driving. *arXiv:2012.03194*, 2020.
- [196] Krzysztof Woloszyk, Pawel Michal Bielski, Yordan Garbatov, and Tomasz Mikulski. Photogrammetry image-based approach for imperfect structure modelling and fe analysis. *Ocean Eng.*, 223:108665, 2021.
- [197] Ewelina Rupnik, Mehdi Daakir, and Marc Pierrot Deseilligny. Micmac—a free, open-source solution for photogrammetry. *Open Geospat. Data Softw. Stand.*, 2:1–9, 2017.
- [198] Chawin Sathirasethawong, Changming Sun, Andrew Lambert, and Murat Tahtali. Foreground object image masking via epi and edge detection for photogrammetry with static background. In *Proc. ISCV*, pages 345–357, 2019.
- [199] David L Donoho. De-noising by soft-thresholding. *IEEE Trans. Inf. Theory*, 41(3):613–627, 1995.
- [200] Jean-Luc Starck, Emmanuel J Candès, and David L Donoho. The curvelet transform for image denoising. *IEEE Trans. Image Process.*, 11:670–684, 2002.
- [201] Sara Zada, Yassine Tounsi, Manoj Kumar, Fernando Mendoza-Santoyo, and Abdelkrim Nassim. Contribution study of monogenic wavelets transform to reduce speckle noise in digital speckle pattern interferometry. *Opt. Eng.*, 58:034109–034109, 2019.
- [202] Yassine Tounsi, Manoj Kumar, Abdelkrim Nassim, and Fernando Mendoza-Santoyo. Speckle noise reduction in digital speckle pattern interferometric fringes by nonlocal means and its related adaptive kernel-based methods. *Applied Optics*, 57(27):7681–7690, 2018.
- [203] Yassine Tounsi, Manoj Kumar, Abdelkrim Nassim, Fernando Mendoza-Santoyo, and Osamu Matoba. Speckle denoising by variant nonlocal means methods. *Applied optics*, 58(26):7110–7120, 2019.

- [204] Fugui Hao, Chen Tang, Min Xu, and Zhenkun Lei. Batch denoising of espi fringe patterns based on convolutional neural network. *Applied optics*, 58(13):3338–3346, 2019.
- [205] Francesco Banterle, Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain. In *Comput. Graph. Forum*, volume 31, pages 19–32.
- [206] Ping Zhou, Wenjun Ye, Yaojie Xia, and Qi Wang. An improved canny algorithm for edge detection. *J. Comput. Inf. Syst.*, 7:1516–1523, 2011.
- [207] A Rosebrock. *Zero-parameter, automatic Canny edge detection with Python and OpenCV*. PyImageSearch, 2015.
- [208] Shou-Ming Hou, Chao-Lan Jia, Ya-Bing Wang, and Mackenzie Brown. A review of the edge detection technology. *Proc. STAIQC*, 1:26–37, 2021.
- [209] Thierry Bouwmans, Sajid Javed, Maryam Sultana, and Soon Ki Jung. Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. *Int. J. Neural Netw.*, 117:8–66, 2019.
- [210] Youzi Xiao, Zhiqiang Tian, Jiachen Yu, Yinshu Zhang, Shuai Liu, Shaoyi Du, and Xuguang Lan. A review of object detection based on deep learning. *Multimed. Tools Appl.*, 79:23729–23791, 2020.
- [211] Konrad Heidler, Lichao Mou, Celia Baumhoer, Andreas Dietz, and Xiao Xiang Zhu. Hed-unet: Combined segmentation and edge detection for monitoring the antarctic coastline. *IEEE Trans Geosci Remote Sens.*, 60:1–14, 2021.
- [212] Xian Tao, Dapeng Zhang, Wenzhi Ma, Xilong Liu, and De Xu. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Appl. Sci.*, 8(9):1575, 2018.
- [213] Rong Su and Richard Leach. Physics-based virtual coherence scanning interferometer for surface measurement. *Light Adv. Manuf.*, 2(2):120–135, 2021.
- [214] Luke Todhunter, Nicola Senin, Richard Leach, Simon Lawes, Francois Blayerton, and Peter Harris. Flexible software framework for the generation of simulated surface topography. In *Proc. euspen Int. Conf.*, 2018.
- [215] Jesus Pineda, Hernando Altamar-Mercado, Lenny A Romero, and Andres G Marugo. Toward the generation of reproducible synthetic surface data in optical metrology. In *Proc. SPIE*, volume 11397, page 113970C, 2020.

- [216] Chi Fai Cheung and Wing Bun Lee. Modelling and simulation of surface topography in ultra-precision diamond turning. *Proc. Inst. Mech. Eng. B J. Eng. Manuf.*, 214(6):463–480, 2000.
- [217] Tong Gao, Weihong Zhang, Kepeng Qiu, and Min Wan. Numerical simulation of machined surface topography and roughness in milling process. *J. Manuf. Sci. Eng.*, 128:96–103, 2006.
- [218] Rafal Reizer, Lidia Galda, Andrzej Dzierwa, and Pawel Pawlus. Simulation of textured surface topography during a low wear process. *Tribol. Int.*, 44(11):1309–1319, 2011.
- [219] Xiangman Zhou, Haiou Zhang, Guilan Wang, and Xingwang Bai. Three-dimensional numerical simulation of arc and metal transport in arc welding based additive manufacturing. *Int. J. Heat Mass Transf.*, 103:521–537, 2016.
- [220] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.
- [221] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proc. ICML*, pages 214–223. PMLR, 2017.
- [222] Lewis Newton, Nicola Senin, Evangelos Chatzivagiannis, Bethan Smith, and Richard Leach. Feature-based characterisation of ti6al4v electron beam powder bed fusion surfaces fabricated at different surface orientations. *Addit. Manuf.*, 35:101273, 2020.
- [223] C Körner. Additive manufacturing of metallic components by selective electron beam melting—a review. *Inter. Mater. Rev.*, 61(5):361–377, 2016.
- [224] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*, chapter Softmax units for multinoulli output distributions, pages 180–184. MIT press, Cambridge, 2016.
- [225] Richard Leach. *Characterisation of areal surface texture*. Springer, 2014.
- [226] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4401–4410. IEEE/CVF, 2019.
- [227] Tobias Hinz, Matthew Fisher, Oliver Wang, and Stefan Wermter. Improved techniques for training single-image gans. In *Proc. WACV*, pages 1300–1309. IEEE/CVF, 2021.

- [228] Poliigon. <https://www.poliigon.com/>, 2020.
- [229] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Proc. NIPS*, pages 9841–9850, 2020.
- [230] Genevieve Diesing. Fixturing 101: Crucial for quality. *Quality*, 61:12–12, 2022.
- [231] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv:1905.05055*, 2019.
- [232] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoonah Asghar, and Brian Lee. A survey of modern deep learning based object detection models. *Digit. Signal Process.*, page 103514, 2022.
- [233] Yong Hong, Jin Liu, Zahid Jahangir, Sheng He, and Qing Zhang. Estimation of 6d object pose using a 2d bounding box. *Sensors*, 21:2939, 2021.
- [234] Tewodros Legesse Munea, Yalew Zelalem Jembre, Halefom Tekle Weldegebriel, Longbiao Chen, Chenxi Huang, and Chenhui Yang. The progress of human pose estimation: a survey and taxonomy of models applied in 2d human pose estimation. *IEEE Access*, 8:133330–133348, 2020.
- [235] Liangchen Song, Gang Yu, Junsong Yuan, and Zicheng Liu. Human pose estimation and its application to action recognition: A survey. *J. Vis. Commun. Image Represent.*, 76:103055, 2021.
- [236] Weiya Chen, Chenchen Yu, Chenyu Tu, Zehua Lyu, Jing Tang, Shiqi Ou, Yan Fu, and Zhidong Xue. A survey on hand pose estimation with wearable sensors and computer-vision-based methods. *Sensors*, 20(4):1074, 2020.
- [237] Guoguang Du, Kai Wang, Shiguo Lian, and Kaiyong Zhao. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artif. Intell. Rev.*, 54:1677–1734, 2021.
- [238] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proc. CVPR*, pages 3343–3352, 2019.
- [239] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proc. CVPR*, pages 431–440.
- [240] Diyar Khalis Bilal, Mustafa Unel, Lutfi Taner Tunc, and Bora Gonul. Development of a vision based pose estimation system for robotic machining and improving its

- accuracy using lstm neural networks and sparse regression. *Robot. Comput Integr. Manuf.*, 74:102262, 2022.
- [241] K Deergha Rao and MNS Swamy. Spectral analysis of signals. In *Digit. Signal Process.*, pages 721–751. 2018.
- [242] Michael JD Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.*, 7:155–162, 1964.
- [243] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16:1190–1208, 1995.
- [244] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods*, 17:261–272, 2020.
- [245] WH Teukolsky, S Vetterling, and W Flannery. Golden section search in one dimension. In *Numerical Recipes: The Art of Scientific Computing*, pages 492–496. Cambridge Univ. Press, 2007.
- [246] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.*, 18:1–43, 2018.
- [247] Robert Paige and Shaye Koenig. Finite differencing of computable expressions. *Proc. TOPLAS*, 4:402–454.
- [248] Lukas Marsalek, Armin Hauber, and Philipp Slusallek. High-speed volume ray casting with cuda. In *Proc. IEEE SIRT*, pages 185–185, 2008.
- [249] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

- [250] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proc. ICCV*, pages 7678–7687, 2019.
- [251] Zongxin Yang, Xin Yu, and Yi Yang. Dsc-posenet: Learning 6dof object pose estimation via dual-scale consistency. In *Proc. CVPR*, pages 3907–3916, 2021.
- [252] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In *Proc. CVPR*, pages 6738–6748, 2022.
- [253] Xin Yu, Zheyu Zhuang, Piotr Koniusz, and Hongdong Li. 6dof object pose estimation via differentiable proxy voting loss. *arXiv:2002.03923*, 2020.
- [254] Jaewoo Park and Nam Ik Cho. Dprost: 6-dof object pose estimation using space carving and dynamic projective spatial transformer. *arXiv:2112.08775*, 2021.
- [255]
- [256] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *Proc. ECCV*, pages 536–551, 2014.

Abbreviations

3D three-dimensional.	MAE mean absolute error.
AM additive manufacturing.	MAPE mean absolute percentage error.
ANN artificial neural network.	MCDDM Midlands Centre for Data Driven Metrology.
API application programming interface.	ML machine learning.
BFGS Broyden–Fletcher–Goldfarb–Shanno.	MMT Manufacturing Metrology Team.
BSDF bi-directional scattering distribution function.	MMT-LS MMT laser speckle.
CAD computer aided design.	MSE mean squared error.
CMM coordinate measurement machine.	MTL multi-task learning.
CMOS complementary metal-oxide semiconductor.	NBV next best view.
CMS coordinate measurement system.	OpenCV open computer vision.
CNC computer numerical control.	OpenMVG open multi-view geometry.
CNN convolutional neural network.	OpenMVS open multi-view stereo.
CPU central processing unit.	PBF powder bed fusion.
CSI coherence scanning interferometry.	PBR physically based rendering.
CUDA compute unified device architecture.	PCA principal component analysis.
DFP digital fringe projection.	PDF probability density function.
DoF degrees-of-freedom.	PG-GAN progressively growing generative adversarial network.
DoG difference of Gaussians.	PMVS patch-based multi-view stereopsis.
EB-PBF electron beam powder bed fusion.	PTM point to mesh.
FCNN fully convolutional neural network.	RANSAC random sample consensus.
FDM fusion deposition modelling.	ReLU rectified linear unit.
FOV field of view.	ResNet residual neural network.
FV focus variation.	RMS root mean squared.
GA genetic algorithm.	RNN recurrent neural network.
GAN generative adversarial network.	SFM structure from motion.
GPU graphics processing unit.	SIFT scale invariant feature transform.
HPC high performance cluster.	STL standard tessellation language.
HPR hidden point removal.	SVM support vector machine.
ICP iterative closest point.	SVR support vector regression.
IO input/output.	Ti64 Ti-6Al-4V.
IRM information rich metrology.	XLA accelerated linear algebra.
KDE kernel density estimation.	
LiDAR light detection and ranging.	
LSF line-spread function.	
LSM learned stereo machine.	

Nomenclature

Coordinate systems	k	Fringe number
(U, V) Image resolution	Z	Height
$[\alpha, \theta, r]$ 3D polar coordinates	Camera modelling	
$[\theta, \phi, \kappa]$ Local rotation	$[c_x, c_y]$	Principal point offset
$[u, v]$ Image coordinates	$[p_u, p_v]$	Pixel pitch
$[X, Y, Z]$ Global coordinate system	γ	Skew angle
$[x, y, z]$ Local coordinate system	A	Projection matrix
a axis-angle representation	E	Extrinsic matrix
D_i camera-to-object distance	K	Intrinsic matrix
P^n A 3D point	R	Rotation matrix
t Time	T	Translation vector
Phase unwrapping	f	Focal length
δ_i Phase shift	k_n	Radial distortion parameters
λ Fringe wavelength	p_n	Tangential distortion parameters
$\mathbf{I}(u, v)$ Intensity image	Q	Reprojection function
$\phi(u, v)$ Wrapped phase	r	Radial distance
$\psi(u, v)$ Unwrapped phase	s	Scale factor

Machine learning	G	Gradient magnitude
η		Learning rate
\mathbf{x}'		Unseen features
$\mathbf{x} \in \mathcal{X}$		Features
$\mathbf{y} \in \mathcal{Y}$		Labels
\mathbf{z}		Intermediate weighted sum
$\phi(\mathbf{x})$		Feature map
$a(\mathbf{z})$		Non-linear activation
A		Categorisation accuracy
D		Discriminator model
E		Loss function
$f(\mathbf{x})$		Target function
G		Generator model
$h(\mathbf{x})$		Hypothesis
h_n		Hidden node
I_n		Input node
$k(\mathbf{x}, \mathbf{x}')$		Kernel function
O_n		Output node
$w_{i,j}^n$		Weight
Image processing		
Θ		Gradient direction
	I	Image matrix
	σ_r	Range smoothing factor
	σ_s	Spatial smoothing factor
		High/low pixel thresholds
	Surface texture	
	Sal	Autocorrelation length
	Sdq	Root mean square gradient
	Sdr	Developed interfacial area ratio
	Sq	Root-mean-square height deviation
	Sz	Maximum height deviation
	Other symbols	
	B	Approximate Hessian matrix
	H	Hessian matrix
	J	Jacobian matrix
	ω	Weighting coefficient
	ϕ_r	Reflected phase
	ϕ_s	Reference phase
	σ	Standard deviation

$A(t)$	Amplitude
c	The speed of light
d	Depth
f_m	Frequency modulation
$S(t)$	Reference signal
S^n	n -dimensional unit sphere

Appendices

Appendix A

GOM system performance verification

GOM GmbH
Schmitzstraße 2
38122 Braunschweig
Germany



GOM Acceptance Test

190401_CP40-320-54605

Certificate No.

Acceptance/Reverification According to VDI/VDE 2634, Part 3

This document may only be distributed in its entirety and without changes. Excerpts and changes require the approval of the issuing company. This document was created electronically and is valid without a signature.

General Data

System: ATOS Core 300 MV300
Serial number: 190323
Measuring volume: MV300 (300x230x230) mm
Date: 01.04.2019
Inspector: Jan Kristen
Measurement temperature: 21.6 °C

Artifact

General

Name: Z0012
Calibration date: 29.11.2018
Calibration ID: 1686/D-K-15007-02-00/2018-11
Calibration laboratory: Carl Zeiss Industrielle Messtechnik GmbH
Calibration temperature: 20.2 °C
Expansion coefficient for sphere spacing: $4.00 \cdot 10^{-6} \text{ K}^{-1}$
Expansion coefficient for diameter: $10.50 \cdot 10^{-6} \text{ K}^{-1}$

Basic dimensions

Sphere spacing: 160 mm
Diameter left sphere: 25 mm
Diameter right sphere: 25 mm

Calibrated nominal dimensions

Sphere spacing: 160.0339 mm
Diameter left sphere: 25.0044 mm
Diameter right sphere: 25.0048 mm

Measurement Parameters**Measurement Settings**

Number of exposure times:	1
Min. fringe contrast:	15 gray values
State: Avoid points at strong brightness differences?:	Yes
State: Avoid Triple Scan points?:	No
State: Avoid Triple Scan points at strong brightness differences?:	Yes
Max. residual:	0.20 pixel
Depth limitation mode:	Automatic depth limitation
Corner mask size:	35
Measurement resolution:	Full resolution

Settings of Checks

State: Check "Sensor movement"?:	Yes
Max. sensor movement:	0.10 pixel
State: Check "Lighting change"?:	Yes

Sensor Calibration**General**

Calibration date:	Mon Apr 1 15:18:14 2019
Measurement temperature:	21.8 °C

Calibration Object

Calibration object type:	Panel (Triple Scan)
Calibration object name:	CP40-320-54605
Test distances:	574.5725 / 574.5314 mm
Certification temperature:	20.0 °C
Expansion coefficient:	22.67 10 ⁻⁶ K ⁻¹

Calibration Settings

Focal length (camera):	12.50 mm
Focal length (projector):	8.00 mm
Light intensity:	100%
Snap mode:	Double snap
Max. ellipse quality:	0.40 pixel

Calibration Result

Calibration deviation:	0.039 pixel (Quality check: Good)
Calibration deviation (optimized):	0.016 pixel
Projector calibration deviation:	0.033 pixel (Quality check: Good)
Projector calibration deviation (optimized):	0.014 pixel
Camera angle:	31.88 °
Height variance:	230.227 mm
Measuring volume:	317 x 238 x 244 mm

Acceptance/Reverification Test**General**

Number of test positions (measurement series): 3
 Nominal diameter of left sphere with temperature correction: 25.0048 mm
 Nominal diameter of right sphere with temperature correction: 25.0052 mm
 Nominal sphere spacing with temperature correction: 160.0348 mm

Parameter Probing Error Form, Left Sphere

Pos ²⁾	M ³⁾	P ⁴⁾	Min. deviation	Max. deviation	Range of deviation	Probing error form (sigma)
1	10	12288	-0.008 mm	0.007 mm	0.014 mm	0.002 mm
2	10	9968	-0.007 mm	0.007 mm	0.014 mm	0.002 mm
3	10	12354	-0.010 mm	0.008 mm	0.018 mm	0.003 mm

Parameter Probing Error Form, Right Sphere

Pos ²⁾	M ³⁾	P ⁴⁾	Min. deviation	Max. deviation	Range of deviation	Probing error form (sigma)
1	10	12529	-0.006 mm	0.007 mm	0.013 mm	0.002 mm
2	10	9741	-0.007 mm	0.007 mm	0.014 mm	0.002 mm
3	10	12467	-0.009 mm	0.008 mm	0.016 mm	0.003 mm

Parameter Probing Error Size, Left Sphere

Pos ²⁾	M ³⁾	P ⁴⁾	Diameter (actual)	Diameter (nominal) ¹⁾	Probing error (size)
1	10	12288	25.006 mm	25.005 mm	0.001 mm
2	10	9968	25.005 mm	25.005 mm	-0.000 mm
3	10	12354	25.002 mm	25.005 mm	-0.003 mm

Parameter Probing Error Size, Right Sphere

Pos ²⁾	M ³⁾	P ⁴⁾	Diameter (actual)	Diameter (nominal) ¹⁾	Probing error (size)
1	10	12529	25.007 mm	25.005 mm	0.002 mm
2	10	9741	25.005 mm	25.005 mm	-0.000 mm
3	10	12467	25.001 mm	25.005 mm	-0.004 mm

Sphere Spacing Error

Pos ²⁾	M ³⁾	Sphere spacing (actual)	Sphere spacing (nominal) ¹⁾	Sphere spacing error
1	10	160.050 mm	160.035 mm	0.015 mm
2	10	160.045 mm	160.035 mm	0.010 mm
3	10	160.048 mm	160.035 mm	0.013 mm

Parameter Length Measurement Error

Pos ²⁾	M ³⁾	Length (actual)	Length (nominal) ¹⁾	Length measurement error
1	10	185.056 mm	185.040 mm	0.016 mm
2	10	185.047 mm	185.040 mm	0.007 mm
3	10	185.045 mm	185.040 mm	0.006 mm

¹⁾ With temperature correction

²⁾ Test position

³⁾ Number of measurements

⁴⁾ Number of points

Summary Acceptance/Reverification Test

Parameter	Maximum deviation	Limit
Probing error form (sigma)	0.003 mm	0.006 mm
Probing error (size)	-0.004 mm	0.027 mm
Sphere spacing error	0.015 mm	0.020 mm
Length measurement error	0.016 mm	0.047 mm

Additional Information

A detailed description of the acceptance test is compiled in the following document:

GOM Acceptance Test - Process Description,
Acceptance test according to the guideline VDI/VDE 2634 Part 3
Document number: 000000476_...

Test requirements are stated, the exact test procedure (incl. software operation) is documented and the used parameters are listed.

In addition, limits are defined with respect to:

- Operation modes (adjustment and configuration possibilities of the measuring system)
- Operation conditions (outside influences)

The documentation in the version valid at the time of the test is part of the GOM Acceptance Test protocol and is enclosed.

The respective current version is available for download under <https://support.gom.com>.

Appendix B

CMM calibration results.

B.1 Calibration certificate

CERTIFICATE OF CALIBRATION		
Issued By	Mitutoyo (UK) Ltd. Calibration Laboratory	
Date of Issue	16 Feb 2021	Certificate No. 310994



0332

Mitutoyo

Calibration Laboratory:

Mitutoyo (UK) Ltd
6 Banner Park, Wickmans Drive
Coventry, West Midlands
CV4 9XA, United Kingdom
T +44 (0)2476 426300
F +44 (0)2476 426339
calibration@mitutoyo.co.uk

Head Office:

Mitutoyo (UK) Ltd
West Point Business Park, Joule Road
Andover, Hampshire
SP10 3UX, United Kingdom
T +44 (0)1264 353123
F +44 (0)1264 354883
enquiries@mitutoyo.co.uk

Page: 1 of 4

Approved Signatory:

V. Enache

CUSTOMER

University Of Nottingham
Nottingham

MANUFACTURER

Mitutoyo

DESCRIPTION

Crysta Apex S7106 Coordinate Measuring Machine

IDENTIFICATION

N/A Serial No. 60681224

CALIBRATION CONDITIONS

Temperature range during calibration 21.5 - 21.7°C

BASIS OF CALIBRATION

To performance verify to the requirements of BS EN ISO 10360-2: 2009 & 10360-5: 2010

DATE OF CALIBRATION

11 Feb 2021

TYPE

Double column design with highly rigid guide rails and air bearing configuration, CNC or joystick controlled utilizing touch trigger type probes.

Decision: Conformation status to requirements/tolerance (for the decision rule basis see page 3 point 6)

$E_{0,MPE}$	1.7 + 3.0 L/M μm	Probably conforms due to uncertainties
$E_{150,MPE}$	1.7 + 3.0 L/M μm	Probably conforms due to uncertainties
$R_{0, MPL}$	1.3 μm	Conforms
P_{FTU}	1.7 μm	Conforms
P_{STU}	1.7 μm	Conforms

Authorised By

This certificate is issued in accordance with the laboratory accreditation requirements of the United Kingdom Accreditation Service. It provides traceability of measurement to the SI system of units and/or to units of measurement realised at the National Physical Laboratory or other recognised national metrology institutes. This certificate may not be reproduced other than in full, except with the prior written approval of the issuing laboratory.

CERTIFICATE OF CALIBRATION



UKAS Accredited Calibration Laboratory No. 0332
 Mitutoyo (UK) Ltd, 6 Banner Park, Wickmans Drive
 Coventry, West Midlands CV4 9XA, United Kingdom

Certificate No: 310994

Page 2 of 4

PROBE HEAD	TYPE	PH10MQ	SERIAL NO.	1P6A40
TOUCH TRIGGER PROBE	TYPE	SP25M	SERIAL NO.	1LEP43
CUSTOMER'S PROBE QUALIFICATION SPHERE	SERIAL NO.	D-11299	CERT.	294290
SOFTWARE	GEOPAK 4.2 R3			
CMM LOCATION	Temperature Controlled Room			
MITUTOYO APPROVED OPERATOR	Jean Memmory			

SUMMARY RESULT: (Measured data)

$E_{0,MPE}$	1.2 + 2.2 L/M μm	where L = Length in metres
$E_{150,MPE}$	1.1 + 2.1 L/M μm	where L = Length in metres
R_0	0.60 μm	
P_{FTU}	0.90 μm	
P_{STU}	0.22 μm	

THE ESTIMATED UNCERTAINTY OF CALIBRATION MEASUREMENT

LENGTH: Estimated uncertainty of measurement at a coverage factor of $k = 2$ is $\pm 0.4 + 2.2 L \text{ m } \mu\text{m}$ where L = Length in metres.


P_{FTU} : Estimated uncertainty of measurement at a coverage factor of $k = 2$ is $\pm 0.14 \mu\text{m}$

P_{STU} : Estimated uncertainty of measurement at a coverage factor of $k = 2$ is $\pm 0.52 \mu\text{m}$

The machine is verified to either the Manufacturer's Specification or the agreed Customer's Specification. The compliance statement is based on ILAC-G8: 09/2019 and ISO 14253-1: 2017.

The reported expanded uncertainty is based on a standard uncertainty multiplied by a coverage factor $k = 2$, providing a coverage probability of approximately 95 %. The uncertainty evaluation has been carried out in accordance with UKAS requirements.

B.2 Method of calibration

CERTIFICATE OF CALIBRATION		
	UKAS Accredited Calibration Laboratory No. 0332 Mitutoyo (UK) Ltd, 6 Banner Park, Wickmans Drive Coventry, West Midlands CV4 9XA, United Kingdom	Certificate No: 310994 Page 3 of 4

METHOD OF CALIBRATION

1. The above referenced Coordinate Measuring Machine and Probing System was performance verified in accordance with the requirements of BS EN ISO 10360 - Part 2: 2009 and BS EN ISO 10360 - Part 5: 2010.

The machine was brought up to its normal operating conditions. The verification was then made using a Step Gauge Length Standard to verify the length measurements and a Precision Sphere to verify the probing system. The recorded measurements were used to calculate the CMM's error and the value(s) were compared to either the Manufacturer's Specification or the Agreed Customer's Specification.

2. The average temperature in the working volume of the CMM was found to be 21.6°C. The maximum temperature change in any 1 hour period was 0.2°C.

The temperature range was 21.5°C to 21.7°C.

3. The probe speed used during the verification was 3 mm/second.

4. See page 4 for diagrams of the measurement lines.

5. Graphs also included for completeness.

6. The calibration certificate includes a conformity statement based on ILAC-G8: 09/2019 – Guidelines on Decision Rules and Statements of Conformity and ISO 14253-1: 2017 (for details please see DOC-006-42.2 which has been included with this calibration certificate

7. 66% measuring length has been measured during the calibration

CERTIFICATE OF CALIBRATION

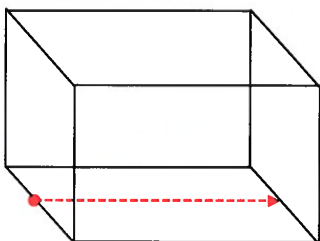
Mitutoyo

UKAS Accredited Calibration Laboratory No. 0332
Mitutoyo (UK) Ltd, 6 Banner Park, Wickmans Drive
Coventry, West Midlands CV4 9XA, United Kingdom

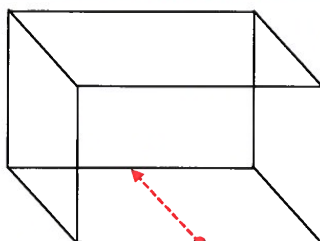
Certificate No: 310994
Page 4 of 4

Ram Axis Stylus Tip Off-Set 0mm

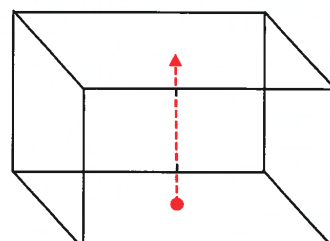
E₀ Position 1



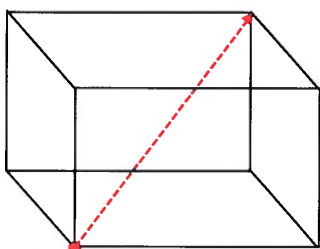
E₀ Position 2



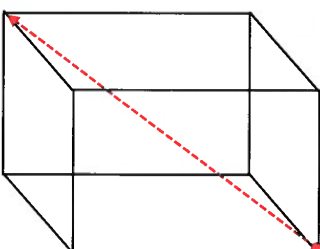
E₀ Position 3



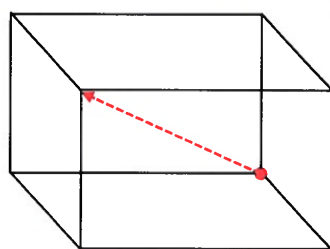
E₀ Position 4



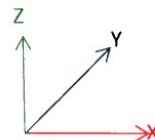
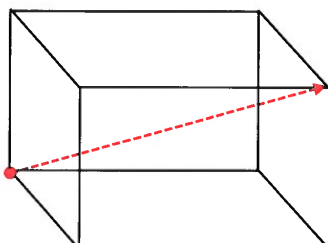
E₀ Position 5



E₀ Position 6

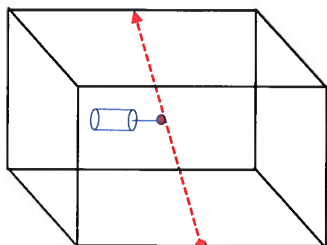


E₀ Position 7

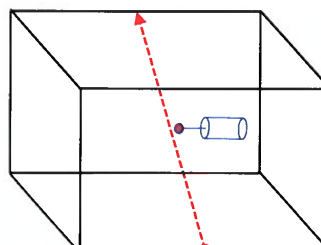


Ram Axis Stylus Tip Off-Set 150mm ±15mm

E₁₅₀ Position 8 (Probe Positive X,Y)



E₁₅₀ Position 9 (Probe Negative X,Y)



** End of report **

ILAC-G8:09/2019 – Guidelines on the reporting of compliance with specification

and

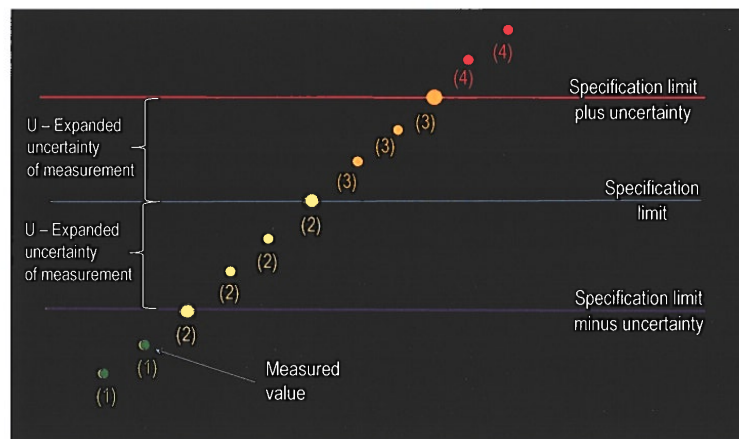
ISO 14253: 2017 – Geometrical product specifications (GPS) – Inspection by measurement of workpieces and measuring equipment

Part 1 - Decision rules for verifying conformity or nonconformity with specifications

When a statement of conformity, according ILAC-G8: 09/2019 and ISO 14253-1: 2017, the uncertainty of measurement is included in the measurement result based on the decision rule displayed below.

On the calibration certificate, one of the following statements will be displayed against each measurement result requiring a conformity statement:

- **CONFORMS** – which means “The measured value complies with the specification limit(s)” displayed below as “**(1)**”
- **PROBABLY CONFORMS** – which means “It is not possible to state compliance using a 95% confidence level for the expanded uncertainty although the measured result is within the specification limit” displayed below as “**(2)**”
- **PROBABLY DOES NOT CONFORM** – which means “It is not possible to state compliance using a 95% confidence level for the expanded uncertainty although the measured result is outside the specification limit” displayed below as “**(3)**”
- **DOES NOT CONFORM** – which means “The measured value does not comply with the specification limit(s)” displayed below as “**(4)**”



NOTE: If the uncertainty of measurement is higher than the required tolerance/specification, no compliance statement will be provided (as specified in **M3003 clause M3.3**). This will be clearly highlighted on the calibration certificate.

B.3 Length error graphs

CRT AS 7106 University of Nottingham ISO10360 Sheet.xlsm

University Of Nottingham

Crysta Apex S7106

60681224

J Memmory

11/02/2021

For Information Only

Customers Expectation Spec	1.70	+	3.00	x L / 1000
----------------------------	------	---	------	------------

Uncertainty	0.40	+	2.22	x L / 1000
-------------	------	---	------	------------

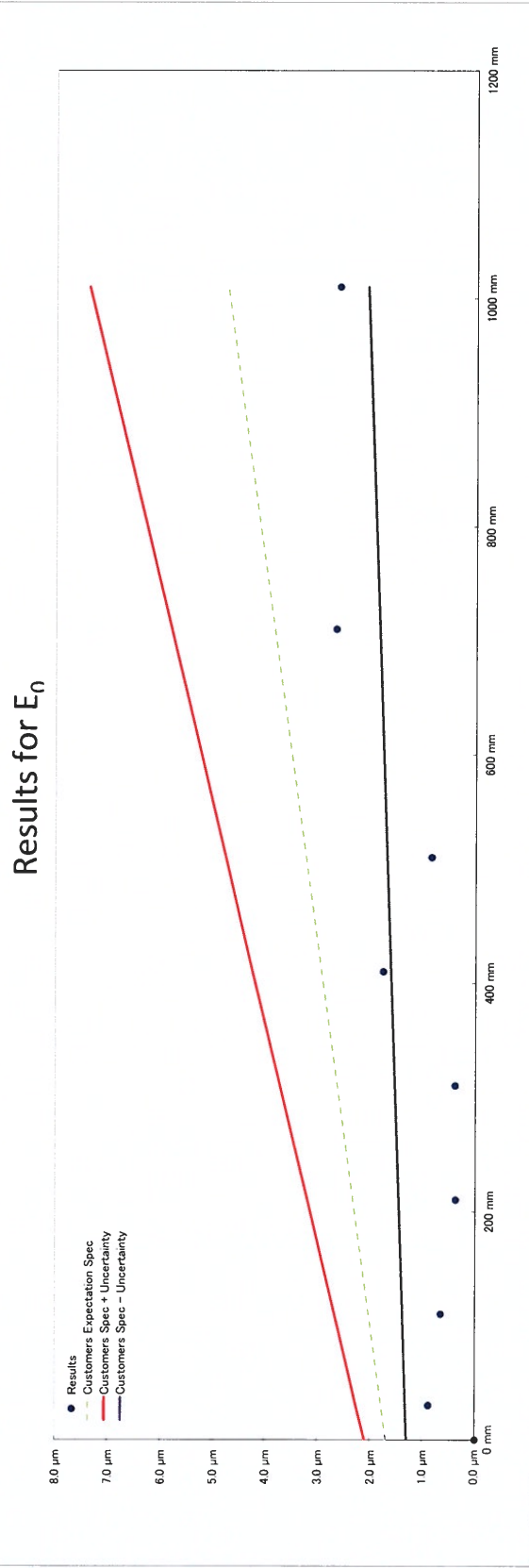
Customers Spec + Uncertainty	2.10	+	5.22	x L / 1000
------------------------------	------	---	------	------------

Customers Spec - Uncertainty	1.30	+	0.78	x L / 1000
------------------------------	------	---	------	------------

Positions	0	30	110	210	310	410	510	710	1010
Results	0.00	0.87	0.64	0.36	0.37	1.76	0.83	2.67	2.61
Customers Expectation Spec	1.70	1.79	2.03	2.33	2.63	2.93	3.23	3.83	4.73

D00-006-04_3-v5

Customers Spec + Uncertainty	2.10	2.26	2.67	2.67	3.20	3.72	4.24	4.76	5.81	7.38
Customers Spec - Uncertainty	1.30	1.32	1.39	1.46	1.54	1.62	1.70	1.85	2.08	2.08



For Information Only

Customers Expectation Spec	1.70	+	3.00	x L / 1000
----------------------------	------	---	------	------------

Uncertainty	0.40	+	2.22	x L / 1000
-------------	------	---	------	------------

Customers Spec + Uncertainty	2.10	+	5.22	x L / 1000
------------------------------	------	---	------	------------

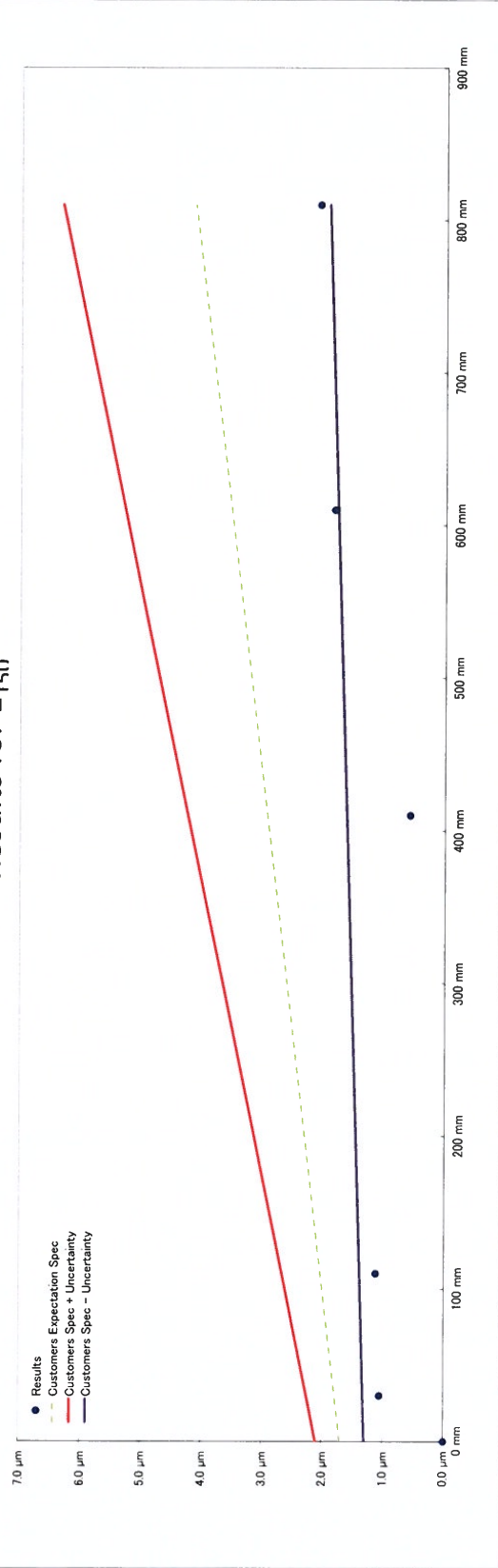
Customers Spec - Uncertainty	1.30	+	0.78	x L / 1000
------------------------------	------	---	------	------------

DOC-006-04.3-v5

Positions	0	30	110	410	610	810
Results	0.00	1.05	1.12	0.58	1.83	2.08
Customers Expectation Spec	1.70	1.79	2.03	2.93	3.53	4.13

Customers Spec + Uncertainty	2.10	2.26	2.67	4.24	5.29	6.33
Customers Spec - Uncertainty	1.30	1.32	1.39	1.62	1.77	1.93

Results for E₁₅₀



Appendix C

Stereo baseline characterisation

The procedure adopted for characterising the baseline of the Taraz system is as follows.

A CMM measurement of the four spherical features on the Tomas artefact, shown in Figure 3.11, was conducted. The spheres are arranged as is shown in Figure C.1 which define a set of 6 sphere-to-sphere distances which are also labelled in Figure C.1.

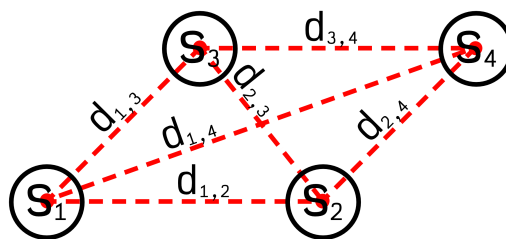


Figure C.1: Sphere arrangement for the characterisation of the stereo baseline distance.

The six sphere-to-sphere distances are extracted from the CMM, the CMM measurement was repeated three times. Then, a 120 image scan

of the artefact was completed using the Taraz system, the output point cloud has an arbitrary unknown scale factor applied to it. This scale factor is given by the term s in Equation 2.9. As there are six known sphere-to-sphere distances across three repeat CMM measurements and a single unknown value s , a least squares approach can be adopted to determine the optimal value of s .

Finally the baseline stereo distance b can be determined from a given pair of cameras c_0 and c_1 as,

$$b = s \cdot |\mathbf{T}_1 - \mathbf{T}_0|, \quad (\text{C.1})$$

where \mathbf{T} is the camera translation vector from the extrinsic matrix. Again, because there are 60 stereo image pairs in the scan the value of b can be optimised using a least squares solution. The results of this procedure are given in the following section.

C.1 Baseline characterisation results

Table C.1 shows the results of sphere fitting to the three sets of CMM data. The nominal dimensions of the artefact were 5 mm spheres spaced at the corners of a 50 mm square.

	Measurement 1/mm					Measurement 2/mm					Measurement 3/mm				
	X	Y	Z	R	RMS	X	Y	Z	R	RMS	X	Y	Z	R	RMS
S1	0.08	-0.12	-0.15	5.05	0.03	0.08	-0.13	-0.19	5.09	0.02	0.06	-0.13	-0.18	5.08	0.031
S2	49.87	-0.30	-0.01	5.06	0.02	49.92	-0.30	-0.11	5.08	0.03	49.87	-0.32	-0.07	5.06	0.024
S3	0.43	49.83	-0.43	5.10	0.02	0.42	49.82	-0.38	5.08	0.02	0.41	49.83	-0.38	5.08	0.022
S4	50.28	49.61	-0.36	5.06	0.02	50.29	49.59	-0.37	5.08	0.02	50.29	49.61	-0.32	5.05	0.018

Table C.1: Sphere fitting to CMM data results.

Table C.2 shows the distances shown in Figure C.1 as extracted from each CMM measurement along with the range and standard deviation across the repeat measurements.

	Measurement			Mean	StDev	Range
	1	2	3			
d_12 /mm	49.79	49.84	49.81	49.812	0.027	0.053
d_13 /mm	49.95	49.96	49.97	49.959	0.010	0.019
d_24 /mm	49.92	49.89	49.93	49.913	0.023	0.046
d_34 /mm	49.86	49.87	49.88	49.871	0.014	0.028
d_23 /mm	70.41	70.44	70.44	70.432	0.016	0.030
d_14 /mm	70.66	70.67	70.69	70.674	0.016	0.031

Table C.2: Sphere-to-sphere distances extracted from repeated CMM measurements.

Table C.3 shows the sphere fitting applied to a 120 image photogrammetric scan from the Taraz system with an unknown arbitrary scale factor.

	X /mm	Y /mm	Z /mm	R /mm	RMS /mm
S1	1.142	1.247	-11.257	0.115	0.027
S2	0.773	0.240	-11.672	0.115	0.023
S3	1.924	0.691	-10.616	0.115	0.016
S4	1.556	-0.315	-11.036	0.115	0.025

Table C.3: Sphere fitting results to a photogrammetric measurement performed by the Taraz system.

Table C.4 shows the determination of the scale factor s . This is first determined individually for each sphere-to-sphere distance, then the mean and standard deviation of the individual scale factors are calculated. The mean scale factor $s = 43.332$ is taken to be the 'true' scale factor for this point cloud.

	Photogram /mm	CMM Mean /mm	Scale Factor (s) /AU
d_12	1.150	49.812	43.330
d_13	1.153	49.959	43.312
d_24	1.151	49.913	43.350
d_34	1.151	49.871	43.338
d_23	1.626	70.432	43.327
d_14	1.631	70.674	43.338
		Mean(s)	43.332
		StDev(s)	0.013

Table C.4: Sphere-to-sphere distances compared and used to calculate the scale factor between the photogrammetric point cloud and the CMM data.

Finally, Table C.5 shows the translation vectors for the stereo camera pair at each imaging position, the unscaled baseline distance is calculated

and then the previously determined scale factor is applied. From the data given in Table C.5 the baseline distance can be determined to be 264.00 mm with a standard deviation of 0.40 mm. In the body of this thesis a baseline distance of $b = 264.00$ mm is therefore used to scale all pointclouds produced by the Taraz system.

N	X_0/mm	Y_0/mm	Z_0/mm	X_1/mm	Y_1/mm	Z_1/mm	Distance/mm	Scaled Baseline/mm
1	2.19	-1.01	-0.22	7.20	-0.89	-3.69	6.10	264.14
2	2.15	0.53	-0.13	7.16	0.94	-3.57	6.10	264.18
3	1.88	2.07	-0.25	6.85	2.78	-3.69	6.08	263.62
4	1.36	3.54	-0.55	6.26	4.53	-4.05	6.09	264.06
5	0.66	4.85	-1.06	5.42	6.10	-4.64	6.09	263.89
6	-0.23	5.96	-1.75	4.38	7.44	-5.44	6.09	263.77
7	-1.25	6.83	-2.57	3.16	8.49	-6.42	6.09	263.74
8	-2.37	7.42	-3.51	1.83	9.20	-7.53	6.09	263.73
9	-3.54	7.70	-4.51	0.44	9.55	-8.73	6.09	263.95
10	-4.73	7.66	-5.55	-0.97	9.53	-9.97	6.10	264.28
11	-5.88	7.31	-6.59	-2.34	9.14	-11.21	6.10	264.45
12	-6.93	6.66	-7.57	-3.60	8.38	-12.38	6.10	264.53
13	-7.85	5.73	-8.46	-4.70	7.30	-13.45	6.10	264.52
14	-8.60	4.58	-9.22	-5.60	5.93	-14.36	6.10	264.54
15	-9.15	3.22	-9.82	-6.26	4.34	-15.09	6.11	264.64
16	-9.47	1.75	-10.24	-6.66	2.58	-15.60	6.11	264.68
17	-9.57	0.19	-10.46	-6.79	0.73	-15.88	6.11	264.92
18	-9.42	-1.38	-10.47	-6.63	-1.13	-15.90	6.11	264.97
19	-9.04	-2.91	-10.27	-6.19	-2.95	-15.68	6.11	264.97
20	-8.45	-4.31	-9.88	-5.50	-4.63	-15.22	6.11	264.60
21	-7.66	-5.54	-9.30	-4.57	-6.11	-14.53	6.10	264.24
22	-6.71	-6.55	-8.57	-3.45	-7.32	-13.65	6.09	263.99
23	-5.64	-7.30	-7.69	-2.18	-8.23	-12.62	6.09	264.05
24	-4.49	-7.76	-6.72	-0.82	-8.80	-11.47	6.09	263.89
25	-3.32	-7.91	-5.69	0.59	-8.99	-10.25	6.10	264.12
26	-2.15	-7.75	-4.65	1.98	-8.82	-9.00	6.10	264.14
27	-1.04	-7.27	-3.63	3.30	-8.27	-7.79	6.09	264.02
28	-0.03	-6.50	-2.68	4.50	-7.37	-6.65	6.09	263.94
29	0.82	-5.47	-1.83	5.53	-6.15	-5.63	6.09	263.89
30	1.49	-4.23	-1.13	6.34	-4.68	-4.79	6.09	263.85
31	0.04	-0.92	-0.06	5.65	-0.97	-2.44	6.09	263.88
32	0.00	0.30	0.02	5.61	0.77	-2.31	6.09	263.86
33	-0.22	1.54	-0.07	5.31	2.53	-2.42	6.09	263.86
34	-0.63	2.73	-0.33	4.75	4.22	-2.77	6.09	263.74
35	-1.20	3.79	-0.74	3.95	5.74	-3.34	6.09	263.74
36	-1.91	4.69	-1.29	2.94	7.03	-4.11	6.08	263.66
37	-2.73	5.39	-1.96	1.77	8.05	-5.06	6.08	263.40
38	-3.64	5.86	-2.71	0.49	8.74	-6.12	6.08	263.47
39	-4.58	6.09	-3.52	-0.85	9.09	-7.28	6.08	263.39
40	-5.53	6.07	-4.36	-2.21	9.07	-8.47	6.08	263.36
41	-6.45	5.79	-5.20	-3.52	8.69	-9.66	6.08	263.42
42	-7.30	5.27	-5.99	-4.74	7.97	-10.80	6.08	263.42
43	-8.65	3.58	-7.32	-6.67	5.61	-12.71	6.09	263.94
44	-9.10	2.49	-7.80	-7.32	4.08	-13.41	6.10	264.17
45	-9.37	1.29	-8.14	-7.71	2.39	-13.91	6.10	264.27
46	-9.44	0.04	-8.32	-7.83	0.61	-14.17	6.10	264.23
47	-9.32	-1.23	-8.33	-7.68	-1.19	-14.21	6.10	264.29
48	-9.02	-2.46	-8.17	-7.27	-2.95	-13.99	6.10	264.24
49	-8.54	-3.59	-7.85	-6.60	-4.56	-13.55	6.09	264.09
50	-7.91	-4.58	-7.39	-5.71	-5.99	-12.89	6.09	263.95
51	-7.14	-5.40	-6.79	-4.63	-7.17	-12.05	6.09	263.93
52	-6.28	-6.01	-6.08	-3.41	-8.05	-11.05	6.09	263.88
53	-5.35	-6.38	-5.30	-2.10	-8.59	-9.95	6.09	263.69
54	-4.40	-6.50	-4.48	-0.74	-8.79	-8.77	6.08	263.58
55	-3.46	-6.37	-3.64	0.60	-8.62	-7.57	6.08	263.56
56	-2.57	-5.99	-2.82	1.87	-8.10	-6.40	6.08	263.53
57	-1.76	-5.38	-2.06	3.03	-7.24	-5.30	6.08	263.53
58	-1.07	-4.55	-1.37	4.02	-6.07	-4.32	6.09	263.68
59	-0.53	-3.54	-0.80	4.80	-4.65	-3.51	6.09	263.81

Table C.5: Determination of the baseline distance for each stereo pair.

Appendix D

Motion characterisation

The purpose of the motion characterisation procedure is to establish a coordinate transformation from global coordinates to machine coordinates. Using this transformation, a list of imaging positions given relative to the centre of the measurement volume can be transformed into motion stage positions and converted to G-code [165]. This G-code can then be executed to conduct the measurement. The process outlined below shows the motion characterisation procedure used for the Taraz system which has five DoFs, but the same procedure can also be used with the MMT system which has only two DoFs.

A checkerboard characterisation target is placed within the measurement volume. The target is imaged at 60 equally spaced positions along the full range of each DoF in the system, while keeping each other DoF constant - in the five DoF Taraz system this leads to 300 images. The OpenCV function `cv2::findChessboardCorners()` is used to detect the characterisation target features, the characterised camera parameters are then used alongside the function `cv2::solve_pnp()` to give the camera extrinsics relative to the characterisation target [175]. For each image it is now known at

what machine coordinates it was imaged from $(M_x, M_y, M_z, M_{\theta_z}, M_{\theta_y})$ and the camera's pose relative to the target (T_x, T_y, T_z) . However, the target coordinate system is misaligned with the global coordinate system, to align these two the procedure shown in Figure D.1 is conducted.

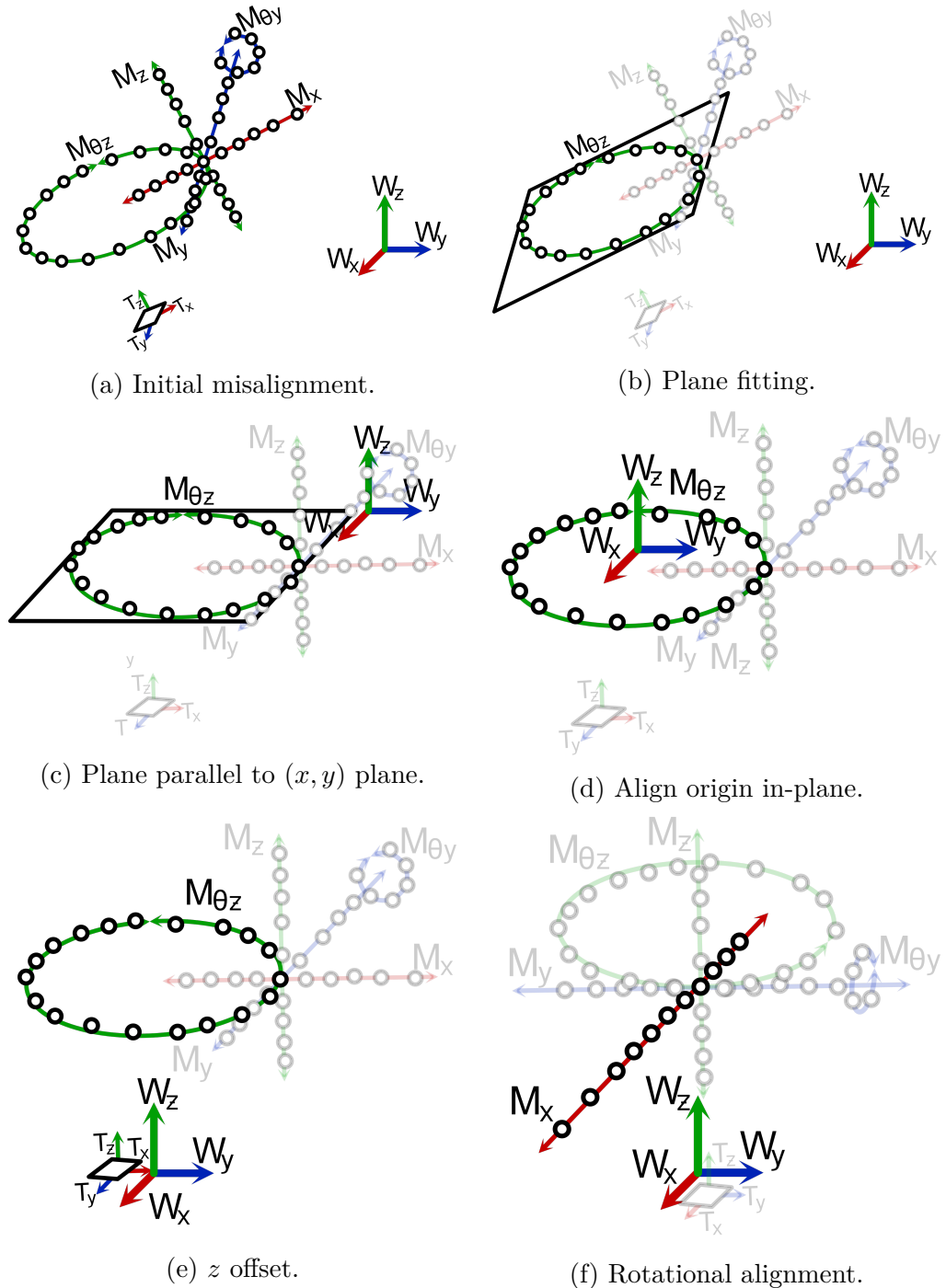


Figure D.1: Global axis alignment for the Taraz system. Each circle indicates an imaging position in the dataset.

Starting with the initial misaligned coordinate systems as shown in Figure D.1a, a plane is fit to the camera positions captured by varying the rotation stage ($M_{\theta z}$) as shown in Figure D.1b. The coordinate system is transformed to make the fit plane parallel with the global x, y plane as shown in Figure D.1c. The centre of rotation is calculated by finding the mean camera location on the fit plane, Figure D.1d shows that the coordinate system is transformed such that the centre of rotation aligns with the global z axis. The coordinate system is then translated such that the origin of the target reference frame is coplanar with the global x, y plane as shown in Figure D.1e. Finally, the coordinate system is rotated such that the mean vector given by camera positions imaged when varying the machine M_x positions is parallel with the global x axis, as shown in Figure D.1f.

As can be seen in Figure D.1f, the relationship between the global coordinate system and the machine coordinate system can now be derived. It can also be seen that, although the mean machine axes are aligned, due to errors in the motion stages the camera positions do not fall exactly on these axes. If very high precision positioning is required, these errors can be individually characterised and accounted for when calculating machine coordinates from a desired global imaging location. In the case of the Taraz system, it was found that the reconstruction results were not sensitive to small errors in the imaging position so only simple linear corrections were applied.

Appendix E

EfficientNet-B5 architecture

Figure E.1 shows the common building blocks of all of the EfficientNet architectures as originally published [174].

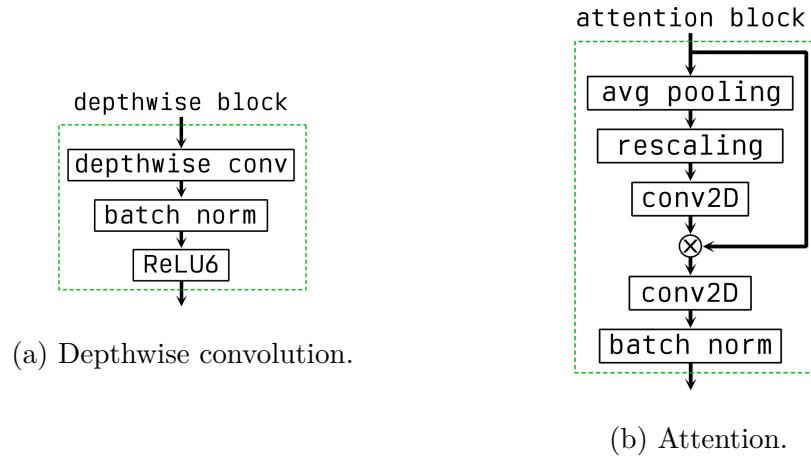


Figure E.1: Building blocks of the EfficientNet model.

Figure E.2 shows the full architecture of the EfficientNet-B5 model as adapted for use in the characterisation procedure outlined in Chapter 4. As the model is very large, for the sake of brevity the sections highlighted in red in Figure E.2 are repeated in series the number of times indicated.

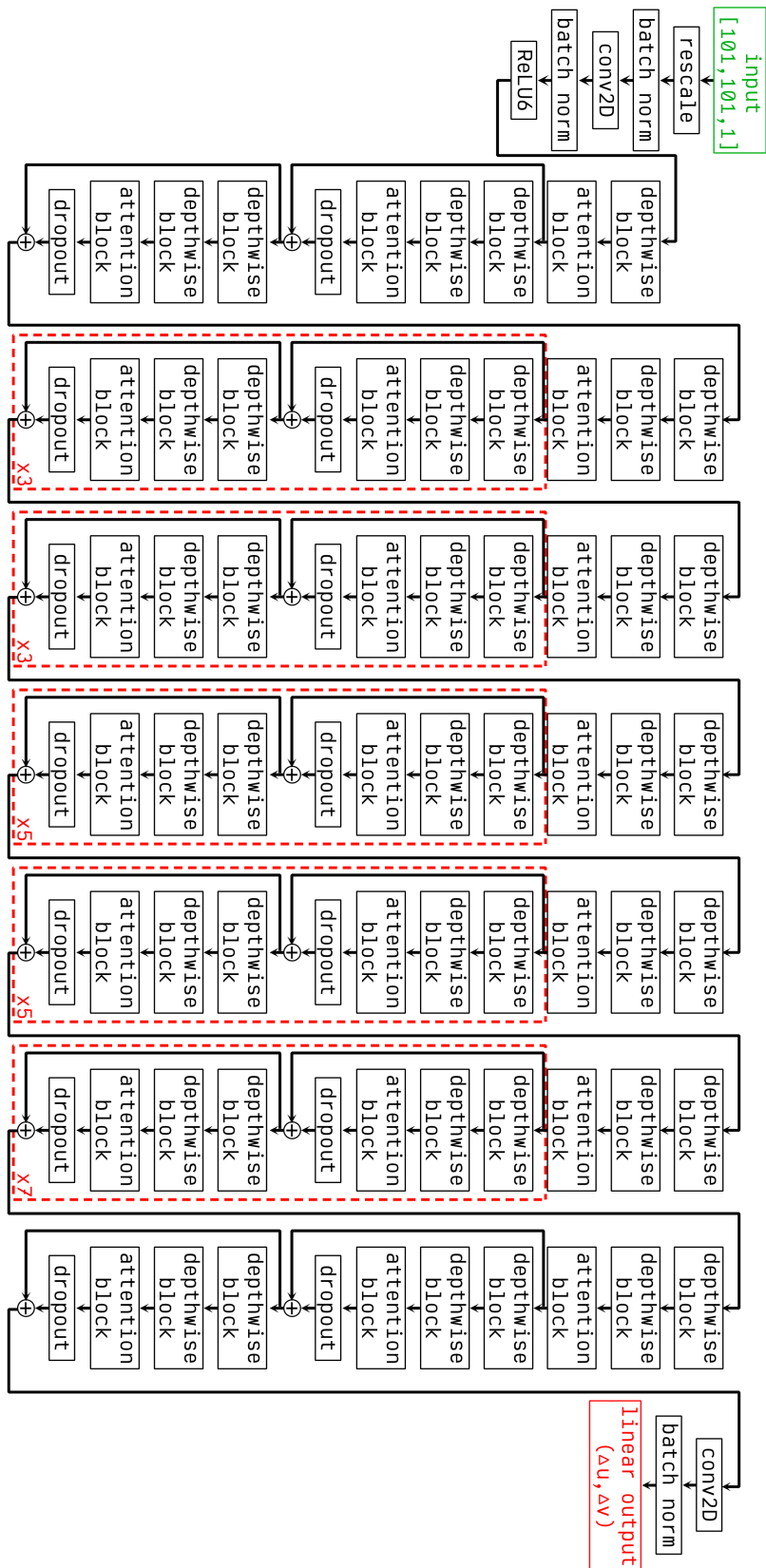
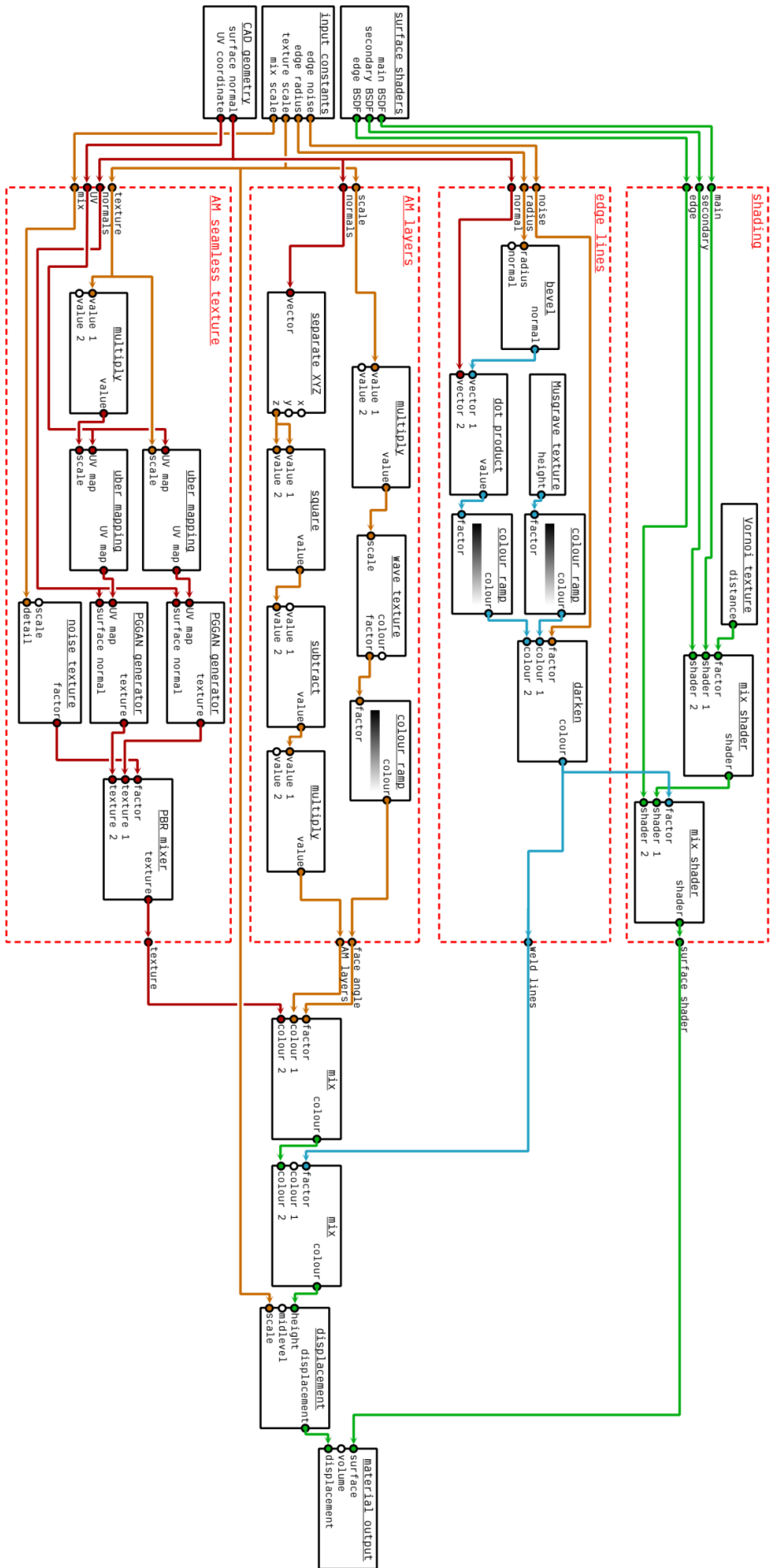


Figure E.2: Full EfficientNetB5 model

Appendix F

GAN based AM material shader.

Shown in this Appendix is the full Blender material shader developed in chapter 7 in Section 7.6. All the nodes used are standard to the Blender shader node editor with the exception of the UberMapping and PBR mixing nodes from Poliigon [228], and the PG-GAN generator nodes. The PG-GAN custom nodes generate a new texture tile from the trained generator model from the appropriate surface type as determined by the surface normal.



Appendix G

Functions

Collected below are a set of useful functions written in Python used within this thesis.

G.1 The closest point between two rays

```
1 # The closest point between two rays, where each ray is
2 # defined by an origin point and a direction vector
3 def closest_point(origin_0: Point,
4                   dir_0: Vector,
5                   origin_1: Point,
6                   dir_1: Vector) → Point:
7     perp: Vector = dir_1.cross(dir_0)
8                     .normalized()
9     rhs: Point = origin_1 - origin_0
10    lhs: ArrayLike = np.array([dir_0, -dir_1, perp]).T
11    [t1: float, t2: float, t3: float] = np.linalg.solve(rhs, lhs)
12    p1: Point = origin_0 + (t1 * dir_0)
13    dist: float = t3 / perp.magnitude
14    closest: Point = p1 + (dist / 2) * perp
15    return closest
```

G.2 Custom 'improvement' metric

```
1 # Custom training metric, the mean difference in Euclidean
2 # distance from the OpenCV estimate to the true centre,
3 # and the model prediction to the true centre
4 class MeanImprovementOverOpenCV(keras.metrics.Metric):
5     # Inherits from the keras Metric base class
6     def __init__(self, name = 'improvement_over_opencv', **kwargs):
7         super(MeanImprovementOverOpenCV, self).__init__(name = name, **kwargs)
8         self.total = self.add_weight(name = 'total', initializer = 'zeros')
9         self.count = self.add_weight(name = 'count', initializer = 'zeros')
10
11     def update_state(self, y_true, y_pred, sample_weight = None):
12         # Method called after each batch, updates internal state
13         y_true = tf.cast(y_true, tf.float32)
14         y_pred = tf.cast(y_pred, tf.float32)
15         dist_opencv = tf.norm(y_true)
16         dist_pred = tf.norm(tf.math.subtract(y_true, y_pred))
17         improv = tf.math.subtract(dist_opencv, dist_pred)
18         self.total.assign_add(improv)
19         self.count.assign_add(tf.constant(1, dtype = tf.float32))
20
21     def result(self):
22         # Method called to return scalar value from current internal state
23         return tf.math.divide(self.total, self.count)
```