euspen's 20th International Conference & Exhibition, Geneva, CH, June 2020
www.euspen.eu

Pose estimation from a monocular image for automated photogrammetry

Joe Eastwood[1], Danny Sims-Waterhouse[2], Samanta Piano[1], Ralph Wier[2] and Richard Leach[1,2]

[1]Manufacturing Metrology Team, Faculty of Engineering, University of Nottingham, UK
[2]Taraz Metrology, UK

Joe.Eastwood@nottingham.ac.uk

Abstract
Close-range photogrammetry is an optical metrology technique for measuring the three-dimensional form of an object. Photogrammetry has many advantages over competing techniques such as structured light projection – the primary being the relative inexpense of the equipment required. However, photogrammetry is currently slow and user dependant at many stages. In order to facilitate automation of the measurement process, the spatial relationship between the camera and the measurand must be known. We present a method of establishing this relationship from a single initial image via a convolutional neural network (CNN). The CNN is trained on synthetic images of an object which are generated from the computer aided design (CAD) data. A simulated texture is applied to the CAD data which is representative of the observed surface created by the manufacturing process and material. By using this representative texture combined with accurately characterised camera intrinsic parameters, near photo-realistic images can be generated. By using the CAD data to generate the training data, training can be conducted in parallel with manufacture removing any additional time penalty that would be otherwise incurred. This method also removes any need for manual labelling of training data. We test this approach on a range of different CNN architectures and select the best performing network based on the model loss and prediction error.

Photogrammetry, machine learning, automation, smart measurement, pose estimation, TensorFlow

1. Introduction

Form metrology is an important aspect of part verification after the completion of a manufacturing process. With the advent of additive manufacture, complex freeform surfaces are becoming more common within industry. These surfaces are often impractical to measure with a traditional tactile coordinate measurement machine; but optical approaches can offer a way forward [1]. Close-range photogrammetry is an optical form measurement process based on detecting features within individual photographic images of an object, then finding correspondences between the images [2] to produce a dense point cloud – the measurement pipeline is shown in Figure 1.

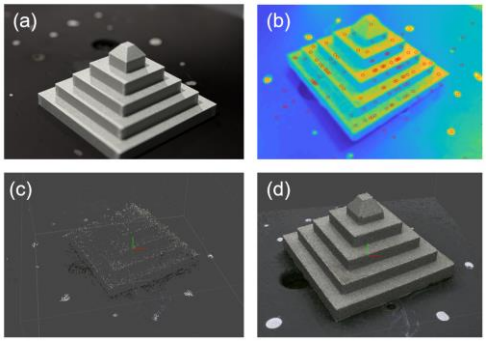


Figure 1. Close-range photogrammetry pipeline. a) Capture images, b) detect features, c) sparse reconstruction, d) dense reconstruction.

Photogrammetry has many advantages over competing methods [3], such as fringe projection, particularly the relative inexpense of the equipment required. However, photogrammetry is often slow and operator dependant. In this paper, we show how to improve the measurement time while removing any user input from the measurement pipeline. In a first step, it is required that the spatial relationship between the camera and the measurement object is obtained. For the sake of simplicity, it would be desirable to detect this relationship from a single initial image. The most successful previous approaches to six degree of freedom pose estimation of objects using CAD data include template matching [4] and DeepIm, an iterative machine learning approach [5]. Both these approaches require a high level of computational power at the time of the prediction, slowing down the pose estimation process. In an initial feasibility study presented elsewhere [6], we have shown that using CAD data of an object to train a convolutional neural network (CNN) to predict this spatial relationship directly can be an effective approach. This has the benefit of 'pre-loading' the computational effort during the training period, allowing rapid predictions to be made, once the model is fully trained. As our application is constrained to a known instrument setup, this allows training of the network on a dataset of more specific images than are found in the large open access datasets. Using this application specific dataset minimises the variation in the possible input images, thus allowing direct regression of the six degrees of freedom (x, y, z, pitch, roll ,yaw) to be an effective approach. Building on this previous work, in this paper we present an effort to improve the pose estimation approach by testing a range of different CNN architectures on four example measurement artefacts. We tested ten initial network configurations before narrowing in on the three best performing options of the original ten. The performace of these three best performing networks are then compared to each other in more detail by comparing the validation loss and error values in the context of other factors, such as training time on a large dataset.

2. Training data

Machine learning approaches require large datasets of training images labelled with known ground-truth values of the variables of interest. Creating a dataset of the required size to train a CNN effectively would be practically impossible to generate from real photographic images. Instead an accurate simulation of the photogrammetry equipment has been developed to produce synthetic images as close to images produced by the real setup as possible – this is shown in Figure 2.

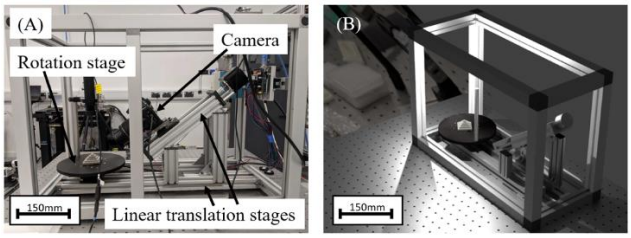


Figure 2. a) Actual experimental setup. b) Simulated experimental setup within Blender Cycles rendering engine.

By placing the CAD file of the object and a simulated camera at random possible positions within the simulated photogrammetry system, the desired dataset can be generated. Simulating the performace of the real camera is important to achieving meaningful predictions. Accurately characterised camera parameters were determined for the real system and consequently used within the simulation. To maintain the levels of photorealism required, a representative surface texture for the manufacturing process used to fabricate the artefact was also generated and applied to the CAD file. This representative texture was generated by detecting the most prominent spatial frequencies from an initial photogrammetry measurement of a surface produced using the same process as the test artefacts. In this case, the manufacturing process and material were laser powder bed fusion (PBF) and titanium-64 (Ti-64). A comparison image showing the real texture and the simulated texture is given in Figure 3.

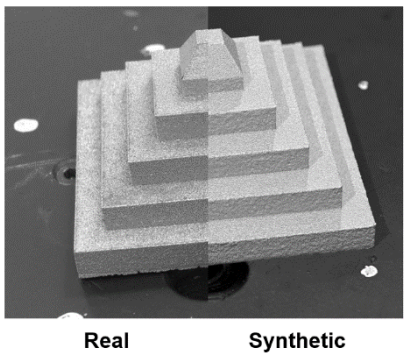


Figure 3. Comparison of real and synthetic image of the pyramid measurement artefact.

Once generated, the simulated texture can then be applied to any CAD file to create an accurate rendering of what the object would look like if manufactured from Ti-64 using PBF.

For this work 2 250 images of 4 artefacts (shown in Figure 4) were generated leading to a dataset of 10 000 images. 90% of the data is used for training with a further 10% used for cross-validation. It is important that the dataset is representative of the entire space of possible input images. Therefore, in each input image, the camera is in a random position on the linear stage, the artefact is positioned randomly on the rotation stage, and the rotation stage is placed with a random rotation. As these images are synthetic, the ground truth labels are known explicitly and are recorded for use during training.

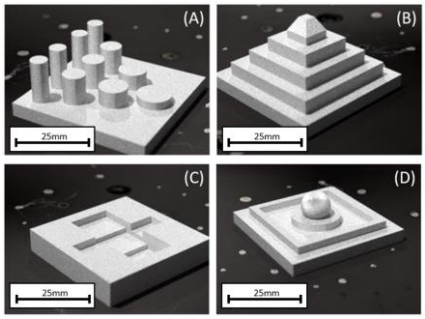


Figure 4. The four test artefact designs.

3. Proposed network architectures

In the original study, a very simple CNN was used to demonstrate the feasibility of the approach. This network consisted of two simple 2D convolutional layers followed by a single densely connected layer. In an effort to improve prediction accuracy and to reduce generalisation error, a range of different architectures, learning rates, batch sizes and optimisation algorithms (hereafter collectively referred to as hyper-parameters) were tested. All the architectures tested were variations of the base architecture shown in Figure 5.

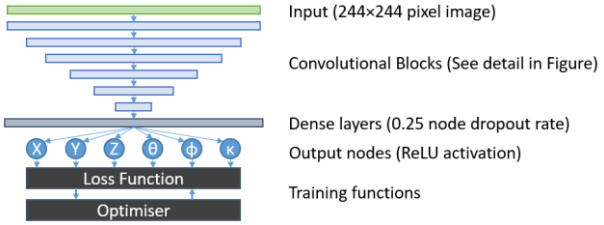


Figure 5. Base CNN architecture

Each of the convolutional blocks shown in Figure 5 was constructed from one of the two layouts shown in Figure 6. Figure 6a) shows a simple convolutional block and Figure 6b) shows a residual convolutional block.

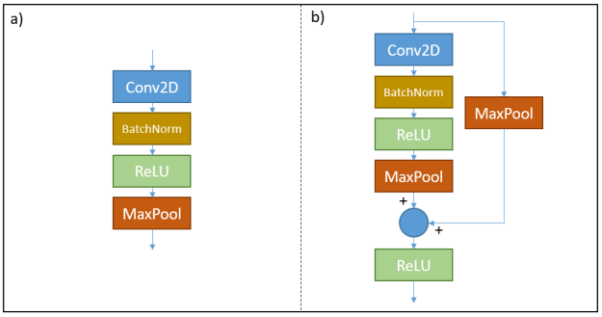


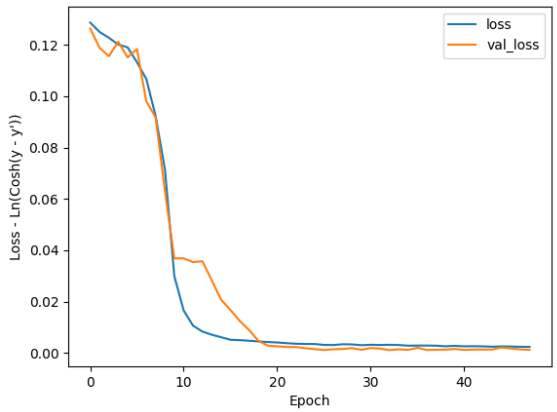
Figure 6. Convolutional blocks used. a) Basic convolutional block – set of convolutional filters (Conv2D), batch normalisation (BatchNorm), activation function - rectified linear operator (ReLU), and downsampling by a factor of two using maximum pooling. b) Same structure as a) but with an added skip connection which downsamples the input, combines the input with the feature maps, and passes the resultant values through a set of ReLU activations.

In order to determine the best performing network, the model loss and prediction mean absolute error (MAE) are calculated for a range of possible architectures and hyper-parameters. Initially, these networks are evaluated on the pyramid dataset, then the best performing configurations are tested on the entire dataset of all four artefacts and the loss and error values compared. An early stopping criteria was used in all cases, to cease training when no decrease has been seen in the validation loss for ten epochs.

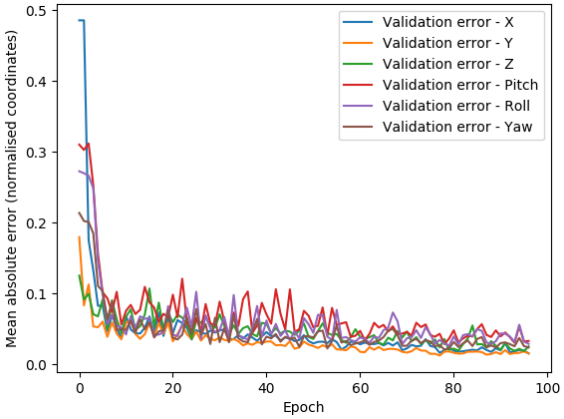
Network	Dense layers	Learning rate	Optimiser	Convolutional block	Batch size	Model loss	Prediction MAE
(-)	(Number of layers)	(AU · 10^{-4})	(-)	(block type)	(number of images)	(AU)	(normalised coordinates)
1	1	1	Adam	Simple	64	0.1373	0.48
2	**2**	**1**	**Adam**	**Simple**	**64**	**0.0024**	**0.039**
3	**1**	**0.1**	**Adam**	**Simple**	**64**	**0.0022**	**0.028**
4	**2**	**0.1**	**Adam**	**Simple**	**64**	**0.0051**	**0.047**
5	1	1	Adam	Residual	64	0.1315	0.50
6	2	1	Adam	Residual	64	0.1301	0.48
7	2	1	SGD	Simple	64	0.1228	0.41
8	2	0.1	SGD	Simple	64	0.1559	0.36
9	1	1	SGD	Residual	64	0.0772	0.24
10	2	1	SGD	Simple	32	0.0669	0.23

Table 1. Results of the initial testing of ten networks on the dataset of 2250 synthetic images of the Pyramid artefact. The best performing networks are shown in bold.

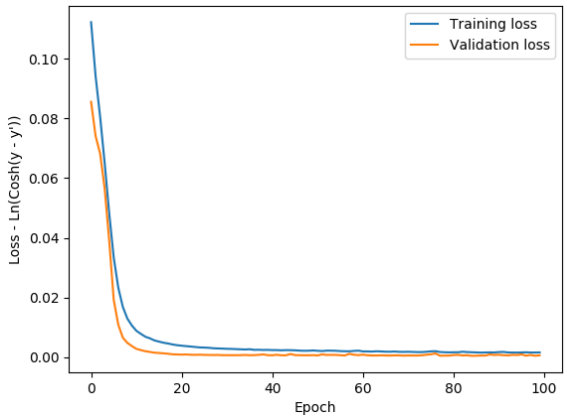
a) Network 2 – training and validation losses



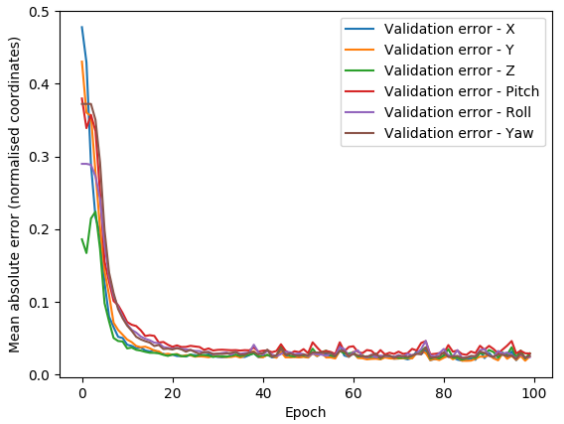
b) Network 2 – MAE in each degree of freedom



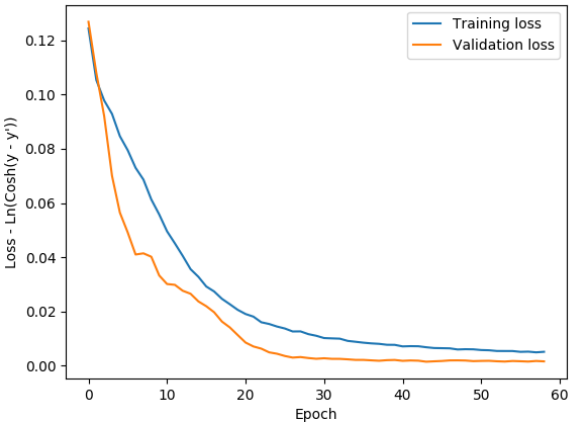
c) Network 3 – training and validation losses



d) Network 3 – MAE in each degree of freedom



e) Network 4 – training and validation losses



f) Network 4 – MAE in each degree of freedom

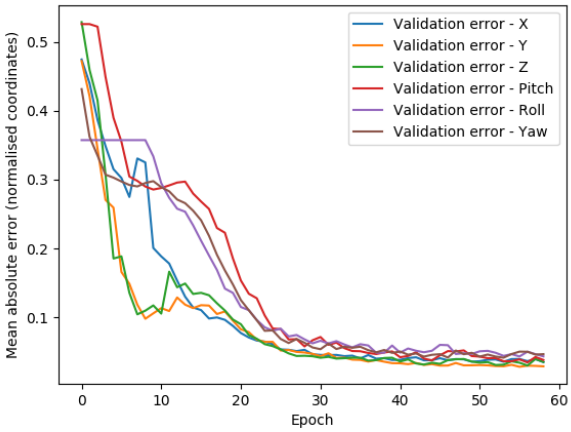


Figure 7. For each network: LogCosh loss function calculated on the training and validation datasets after each training epoch, MAE in predictions on validation data for each degree of freedom.

4. Results

Table 1 shows the ten initial networks tested. The parameters varied were: number of densely connected layers, training batch size, optimisation learning rate, optimiser type (Adam [7] or stochastic gradient descent (SGD) [8]), and convolutional block type (as shown in Figure 6). In all cases, a LogCosh loss function was used in the optimisation process. Error in the predictions was calculated by taking the mean absolute difference between the prediction and the known ground truth value.

It can clearly be seen that three of the networks (networks 2, 3 and 4) lead to a decrease in the loss value by one to two orders of magnitude over the competing networks. These highlighted networks also showed similar improvements on the mean absolute prediction error. The three best performing networks all share two main similarities: they use simple convolutional blocks and the Adam optimiser. Note that the use of residual blocks has been previously shown to decrease error in deep networks [9]. It is likely the case that the six convolutional blocks used here are not deep enough to respond to the benefit of this approach, so the use of residual blocks may decrease performance.

Networks 2, 3 and 4 were given the entire dataset of 10 000 images of all four artefacts. The model loss and validation errors over the training periods are shown in Figure 7. As can be seen, all three networks perform similarly with a few minor but important distinctions. Firstly. the variance in the prediction error during the final training epochs for network 3 is far lower than for the other two approaches. Although the mean of all three errors has converged well in networks 2 and 4 (as can be seen by the smooth loss curves), the variance in the individual components of the error are more varied when compared to network 3. This increase in prediction variance is likely caused by overfitting due to the extra densely connected layers. Furthermore, because of the lower learning rate used in networks 3 and 4, the training time per epoch is increased. For example, the time to compute the first epoch was 1290 s for network 2, 1624 s for network 3, and 2190 s for network 4. Network 3 also provided the lowest prediction error in the initial study of 0.03 compared to 0.04 and 0.05 from networks 4 and 5 respectively. For these reasons, network 3 was selected as the best performing network. This is due to the more consistant prediction errors than both of the other networks and the more efficient training schedule than network 4. Furthermore, when comparing the performance of network 3 to the network used in the initial study [6], we see an order of magnitude decrease in the calculated loss values, from 0.01 to 0.002. Figure 8 shows two example predictions produced by network 3 – the input image is shown with the prediction overlaid.

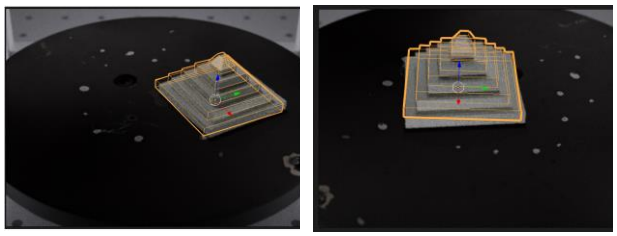


Figure 8. Example predictions given by network 3 – prediction outlined and center of mass location and rotation shown.

5. Conclusions

Building on previously completed work, which showed the feasibility of direct regression of six degrees of freedom for object pose estimation, a set of ten possible networks were proposed. Through testing on an initial smaller dataset, the best three solutions were shown to have simple convolutional layers over residual blocks and use the Adam optimiser over SGD. These three networks were: network 2, which uses two densely connected layers; network 3, which has a lower learning rate than network 2 by a factor of ten; and network 4, which has the same lower learning rate as used in network 2 and an extra dense layer.

These three networks were then tested on a larger set of data consisting of the four test artefacts from Figure 4. It was shown that, while all the networks produced comparatively low model losses of around 0.002, network 3 produced consistently lower errors than both networks 2 and 4, while also training in less time per epoch than network 4. Network 3 was, therefore, selected as the best performing CNN strategy overall. This approach was shown to provide an order of magnitude improvement in performace over the original CNN design, based on the calculated loss values.

6. Future work

Until this point the best performing network has been determined for synthetic data; further testing on actual photogrammetric images is required. There is a chance that the network which performs the best on synthetic data may not be the network which generalises with the least increase in error onto photographic images.

Once complete, this pose estimation procedure will be incorporated into the automation algorithms currently in development within our team.

References

[1] Leach R K, Bourell D, Carmignato S, Donmez M, Senin N and Dewulf W 2019 Geometrical metrology for metal additive manufacturing *Ann. CIRP 68* 677-700

[2] Luhmann T, Robson S, Kyle S and Harley I 2007 *Close range photogrammetry* (Wiley)

[3] Luhmann T 2010 Close range photogrammetry for industrial applications *ISPRS Journal of Photogrammetry and Remote Sensing 65(6)* 558-569

[4] Tejani A, Tang D, Kouskouridas R and Kim T K 2014 Latent-class hough forests for 3D object detection and pose estimation *Proc. ECCV* 462-477

[5] Li Y, Wang G, Ji X, Xiang Y and Fox D 2018 Deepim: Deep iterative matching for 6d pose estimation *Proc. ECCV* 683-698

[6] Eastwood J, Sims-Waterhouse D, Piano S, Weir R, and Leach R K 2019 Autonomous close-range photogrammetry using machine learning *Proc. ISMTII2019*

[7] Kingma D P and Ba J 2015 Adam: A method for stochastic optimization *Proc. ICLR*

[8] Zhang T 2004 Solving large scale linear prediction problems using stochastic gradient descent algorithms *Proc. ICML* 116

[9] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. IEEE CVPR* 770-778