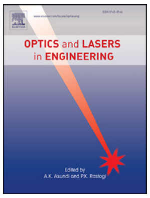
Optics and Lasers in Engineering 114 (2019) 7–14

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng



Rapid tracking of extrinsic projector parameters in fringe projection using machine learning

Petros Stavroulakis[a,*], Shuxiao Chen[a], Clement Delorme[b], Patrick Bointon[a], Georgios Tzimiropoulos[c], Richard Leach[a]

[a] Manufacturing Metrology Team, Faculty of Engineering, University of Nottingham, Nottingham, UK
[b] Ecole Nationale d'Ingénieurs de Saint Etienne, Saint Etienne, France
[c] Department of Computer Science, University of Nottingham, Nottingham, UK

ABSTRACT

In this work, we propose to enable the angular re-orientation of a projector within a fringe projection system in real-time without the need for re-calibrating the system. The estimation of the extrinsic orientation parameters of the projector is performed using a convolutional neural network and images acquired from the camera in the setup. The convolutional neural network was trained to classify the azimuth and elevation angles of the projector approximated by a point source through shadow images of the measured object. The images used to train the neural network were generated through the use of CAD rendering, by simulating the illumination of the object model from different directions and then rendering an image of its shadow. The accuracy to which the azimuth and elevation angles are estimated is within 1 classification bin, where 1 bin is designated as a ±10° patch of the illumination dome. To evaluate use of the proposed system in fringe projection, a pyramidal additively manufactured object was measured. The point clouds generated using the proposed method were compared to those obtained by an established fringe projection calibration method. The maximum dimensional error in the point cloud generated when using the convolutional network as compared to the established calibration method for the object measured was found to be 1.05 mm on average.

1. Introduction

Estimating scene illumination from a single image is useful in many computer vision applications, such as shape-from-shading [1–3]. Coarsely estimating the light source direction relaxes the reliance of shading algorithms [4] on exact a priori information regarding the light source configuration and the surface reflectance properties [1–3]. Another application, which requires photometric registration of scenes from images, is augmented reality [5,6]. In augmented reality applications, a virtual object is overlaid onto a real scene, and, in order to make the object blend with the scene realistically, the illumination of the scene needs to be estimated and applied onto the virtual object whilst it is being rendered. The system should, therefore, be able to estimate in real-time, both the photometric and geometric characteristics of the virtual object in the scene [7].

Because cameras use the same theoretical framework for calibration as projectors, it is worth mentioning that machine learning and in particular neural networks have been used to calibrate camera calibration parameters (both intrinsic and extrinsic) either by directly training a network to perform the numerical parameter extraction [8] or indirectly by using the a neural network to identify checkerboard cross-sections commonly used in camera calibration techniques to identify the correct correspondences between images [9]. Other machine learning techniques used for photogrammetry bundle the camera calibration and 2D to 3D pixel to world coordinate mapping into one procedure [10,11] with the disadvantage of the model being useful only for the particular camera configuration after it has been trained. Even though the calibration model between camera and projector are theoretically similar, these techniques cannot be directly applied to projector calibration as the projector itself cannot capture images. The procedure proposed here for extrinsic projector parameter calibration in a fringe projection system infers the projector location and orientation in the particular setup by use of a camera which records the shadow image cast by the object onto the measurement surface.

In this work, therefore, near real-time coarse estimation of the light source orientation using shadow cues is shown for additively manufactured (AM) objects and realised through machine learning. The algorithm proposed can run on cost-effective hardware and be used for objects without specular reflection cues, reference objects of known geometry or light probes. The proposed CNN algorithm is implemented in a fringe projection system and used to continuously estimate the position and orientation difference between a projector and camera during measurement process, thus allowing for the camera and the light source to become completely decoupled during the measurement procedure. Decoupling the camera and the projector has been shown to have benefits when measuring objects with high aspect ratio occlusions [12]. Without the ability to continuously estimate the position between a camera and a projector in a decoupled fringe projection system, the system would need to be re-calibrated after each change in relative position

[*] Corresponding author.
E-mail address: petros.stavroulakis@nottingham.ac.uk (P. Stavroulakis).

https://doi.org/10.1016/j.optlaseng.2018.08.018
Received 23 April 2018; Received in revised form 27 August 2018; Accepted 28 August 2018

between the camera and projector using one of the established techniques [13,14]. Using the established techniques usually involves stopping the measurement, removing the measurement object and inserting a calibration plane textured with a circular or checkerboard pattern, and the acquisition of multiple test measurements, which would make the measurement procedure impractical and time-consuming. Repetitive pre-calibration is currently required in semi-decoupled fringe projection systems (for example, SIDIO XR by NUB3D), where the position and orientation between the camera and projector are allowed to vary in a collection of pre-set positions, in order to allow for multiple scan volumes and scan resolutions. With the ability for continuous position estimation between the camera and projector, these semi-decoupled systems can become fully decoupled and allow for changes in configuration during the measurement without the need to pause for recalibration.

2. Background

One of the earliest methods of performing photometric estimation is described by Pentland [15], who statistically computed the illumination direction of the environment using a maximum-likelihood estimator. Improving upon Pentland's solution, the method of Chojnacki et al. [16] provides better performance at higher resolutions and with higher accuracy. For light source estimation from images in augmented reality applications, three general methods can be identified [6], namely: (1) using a light probe in the scene in the field of view of the camera, (2) detecting the environmental illumination directly using a fish eye lens, and (3) using shadows cast by known objects. Out of the three aforementioned methods, the most efficient one in fringe projection applications is shadow cue estimation [6]. The methods shown elsewhere [6,17,18], however, either require an object of known geometry to perform the estimation or are too slow to run in real-time.

Recent approaches, initially thought impractical for real-time light source estimation, such as methods using specular illumination cues [19] and light probes in the scene [5], have been shown to work in real-time with modern hardware. Using specular reflection cues would not be efficient for AM objects as their surfaces are optically rough [20–22], thus reflect light diffusely and do not provide obvious light source cues for calculating the scene illumination. Light probes are also a hindrance in general because they either require a separate camera pointing at the light probe, or a specific pixel real-estate on the measurement camera to monitor the probe [5]. Using some pixels of the measurement camera for this purpose reduces the number of pixels available to perform the measurement and, therefore, reduces the system's resolution.

In this work, we propose a method which avoids the need to use reflectance cues on the object and the need for a light probe, by training a convolutional neural network (CNN) [23] to recognise the position of the light source from the shadowed version of the measured object. We also evaluate and discuss the method's accuracy and applicability.

3. Projector calibration

To calibrate the camera and the projector in a fringe projection system, the correspondence between both the projector's and camera's pixel arrays and the 3D projected points in space needs to be calculated. Eq. (1) is used to describe the relationship between a pixel array and its corresponding 3D world coordinates using the pinhole model, which does not consider optical distortion:

$$[\boldsymbol{x}\ \ \boldsymbol{y}\ \ 1]\boldsymbol{w} = [\boldsymbol{X}\ \ \boldsymbol{Y}\ \ \boldsymbol{Z}\ \ 1]\boldsymbol{P} \tag{1}$$

where w is the scale factor, x and y are the coordinates of the image along the horizontal and vertical directions respectively, X, Y and Z are the spatial coordinates of the corresponding pixel in the world coordinate system and P is known as the projection matrix.

The projection matrix P contains the intrinsic and extrinsic calibration parameters of the system which are determined during the calibration procedure. The intrinsic parameters refer to the optical system used to project the image onto the pixel array (optical centre, focal distance, pixel size, etc.), and the extrinsic parameters relate to the position and orientation of the optical system with respect to the world coordinate system.

To account for optical distortion of the lens, in a similar manner to that used for camera calibration, an additional non-linear radial and tangential calibration step is required to enhance the accuracy of the pixel locations. In projectors and cameras with poor optical lenses and alignment, this step is important as the distortions can be relatively large. The equations which are used to describe the non-linear distortions in the projector (Eqs. (2)–(6) from [13]) are the following:

$$\tilde{\boldsymbol{u}} = \begin{pmatrix} \tilde{\boldsymbol{u}}_x \\ \tilde{\boldsymbol{u}}_y \end{pmatrix},\ \boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_x \\ \boldsymbol{u}_y \end{pmatrix}. \tag{2}$$

$$\boldsymbol{r}^2 = \tilde{\boldsymbol{u}}_x^2 + \tilde{\boldsymbol{u}}_y^2. \tag{3}$$

$$\boldsymbol{L}(\tilde{\boldsymbol{u}}) = \begin{bmatrix} \tilde{\boldsymbol{u}} \cdot \left(1 + \boldsymbol{k}_1\boldsymbol{r}^2 + \boldsymbol{k}_2\boldsymbol{r}^4\right) + \Delta_t(\tilde{\boldsymbol{u}}) \\ 1 \end{bmatrix} \tag{4}$$

$$\Delta_t(\tilde{\boldsymbol{u}}) = \begin{bmatrix} 2\boldsymbol{k}_3\tilde{\boldsymbol{u}}_x\tilde{\boldsymbol{u}}_y + \boldsymbol{k}_4\left(\boldsymbol{r}^2 + 2\tilde{\boldsymbol{u}}_x^2\right) \\ \boldsymbol{k}_3\left(\boldsymbol{r}^2 + 2\tilde{\boldsymbol{u}}_y^2\right) + 2\boldsymbol{k}_4\tilde{\boldsymbol{u}}_x\tilde{\boldsymbol{u}}_y \end{bmatrix} \tag{5}$$

$$\boldsymbol{u} = \boldsymbol{K}_c \cdot \boldsymbol{L}(\tilde{\boldsymbol{u}}) \tag{6}$$

where k_1, k_2 are the radial distortion calibration coefficients, $k_{3,}$ k_4 are the tangential distortion calibration coefficients, $\tilde{u}_x$ and $\tilde{u}_y$ are the undistorted x, y pixel coordinates of point $\tilde{u}$, K_c is the intrinsic calibration coefficient matrix, u_x, u_y are the distorted image x, y pixel coordinates of point $\tilde{u}$ (which define the observed position u on the projected image), r is the radial distance of the undistorted point $\tilde{u}$, $L(\tilde{u})$ is the overall effect of the radial and tangential error on the x, y pixel positions and $\Delta_t(\tilde{u})$ is the effect of the tangential error on the x, y pixel positions.

Out of all of the calibration parameters required for projector calibration described above, the only ones that vary when a projector is moved around physically are the extrinsic parameters which are included in matrix P (Eq. (1)). By initially calibrating the projector using one of the established techniques [13], we can ascertain the intrinsic and distortion-related parameters of the projector, which do not change during the measurement. To complete the calibration, whilst allowing for independent projector movement during the measurement procedure, we simply need to update the extrinsic parameters relating to the position and orientation of the projector in real time as the projector moves around the object.

4. Methodology

Machine learning is a subset of artificial intelligence which employs statistical techniques to iteratively 'learn' the relationship between a large number of known and labelled input-output data without explicit knowledge of the specific underlying function. One of the learning techniques available in machine learning is what is called an artificial neural network (ANN) or usually simply called neural network (NN). An NN models the way biological neural networks operate and excels at ascertaining non-linear input-output relationships in a statistical sense when trained on a large amount of data, and they are widely used for classification problems. A type of NN which is widely used in computer vision, because of its ability to work well with image inputs, is what is called a convolutional neural network (CNN). As previously mentioned, we will be using a CNN in this work in order to classify the projector's azimuth and elevation angle in real time from a collection of labelled shadow images trained on a specific object.

There are various methods which can be used to initially calibrate the projector and set the intrinsic and extrinsic parameters of the system [18,19]. In our case, we used the calibration procedure developed

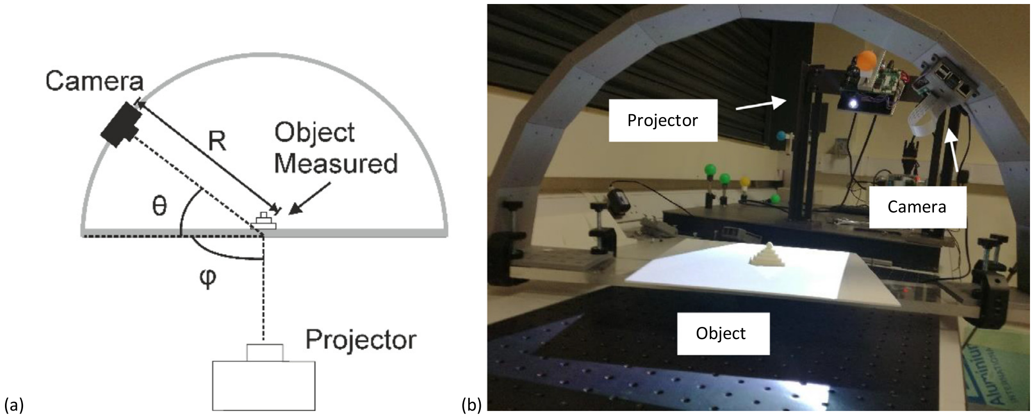


Fig. 1. (a) Schema of the setup denoting the azimuth (φ) and elevation (θ) angles for the camera in the setup shown on the photo on the right; (b) photograph of the setup.

by Moreno [13]. In particular, a checkerboard pattern in different orientations is used in combination with a projected binary pattern to both calibrate the camera and associate the corresponding pixels of the projector. The projector is then calibrated as an 'inverse camera'.

After the system is initially calibrated, as was discussed in the previous section, if the projector is moved, the calibration would normally need to be repeated. However, we propose to track the projector's azimuth and elevation angles, so that the calibration does not need to be repeated, by using a CNN which has been trained to classify the projector's azimuth and elevation by an input image of the shadow of an object. It is worth noting that inferring all extrinsic parameters relating to the position and orientation of a projector by simply tracking it's azimuth and elevation angles is only possible in systems where the projector's movement is constrained so that the distance of the projector from the object is either stationary or can be calculated in some way from the azimuth and elevation angles, and the projector's centre of projection is fixed to a particular point in space, which in our case is the centre of rotation of the object mount. To test the CNN, the relative azimuthal angular position of a projector is changed by rotating a rig holding the camera and the measured object, as shown in Fig. 1. To test the elevation angle, the projector was simultaneously tilted and moved on a vertical rail (Fig. 1b); in both cases the centre of projection was always centred on the centre of rotation of the rotation stage on which the object was mounted, and the position and orientation of the projector could, therefore, be calculated by simply tracking the azimuth and elevation projector angles. After each projector movement, a new image of the object is taken and sent to the CNN, which responds with the new projector azimuth and elevation position, thus making it possible to calculate the new projection matrix, and recalibrating the system on the fly.

To train our CNN classifier, we propose the following method:

(1) Use a rendered CAD model of the object to generate a complete library of images of the object illuminated at different elevations and azimuth (for more details see below in Section 4.1).
(2) Convert the images into binary to extract the shadow from the simulated images.
(3) Train a CNN to classify images in the illumination dome (for more details see below in Section 4.3).

Then, during the estimation operation cycle, the system:

(1) Acquires live images from the setup, segments and thresholds the image to extract the object shadow.
(2) Sends the image to the CNN classifier to extract the light source position.

One of the novelties of our approach is in the use of simulated data to train the CNN. As the CAD model was available in advance, we used it to render a large number of synthetically-generated training images with known illumination. However, training a model on synthetic data and then using it on real data typically results in poor performance (this is known as domain shift [24]). To this end, we propose to binarise the synthetic images prior to training the CNN, and to do the same for the real images during test time. The produced binary images look (visually) quite similar, thus greatly reducing the domain shift, and enabling robust estimation during test time.

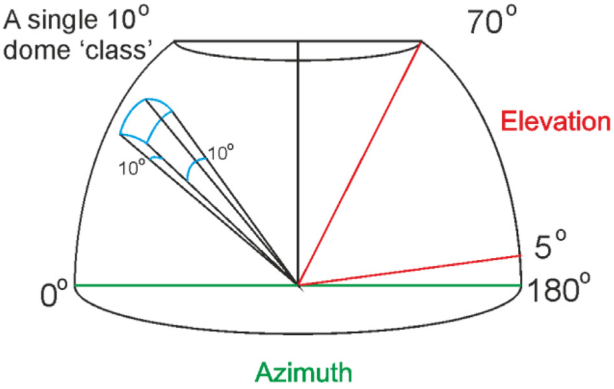


Fig. 2. Depiction of a single 'class' of the classification procedure, a 10°×10° patch of the illumination dome.

4.1. Rendering the images

The simulated illumination region was set such that there was an 180° azimuth angle illumination span (±90° in relation to the camera azimuth) and a 65° elevation span (5° to 70° from the stage). The region was segmented in 10° horizontal and vertical bins, each of which defines a different class with regards to the training of the CNN (Fig. 2). Each class contained 100 rendered samples within the same area, out of which 80 were used for training and 20 for validation.

The images were rendered by illuminating a CAD model of the object using a projector which was modelled as a point source (Fig. 3).

The generation of the images was performed in an open source rendering software package called Blender. The CAD model of the pyramid was loaded as an STL file which is readily supported by Blender (shown in Fig. A.1). Blender also supports automation through the use of Python scripts. A Python script was then written to automate the movement of the point light source (representing the projector); the rendering and extraction of the images which followed the pipeline is presented in Appendix A.

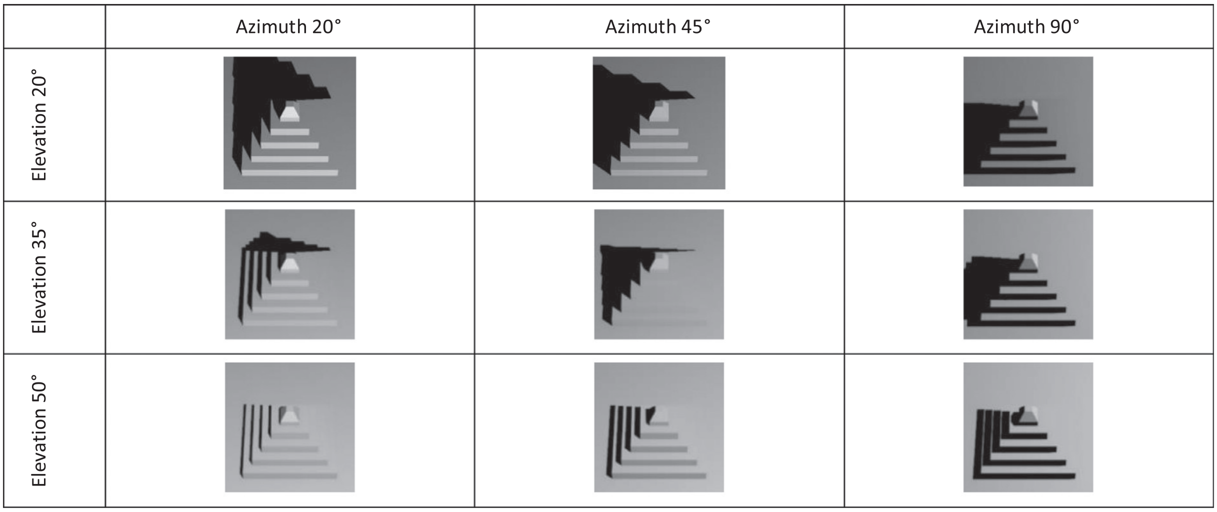


Fig. 3. Examples of simulated camera images created by changing the light source to nine different locations in azimuth and rotation.

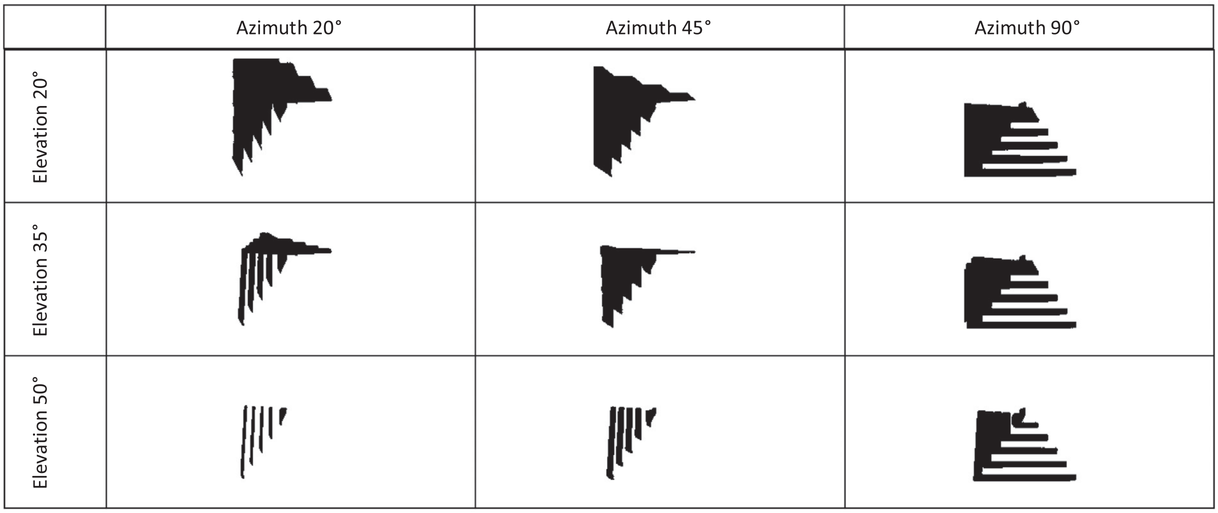


Fig. 4. Example of pre-processing the images in Fig. 3 to isolate the shadow.

4.2. Image pre-processing

For each simulated image, the object's shadow was extracted through image thresholding. The binary images with the shadows extracted from the images of Fig. 3 are shown in Fig. 4.

4.3. Network training

The illumination direction prediction network was trained for 6370 iterations and achieved an accuracy of greater than 90% in approximately 45 min. The hardware used for this purpose was an NVidia Titan X graphics card and the deep learning framework used was Caffe [23] network. It is worth noting that the training time was not the actual bottleneck of the process; the most time-consuming part of the procedure was the image generation process used to create the training dataset of the approximately 20,000 images (Section 4.1), which took around 6 h to complete. Appendix B lists the settings, network architecture and other details used in the CNN training process.

4.4. Acquire image from setup

The images were acquired by a Raspberry Pi camera and saved at intervals of 2 s. The automation of the process was enabled by using Linux scripting and was generally slow as the acquisition and processing of the images could not be carried out synchronously. Hence, a second script was run on the Linux server that hosted the CNN, which would perform illumination direction predictions every 5 s. Due to all the bottlenecks involved in later stages of processing the images, the estimation interval was approximately 5 s.

4.5. Send image to classifier to extract light source position

The images after thresholding were sent to the CNN classifier. As previously discussed, this whole process took around 5 s to complete as Linux automation scripts were used to pass command line options to programs and Python scripts used in the process.

5. Experiment

5.1. Testing accuracy of point source angle prediction

To test the accuracy of our system, an experimental rig was built, whereby the relative position of the light source is altered in relation to both the camera and object, in azimuth and elevation. The change in the azimuth rotation was verified using the markings on the rotation stage, whereas the elevation was verified using a digital inclinometer

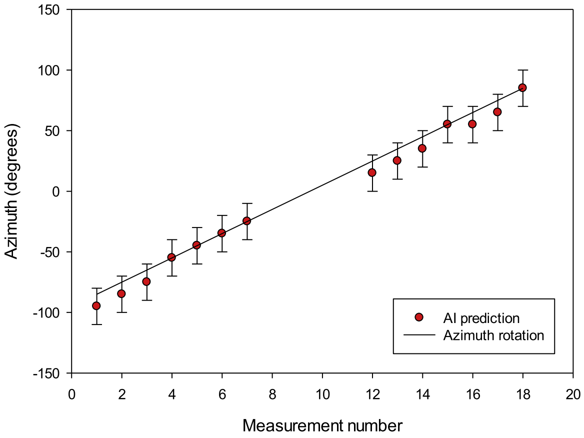


Fig. 5. Azimuth angle against CNN prediction as the relative light source is rotated in azimuth between −90° and 90° relative to the camera's azimuth. The error bars are set to ±15° from the reported class median point.

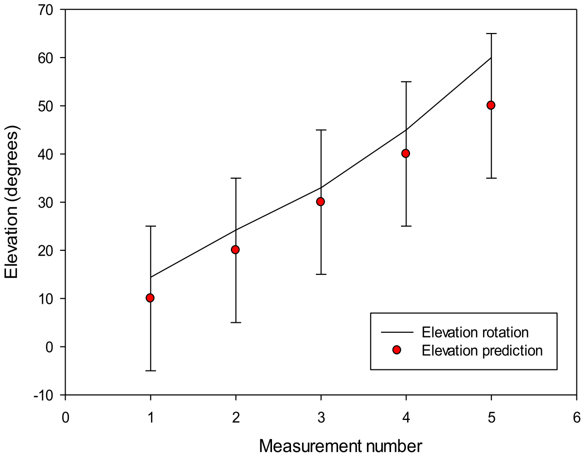


Fig. 6. Elevation angle against CNN prediction and the relative light source angle is tilted in elevation between 14° and 65° with respect to the measurement surface (and adjusting height as necessary). The error bars are set to ±15° from the reported class median point.

placed on the top surface of the projector. The projector light source was selected to be static and the camera and the measured object were moved simultaneously by mounting them on a rotation frame to create the relative effect of the light source changing position (Fig. 1).

Fig. 5 shows the results of testing the prediction accuracy of the CNN by rotating the camera-object frame in set intervals of 10° from −90° to +90° with respect to the camera azimuth. There is a gap between measurements eight and twelve because the structure on which the camera was mounted would completely shadow the object between ±20°, so no measurements could be taken in that range. The straight line depicts the true rotation measured by the markings on a rotation stage and the points show the predictions made by the CNN. It can be seen that the predictions follow the true rotation trend and the maximum deviation between the true and estimated prediction is ±15°.

A similar trend can be seen when altering the elevation angle on the sample. When increasing elevation angle direction, the predictions follow the trend. For the elevation results, a more accurate prediction is shown, with most of the results being within ±10° from the median prediction (Fig. 6).

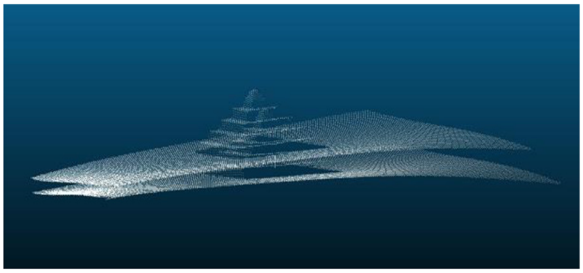


Fig. 7. Difference between point clouds generated via the Moreno [13] calibration routine and the CNN software estimation.

5.2. Ascertaining point cloud accuracy

Our measuring system, comprising a Raspberry Pi camera and a TI DLP® LightCrafter^{TM} 4500 projector, was initially calibrated using a method described elsewhere [13]. A pyramidal 3D object (50 mm × 50 mm × 30 mm) was measured from a single point of view using the structured light method described in [13] and a point cloud was generated. To investigate the effect of CNN projector–camera angle estimation error on the point cloud accuracy (±15°), a study into the point cloud error observed by incrementally injecting angular errors of 0° to 20° into the measurement process. When adding erroneous projector–camera angular values to the calibration file, the two point clouds are displaced when they are superimposed onto the same frame of reference (Fig. 7).

To measure the effect of the error in estimating the angle between the camera and projector on the actual accuracy of the measured structure, the two point clouds were first aligned using an iterative closest point (ICP) algorithm (CloudCompare [25]) and the average distance between the two point clouds was calculated (Fig. 8). Initially aligning the point clouds was required, as we are not concerned about the offset of the object in space but rather by the effect on the actual measurement accuracy of the measured object.

The measurement of the average point cloud distance to that of the calibrated point cloud was performed for error values in projector–camera angle between 0° to 20° in 1° increments. The resulting effect of projector–camera angle estimation error to the average distance of the generated point cloud to that of the reference point cloud taken after calibration is shown in Fig. 9.

6. Discussion

The illumination direction prediction method described can be used to infer all extrinsic parameters of the projector's position and orientation in a fringe projection application for setups where the distance of the projector to the object is stationary and the projector's centre of projection is fixed to a particular point in space, whilst rotated around the object (which in our case is the centre of rotation of the object mount). It is, therefore, not advised to use the technique described to calibrate the extrinsic parameters of a projector in a generic fringe projection measurement scenario where the distance of the projector to the object is not stationary during the measurement and the centre of projection is not fixed to a specific point in space.

The experimental results show that the accuracy of the illumination direction prediction achieved using the trained CNN network in azimuth and elevation is within ±1 bin of the angle reported by the rotation stage and inclinometer. The trend of predictions closely follows the real positioning of the light source. Each class trained is a 10° × 10° bin of the illumination dome (Fig. 2) hence, in angular terms, the maximum error is ±10° from the correct class area or ±15° from the class median. A study into the actual point cloud error incurred from this angle estimation error is of the order of 1 mm. The advantages of using CNN-estimated illumination directions to calibrate the measurement are that,

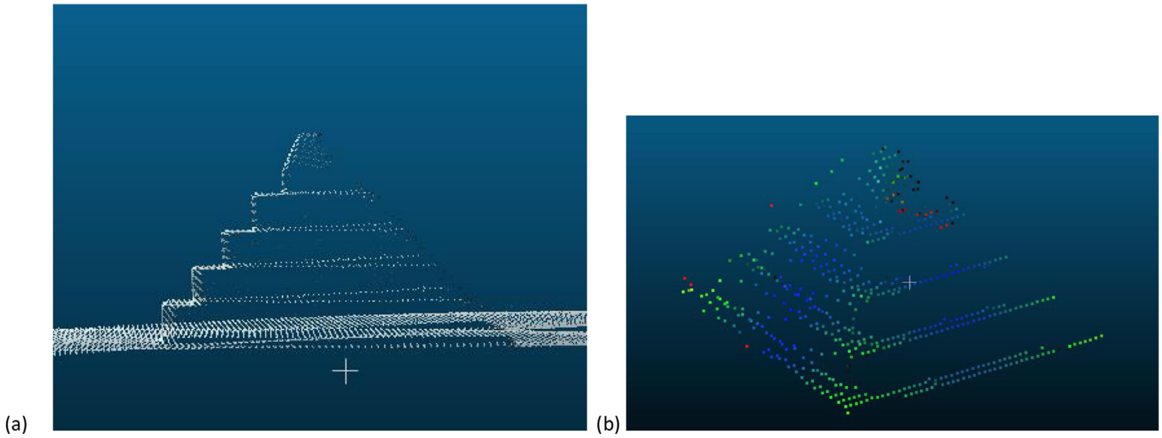


Fig. 8. Image of the two point clouds after aligning (a) and after measuring the distances between them on the points acquired from the pyramid structure (b).

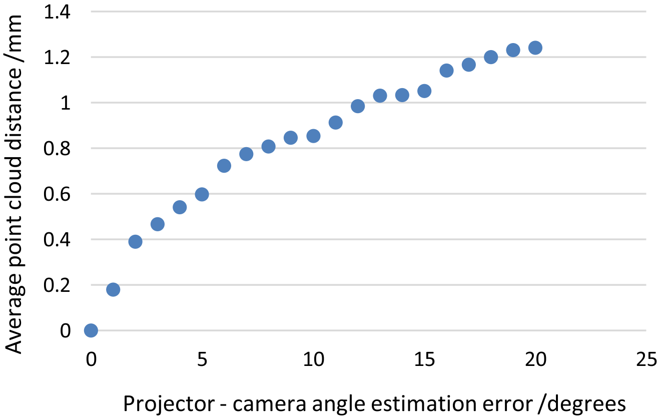


Fig. 9. Average point cloud distance against projector– camera angle estimation error. We can observe that for the maximum error achieved by the CNN network of 15°, the pyramid object measured has approximately 1 mm average point cloud error compared to that calibrated by the method presented elsewhere [13].

when CNN-specific hardware is used (for example, the Movidius USB stick [26]), the network can provide estimates of the true illumination direction very quickly (less than 1 s) without large computational overheads and, therefore, can operate on cost-effective hardware such as Raspberry Pi in near real time. Additionally, the quality of the angular estimation is irrelevant of the calibration procedure, unlike the classical method, whereby a checkerboard pattern must be placed in various orientations in the measurement volume and often does not complete successfully or completes with variable re-projection errors.

The disadvantages of the method are that it is sensitive to the threshold used in isolating the shadow for it to operate with high accuracy. This could be easily alleviated by training on binary images produced by different amounts of thresholding. Another weak point in our approach is that it is object-specific: because the model is trained on the shadow profile of a specific object, when applied on a different object the system has to be re-trained. As discussed previously, the estimation of the illuminated direction of a trained network can be fast, but the generation of the images required for network training takes approximately 6 h and the actual network training takes another thirty to 45 min. Finally, the number of classes used was relatively low, resulting in a class size of 10°×10°, which limits the resolution of the predicted illumination angle.

Future work will address the aforementioned drawbacks, such as the model's specificity to a particular object, which can be overcome by training the model including different objects and hence generating a generalised illumination direction prediction network. Training on multiple objects would also mean that the CAD data to train the object-specific network will no longer be required, so it would apply for objects for which no CAD data is available. Furthermore, the need for shadow isolation and image thresholding can be circumvented by directly training the network on realistic simulated images by using a model-based approximation [24]. CNN-specific hardware, such as the Movidius USB stick [26], will be used to reduce estimation intervals from 5 s to less than 1 s and hence allow faster real-time operation. Finally, increasing the number of classes by reducing the class size or switching to a regression model for predicting continuous values can lead to higher estimation precision and higher accuracy, and consequently lower the average error of the point clouds generated using this technique.

Funding

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/M008983/1].

Acknowledgements

We would like to thank Alexander Jackson-Crisp and Dr Matthias Hirsch for their assistance in machining and creating various parts of the setup and the measured objects.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.optlaseng.2018.08.018.

Appendix A

An example of the rendering environment in the software package Blender with the CAD model of the pyramid used is shown in Fig. A.1:

The pipeline followed in generating the simulated images in Blender is as follows:

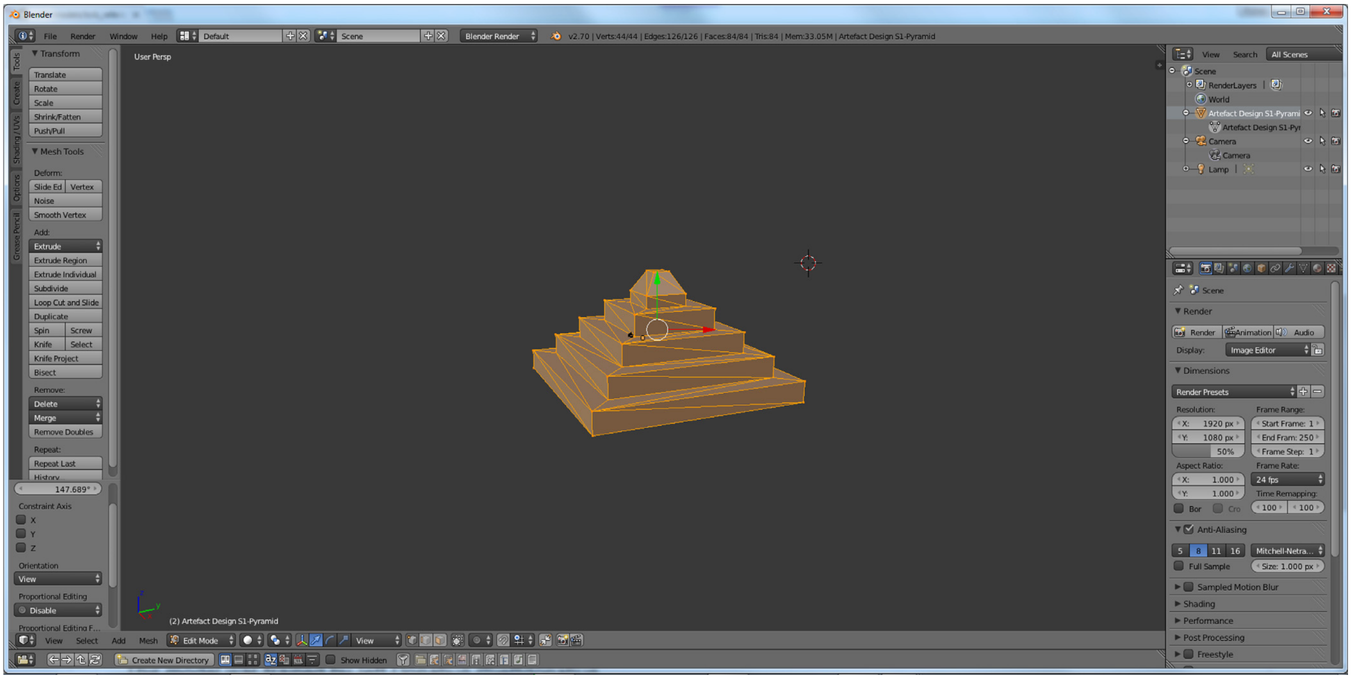
(1) Load pyramid STL file in Blender
(2) Rotate the object to sit flat on the *X*–*Y* plane, this is done because some STL files have different axes parameters or the object's native world coordinates are not similar to the *X*–*Y* axes in Blender.
(3) Scale the model to the correct size
(4) Create a camera with the following characteristics:
   a. Camera distance to object: 400 mm (this was the actual distance of the camera in our setup)
   b. Camera focal length: 200 mm
   c. Camera clip (start = 1 mm, end 5000 mm) bring all objects into view
   d. Camera sensor (width = 36 mm, height = 24 mm) modelled for full frame camera but this is not essential as image can be cropped later on.
(5) Create a point source with the following characteristics:
   a. Type = point source
   b. Distance = 700 mm – we found that the projector behaved essentially as a point source since moving it closer or further away to the object did not change its shadow projection significantly so this number is not very important but it has to be relatively far away as to not distort the shadow.
   c. Lamp energy = 5 (this number had to do with the source intensity and needs to be adjusted according to the source distance from the object in order to give a proper image intensity)
(6) Create a background thin background plane (for the shadow to be cast on) this was selected to be 50 times that of the object but can be any size as long as it fills the camera scene.
(7) The rendering specifications were set as follows:
   a. Output file type: JPEG
   b. Image size: 227 pixels (horizontal) × 227 pixels (vertical) – this size had to match up with the input of the CNN used in the training phase.

Appendix B

The network used for training the CNN on the model images was the popular CaffeNet network. A diagram of the network layers used is shown below (Fig. B.1):

The pipeline followed for training the CNN using the CaffeNet network was as follows:

(1) Compile all the binary images created in an (Lightning Memory-Mapped Database) LMDB database.
(2) Next calculate the mean image by averaging the average intensity value for each pixel across the training set.
(3) Subtract the mean image from all the images in the dataset to obtain a normalised dataset.
(4) Reset only layer fc8 (Fig. B.1) to employ transfer learning as the rest of the layers will not be trained from scratch but will begin optimisation from the default CaffeNet weights.
(5) Train the network on the GPU and monitor its training curve to observe training progress. The maximum allowed number of iterations were set to 10,000. That is, after 10,000 iterations the network training would stop no matter how much accuracy was achieved. The maximum number of iterations was set out of experience as the maximum number of iterations was increased incrementally whilst monitoring the training curve progress.
(6) After allowing the network to train for some time it was deemed that 6370 iterations were enough as the training accuracy rose to above 90%. If the training accuracy is too high there is danger that the network will fit 'too well' or overfit to the training data and therefore would not work with high prediction accuracy when presented with real data which would be inevitably a bit different to the training data.



v2.70 | Verts:44/44 | Edges:126/126 | Faces:84/84 | Tris:84 | Mem:33.05M | Artefact Design S1-Pyramid

147.689°

Constraint Axis
X
Y
Z

Orientation: View

Proportional Editing: Disable

(2) Artefact Design S1-Pyramid

Display: Image Editor

Resolution: X: 1920 px, Y: 1080 px, 50%

Frame Range: Start Frame: 1, End Fram: 250, Frame Step: 1

Aspect Ratio: X: 1.000, Y: 1.000

Frame Rate: 24 fps

Time Remapping: 100, 100

Anti-Aliasing: 5 8 11 16; Size: 1.000 px

Fig. A.1. Blender software rendering environment showing pyramid CAD structure loaded from and STL file.

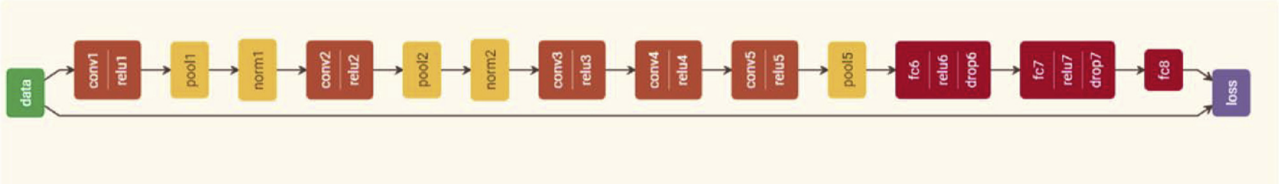


Fig. B.1. Convolutional neural network layers in CaffeNet (created using the neural network visualizer Ethereon Netscope [27]).

References

[1] Lee C-H, Rosenfeld A. Improved methods of estimating shape from shading using the light source coordinate system. Artif Intell 1985;26:125–43. doi:10.1016/0004-3702(85)90026-8.

[2] Pentland AP. Linear shape from shading. Int J Comput Vis 1990;4:153–62. doi:10.1007/BF00127815.

[3] Y. Yang, A. Yuille. Sources from shading. Proceedings of 1991 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. IEEE Comput. Soc. Press; n.d., p. 534–9. doi:10.1109/CVPR.1991.139749.

[4] B.K.P. Horn, M.J. Brooks. Shape from shading. 1989.

[5] Kanbara M, Yokoya N. Real-time estimation of light source environment for photorealistic augmented reality. Proc Int Conf Pattern Recognit 2004;2:911–14. doi:10.1109/ICPR.2004.1334407.

[6] Arief I, McCallum S, Hardeberg JY. Realtime estimation of illumination direction for augmented reality on mobile devices. In: Final Progr Proc - IS T/SID Color Imaging Conf; 2012. p. 111–16.

[7] Kanbara M, Yokoya N. Geometric and photometric registration for real-time augmented reality. In: Proceedings. Int. Symp. Mix. Augment. Real. IEEE Comput. Soc; 2002. p. 279–80. doi:10.1109/ISMAR.2002.1115112.

[8] Ahmed MT, Hemayed EE, Farag AA. Neurocalibration: a neural network that can tell camera calibration parameters. In: Proc. Seventh IEEE Int. Conf. Comput. Vis., 1. IEEE; 1999. p. 463–8. doi:10.1109/ICCV.1999.791257.

[9] Donné S, De Vylder J, Goossens B, Philips W. MATE: machine learning for adaptive calibration template detection. Sensors 2016;16:1858. doi:10.3390/s16111858.

[10] Memon Q, Khan S. Camera calibration and three-dimensional world reconstruction of stereo-vision using neural networks. Int J Syst Sci 2001;32:1155–9. doi:10.1080/00207720010024276.

[11] Jun J, Kim C. Robust camera calibration using neural network. In: Proc IEEE Reg 10 Conf TENCON 99 “Multimedia Technol Asia-Pacific Inf Infrastructure” (Cat No99CH37030), 1; 1999. p. 694–7. doi:10.1109/TENCON.1999.818509.

[12] Stavroulakis P, Sims-Waterhouse D, Piano S, Leach R. Flexible decoupled camera and projector fringe projection system using inertial sensors. Opt Eng 2017;56:1. doi:10.1117/1.OE.56.10.104106.

[13] Moreno D, Taubin G. Simple, accurate, and robust projector–camera calibration supplementary material. In: 2012 Second Int Conf 3D Imaging, Model Process Vis Transm; 2012. p. 3–7. doi:10.1109/3DIMPVT.2012.77.

[14] Luo H, Xu J, Hoa Binh N, Liu S, Zhang C, Chen K. A simple calibration procedure for structured light system. Opt Lasers Eng 2014;57:6–12. doi:10.1016/j.optlaseng.2014.01.010.

[15] Pentland AP. Finding the illuminant direction. J Opt Soc Am 1982;72:448. doi:10.1364/JOSA.72.000448.

[16] Chojnacki W, Gibbins D, Brooks MJ. Revisiting Pentland’s estimator of light source direction. J Opt Soc Am A 1994;11:118. doi:10.1364/JOSAA.11.000118.

[17] Panagopoulos A, Samaras D, Paragios N. Robust shadow and illumination estimation using a mixture model. In: 2009 IEEE Comput Soc Conf Comput Vis Pattern Recognit Work CVPR Work; 2009. p. 651–8. doi:10.1109/CVPRW.2009.5206665.

[18] Panagopoulos A, Wang C, Samaras D, Paragios N. Illumination estimation and cast shadow detection through a higher-order graphical model. In: Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit; 2011. p. 673–80. doi:10.1109/CVPR.2011.5995585.

[19] Plopski A, Mashita T, Kiyokawa K, Takemura H. Reflectance and light source estimation for indoor AR Applications. In: Proc. - IEEE Virtual Real; 2014. doi: 10.1109/VR.2014.6802072.

[20] Townsend A, Senin N, Blunt L, Leach RK, Taylor JS. Surface texture metrology for metal additive manufacturing: a review. Precis Eng 2016;46:34–47. doi:10.1016/j.precisioneng.2016.06.001.

[21] Stavroulakis PI, Leach RK. Invited review article: review of post-process optical form metrology for industrial-grade metal additive manufactured parts. Rev Sci Instrum 2016;87:0411011–04110115. doi:10.1063/1.4944983.

[22] Triantaphyllou A, Giusca CL, Macaulay GD, Roerig F, Hoebel M, Leach RK, et al. Surface texture measurement for additive manufacturing. Surf Topogr Metrol Prop 2015;3:024002. doi:10.1088/2051-672X/3/2/024002.

[23] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, et al. Caffe: convolutional architecture for fast feature embedding. In: Proc. ACM Int. Conf. Multimed. - MM ’14. New York: ACM Press; 2014. p. 675–8. doi:10.1145/2647868.2654889.

[24] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping 2017.

[25] D. Girardeau-Montaut. Cloud Compare n.d. http://www.danielgm.net/cc/ (accessed June 28, 2018).

[26] Ionica MH, Gregg D. The Movidius Myriad architecture’s potential for scientific computing. IEEE Micro 2015;35:6–14. doi:10.1109/MM.2015.4.

[27] Ethereon Netscope n.d. https://github.com/ethereon/netscope (accessed June 5, 2018).