euspen's 22nd International Conference & Exhibition, Geneva, CH, May/June 2022
www.euspen.eu

Efficient close-range photogrammetry reconstruction through autonomous image background removal

Joe Eastwood[1], Richard Leach[1,2], Samanta Piano[1]

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

Joe.Eastwood@nottingham.ac.uk

Abstract
Optical coordinate measurement techniques are growing in popularity due to their high surface coverage and fast data acquisition time, and photogrammetry in particular has the further advantage of inexpensive component requirements. Although data acquisition is fast, data processing speeds can be relatively slow, particularly when high point cloud densities are required, such as for metrology applications. Reconstructed point clouds often also contain superfluous points due to feature matching in the background of images. In this paper, we present a method for exploiting the a-priori knowledge of a measurement system to efficiently and autonomously segment the object of interest from the background within a photogrammetric reconstruction pipeline. We show that using this technique leads to reduced reconstruction times and reduced background feature matching, while maintaining the quality and point density of the point cloud. We show a time reduction of up to 71% and a reduction in background points of up to 77%. The time required to segment the images is also shown to be lower than the time saved during feature matching leading to an overall more efficient measurement pipeline, with potential further gains with a more optimised implementation of the techniques presented.

Coordinate metrology, form metrology, photogrammetry

1. Introduction

Close-range photogrammetry is a coordinate metrology technique that reconstructs an object from a series of overlapping photographic images of that object [1]. By extracting features which are invariant under affine transformations from each image, such as through the scale invariant feature transform (SIFT) algorithm [2], surface features can be matched between images and then triangulated to produce a sparse point cloud of an object's surface. This sparse point cloud is refined through a process of bundle adjustment [3] which iteratively seeks to minimise the reprojection error of the camera network; optionally the sparse point cloud can then be 'densified' to create a dense point cloud using an algorithm such as semi-global matching [4]. The photogrammetric pipeline as described above has one major disadvantage over competing optical coordinate measuring techniques, mainly fringe projection, in that it can be relatively slow. Although exact times depend on quality settings and the number and resolution of the images used, typical sparse reconstructions can take on the order of minutes and dense reconstruction can take on the order of hours.

1.1. Previous work

There are two main approaches to reducing data processing time for photogrammetry. The first is to reduce the number of images required for the reconstruction. Zhang et al. [5] and Eastwood et al. [6] attempted to reduce the number of images needed to reconstruct an object while maintaining reconstruction accuracy. This is achieved by performing a global minimisation procedure on the computer aided design (CAD) data of the object to be measured. They show that using relatively few optimised imaging locations can produce higher quality reconstruction results than using many more unoptimised images.

The other approach to reducing overall processing time is to improve the per-image processing time. Removing the background from an image, and as such reducing the number of points detected and matched, has been used as a method for speeding up reconstruction and reducing superfluous data. Most current approaches to background removal rely on manual masking of images by the user [7,8]. If the background is static relative to the camera, such as in measurement systems using a rotation stage, this can be exploited to remove the background in an automated way (see [9]). 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, commercial photogrammetry software packages can accept masks as part of their reconstruction algorithms. OpenMVG [10], an open-source structure-from-motion 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. Agisoft Metashape [11], a commercial photogrammetry package, can generate image masks but requires the user to manually outline the object in a sub-set of the images used for reconstruction.

In this paper, we first present an algorithm for autonomously segmenting the object from the background of an image taken by a Taraz Metrology P2 photogrammetry system [12]. We show that when these segmented images are used for reconstruction, the overall processing time is significantly reduced. Furthermore, we show that removing the background has the added benefit of reducing unwanted background matches, while maintaining the point density on the object's surface. We use the measurement system to acquire images of two artefacts, then sparse photogrammetric reconstruction is performed on the data, both with and without background removal. The two sets of data are then compared.

2. Methodology

Figure 1(a) shows the measurement system used in this paper. The system has five degrees of freedom of motion with a pair of stereo machine vision cameras for data capture. In a typical scan sixty pairs of images will be taken in an equally spaced ring around the object that is to be measured.

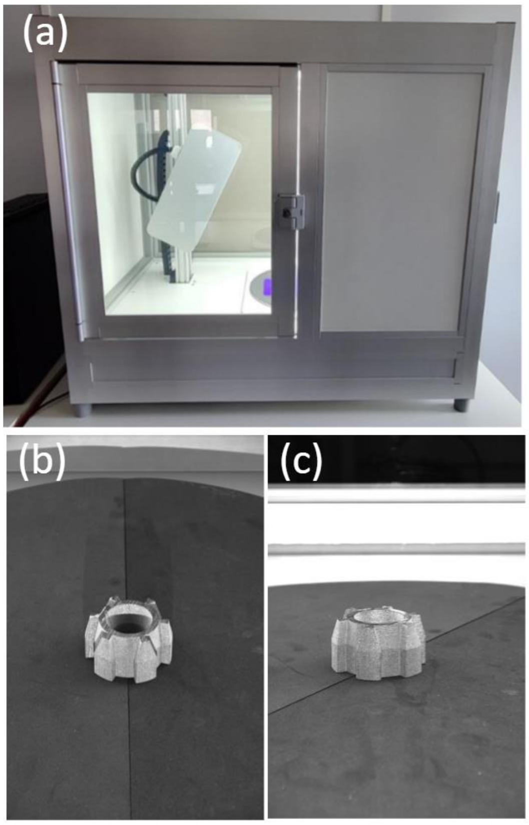

Figure 1. Taraz Metrology P2 photogrammetry system: (a) the measurement system, (b) an example image from a high angle, (c) an example image from a low angle.

As can be seen in Figures 1(b) and 1(c), the background of the image contains no closed contours and the object to be measured is fully contained in the cameras' field of view. Our method for background removal assumes that this will be the case for all objects and all imaging positions. Therefore, if we can detect the largest closed convex contour in the image, this contour will describe the outer bounds of the object within the image. The algorithm consists of five steps: bilateral filtering, Canny edge detection, Gaussian filtering, contour extraction and selection, and boundary dilation.

First, a bilateral filter [13] is applied to the image. A bilateral filter is an edge-preserving smoothing filter and is used here to smooth high spatial frequency texture information from the image which improves the performance of edge detection in the next stage of the algorithm. A bilateral filter combines a spatial filter with an intensity filter, both are Gaussian kernels which are convolved over the image. The spatial filter acts as a normal Gaussian blur and the intensity filter (called the range filter) acts over the space of pixel values. The combined filter is formulated as,

$$I_f(x) = \frac{1}{W_p} \sum_{x_i \in K} [I(x_i)\, G_r(|I(x_i) - I(x)|)\, G_s(|x_i - x|)]\,, (1)$$

where I_f is the filtered image, I is the original image, x is the current pixel coordinate, K is the sliding kernel window centred at x such that x_i is another pixel, G_r is the range kernel with standard deviation σ_r, G_s is the spatial kernel with standard deviation σ_s and W_p is a normalisation term given by,

$$W_p = \sum_{x_i \in K} [G_r(|I(x_i) - I(x)|)\, G_s(|x_i - x|)]\,. \qquad (2)$$

By combining the range and spatial filters, pixels which are close together but have very different values, such as across an edge in an image, are weighted much less than pixels on the same side of the edge which will be closer in value. Figure 2 shows the performance of an edge detection algorithm on an example image with and without bilateral filtering.

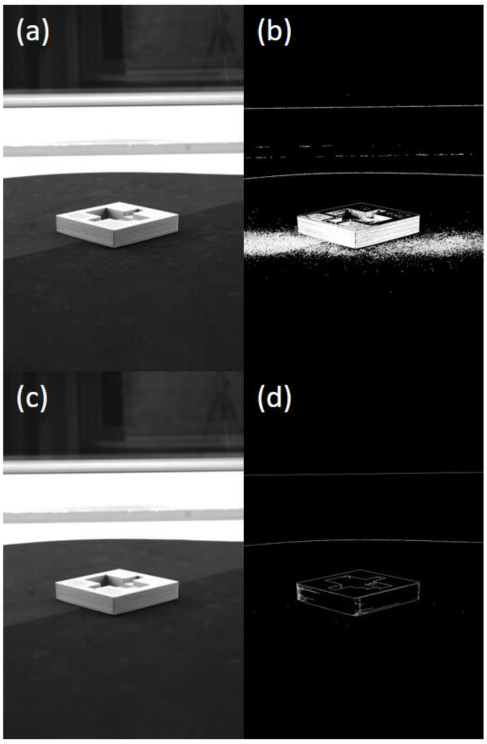

Figure 2. Canny edge detection with and without bilateral filtering: (a) original image, (b) edge detection on original image, (c) filtered image, (d) edge detection on filtered image.

As can be seen in Figure 2(d), the bilateral filter improves the edge selection, reducing the number of minor edges detected in the texture of the rotation stage and object, which can be seen in Figure 2(b).

Next, edges are extracted using a Canny edge detection algorithm [14]. The Canny edge detection algorithm takes the filtered image as input and uses the Sobel operator [15] to detect the image gradients at each pixel, where areas of high gradient are considered edges. These edges are then thinned using non-maximum suppression, a process of selecting only the maximum gradient values in the direction of that gradient at each pixel. Finally, a process of hysteresis selection is used to further refine the edge selection. The hysteresis selection takes two parameters which are thresholds used to determine whether a pixel lies on an edge. If a pixel is above the upper threshold, it is considered an edge, if it is above the lower threshold and neighbours other edge pixels, it is considered an edge, otherwise it is not considered an edge. Usually, these parameters are set manually on a per image basis; however, to ensure the process is autonomous these parameters are selected based on the current image by,

$$t_{lower} = max\left(\left[0, (1 - 0.33) \cdot I\widetilde{(x)}\right]\right), \qquad (3)$$

$$t_{upper} = min\left(\left[255, (1 + 0.33) \cdot I\widetilde{(x)}\right]\right), \qquad (4)$$

where $I\widetilde{(x)}$ is the median grayscale pixel intensity value over the image, and t_{lower} and t_{upper} are the upper and lower thresholds for edge selection. Distributing the threshold values about the median pixel intensity, as in Equations 3 and 4, was evaluated through testing on a range of artefacts and was found to perform well.

Once the edges have been extracted, a smoothing kernel is convolved over the image to connect any discontinuities in the extracted edges.

From the edges, continuous contours are extracted using a pixel following algorithm. These contours are sorted by area and the convex contour with the largest area is selected to be used in the background masking. As a final step, the boundary of this contour is dilated; this ensures the entire object is included in the mask, as the shadow at the base of the object can sometimes lead to the contour being slightly misplaced. This dilated contour is then used to segment the object from the background. Figure 3 shows the entire pipeline applied to an example artefact.

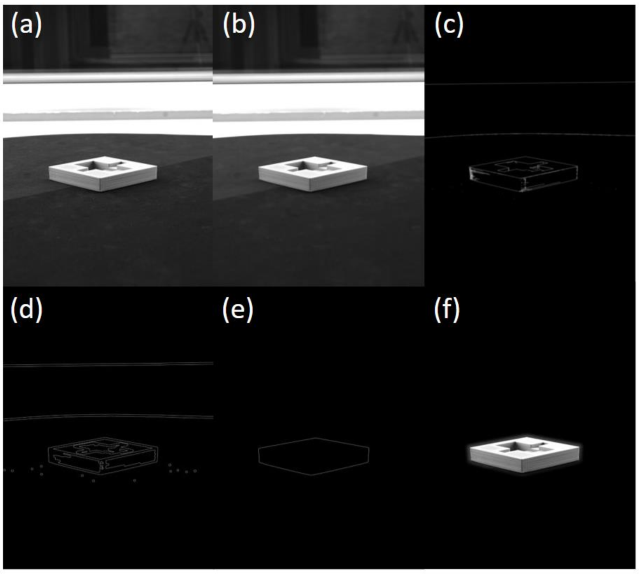

Figure 3. Background removal pipeline stages applied to scan data of a 3D printed polymer artefact: (a) original image, (b) bilateral filter, (c) edge detection, (d) contour extraction, (e) contour selection, (f) dilation and masking .

3. Results

The proposed background removal pipeline was tested for two different artefacts, which are shown in Figure 4.

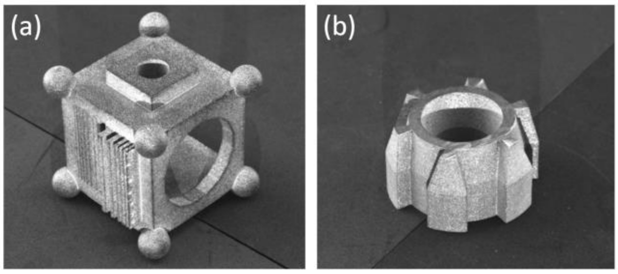

Figure 4. Test artefacts: (a) artefact 1 manufactured by polymer powder bed fusion (PBF), (b) artefact 2 manufactured from Ti64 using metal PBF.

In both cases, 60 stereo pairs of images were captured on a ring of equally spaced points around each object leading to a total of 120 images. Each image was segmented using the proposed method which was implemented using the OpenCV library [16] for Python 3.7, this took 10 s per image. These images were then reconstructed with OpenMVG, using a sequential reconstruction method [10].

Figure 5 shows the reconstruction results for artefact 1.

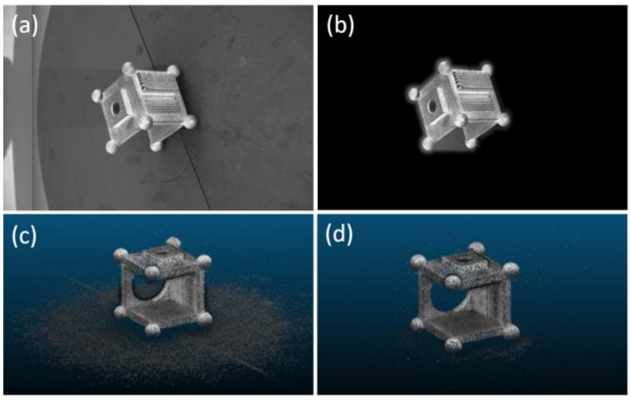

Figure 5. Open MVG reconstruction results on 120 images of artefact 1: (a) example scan image, (b) example masked image, (c) normal sparse reconstruction, (d) sparse reconstruction with masking.

As can be seen in Figure 5, artefact 1 was reconstructed well in both cases with approximately 130,000 surface points being reconstructed using both techniques. However, the normal reconstruction shown in Figure 5(c) also reconstructed 110,198 background points. In comparison, using the proposed method, the reconstruction shown in Figure 5(d) reconstructed only 5465 background points. This led to a decrease in processing time from 2353 s to 1130 s, a reduction in time of 52%.

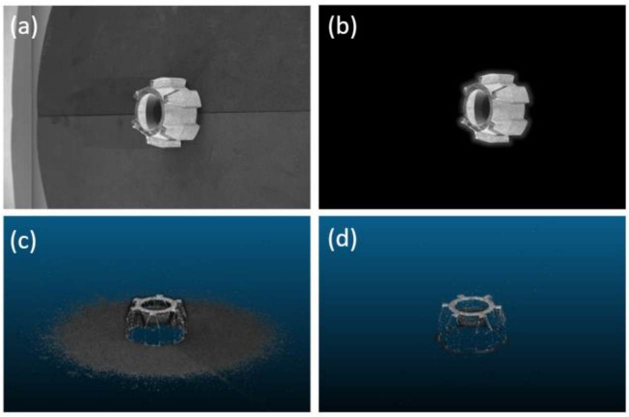

Figure 6. Open MVG reconstruction results on 120 images of artefact 2: (a) example scan image, (b) example masked image, (c) normal sparse reconstruction, (d) sparse reconstruction with masking.

To provide further evidence of the effectiveness of the proposed approach, the same test was conducted on a second artefact as shown in Figure 6. Once again, the reconstruction of the object was similar in both methods, but the number of background matches was significantly decreased when masking was used. In this case, when using conventional reconstruction, there were 55,923 object points and 340,639 background points, compared to 54,041 object points and 3,085 background points when the proposed image masking method is utilised. This over 100 times reduction in background points leads to a reduction in reconstruction time from 11,524 s to 3302 s, which is a reduction of 71%.

4. Discussion

It is shown in Figures 5 and 6 that the quality of the sparse reconstruction is maintained when the background removal is utilised. In both cases the masked reconstructions are within 3% of the object points, as compared to using the un-masked images.

When reconstructing smaller objects, such as artefact 2, it was found that the vast majority of the reconstructed points are background points, which are not useful to the measurement of the object. In the specific case of artefact 2 only 18% of the reconstructed points were related to the measured object. This explains the even greater time reduction when compared to artefact 1 in which 54% of the reconstructed points were on the object surface.

When using the masked images, the reconstruction of artefact 1 had 95.9% object points and the reconstruction of artefact 2 had 94.6% object points. This result is clearly a significant improvement over the un-masked reconstructions. The reason that there are still some background points is because of the dilation of the contour used to mask the object. However, this leads to better reconstruction results than the cases when the un-dilated boundary would remove some of the object pixels along with the background. Furthermore, there are some erroneous 'floating' points both with and without the masking process. However, these points will be filtered out before densification by setting a maximum reprojection error threshold for a point to be included.

There is potential to further reduce the number of background points carried forward to reconstruction by refining the background removal. For example, Figure 5(b) shows how, if a part has concave features, a large amount of background can fail to be masked out. However, during testing, it was found that attempts to increase the amount of background pixels removed lead to the algorithm becoming more unstable and frequently removing object pixels on more complex objects. Therefore, in this paper we chose to include more background pixels to ensure all object pixels were included for complex parts.

The current Python implementation of the algorithm, when run on an Intel Xeon W-2723 CPU, takes around 10 s per image of processing time to complete the image segmentation. It is likely this time can be significantly reduced with a more efficient implementation using optimised code and more powerful and effective hardware, such as GPU acceleration. However, with the current implementation, the sum of the masking time and reconstruction time is still lower that the time taken for reconstruction using the un-masked images.

5. Conclusions

We have presented a method to robustly and autonomously remove the background from images. Our method exploits the knowledge that there are no large, closed contours in the background of images captured by the measurement system (in this case a Taraz Metrology P2). Therefore, the largest closed contour in the image will define the outline of the object which is being measured.

We show that using these masked images in sparse photogrammetric reconstruction can lead to up to 71% time savings and the point density on the object surface is maintained. Furthermore, we show that the number of superfluous background matches is significantly reduced with around 95% of reconstructed points constituting the object surface in both case studies presented.

6. Future work

Immediate future work is to determine the impact of the proposed approach on dense reconstruction through comparison to CMM data. This is important to verify that masking the background does not lead to form or dimensional errors in the final measurement result.

Further work has begun on a more optimised version of the algorithm implemented in efficient Rust code. Early results show this is likely to lead to significantly reduced processing time at the image masking stage, compared to the Python implementation used to generate the results shown in this paper.

Acknowledgements

This work was supported by the EPSRC (grants EP/M008983/1 and EP/L016567/1) and Taraz Metrology Ltd. The authors would also like to thank Dr Danny Sims-Waterhouse for his feedback on the project.

References

[1] Luhmann T, Robson S, Kyle S, Harley I 2007 *Close Range Photogrammetry* (Wiley)

[2] Lowe D G 1999 Object recognition from local scale-invariant features *Proc. ICCV (Kerkyra, Greece)* **2** 1150–1157

[3] Triggs B, McLauchlan P, Hartley R, Fitzgibbon A 1999 Bundle adjustment — A modern synthesis *Proc. ICCV (Kerkyra, Greece)* 298–372

[4] Hirschmüller H, Maximilian B, Ines E 2012 Memory efficient semi-global matching *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci* **3** 371-376

[5] Zhang H, Eastwood J, Isa M, Sims-Waterhouse D, Leach R K, Piano S 2021 Optimisation of camera positions for optical coordinate measurement based on visible point analysis *Prec. Eng.* **67** 178-188

[6] Eastwood J, Zhang H, Isa M, Sims-Waterhouse D, Leach R K, Piano S 2020 Smart photogrammetry for three-dimensional shape measurement *Proc. SPIE* 1135A

[7] Woloszyk K, Bielski P M, Garbatov Y, Mikulski T 2021 Photogrammetry image-based approach for imperfect structure modelling and FE analysis *Ocean Eng.* **223** 108665

[8] Rupnik E, Daakir M, Deseilligny M P 2017 MicMac–a free, open-source solution for photogrammetry *Open Geospat. Data, Softw. Stand.* **2** 1-9.

[9] Sathirasethawong C, Sun , Lambert A, Tahtali M 2019 Foreground object image masking via EPI and edge detection for photogrammetry with static background *Proc. ISVC (Lake Tahoe, USA)* 345-357

[10] Moulon P, Monasse P, Perrot R, Marlet R 2016 OpenMVG: Open multiple view geometry *Proc. RRPR (Cancún, Mexico)* 60-74

[11] Agisoft Metashape: Standard edition, v. 1.5.5 www.agisoft.com

[12] Taraz Metrology Ltd https://taraz-metrology.com/

[13] Banterle F, Corsini M, Cignoni P, Scopigno R 2012 A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain *Comput. Graph.* **31** 19-32

[14] Zhou P, Ye W, Xia Y, Wang Q 2011 An improved canny algorithm for edge detection *J. Comput. Inf. Syst* **7** 1516-1523.

[15] Gao W, Zhang X, Yang L, Liu H 2010 An improved Sobel edge detection *Proc. CSAE (Sanya, China)* **5** 67-71

[16] Bradski G 2000 The OpenCV Library *DDJ* 2236121