

Deep Learning for Accurate Corner Detection in Computer Vision-Based Inspection

M. Fikret Ercan(✉) and Ricky Ben Wang

School of Electrical and Electronic Engineering, Singapore Polytechnic, 500 Dover Rd, Singapore, Singapore
M_Fikret_Ercan@sp.edu.sg

Abstract. This paper describes application of deep learning for accurate detection of corner points in images and its application for an inspection system developed for the worker training and assessment. In our local built and construction industry, workers need to be certified for their technical skills through a training and assessment process. Assessment involves trainees to understand a task given with a technical drawing, e.g. electrical wiring and trunking wall assembly, and implement it accurately in a given period of time. Typically experts manually/visually evaluate the finished assembly and decide if it's done correctly. In this study, we employed computer vision techniques for the assessment process in order to reduce significant man hour. Computer vision based system measures dimensions, orientation and position of the wall assembly and produces a report accordingly. However, analysis depends on accurate detection of the objects and their corner points which are used as reference points for measurements. Corner detection is widely used in image processing and there are numerous algorithms available in the literature. Conventional algorithms are founded upon pixel based operations and they return many redundant or false corner points. In this study, we employed a hybrid approach using deep learning and Minimum Eigen value corner detection for this purpose and achieved highly accurate corner detection. This subsequently improved the reliability of the inspection system.

Keywords: Computer vision · Deep learning · Corner detection · Quality construction

1 Introduction

The application of computer vision for inspecting and quality control has been widely used in industry. It has wide and diverse range of applications extending from manufacturing [1] to farming [2], civil engineering [3] to biomedical engineering [4]. Most of these applications concerned with detecting patterns, irregularities or deriving information from the objects analyzed which can be broadly defined as recognition tasks. Computer vision is also widely used in automated assembly [5] and vision based control of mobile robots [6]. Among these, manufacturing is a typical application where geometric measurement tests for parts and objects are performed. In such computer vision application, inspection of parts or an assembly achieved by automatic recognition of the

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12950, pp. 45–54, 2021.
https://doi.org/10.1007/978-3-030-86960-1_4

object and then analyzing geometric measurements. Using computer vision in manufacturing for this purpose saves time, minimize errors and improve the automation process. There is a vast number of application examples presented in literature [7, 8].

The application concerned in this paper is an automated inspection and assessment system developed to be used in construction industry. In the local construction industry, workers are required to go through a training and certification process in order to ensure quality of building and construction standards. After the training period, workers go through an assessment process for their certification in which they are expected to deliver an assembly from its technical drawing in a given time period. The quality of the work produced is examined by a team of experts manually and visually. The assessment involves checking measurements, alignments, and positions of the assembly and rating is based on its compliance with the technical drawing given. Naturally, it is an arduous and time consuming process for the training providers. Therefore, a computer vision solution is developed to automate the assessment process. Computer vision and image processing techniques in built and construction industry has long been utilized typically for construction safety, resource tracking and activity monitoring, surveying, inspection and condition monitoring. For a comprehensive survey on computer vision application in built and construction industry refer to Martinez et al. [3].

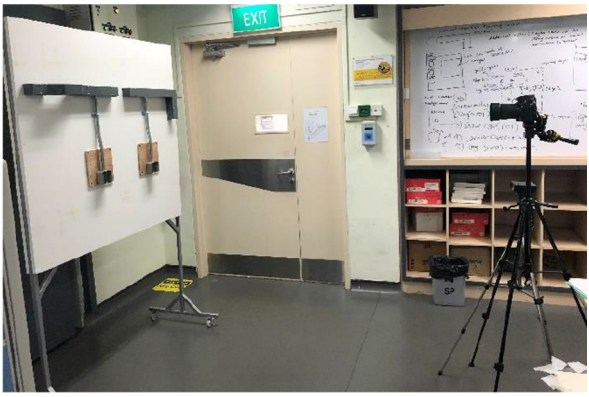
In our application, vision system inspects the work produced by a trainee in terms of the geometric measures, orientation and position. It produces a report highlighting discrepancies from the specifications. A fundamental object feature used in the process is corner detection. Performance of the system is therefore highly dependent on accurate detection of the corner points of the objects. Corner detection is widely used in image processing and there are numerous algorithms available in the literature most popular ones being FAST [9], SUSAN [10] and Harris corner detection algorithms [11]. These algorithms are pixel based operations and they often return redundant or false corner points. In this study, we experimented with FAST, Harris and Minimum Eigen value corner detection and found that Minimum Eigen value performed well. However, it is computationally costly to filter out noisy detections in order to identify corners that are belong to the objects. On the other hand, corner detectors utilizing neural network and evolutionary techniques present competitive results especially with noisy images [12]. Recently, deep learning algorithms become very popular in computer vision due to their remarkable performance. Our experiments showed that deep learning can identify object corners effectively however it is within a bounded box. Using a hybrid approach that is filtering Minimum Eigen value result with deep learning output, we are able to detect object corners very accurately which makes the geometric measurement inspection robust.

In the following an overview of the system, together with hybrid solution and performance results will be presented.

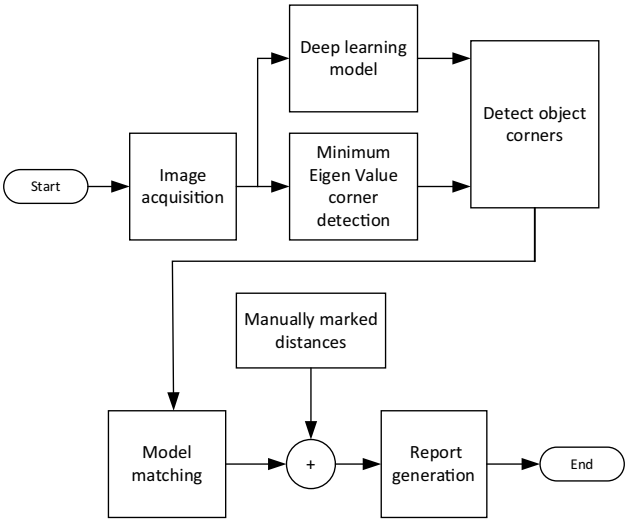
2 System Overview

Computer vision hardware used in this study is made of off –the-shelve components and the hardware setup is rather straightforward as shown in Fig. 1(a). The camera is a Nikon D800 DSLR with an AF-S Micro NIKKOR 60 mm f/2.8G ED Lens. Optoma

LED projector serves as an illuminator, providing a constant lighting condition which improves image processing accuracy. A set of wall assembly developed by trainee is placed 211 cm from the camera lens. The camera is set at the same height as the midline of the assembly. Matlab is used for the software development.



(a)



(b)

Fig. 1. (a) Hardware setup. (b) A flow chart of the operations.

Figure 1 (b) shows a flow of operations. After image acquisition, the first step of image processing algorithm is to detect corner points using hybrid method. These corner points will be used as reference points in measuring objects dimensions, alignments, angles, and so on. A model of the objects in the wall assembly is available as CAD drawing. The next step of the process is to match detected features with the model and determine discrepancies. Furthermore, a user can also manually mark any points of interest in

the image and get the measurements for them. Finally, a report is generated providing assessment results.

2.1 Detecting Corner Points

There is a vast literature on corner detection algorithms. Amongst those, we experimented with three popular algorithms namely Minimum Eigen, Harris and FAST. After experimenting with these three functions by testing them with numerous trunking images, we observed that reference corner points that we would like to detect was mostly captured by Minimum Eigen algorithm. Output from these three algorithms are shown in Fig. 2 for comparison.

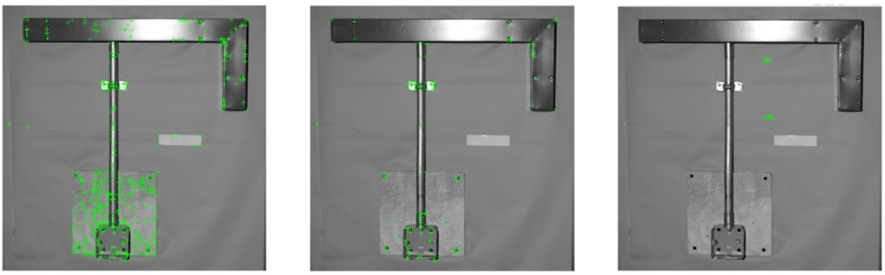
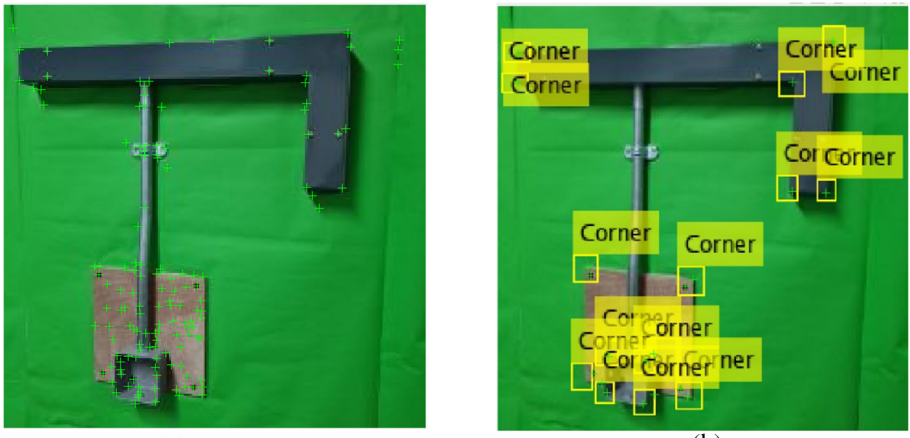


Fig. 2. Corner detection algorithms. Results for (a) MinEigen (b) Harris (c) Fast

Although the Minimum Eigen algorithm is capable of detecting the corner points of the objects that we are interested, there are still redundancy and errors present as seen from Fig. 3(a). The quality of the image including its resolutions, lighting conditions also contribute to these errors. On the other hand, recent developments in computer vision demonstrate that deep learning algorithms are very effective in object detection, particularly in natural environment with many background clutter. In the next phase of corner detection, a trained deep learning network is used for detecting key reference points in the image. The bounding boxes generated by the trained network as the region of interest (ROI) are not as accurate as the pixel level detection of corner points though using the ROI we are able to filter out erroneous corner points as seen on Fig. 3(b).

2.2 Deep Learning for Detecting Trunking Corners

Deep Learning network architecture used in this study is YOLO version 2. Other meta architectures that are light weight and fast such as SSD (Single Shot Multibox Detector), Faster R-CNN were also experimented but not considered in this application since accuracy is more imperative than the real-time performance. YOLO v2 is trained to do classification and bounding box regression at the same time. Additionally, YOLO v2 learns generalizable representations of objects therefore it can perform better when applied to new domains or unexpected inputs. For the feature extraction network, we



(a) (b)

Fig. 3. Error filtering using trained deep learning networks. (a) Detected corner points before filter (b) Corners detected after error filtering

employed Resnet-50 based on our earlier empirical study in comparing network and feature extractor architectures [13]. Resnet-50 consists of 50 layers and the pre-trained network can classify images in up to a thousand object categories. The detection sub-network is a small CNN compared to the feature extraction network and is composed of a few convolutional layers and layers specific for YOLO v2.

For the training, we have used 500 images of trunking wall assembly as shown in Fig. 1 under different light conditions and angles. After multiple iterations of training, a model with a precision of 92% achieved. To test the accuracy and consistency of the trained network, we used 20 test images with different light and camera angles. The accuracy is defined as the ratio of accurately detected corner points over actual number of corner points in the image. Table 1 tabulates these test results. Majority of test results lie above 93% accuracy, however there were also some outliers with accuracies as low as 75% due to poor image quality. Overall a median of 93% with a confidence interval of 95% was satisfactory to use such model in the corner detection problem.

Table 1. Network performance on test images

Image	Corners expected	Corners detected	Corners missed	Accuracy	Wrong detection
1	12	12	0	100%	1
2	12	12	0	100%	1
3	12	12	0	100%	1
4	12	12	0	100%	3

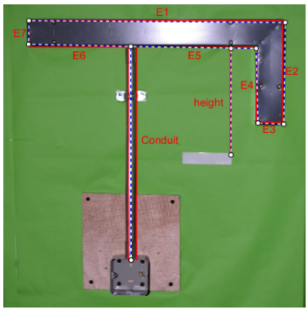
(*continued*)

Table 1. (*continued*)

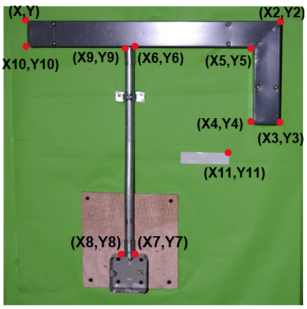
Image	Corners expected	Corners detected	Corners missed	Accuracy	Wrong detection
5	12	12	0	100%	1
6	12	12	0	100%	1
7	12	12	0	100%	1
8	12	12	0	100%	0
9	12	12	0	100%	1
10	12	12	0	100%	2
11	12	11	1	91.7%	0
12	12	10	2	83.3%	0
13	12	11	1	91.7%	0
14	12	10	2	83.3%	1
15	12	9	3	75%	1
16	12	8	4	66.7%	0
17	12	10	2	83.3%	0
18	12	10	2	83.3%	0
19	12	12	0	100%	0
20	12	12	0	100%	0
Average				92.2%	

2.3 Template Matching and Operation of the System

Hybrid approach provided a robust detection of corner points that are needed for further analysis. However, there may still be a few remnant stray corner points. As the technical drawing of the wall assembly is known prior, in the final step of analysis a template matching method is used to establish the most suitable corner points to represent the corners of the trunking object. Template is constructed starting from the top left corner point using geometric methods. The selection of valid corner points is done by accepting the corner points that are closest to the template's corner points, thereby completing the corner point selection process. Figure 4(a) shows the template and Fig. 4(b) the final reference corner points extracted. Once these reference points obtained measurements for assessment are done using simple geometric techniques. Finally, a report as an excel spreadsheet is created listing the result of analysis as shown in Fig. 4(c).



(a)



(b)

	A	B	C	D	E	F
1	Row	Auto_results	actual	Error_a		
2	Edge1	504.9150085	500	0.983%	pass	
3	Edge2	200.1002808	200	0.0501%	pass	
4	Edge3	50.66304016	49	3.394%	pass	
5	Edge4	146.8341217	148	0.7878%	pass	
6	Edge5	247.0778198	240	2.9491%	pass	
7	Edge6	201.464798	200	0.7324%	pass	
8	Edge7	46.49208832	48	3.1415%	pass	
9	Conduit	420.6500549	420	0.1548%	pass	
10	Referenc	209.5552216	209	0.2657%	pass	
11						
12						
13	Row	Manual_resul	actual	Error_m		
14	Edge1	502.701455	500	0.5403%	pass	
15	Edge2	203.6600535	200	1.83%	pass	
16	Edge3	51.53149243	49	5.1663%	fail	
17	Edge4	148.3876106	148	0.2619%	pass	
18	Edge5	241.5190867	240	0.633%	pass	
19	Edge6	193.813868	200	3.0931%	pass	
20	Edge7	49.99019318	48	4.1462%	pass	
21	Conduit	424.7243804	420	1.1249%	pass	
22	Referenc	212.7393805	209	1.7892%	pass	
23						
24	Row	Error				
25	Overall	0.33%				

(c)

Fig. 4. Model matching with template. (a) Template of the object. Edges labeled 1 to 6 and conduit length are some of the key measurements (b) Final reference corner points marked. (c) Final report generated for the trainee assessment.

3 Performance of the System

In order to evaluate the performance of our system, we used a test setup which is fabricated accurately as indicated in the technical drawings given to trainees. We have taken images from the fixed point under various lighting conditions as lighting will immediately affect the image processing performance. Table 2 shows measurement results for the selected assessment points using five images taken under good lighting conditions. These results compared with the actual dimensions of the test setup. Difference between them implies the measurement error of the vision system.

Table 2 shows all the dimensions detected by the program for each of the test images. As seen on the Table 2, most of the average percentage of error lies below 1%, which implies a negligible measurement error. However, it is also noticeable that the average percentage of error for “Edge 4”and “Edge 5” is relatively higher than the ones of the other edges. These larger errors are due to the inaccuracies generated during template matching particularly caused by shadows of the trunking.

The template matching method helped to find the most suitable/correct corner points to be used for the calculation of the trunking’s dimensions, however if there are errors present, such as an inaccurate camera calibration, it will affect the selection of the most suitable corner points, hence causing the chosen corner points to be inaccurate and resulting a miscalculation of the trunking’s dimensions.

A function namely “display template” added to the GUI of the system. It enables user to have a clear visual of the areas of error. And displays the overall error at the bottom left side for reference as shown on Fig. 5(a). Standard trunking’s area displayed in white pixels based on the dimensions of the template. While the display of the test trunking’s area (in red) is based on the detected corner points of the test trunking. Hence, the differences in area (the error) can be spotted easily by overlapping these two areas together.

Table 2. Average measurement (in millimeters) error for test images

Feature	Image 1	Image 2	Image 3	Image 4	Image 5	Actual	Average measurement error (%)
Edge 1	504.6	504.3	502.7	504.2	502.9	500	0.75
Edge 2	199.8	199.6	199.5	199.8	199.3	200	0.20
Edge 3	50.1	50.7	50.5	50.4	50.4	50	0.84
Edge 4	145.7	146.6	146.4	146.6	146.9	150	2.37
Edge 5	231.7	246.4	231.9	231.8	231.9	240	3.26
Edge 6	201.1	201.2	200.6	200.5	200.6	200	0.40
Conduit	419.9	410.3	408.5	421.0	420.3	420	1.08
Angle	130.7°	130.5°	130.4°	131.3°	130.8°	131°	0.29

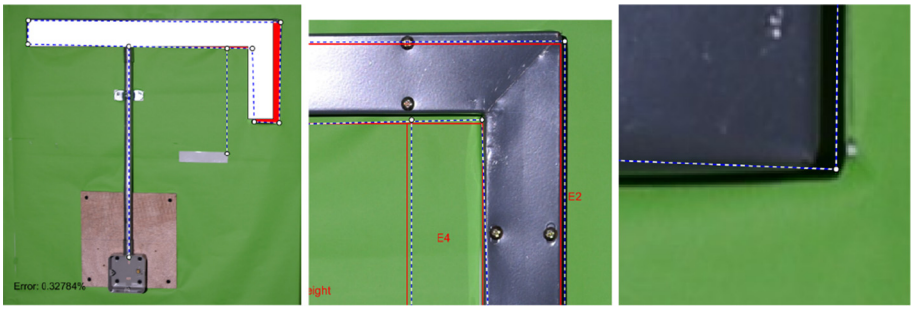


Fig. 5. Example of result (a) with error (percentage error calculated and indicated at the bottom left) (b) error due to image quality (c) due to shadows. (Color figure online)

The image in Fig. 5 (b) shows an example of inaccuracy when detecting suitable corner points. The picture of the trunking shown in this figure is slightly tilted by 1° anti-clockwise. Since the program is unable to detect such shortcomings, the template is not tilted to fit the image. Hence, the selection for the correct corner points will be based on the inaccurate template match therefore producing errors in its dimensions. Based on our experiments, we note that this error in inaccuracy usually within 5%. If the error in accuracy is more than 5%, we observed other external factors are contributing to the error in measurement accuracy. Figure 5(c) shows the error in corner detection due to the presence of shadows. Shadows appear when there isn't an even distribution of light source on the trunking when taking the image using a camera. Therefore, if the difference in shade between the shadow and the trunking itself isn't stark enough, the corner detector algorithm will detect the shadow's corner point, instead of the trunking's. Hence, if the process of template matching will not be unable to correct this error, the shadow's corner will be used to calculate the trunking's dimensions, producing an inaccurate result. This can error can easily be prevented with good lighting arrangement in hardware.

In conclusion, test results presented in Table 2 imply that the computer vision based system delivers accurate measurement results. Minor inaccuracies are mainly due to environmental factors. Current set-up utilizes a commercial camera and light system. The performance can be further improved if the vision set up is replaced with an industrial grade lighting and camera hardware.

4 Conclusions

This paper described a computer vision based inspection system for automated assessment of work done by trainees in construction industry for their certification. A key feature used in the geometric measurement of the assembly was corner points of the objects. Accurate detection of them significantly contributes to the system performance. Minimum Eigen Value corner detection algorithm combined with a deep learning model for the object corners provided accurate corner detection. Using template matching, detected objects dimensions are compared with the model obtained from technical drawings. Our experimental results revealed that vision system has minor measurement error under good lighting conditions. Our future study involves developing a generic system where various assessment configurations can be programmed by the instructors creating a wide range of scenarios that can be used for examining trainees work and grading them.

Acknowledgement. This project is sponsored by Singapore Polytechnic on under the grant number 03-11000-36-J723 and cosponsored by Fonda Global Engineering Pte Ltd. Authors acknowledge the contributions of interns Fong Kah Kian and Jacky Wijaya.

References

1. Batchelor, B.G.: Machine vision for industrial applications. In: Batchelor, B.G. (eds.) Machine Vision Handbook. Springer, London (2012). https://doi.org/10.1007/978-1-84996-169-1
2. Mavridou, E., Vrochidou, E., Papakostas, G.A., Pachidis, T., Kaburlasos, V.G.: Machine vision systems in precision agriculture for crop farming. J. Imag. **5**(89), 1–32 (2019)
3. Martinez, P., Al-Hussein, M., Ahmad, R.: A scientometric analysis and critical review of computer vision applications for construction. Autom. Constr. **107**, 102947 (2019)
4. Park, C., Took, C.C., Seong, J.-K.: Machine learning in biomedical engineering. Biomed. Eng. Lett. **8**(1), 1–3 (2018)
5. Pierleoni, P., Belli, A., Palma, L., Palmucci, M., Sabbatini, L.: A machine vision system for manual assembly line monitoring. In: International Conference on Intelligent Engineering and Management (ICIEM), London, UK, pp. 33–38 (2020)
6. Sankowski, D., Nowakowski, J.: Computer Vision in Robotics and Industrial Applications. World Scientific Publishing, New Jersey (2014)
7. Okarma, K., Fastowicz, J.: Computer vision methods for non-destructive quality assessment in additive manufacturing. In: Burduk, R., Kurzynski, M., Wozniak, M. (eds.) CORES 2019. AISC, vol. 977, pp. 11–20. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-19738-4_2
8. Rajan, A.J., Jayakrishna, K., Vignesh, T., Chandradass, J., Kannan, T.T.M.: Development of computer vision for inspection of bolt using convolutional neural network. Mater. Today Proc. **45**(7), 6931–6935 (2021)

9. Trajković, M., Hedley, M.: Fast corner detection. Image Vis. Comput. **16**(2), 75–87 (1998)
10. Smith, S., Brady, J.: SUSAN-A new approach to low level image processing. Int. J. Comput. Vis. **23**(1), 45–48 (1997)
11. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference, pp. 147–151 (1988)
12. Cuevas, E., Rodríguez, A., Alejo-Reyes, A., Del-Valle-Soto, C.: Corner detection algorithm based on Cellular Neural Networks (CNN) and Differential Evolution (DE). In: Recent Meta-heuristic Computation Schemes in Engineering. Studies in Computational Intelligence, vol. 948. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-66007-9_4
13. Ercan, M.F., Qiankun, A.L., Sakai, S.S, Miyazaki, T.: Circle detection in images: a deep learning approach. In: Proceedings of IEEE OCEANS Conference (2020)