



Computing the Orientation of Hardware Components from Images using Traditional Computer Vision Methods

Gambale, A., Coleman, S., Kerr, D., & Kerr, E. (in press). Computing the Orientation of Hardware Components from Images using Traditional Computer Vision Methods. In *The 39th International Manufacturing Conference: Smart Manufacturing: The Next Generation* (pp. 1-2)

[Link to publication record in Ulster University Research Portal](#)

Published in:

The 39th International Manufacturing Conference

Publication Status:

Accepted/In press: 25/08/2023

Document Version

Publisher's PDF, also known as Version of record

General rights

Copyright for the publications made accessible via Ulster University's Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact pure-support@ulster.ac.uk.

Computing the Orientation of Hardware Components from Images using Traditional Computer Vision Methods

Antonio R. Gambale* Emmett Kerr, Dermot Kerr, Sonya Coleman

School of Computing, Engineering and Intelligent Systems

Ulster University, Magee Campus, Northern Ireland

*Contact: Gambale-A@ulster.ac.uk

keywords: Manufacturing, Vision, PCA, Machine Learning

Abstract—This paper introduces a methodology for precise object orientation determination using Principal Component Analysis, with robust performance under significant noise conditions. It validates the potential to mitigate the challenges associated with Axis-Aligned Bounding Boxes in smart manufacturing environments. The proposed approach paves the way for improved alignment in robotic grasping tasks, positioning it as a computationally efficient alternative to ML methods employing Oriented Bounding Boxes. The methodology demonstrated a maximum angle deviation of 3.5 degrees under severe noise conditions through testing with bolts in orientations of 0 to 180 degrees.

I INTRODUCTION

It is aspired in both academia and industry to enhance production lines with vision systems capable of processing real-time data [1]. This enhancement aids in the accurate identification of objects which can be used to facilitate robotic grasping operations with dynamic or moving infeeds [2]. Consequently, it provides substantial flexibility to production lines, fostering an adaptable manufacturing environment [3]. The subsequent discourse in the current literature highlights a growing fascination in employing machine learning (ML) techniques, particularly Convolutional Neural Networks (CNNs), to enhance object recognition for robotic grasping operations [4]. These ML-driven object detection algorithms, such as Faster R-CNN, YOLO, and SSD, operate under two distinct methodologies, discernible by the nature of their bounding boxes (bboxes).

The first of these methodologies can be seen in models such as, Faster R-CNN, YOLO, and SSD, which implement Axis-Aligned Bounding Boxes (AABBs) [1]. AABBs use bboxes that align with the axes of the image, assuming a vertical and horizontal position. AABBs provide a simple and computational efficient way to outline the objects in the scene making them a preferred choice for many real-time applications [1]. On the contrary, certain models like R-FCN and RPN (Rotated Region Proposal Networks) employ Oriented Bounding Boxes (OBBs) [5]. OBBs use bboxes, that are adaptable to the object's orientation within the image, therefore providing a tighter fit around the object. This feature becomes particularly advantageous in specific applications where the object's orientation is of importance such as in detecting objects to be grasped by robot arms from moving platforms or conveyors [2].

AABBs, while being less computationally demanding than OBBs, may encapsulate significant background area,

particularly when dealing with rotated or irregularly shaped objects. This inclusion could potentially introduce noise into a model's predictions, detracting from overall accuracy. Conversely, OBBs have the advantage of providing a more precise fit around the object and minimizing background noise, thus potentially enhancing accuracy for certain tasks. However, this improvement in performance comes with increased computational requirements and complexity. For these reasons, K. Nguyen [4] highlights that YOLO is considered the optimal choice for real-time object detection applications requiring high frames per second and moderate precision. Nevertheless, this model has limitations as it uses AABBs, which are insufficient for autonomous robotic grasping of objects that are asymmetrical along multiple axes, such as bolts. Hence, we present a computationally efficient approach for determining the orientation of components, such as bolts. The objective of this paper is to introduce a methodology for deriving the orientation of an object post-detection using ML techniques that employ AABBs. By doing so, it seeks to mitigate the challenges associated with AABBs, thereby enhancing the alignment capabilities of robotic grippers with the grasped parts. Notably, this proposed methodology also aims to provide a more computationally efficient alternative to methods utilising OBBs.

II METHODS

The proposed method loads the test images as grayscale then applies a Gaussian blur (kernel size of 5x5) to reduce noise. The image is segmented using an adaptive binarisation method to separate bolt pixels from the background. This is accomplished using the mean adaptive thresholding technique in OpenCV, `cv2.ADAPTIVE_THRESH_MEAN_C` [7], where the threshold value corresponds to the neighbourhood's average intensity. The neighbourhood size is determined by the `blockSize` parameter, set at 251. From this mean, a constant `C` value of 15, optimised through systematic trial and error, is subtracted to refine the threshold. The binary output image, achieved by applying the computed threshold to each block, signifies bolt pixels as 1's (black) and the remainder as 0's (white). To ensure a more continuous bolt shape, a morphological 'closing' operation is performed on the binary image. This operation fills small gaps within the bolt shape using a rectangular kernel with a size of 21x21 pixels. The bolt's pixel locations within the image were identified and

centralised by examining the non-zero intensity values. The mean was then subtracted to centralise their positions. This process yields a new set of points referred to as centralised points, which have a zero mean. Principal Component Analysis (PCA) is then applied to determine the orientation of the bolt. The first step involves calculating the covariance matrix, Σ of the centralized points. The covariance matrix is computed using the equation (1).

$$\Sigma = (1 / (n - 1)) * \Sigma ((x_i - \bar{x}) * (x_i - \bar{x})^T) \quad (1)$$

Where, x_i represents the centralized points, n is the number of points, and \bar{x} is the mean of the points. Eigen decomposition is performed on the covariance matrix, resulting in eigenvalues λ and their corresponding eigenvectors v . The eigenvalues are sorted in descending order, and their corresponding eigenvectors are likewise ordered. The eigenvectors represent the principal components, which signify the directions in which the data varies the most. To determine the orientation, the dot product between the second eigenvector (representing the minor axis of variation) and the centralised points is computed. By determining the angle of the principal eigenvector relative to the image's vertical axis and gauging the relative position of each point to the minor axis line, the bolt's orientation and shape can be accurately established. These calculations facilitate subsequent image analyses, thereby enhancing the ability to characterise the bolt.

III RESULTS & DISCUSSION

The methodology was evaluated using colour images of $M12 \times 40$ hex bolts. To assess performance across a range of bolt orientations, 19 synthetic images were generated with variations in the bolt's angle at 10-degree intervals from 0 to 180 degrees. Furthermore, to examine the method's robustness, these images were augmented using Gaussian noise, following the approach outlined by G.B.P. da Costa [8]. The augmentation process resulted in the creation of six distinct test sets, each comprising 19 images. These test sets were generated by applying different levels of Gaussian noise, with standard deviations (σ) ranging from 0 to 50. This experimental setup enabled comprehensive testing and analysis of the methodology's performance under different bolt orientations and varying levels of image noise. An example of the image noise range can be seen in Fig. 1.



Fig. 1 Gaussian Noise Levels in Test Images: $\sigma = 0$, $\sigma = 20$, and $\sigma = 50$

Successful orientation detection was determined if the observed angle was within a 10-degree range of the known one, factoring in the robot gripper's ability to effectively handle minor misalignments. The proposed method reliably identified the bolt's angle within this tolerance across all test images, as depicted in Fig. 2. While no measured angle precisely matched the known angle in all rotations and noise

levels a high degree of accuracy was achieved. Notably, the most significant deviations were observed in angles from 0 and 80 degrees, with the maximum average deviation of 3.5 degrees (across all noise levels) recorded for the bolt at 40 degrees. In contrast, the precision improved for bolts oriented past 90 degrees, with the highest average deviation (across all noise levels) being 2.2 degrees for bolts at 140 and 150 degrees.

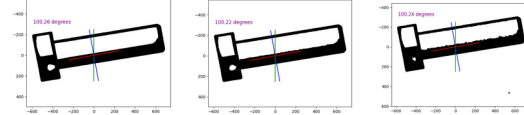


Fig. 2 Output Images (100 degrees): $\sigma = 0$, $\sigma = 20$, and $\sigma = 50$

Testing has shown that lighting conditions appear to cause this change in accuracy, particularly once the bolt image is rotated beyond 90 degrees. This rotation reduces bright pixels caused by light reflecting off the bolt's surface, leading to more pixels being converted to black during binarization. Consequently, a larger number of pixels is available for PCA application, improving the accuracy of the bolt orientation determination past 90 degrees.

IV CONCLUSIONS

In conclusion, the technique demonstrates potential for accurately identifying bolt orientation in low to highly noisy conditions with an maximum angle deviation of 3.5 degrees (averaged across all noise levels for a bolts at a known angle of 40 degrees). Future research hopes to extend this work into being used with with a YOLO system for a comparison with models that utilise orientated boundary boxes, evaluating both operational time and computational requirements.

REFERENCES

- 1 D.L. Pham and T.W. Chang, "A YOLO-based real-time packaging defect detection system," in *Procedia Computer Science*, vol. 217, pp. 886-894, 2023.
- 2 C.C. Wong, M.Y. Chien, R.J. Chen, H. Aoyama, and K.Y. Wong, "Moving object prediction and grasping system of robot manipulator," in *IEEE Access*, vol. 10, pp. 20159-20172, 2022.
- 3 H. Giberti, T. Abbattista, M. Carnevale, L. Giagu, and F. Cristini, "A methodology for flexible implementation of collaborative robots in smart manufacturing systems," in *Robotics*, vol. 11, no. 1, p. 9, 2022.
- 4 K. Nguyen, N. T. Huynh, P. C. Nguyen, K. D. Nguyen, N. D. Vo, and T. V. Nguyen, "Detecting objects from space: An evaluation of deep-learning modern approaches," *Electronics*, vol. 9, no. 4, pp. 583, 2020.
- 5 M. Zand, A. Etemad, and M. Greenspan, "Oriented Bounding Boxes for Small and Freely Rotated Objects," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-15, 2022, Art. no. 4701715, doi: 10.1109/TGRS.2021.3076050.
- 6 H. Abdi and L.J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433-459, 2010.
- 7 "Image Thresholding," OpenCV. Available: https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html. Accessed on: July 14, 2023.
- 8 G.B.P. da Costa, W.A. Contato, T.S. Nazare, J.E. Neto, and M. Ponti, "An empirical study on the effects of different types of noise in image classification tasks," *arXiv preprint arXiv:1609.02781*, 2016.