Abd Alrazzak Dabbour
Rabie Habib
Mariam Saii 🄲

OBJECT POSE ESTIMATION IN MONOCULAR IMAGE USING MODIFIED FDCM

Abstract

In this paper, a new method for multi-object detection and pose estimation in a monocular image is proposed based on the FDCM method. This method can detect an object with a high-speed running time even if the object was under partial occlusion or bad illumination. Additionally, it only requires a single template without any training process. In this paper, a new method (MFDCM) for 3D multi-object pose estimation in a monocular image is proposed, which is based on the FDCM method with major performance improvements in accuracy and running time. These improvements were achieved by using the LSD method instead of a simple edge detector (Canny detector), using an angular Voronoi diagram instead of calculating the 3D distance transform image, a distance transform image, and an integral distance transform image at each orientation. In addition, the search process in the proposed method depends on a line segment-based search instead of the sliding window search in the FDCM. As a result, the proposed method is more robust and much faster than the FDCM method, and the position, scale, and rotation are invariant. In addition, the proposed method was evaluated and compared to different methods (COF, HALCON, LINE2D, and BOLD) using a D-textureless dataset. The comparison results show that the MFDCM has the highest score among all of the tested methods (with a slight advantage from the COF and BLOD methods) while it was a little slower than LINE2D (which was the fasted method among the compared methods). Furthermore, it was at least 14-times faster than the FDCM in the tested scenarios. The results prove that the MFDCM is able to detect and 3D pose estimate of object in a clear or clustered background from a monocular image with a high-speed running time, even if the objects are under partial occlusion; this makes it robust and reliable for real-time applications.

1. Introduction

In many industrial application cases, the accuracy and automation level would be significantly improved if the objects were robustly detected and pose estimated using a monocular image.

Although there has been much research in the 3D object pose-estimation field, the problem of robustly recognizing and pose estimating a textured or untextured 3D object in a clustered scene from a monocular image in a reasonable amount of time is still unsolved. Besides, most of the previously proposed methods depend on two or more cameras or RGBD cameras, which are difficult to use because they are too expensive, the calibration process for multi-camera system is too complex, or the size of the equipment is too large.

In this paper, a new method is presented to detect and 3D pose estimate multi-objects in a monocular image that is based on the FDCM method with effective modifications.

To evaluate the proposed method, it was compared to the FDCM [21] in a clustered background and under partial occlusion. It was also compared to the COF [20], HALCON [31], LINE2D [11], and BOLD [33] methods using a D-textureless dataset [33] consisting of 9 different objects (with a total of 54 images for all objects). The objects appeared in different rotation angles with scale variants and while featuring cluttered backgrounds and occlusions.

2. Related works

Many methods for 3D object pose estimation from monocular images have previously been proposed. They can be classified into main groups, which are view-based methods, template-based methods, feature-based methods, and descriptor-based methods.

In view-based methods, the comparison result among precomputed 2D views of the object and the query image defines the position of the object. To accelerate the running time in these methods, the similar viewpoints will be merged [15, 17, 24, 28, 36]. Some of the view-based methods fixed the distance between the camera and the object (like the object exists on a table or conveyor belt) with a known constant distance from the camera. This leads to a reduction in the search space [34, 35].

The template-based methods depend on template-matching methods to detect and pose estimate an object in a monocular image [3, 11–13, 22, 23, 27, 29, 32, 37]. These use whole 2D projection images from various viewpoints, as their model templates successfully deals with textureless objects. However, they suffer from speed degradation when the numbers of templates are increased for covering a wider range of 3D object posing.

Some of the existing methods reduce the search space to a planar object scenario with scale and small affine transformation; in this case, they depend on contour-matching methods and contour descriptors to detect and pose estimate [7, 8, 16, 25, 30, 38, 39, 41].

The feature-based methods used feature-matching techniques between the query image and 3D object features, which are called feature-based methods [1, 5, 20, 26, 40]. By using the corresponding points of an extracted feature (like the gray value, edges, or the intersections of straight lines), the position of the object in 3D space is estimated. They must deal with a large number of corresponding points between the image features and the object features. In addition, the clustered scene will increase the number of extracted features rapidly and, thus, the search space and consumed time.

In descriptor-based methods, the artificial views of an object are created and used to train the classifier, which is used to detect and pose estimate the object in a query image by corresponding the process between the object descriptors and the descriptors derived from the query image [4, 9, 12, 14, 15, 19, 20, 24, 43]. The descriptor-based methods have a great advantage (which is outstanding performance in several scenarios), but they are restricted to the recognition of textured objects, as only then can meaningful descriptors be determined (while most of the industrial objects are untextured).

Besides the previous groups, recent methods have been used to detect and 6DOF pose estimate textureless objects in a monocular image [2, 6, 18, 42]; they use a convolutional neural network to detect and pose estimate. They provide high performance even if the object feature a clustered background or was under partial occlusion, but these methods need a training process that takes a long time.

3. Background and notations

3.1. FDCM

The DCM cost can be calculated based on the distance and direction of the edges between the query image and template as follows [14, 21]:

$$d_{DCM}(U, V) = \frac{1}{n} \min_{v_j \in V} (|u_i - v_j| + \lambda |\phi(u_i) - \phi(v_i)|), \tag{1}$$

where $U = \{ui\}$ and $V = \{vj\}$ are be the template sets and the query image edge maps respectively, the chamfer distance between U and V is given by the average of distances between each point $u_i \in U$ and its nearest edge in V, $n = |U|$ and λ weighting factors, and $\phi(x)$ means the edge direction at x.

To increase the speed of the matching process in the DCM, the directions of the edges will be quantized into k directions and using distance transform to simplify the calculation. Besides, a sliding window is used, and chamfer distance d_{DCM} is calculated at each window position. If the chamfer distance value is less than the thresholding value, then the current sliding window position will be stored as a detected target object in the query image [14, 21].

The FDCM has some improvements from the DCM (as follows): Line Segments: In the FDCM, to approximate the template and the query edge image to the line

segments, the RANSAC line-fitting method was used. Integral Distance Transform: Figure 1 describes the distance transform calculation steps in the DCM and FDCM as follows:

1) Line Segments: In the FDCM, to approximate the template and the query edge image to the line segments, the RANSAC line-fitting method was used.

2) Integral Distance Transform: Figure 1 describes the distance transform calculation steps in the DCM and FDCM as follows:

   - Depending on its direction, the edges in the image are grouped into k directions as shown in Figure 1b.
   - For each orientation, the corresponding 2D distance transform is calculated – Figure 1c.
   - By using orientation cost, the distance transform is updated – Figure 1d.
   - The final step is computing the integral distance transform for each orientation as shown in Figure 1e, then the d_{DCM} cost can be calculated as follows:

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{l_j ? L_u} |IDT3_v(e_j.\phi_j) - IDT3_v(s_j.\phi_j)| \tag{2}$$
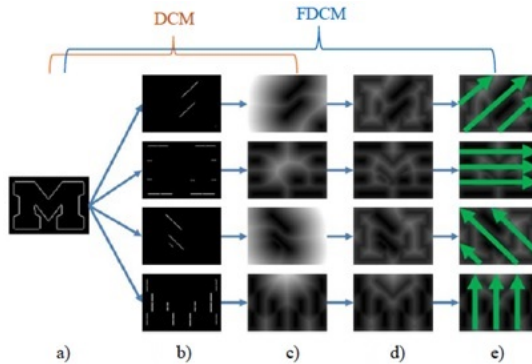

Figure 1. Steps of distance transform calculation using DCM and FDCM [21]

3) Region Search Optimization: To accelerate the searching process, the FDCM suggests that, if the chamfer distance cost at the current location is much larger than the thresholding value of the target object, then the nearby regions could be skipped. As a result, the search process will be greatly accelerated [21].

Although the FDCM is much faster than the DCM, it still has increased consuming time and cannot be used in real-time applications because it still has two time-consuming steps (the line fitting and distance transform).

In this paper, the FDCM was improved by modifying some steps and used to detect and pose estimate an untextured object in a monocular image based on its single template image.

4. Modified FDCM

Although the FDCM has many advantages over the DCM, it still has large consuming time and cannot be used in real-time applications. In this paper, the FDCM was modified and used to detect and pose estimate the object in a monocular image using a single template image.

In this paper, the FDCM was modified and improved using the following steps.

4.1. Replace the line fitting method

In the FDCM method, the line-fitting stage consists of the following steps [21]:

1) Detect edges in image using canny detector.
2) Several points and their orientation will be selected randomly from edge image to define lines.
3) Support for each edge includes its points that close to line are closed and connected to these points.
4) Line segment that has greatest support will be stored if it has enough points.
5) Repeat 1)–4) until there are not enough points to support any new line.

In previous steps, the edge-detection process generates lines randomly with a low probability of finding strong support. This causes many false attempts, which is very time-consuming.

In the proposed method, a line segment detector (LSD) [10] for edge extraction is used because it is faster and more robust than a normal edge detector (Canny edge detector).

4.2. Angular Voronoi diagram

The continuous angle is quantified into k discrete orientations. In each orientation, a distance transform image, a 3D distance transform image, and an integral distance transform image are calculated. So $3 \cdot k$ times pixel traversal of the query image is required. The author of FDCM suggests that 60 orientations are sufficient. However, 180 times pixel traversal is very time-consuming. In addition, the angle discretization will lose some angle accuracy [21].

In the proposed method, this step was replaced by calculating the Voronoi diagram. The angle of a pixel is assigned by the inclination angle of its closest line segment, and each angle value will be represented by different-colored regions as shown in Figure 2c.

The chamfer distance in the proposed method between the template and point set in the query image can be computed using the 2D distance transform tensor and angular cost as follows:

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{l_j \in L_u} \sum_{p_i \in l_j} [DT(p_i) + \lambda|\phi_j - VD(p_i)|], \qquad (3)$$

where p_i is the *ith* point from line l_j, ϕ_j the orientation of l_j, and DT and VD the distance transform image and Voronoi image, respectively. Only two tensors are necessary. By using the previous step, the proposed method is $k \cdot 3/2$ times faster than the FDCM.
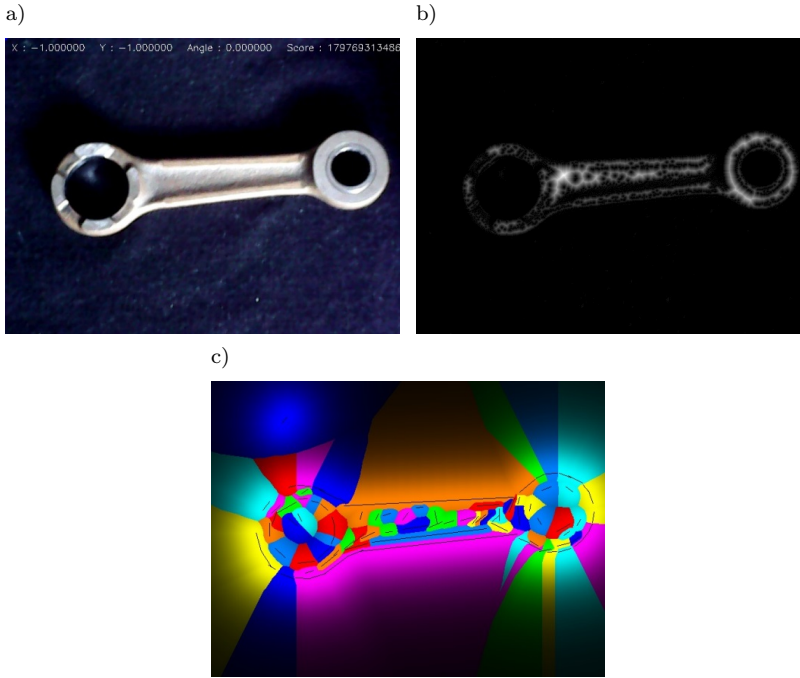
a)

b)

c)

Figure 2. Distance transform and angular Voronoi diagram: a) original image; b) distance transform; c) Voronoi diagram

4.3. Line-based matching

In the DCM method, the search process was very time-consuming; for an $r \cdot c$ query image with k directions, there are $r \cdot c \cdot k$ candidate locations (which could number in the millions). Despite the fact that the FDCM reduces this number by skipping some nearby regions, it is still very large.

In the proposed method, searching for the target object in the query image consists of the following steps:

1) The extracted lines from the template image and the query image by using the line segment detector will be stored and sorted from long to short, respectively.
2) For each sorted line in the query image, the template lines are scaled and rotated as the longest template line segment is aligned with the tested line in the query image.
3) Then, the cost is calculated between the template and the current position in the query image to decide whether there is a target in the current location.

4) If no target is found using the longest line segment, a new search begins using the second-longest line segment in the template image.

5) The process will continue to the fifth-longest line segment.

By using the previous steps, the number of tested locations drops from the millions to the thousands in addition to making the proposed method scale- and rotation-invariant.

Although the running time of the proposed method depends on the complexity of the background, it still much faster than the FDCM and DCM (even with clustered background scenarios).

The difference between the proposed method (MFDCM) and original method (FDCM) [21] can be summarized with the following points:

- Edge detector (Canny detector) in the FDCM was replaced by a line segment detector (LSD), which allows the proposed method to detect edges faster and more robustly than with a normal edge detector.

- In the FDCM method's 3D distance transform image, a distance transform image and integral distance transform image at each orientation should be calculated (which is very time-consuming), while in the MFDCM, this step was replaced by calculating the Voronoi diagram (which makes the MFDCM more accurate and much faster than the FDCM).

- To make proposed method (MFDCM) position, scale, and rotation invariant with high-speed running time, the line-based matching method was used instead of sliding window method.

5. Object pose estimation using MFDCM

To detect and pose estimate the object using the MFDCM method, only the template image from the top viewpoint is needed as illustrated in Figure 3b. As a result, the coordinates of the object's center and rotation angle are calculated as illustrated in Figure 3a.
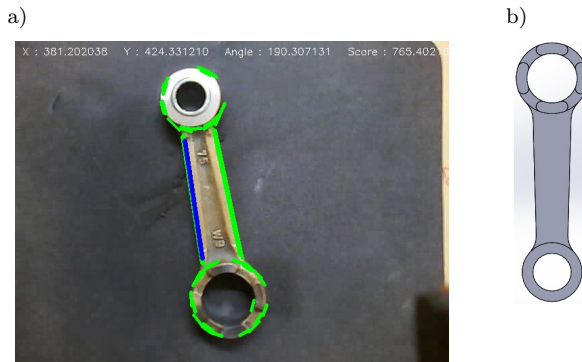

Figure 3. Distance transform and angular Voronoi diagram: a) pose estimation using MFDCM; b) template image

6. Results and discussion

To evaluate the proposed method, we compared our method with the FDCM using its code provided by the author [21] in an unclustered black background as well as a clustered background using a computer with Windows 7 46 bit, core i5, and 6 Gbytes memory. Table 1 shows a comparison of the results in the planer object with a black background as shown in Figure 4 and Figure 5.

Table 1

Consumed time in black background scenario

Algorithm	Consumed Time (ms)			
	Line Fitting	Distance Transform	Matching	Total
FDCM	70.25	52.32	20.67	143.24
Proposed(MFDCM)	3.25	3.5	2.19	8.94

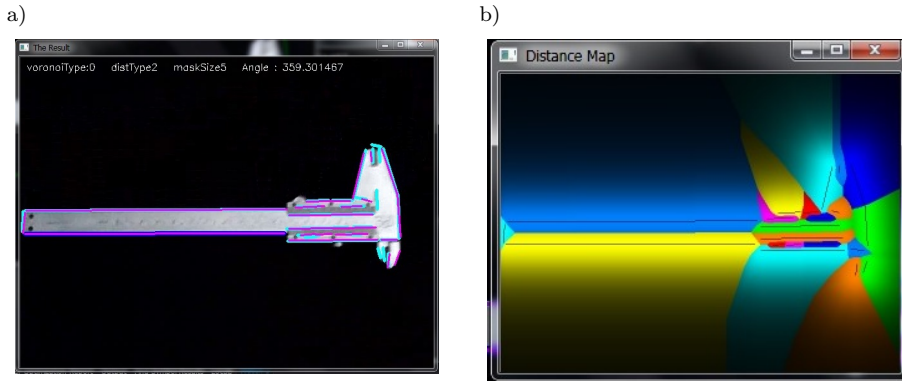a)   b)

Figure 4. Object 1 with black background: a) detected object; b) angular Voronoi diagram
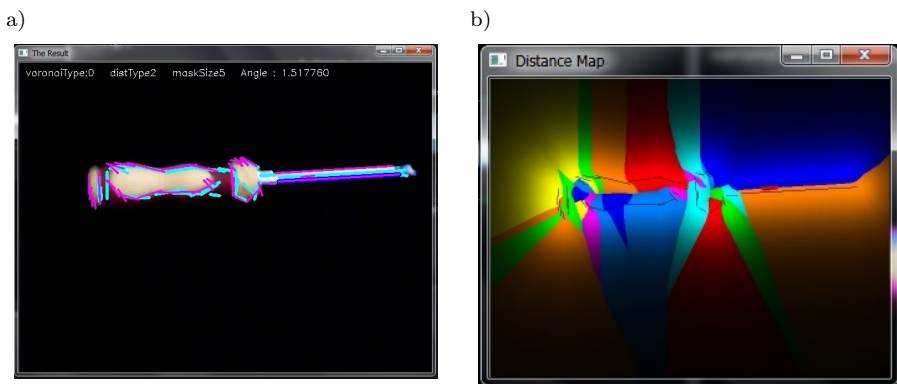
a)   b)

Figure 5. Object 2 with black background: a) detected object; b) angular Voronoi diagram

Table 2 shows a comparison of the results in the planar object with a clustered background and under partial occlusion as shown in Figure 6 and Figure 7.

Table 2

Consumed time with clustered background and under partial occlusion scenario

Algorithm	Consumed Time [ms]			
	Line Fitting	Distance Transform	Matching	Total
FDCM	74.27	59.51	30.63	164.41
Proposed (MFDCM)	2.18	4.6	4.57	11.35

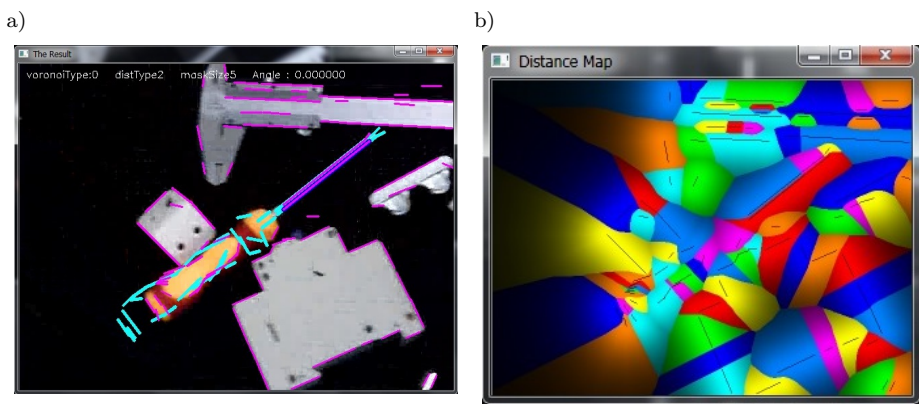a)   b)

Figure 6. Object 2 with clustered background and under partial occlusion: a) detected object; b) angular Voronoi diagram

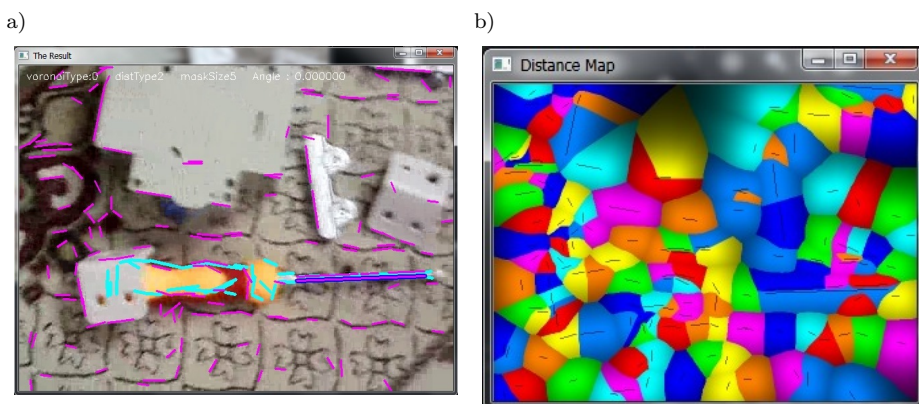a)   b)

Figure 7. Object 2 with very clustered background and under partial occlusion: a) detected object; b) angular Voronoi diagram

By comparing the consumed time of our proposed method (the MFDCM with Scale-Invariant) with the FDCM in each scenario (black background and clustered background), our method can detect and pose an estimation of a textured and untextured object approximately 14 times faster than the FDCM in both scenarios.

In addition, the proposed method was compared with the other four methods (the COF [20], HALCON [31], LINE2D [11], and BOLD [33] methods) using a D-textureless dataset [33] on the same PC (Core i5 2.4GHz, 6 GB RAM). The D-Textureless Dataset contains 54 images of 9 kinds of texture-less objects such as spanners and nippers; these appeared in various rotation angles and scales under cluttered backgrounds and occlusions. Examples of the detection results based on the MFDCM are presented in Figure 8 and Figure 9.

Table 3 shows the score and processing time when the false positives per image (FPPI) is 1. From a comparison of the results, the MFDCM (proposed) has the highest correct detection rate among all of the tested methods (with a small advantage over the COF and BLOD methods), while it outperformed LINE2D and HALCON. The MFDCM was a little slower than LINE2D (which was the fastest among the compared methods), and the MFDCM was much faster than BLOD and HALCON. According to the previous results, the MFDCM shows top-class performance in speed and accuracy.
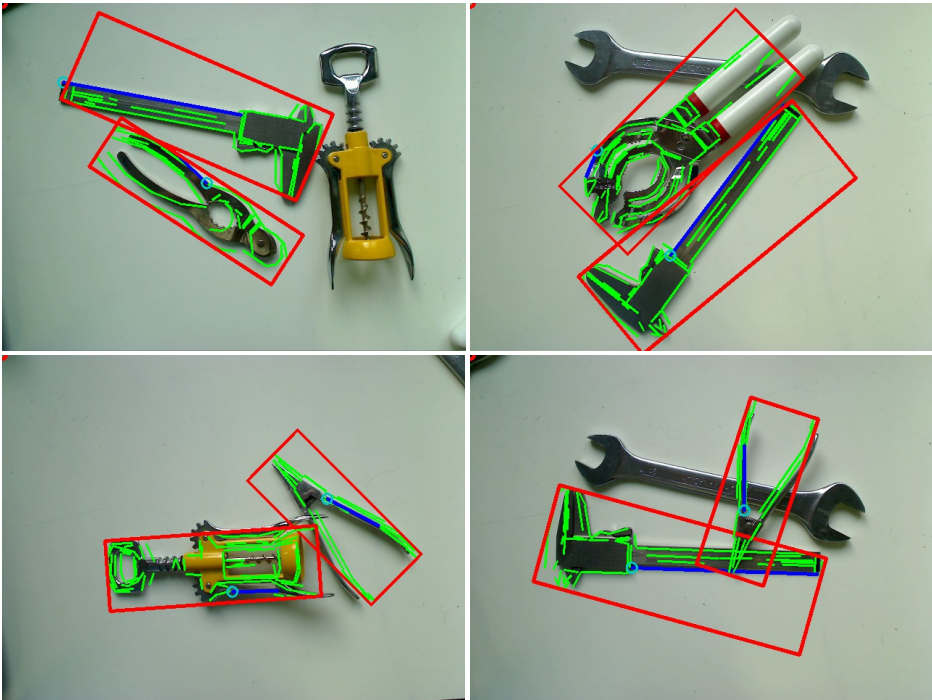

Figure 8. Example of pose estimation results on D-Textureless dataset
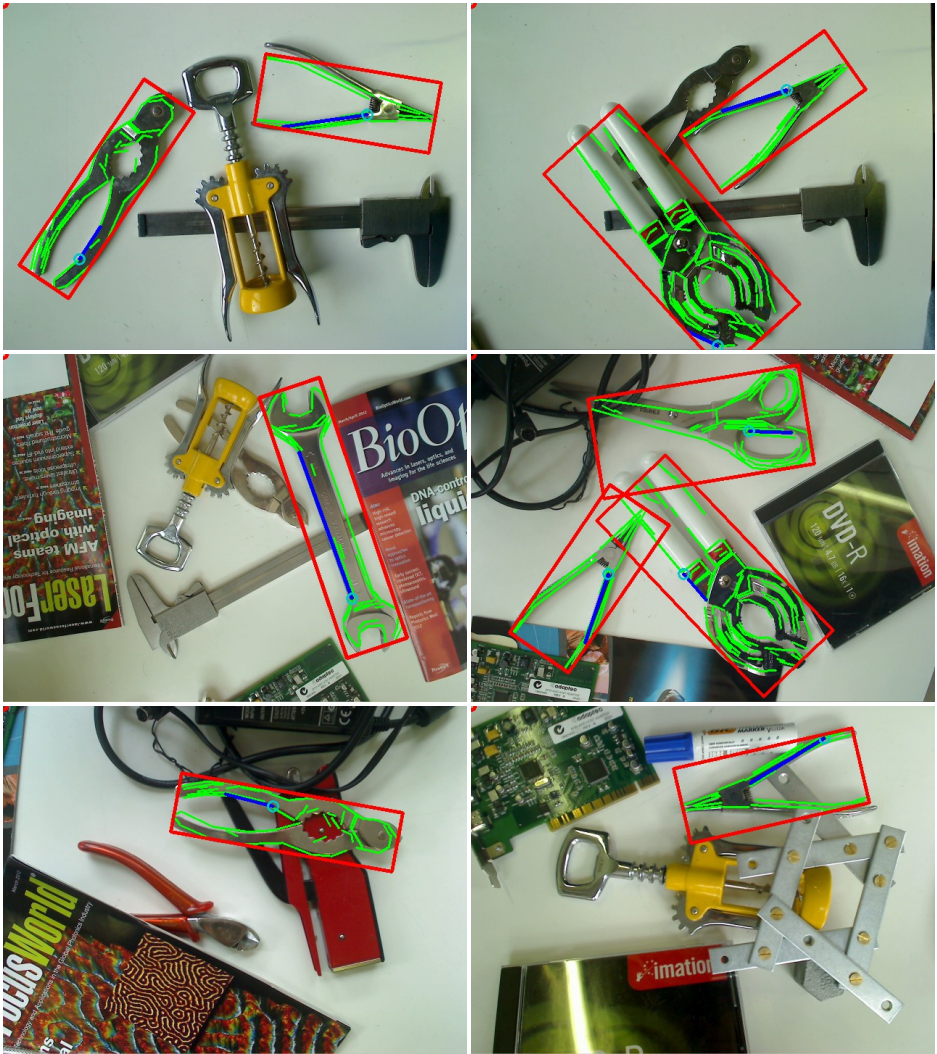
Figure 9. Example of pose estimation results on D-Textureless dataset. These were marked using green lines

Table 3
Consumed time in black background scenario

	HALCON [31]	LINE2D [11]	BOLD [33]	COF [20]	MFDCM (Proposed)
DR [%]	45.8	51.0	85.1	85.8	89.4
Time [ms]	380.5	47.8	177.9	53.9	51.2

Our proposed method (MFDCM) requires no training process – it only needs a template image and a global chamfer distance thresholding value. The proposed method can detect and pose estimate textured and untextured objects in a clear or cluttered background under partial occlusion and bad illumination with a fast running speed (approximately 40 fps), which allows the proposed methods to be used robustly in real-time applications.

7. Conclusion

In this paper, a robust and fast method for object 3D pose estimation in a monocular image was presented. It only requires a template image without any preprocessing steps.

The proposed method was faster than FDCM and more robustness because of major improvements achieved by following steps:

- Replacing Edge detector (Canny detector) with line segment detector (LSD), which allows the proposed method to detect edges faster and more robustly than with a normal edge detector.
- Using Voronoi diagram instead of calculating the integral distance transform at each orientation.
- Using the line-based matching method in the searching step instead of the sliding window method.

These improvements make the proposed method position, scale, and rotation invariant with high-speed running time, as the two most time-consuming steps in the FDCM method were simplified and accelerated.

Additionally, by comparing the proposed method with the COF, HALCON, LINE2D, and BOLD methods, it has the highest detection score among all of the tested methods (with a small advantage over the COF and BLOD methods), while it was a little slower than LINE2D the which was fasted among the compared methods).

The experimental results show that the proposed method is able to detect and pose estimate textured and untextured objects in a clear or clustered background from a monocular image with high-speed running time, even if the object is under partial occlusion or bad illumination; this makes the proposed method robust and reliable for real-time applications.

References

[1] Barrois B., Wöhler C.: 3D pose estimation based on multiple monocular cues. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007. https://doi.org/10.1109/CVPR.2007.383352.

[2] Brachmann E., Michel F., Krull A., Yang M.Y., Gumhold S., Rother C.: Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 49, pp. 3364–3372. IEEE, 2016. https://doi.org/10.1109/CVPR.2016.366.

[3] Bratanič B., Pernuš F., Likar B., Tomaževič D.: Real-time pose estimation of rigid objects in heavily cluttered environments, *Computer Vision and Image Understanding*, vol. 141, pp. 38–51, 2015. https://doi.org/10.1016/j.cviu.2015.09.002.

[4] Cai H., Werner T., Matas J.: Fast Detection of Multiple Textureless 3-D Objects. In: *Computer Vision Systems. ICVS 2013. Lecture Notes in Computer Science*, vol. 7963, Springer, Berlin, Heidelberg, pp. 103–112. 2013. https://doi.org/10.1007/978-3-642-39402-7_11.

[5] Damen D., Bunnun P., Calway A., Mayol-Cuevas W.: Real-time learning and detection of 3D texture-less objects: A scalable approach. In: *BMVC 2012 – Electronic Proceedings of the British Machine Vision Conference 2012*, pp. 1–12, 2012. https://doi.org/10.5244/C.26.23.

[6] Do T.T., Cai M., Pham T., Reid I.: Deep-6DPose: Recovering 6D Object Pose from a Single RGB Image, 2018. http://arxiv.org/abs/1802.10367.

[7] Fan B., Wu F., Hu Z.: Aggregating gradient distributions into intensity orders: A novel local image descriptor. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2377–2384, 2011. https://doi.org/10.1109/CVPR.2011.5995385.

[8] Fan B., Wu F., Hu Z.: Rotationally invariant descriptors using intensity order pooling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34(10), pp. 2031–2045, 2012. https://doi.org/10.1109/TPAMI.2011.277.

[9] Gálvez-López D., Salas M., Tardós J.D., Montiel J.M.: Real-time monocular object SLAM. In: *Robotics and Autonomous Systems*, vol. 75, pp. 435–449, 2016. https://doi.org/10.1016/j.robot.2015.08.009.

[10] Grompone Von Gioi R., Jakubowicz J., Morel J.M., Randall G.: LSD: a Line Segment Detector Rafael, *Image Processing On Line*, vol. 2, pp. 35–55, 2012. https://doi.org/10.5201/ipol.2012.gjmr-lsd.

[11] Hinterstoisser S., Cagniart C., Ilic S., Sturm P., Navab N., Fua P., Lepetit V.: Gradient response maps for real-time detection of textureless objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34(5), pp. 876–888, 2012. https://doi.org/10.1109/TPAMI.2011.206.

[12] Hinterstoisser S., Holzer S., Cagniart C., Ilic S., Konolige K., Navab N., Lepetit V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 858–865, 2011. https://doi.org/10.1109/ICCV.2011.6126326.

[13] Hinterstoisser S., Lepetit V., Ilic S., Holzer S., Bradski G., Konolige K., Navab N.: Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In: Lee K.M., Matsushita Y., Rehg J.M., Hu Z. (eds.) *Computer Vision – ACCV 2012*, pp. 548–562, 2013. https://doi.org/10.1007/978-3-642-37331-2_42.

[14] Imperoli M., Pretto A.: D²CO: Fast and Robust Registration of 3D Textureless Objects Using the Directional Chamfer Distance. In: Nalpantidis L., Krüger V., Eklundh JO., Gasteratos A. (eds.), *Computer Vision Systems. ICVS 2015. Lecture Notes in Computer Science*, vol. 9163, pp. 316–328, Springer, Cham, 2015. https://doi.org/10.1007/978-3-319-20904-3_29.

[15] Imperoli M., Pretto A.: Active Detection and Localization of Textureless Objects in Cluttered Environments, 2016. http://arxiv.org/abs/1603.07022.

[16] James S., Collomosse J.: Interactive video asset retrieval using sketched queries. In: *CVMP '14: Proceedings of the 11th European Conference on Visual Media Production*, pp. 1–8, 2014. https://doi.org/10.1145/2668904.2668940.

[17] Jaynes C., Hou J.: Temporal Registration using a Kalman Filter for Augmented Reality Applications. In: *Proceedings of Vision Iterface Conference Journal*, 2000.

[18] Kehl W., Manhardt F., Tombari F., Ilic S., Navab N.: SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2017 October, pp. 1530–1538. IEEE, 2017. https://doi.org/10.1109/ICCV.2017.169.

[19] Konishi Y., Hanzawa Y., Kawade M., Hashimoto M.: Fast 6D Pose Estimation from a Monocular Image Using Hierarchical Pose Trees. In: Leibe B., Matas J., Sebe N., Welling M. (eds.), *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*, vol. 9905, pp. 398–413, Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-46448-0_24.

[20] Konishi Y., Ijiri Y., Suwa M., Kawade M.: Textureless object detection using cumulative orientation feature. In: *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 1310–1313, 2015. https://doi.org/10.1109/ICIP.2015.7351012.

[21] Liu M.Y., Tuzel O., Veeraraghavan A., Chellappa R.: Fast directional chamfer matching. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1696–1703. IEEE, 2010. https://doi.org/10.1109/CVPR.2010.5539837.

[22] Liu M.Y., Tuzel O., Veeraraghavan A., Taguchi Y., Marks T.K., Chellappa R.: Fast object localization and pose estimation in heavy clutter for robotic bin picking, *International Journal of Robotics Research*, vol. 31(8), pp. 951–973, 2012. https://doi.org/10.1177/0278364911436018.

[23] Mörwald T., Zillich M., Vincze M.: Edge tracking of textured objects with a recursive particle filter. In: *19th International Conference on Computer Graphics and Vision, GraphiCon'2009 – Conference Proceedings*, pp. 96–103, 2009.

[24] Muñoz E., Konishi Y., Beltran C., Murino V., Del Bue A.: Fast 6D pose from a single RGB image using Cascaded Forests Templates. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4062–4069, 2016. https://doi.org/10.1109/IROS.2016.7759598.

[25] Nagarajan B., Rajathilagam B.: Object Shapes from Regular Curves through Sparse Representations, *Procedia Computer Science*, vol. 133, pp. 635–642, 2018. https://doi.org/10.1016/j.procs.2018.07.098.

[26] Pavlakos G., Zhou X., Chan A., Derpanis K.G., Daniilidis K.: 6-DoF Object Pose from Semantic Keypoints. In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. https://doi.org/10.1109/ICRA.2017.7989233.

[27] Peng X.: Combine color and shape in real-time detection of texture-less objects, *Computer Vision and Image Understanding*, vol. 135, pp. 31–48, 2015. https://doi.org/10.1016/j.cviu.2015.02.010.

[28] Phillips C.J., Lecce M., Daniilidis K.: Seeing glassware: From edge detection to pose estimation and shape recovery, *Robotics: Science and Systems*, vol. 12, 2016. https://doi.org/10.15607/rss.2016.xii.021.

[29] Rios-Cabrera R., Tuytelaars T.: Discriminatively trained templates for 3D object detection: A real time scalable approach. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2048–2055, 2013. https://doi.org/10.1109/ICCV.2013.256.

[30] Rios-Cabrera R., Tuytelaars T.: Boosting masked dominant orientation templates for efficient object detection, *Computer Vision and Image Understanding*, vol. 120, pp. 103–116, 2014. https://doi.org/10.1016/j.cviu.2013.12.008.

[31] Steger C.: Similarity Measures for Occlusion, Clutter, and Illumination Invariant Object Recognition. In: Radig B., Florczyk S. (eds.), *Pattern Recognition. DAGM 2001. Lecture Notes in Computer Science*, vol. 2191, pp. 148–154, Springer, Berlin, Heidelberg, 2001. https://doi.org/10.1007/3-540-45404-7_20.

[32] Tejani A., Tang D., Kouskouridas R., Kim T.K.: Latent-Class Hough Forests for 3D Object Detection and Pose Estimation. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds.), *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science*, vol. 8694, pp. 462–477, Springer, Cham, 2014. https://doi.org/10.1007/978-3-319-10599-4_30.

[33] Tombari F., Franchi A., Di L.: BOLD features to detect texture-less objects. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1265–1272. IEEE, 2013. https://doi.org/10.1109/ICCV.2013.160.

[34] Tsarouchi P., Matthaiakis S.A., Michalos G., Makris S., Chryssolouris G.: A method for detection of randomly placed objects for robotic handling, *CIRP Journal of Manufacturing Science and Technology*, vol. 14, pp. 20–27, 2016. https://doi.org/10.1016/j.cirpj.2016.04.005.

[35] Tsarouchi P., Michalos G., Makris S., Chryssolouris G.: Vision system for robotic handling of randomly placed objects, *Procedia CIRP*, vol. 9, pp. 61–66, 2013. https://doi.org/10.1016/j.procir.2013.06.169.

[36] Ulrich M., Wiedemann C., Steger C.: CAD-based recognition of 3D objects in monocular images. In: *2009 IEEE International Conference on Robotics and Automation*, pp. 1191–1198, 2009. https://doi.org/10.1109/robot.2009.5152511.

[37] Ulrich M., Wiedemann C., Steger C.: Combining scale-space and similarity-based aspect graphs for fast 3D object recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34(10), pp. 1902–1914, 2012. https://doi.org/10.1109/TPAMI.2011.266.

[38] Wei H., Yang C., Yu Q.: Contour segment grouping for object detection, *Journal of Visual Communication and Image Representation*, vol. 48, pp. 292–309, 2017. https://doi.org/10.1016/j.jvcir.2017.07.003.

[39] Wei H., Yang C., Yu Q.: Efficient graph-based search for object detection, *Information Sciences*, vol. 385–386, pp. 395–414, 2017. https://doi.org/10.1016/j.ins.2016.12.039.

[40] Wu J., Xue T., Lim J.J., Tian Y., Tenenbaum J.B., Torralba A., Freeman W.T.: Single Image 3D Interpreter Network. In: Leibe B., Matas J., Sebe N., Welling M. (eds.), *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*, vol. 9910, pp. 365–382, Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-46466-4_22.

[41] Xu X., Tian L., Feng J., Zhou J.: OSRI: A rotationally invariant binary descriptor, *IEEE Transactions on Image Processing*, vol. 23(7), pp. 2983–2995, 2014. https://doi.org/10.1109/TIP.2014.2324824.

[42] Zhao Z., Peng G., Wang H., Fang H.S., Li C., Lu C.: Estimating 6D Pose From Localizing Designated Surface Keypoints, 2018. http://arxiv.org/abs/1812.01387.

[43] Zhu M., Derpanis K.G., Yang Y., Brahmbhatt S., Zhang M., Phillips C., Lecce M., Daniilidis K.: Single image 3D object detection and pose estimation for grasping. In: *Proceedings – IEEE International Conference on Robotics and Automation*, pp. 3936–3943, IEEE, 2014. https://doi.org/10.1109/ICRA.2014.6907430.

Affiliations

Abd Alrazzak Dabbour
   Tishreen University, Department of Mechatronics Engineering, Lattakia, Syria, abddabbour@outlook.com

Rabie Habib
   Tishreen University, Department of Mechatronics Engineering, Lattakia, Syria, Rabie.habib967@yahoo.com

Mariam Saii
   Tishreen University, Department of Computer and Automatic Control Engineering, Lattakia, Syria, dr.mariam.saii@gmail.com, ORCID ID: https://orcid.org/0000-0002-5222-8176