sciendo

ISSN 1335-8871

MEASUREMENT SCIENCE REVIEW

Journal homepage: https://content.sciendo.com

Robust Pose Estimation by Fusing Partial Color and Depth Imagery

Mehmet Akif Alper

Department of Cybersecurity, College of Engineering and Technology, Eastern Michigan University, Ypsilanti, MI, USA, malper@emich.edu

Abstract: Pose estimation algorithms are an extensively studied research topic in the field of computer vision and machine learning. Even though many algorithms attempt to solve the problem, most algorithms are still not accurate enough to recover poses in real-world applications. Therefore, we have developed a new approach that utilizes depth cues and optical flow measurements that presents improved pose recovery in real-world pose estimation applications. We also present a camera calibration method that creates projection matrices for pose estimation from cameras, which enables angular comparison for relative pose estimates from two sensor systems positioned at different locations. We applied and tested the proposed algorithm in the laboratory settings and compared our findings with a commercial and a gold standard pose estimation system. Angular pose errors were reported.

Keywords: point cloud, transformation, relative pose, coherent point drift, Kinect II, Vicon

1. Introduction

Pose estimation algorithms are a challenging and widely studied research field to estimate pose, motion, and trajectory of target objects. Thanks to recent developments in digital technology, we are able to capture high resolution color images. Nevertheless, depth cameras are more expensive and have lower resolution, although they can acquire 3D point cloud data from the real world that can be used to find poses of objects in 6 degrees of freedom (6-DoF). Pose estimation algorithms are utilized in a wide range of vision applications such as robot navigation, augmented reality (AR), camera calibration, action recognition, and human-machine interaction [1]-[5]. There is no typical method that defines model and target similarity. A variety of methods can be used to calculate poses such as matching image features, point clouds, or deep learning methods. Pose estimation algorithms process model and target frames by using pre-calibrated camera systems and aim to estimate the 6-DoF accurately. Even though many researchers have proposed pose estimation algorithms, these algorithms are still unsolved due to local minima and false matches in real-world applications. Therefore, we have developed an algorithm that presents enhanced pose estimation and a method that calibrates with external camera sensors.

The accuracy of the relative pose can be very important for many real-world applications. Applications include robot vision, autonomous vehicles that are subject to noise and outliers, so that outliers can be eliminated to find accurate estimation. A robot in a spacecraft needs to complete active debris removal tasks in space, micro robot arms are utilized for medical operations that require high accuracy and sensitivity, autonomous vehicles drive passengers and are required to estimate the relative pose of nearby vehicles [6].

Our proposed algorithm finds and fuses the depth and flow measurements captured by the Kinect II. The relative pose is compared with the ground truth for evaluation. Microsoft Kinect II created a great impact on the computer vision and deep learning field, which provides RGB-D imaging and rough pose estimates for indoor scenarios. Raw RGB-D images are acquired by the Kinect II, which is released for gaming and presents an inexpensive and commercial human motion capture device, released to the market in 2014.

The rigid object pose defines finding the best alignment of the model and the target object image. The rigid transformation of an object can be quantified in terms of R and T. Color image-based methods propose pose estimations limited to 2D, depth can be used to upgrade estimation to higher dimensions. However, depth cameras are expensive and have lower resolution with respect to color. Another problem is that partial object frames can impede accuracy. Point cloud registration is one approach to solve the problem, as ICP is a widely studied approach [7] that refines the pose by seeking local minimum. Myronenko et al. [8] developed the coherent point drift (CPD) algorithm, their algorithm models point sets as a Gaussian mixture model (GMM). The

DOI: 10.2478/msr-2025-0033

Corresponding author: malper@emich.edu (M. A. Alper) ground truth pose is captured by Vicon. We upgraded the CPD algorithm by rejecting outlier depth points and projecting the depth onto color points fused with the optical flow. The proposed method, termed Flow-CPD, eliminates the noisy measurements that can reduce the performance of the algorithm. Due to the distinctiveness of real-world applications, we tested Flow-CPD with a mm-level pose estimation captured with Vicon.

In the remainder of our paper, we review related studies in Section 2 and explain the details of our relative pose estimation algorithm in Section 3. In Section 4, we present quantitative results and in Section 5, we explain our findings and future studies.

2. Related studies

Pose estimation algorithms can be divided into three main groups: Template based methods, feature based methods, and machine learning based methods. One solution to the pose estimation problem is the template matching method, in which the template is created by rendering a 3D shape model of an object. Template-based pose estimation algorithms such as ICP have been widely studied in the literature [7]. The ICP algorithm iteratively converges to a local minimum, and proposes high accuracy estimates in some cases. Upon reaching the local minimum distance, ICP computes pose estimates by determining the closest distance and computing spatial transformations for point sets. Myronenko [8] proposed a probabilistic registration method called the CPD algorithm. CPD finds the registration of point clouds by modeling one point cloud as a GMM and the other point cloud as a data point and finding the maximum GMM posteriori probability. Delavari et al. [9] utilized the mesh construction of objects and added new model parameters to the CPD algorithm. Their modified CPD algorithm was applied to medical liver data and achieved improved registration accuracy. Biber et al. [10] developed the normal distance transform (NDT). The NDT models the point cloud as a set of 2D normal distributions and the second scan of the NDT is defined as maximizing the sum that defines the score for the density of the second scan. LIDAR sensors are also widely used sensors that enable autonomous operation of vehicles. LIDAR sensors can take measurements over long distances where typical camera sensors cannot. In addition, LIDAR sensors can detect depth with high accuracy, as the accuracy of laser sensors is higher than that of depth cameras. Opromolla et al. [11] used LIDAR point clods to find the centroid of the LIDAR measurements and calculate the pose based on a defined correlation measure. Their algorithm requires template models and finds the pose for space robot applications. Picos et al. [12] use correlation filters to estimate the locations and orientation of the target frame by iteratively finding the highest correlation between the model and the target frames.

There are a variety of algorithms for feature-based pose estimation methods. The general idea is to estimate distinctive feature matches and descriptors from model and target frames that are expected to be robust to image deformations in an object, and then estimate the pose measures of the object by error minimization, voting scheme, etc. Feature-based pose estimation methods can be divided into local and global methods. To capture accurate poses, the image frames must have sufficient texture of the model and the target object of interest. Chen et al. [13] used optical flow measurements that help to find large displacements. Their algorithm finds the pose by combining template warping and using the scale invariant feature transform (SIFT) feature correspondences. Feature-based pose estimation methods can find local minima, which can lead to incorrect pose refinement. Contour-based methods are also widely studied in pose estimation algorithms, as contours can present accurate edge information about a model object. Leng et al. [14] proposed a pose estimation algorithm that extracts the model and target contours from a gray image and iteratively searches for a match until convergence. Schlobohm et al. [15] utilized the contours and proposed projected features that increased the accuracy of pose estimation. Their algorithm finds the pose through a global optimization method. Zhang et al. [16] proposed an algorithm that utilizes the shape and image contour. Their algorithm finds inliers, rejects outlier points intensively and finds the pose of the object. Similarly, Wang et al. [17] also used image contours and edge features. Their algorithm applies particle filter searches for improved matches. In this way, their algorithm produces robust pose estimations in cluttered conditions.

The CAD model-based methods capture the 3D environment and use CAD models for shape matching. They present a noiseless and ideal representation of the object model, which can be enhanced to pose estimation accuracy. CAD models allow the use of a whole part of the object model. He et al. [18] developed a template-based pose estimation algorithm that extracts key points from the CAD model and finds the pose by an error minimization method. Tsai et al. [19] integrated template matching and perspective-n-point (PnP) pose estimation, their algorithm extracts and matches image key points and can be used in AR applications. Song et al. [20] have developed a CAD model-based pose estimation algorithm, their pose estimation algorithm filters depth images to remove outliers, and random bin picking infers pose from RGB images.

Recently, numerous machine learning based pose estimation algorithms have been proposed. These methods need pre-training and present automatic segmentation and pose estimation. Machine learning based methods aim to learn feature descriptors or find the pose of the object with CNNs. Zeng et al. [21] developed a convolutional neural networks (CNN) based pose estimation algorithm for robot manipulators. The algorithm was implemented for a robot that can automatically pick and place tasks. Le et al. [22] proposed a CNN network that segments objects and applies the pose estimation task to robotic applications. Brachmann et al. [23] developed a pose estimation algorithm method using a random forest algorithm for pixel classification of RGBD frames. Deep learning based algorithms can also be trained with synthetic data [24]. They developed a series of convolutional layers to ensure sufficient encoding of the pose. Although learning based pose estimation methods have a high potential, they are limited in learning different geometric poses, invariances, and computational time.

3. METHOD

A. Calibration

We proposed an algorithm that uses depth and color and is integrated with the Kinect II, which acquires raw RGB-D measurements. Kinect II was calibrated with a checkerboard. Then the extrinsic camera parameters were calculated. Using the extrinsic camera parameters, the depth measurements were projected onto color imagery, resulting in smearing and outliers at the object boundaries as well as sparse depth measurements that need to be further processed for robust pose recovery.

B. Boundary estimation:

The proposed algorithm uses depth and color correspondences. Subsequently, the target objects need to be estimated and point clouds need on the object to be extracted. Depending on the application, deep learning networks for objects can be trained to detect objects. CNN algorithms consist of convolutional layers, pooling layers, activation layers, and fully connected layers. Convolutional layers apply a convolutional kernel to the image, which reduces the training complexity of the network. The pooling layer reduces the size of the region in the image. Activation layers apply mathematical operations to the image pixel values. The fully connected layer weighs and connects each neuron to the following layer. We used deep learning techniques for object estimation and a convolutional neural network-based object detector, so the single shot video object detector (SSVD) was used for object detection. The SSVD was trained with our test objects [25]. The SSVD is a fast detector that extracts multiscale object features along the object motion path using a pyramid network, and the CNN-based detector estimates the target objects based on the aggregated target object features. Then, the proposed algorithm finds the rough object boundary from the SSVD and the sharp object boundary required for accurate pose refinement. Then, the sharp object boundary is extracted using optical flow estimation [26], which provides motion estimates within the region of interest.

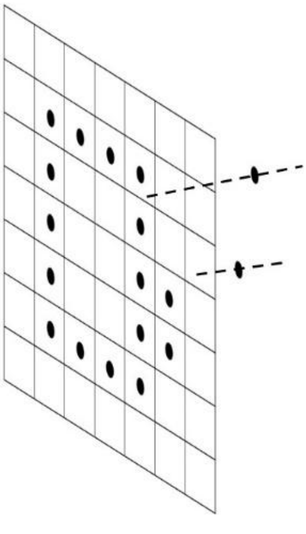

Fig. 1. Outlier depth points detected within the filter that showed extreme depth changes, so the filter eliminates extreme depths to create a sharp object surface.

Since the depth measurements are sparse, the proposed algorithm interpolates the depth on the color imagery, which may lead to outliers. Therefore, we applied a grid to the depth estimates that eliminates anomalous depth estimates, since the depth measurements should change linearly in neighboring pixels of the target object. Extreme depth shifts were eliminated within the grid, which filters and interpolates the sparse depth on the object, see Fig. 1.

C. Pose estimation:

Flow CPD refines the relative pose of the rigid object by fusing depth and optical flow. We compared our results with ICP and Vicon, which provide the gold standard for pose estimation by tracking a non-symmetric plate attached to the tracked object. Then the poses of the object can be tracked in samples. Vicon can provide ground truth pose estimates. The model point cloud data (P_m) and the target point cloud data (P_t) can be transformed using pose estimates (R and T), see (1). The CPD algorithm is an efficient algorithm to align two-point cloud sets. The CPD algorithm considers pose estimation as a probability density estimation problem. The points in the 3D world are defined by the 3-dimensional coordinate system (X, Y, Z), the color imagery correspondences (x, y) are defined in (2), and the optical flow function is defined by f. The flow vectors can be defined as x_f, y_f, see (3). One set is defined as GMM centroid and the other point set is defined as data points. The CPD algorithm calculates the spatial transformation between two-point clouds by maximizing the GMM likelihood function. It can calculate the pose of the objects by modeling the objects as rigid or non-rigid objects. The best matches between the model and the target imagery are found by calculating P_{mt}, where s defines the scale, and w defines the weighs of noise and outliers, see (4). Here, we rejected outliers and fused the optical flow with depth, which leads to improved pose recovery and is referred to as Flow-CPD. The accuracy of our algorithm can be compared with the ground truth pose. However, we need to calibrate Kinect II and Vicon, which is explained next.

$$P_t = R.P_m + T \tag{1}$$

$$(x, y) = P(X, Y, Z) \tag{2}$$

$$\left(x_f, y_f\right) = f(x, y) \tag{3}$$

$$P_{mt} = \frac{\exp^{-\frac{1}{2\sigma^2}\|P_t-(sRP_m+T)\|^2}}{\sum \exp^{-\frac{1}{2\sigma^2}\|P_t-(sRP_m+T)\|^2} + (2\pi\sigma^2)^{0.5.D}\dfrac{w.m}{(1-w)t}} \tag{4}$$

D. Calibration of Kinect and Vicon:

We have compared the pose estimates with the Vicon. Since the ground truth pose is captured by the gold standard tracker (Vicon), the pose estimated by the Kinect was calibrated with the Vicon. The pose changes are the same at all locations, but the pose is transferred differently in the 3-axis. Multiple pose estimates can be used to find a calibration matrix.

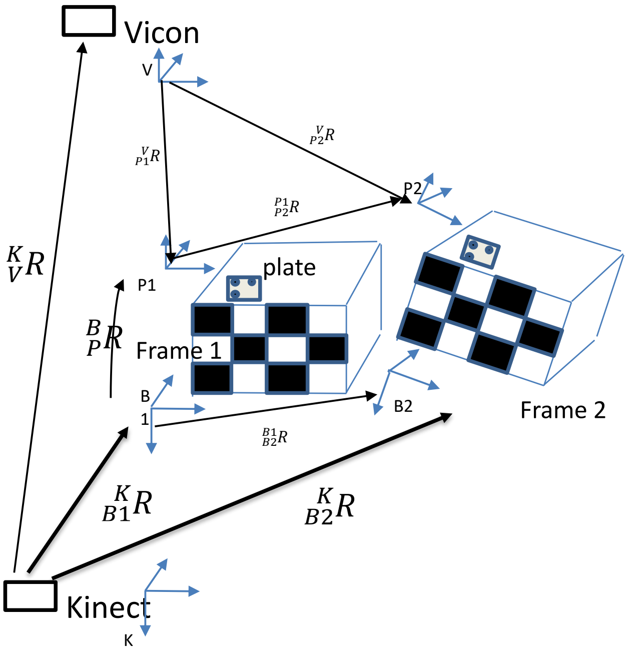

Fig. 2. The calibration framework and the transformations between the coordinate systems have been specified.

In the case of a rigid object rotation observed with respect to 2 cameras with different orientations, the poses of the cameras can be identified as R_1 and R_2. Then the relationship between the poses can be formulated, see (5).

$$ {}^{k}_{b1}R^{T}\, {}^{k}_{b2}R\, {}^{b}_{p}R\, {}^{v}_{p2}R^{T}\, {}^{v}_{p1}R\, {}^{b}_{p}R^{T} = I \tag{5} $$

We can simplify the above equation. Therefore, we noted the R_x calibration matrix which transforms the pose from R_1 to R_2. Then we create (5) by tracing the coordinate system in Fig. 2, and we can define (6) and (7). The transformation between the base of the plate and Kinect's reference coordinates can be simplified, see (8). Multiplications of the poses by the transpose of the same amount of rotation equals to identity matrix I, see (9), and can be redefined with the relative pose (10).

$$ R_1 = {}^{k}_{b1}R^{T}\, {}^{k}_{b2}R \tag{6} $$

$$ R_2 = {}^{v}_{p1}R^{T}\, {}^{v}_{p2}R \tag{7} $$

$$ R_x = {}^{b}_{p}R \tag{8} $$

$$ R_1\, R_x\, R_2^T\, R_x^T = I \tag{9} $$

$$ R_1 = R_x\, R_2\, R_x^T \tag{10} $$

The same physical rotation R_θ can be quantified from two different cameras. We can decompose the rotation into U_1, U_2, see (11) and (12).

$$ U_1\, R_1\, U_1^T = R_\theta \tag{11} $$

$$ U_2\, R_2\, U_2^T = R_\theta \tag{12} $$

If you write $R_\theta$with two derivations, which are defined in different matrices, see (13) and (14).

$$ U_1\, R_1\, U_1^T = U_2\, R_2\, U_2^T \tag{13} $$

$$ R_1 = U_1^T\, U_2\, R_2 (U_1^T\, U_2)^T \tag{14} $$

Then the calibration matrix ($R_x = {}^{b}_{p}R$) can be derived, and the rotation can be decomposed, see (15).

$$ R_x = {}^{b}_{p}R = U_1^T\, U_2 \tag{15} $$

We can find ${}^{K}_{V}R$ and compare the pose estimations of Vicon and Kinect directly with ${}^{b}_{p}R$, see (16):

$$ {}^{K}_{V}R = {}^{K}_{b1}R\, {}^{b}_{p}R\, {}^{b}_{V}R \tag{16} $$

The calibration matrices (${}^{b}_{p}R, {}^{K}_{V}R$) give the relative pose difference between the camera orientations for the 3-dimensional rotation angles θ_1 and θ_2 for the rotation matrices: R_1, R_2, see (17) and (18).

$$ \theta_1 = [\alpha_1, \beta_1, \gamma_1] \tag{17} $$

$$ \theta_2 = [\alpha_2, \beta_2, \gamma_2] \tag{18} $$

Angular rotations can be formulated as follows, see (19), (20), and (21).

$$ R_1 = \mathrm{Rot}\, (\varDelta_1, \theta_1) \tag{19} $$

$$ R_2 = \mathrm{Rot}\, (\varDelta_2, \theta_2) \tag{20} $$

$$ R_1 = R_x\, R_2\, R_x^T \tag{21} $$

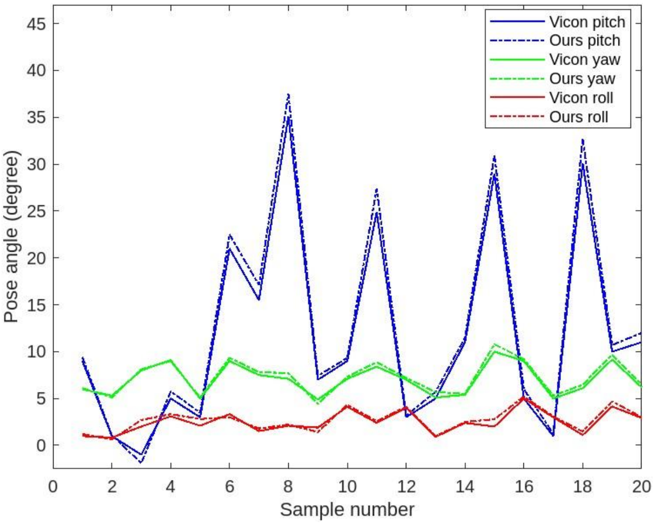

Fig. 3. After calibration, Vicon and the pose estimates show close alignment.

The calibration matrix R_x transforms the same amount of rotation from R_1 to R_2 with 2π mod. Camera poses can be transformed using R_x, see (21). In this way, the relative poses of two differently oriented cameras can be directly compared using the calibration matrices ${}_{p}^{b}R, {}_{V}^{K}R$. To compare the relative pose estimates of cameras, we only need the calibration matrix ${}_{p}^{b}R$. Using the calibration matrices ${}_{p}^{b}R, {}_{V}^{K}R$, we can compare the pose estimates and the pose estimates plotted with respect to Vicon, see Fig. 3.

4. Experiments

We tested the proposed algorithm to evaluate R and T. Laboratory tests using a Vicon motion capture device and a public pose dataset were also used to evaluated the algorithms. The results of the Flow-CPD algorithm were also compared with the Vicon. Flow-CPD provided a good alignment with the pose estimated by Vicon. Model and target objects are shown in Fig. 4. CPD finds the pose from model to target, see pose matches in Fig. 5. Pose matches are given for Flow-CPD, see Fig. 6. Model point clouds are transformed to target point cloud data using the estimated pose parameters. We have tested the pose estimation methods in a series of experiments and published the results.

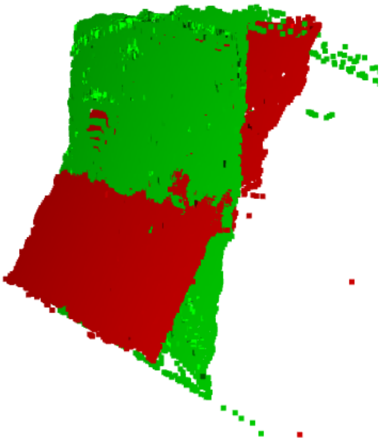

Fig. 4. Rigid object was rotated by 15 degrees, model and target point clouds are given.

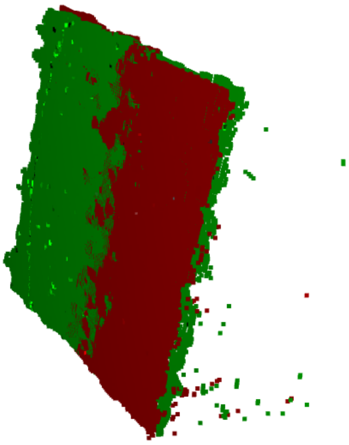

Fig. 5. Pose is calculated using the CPD algorithm and the point clouds were aligned based on estimates leading to errors.

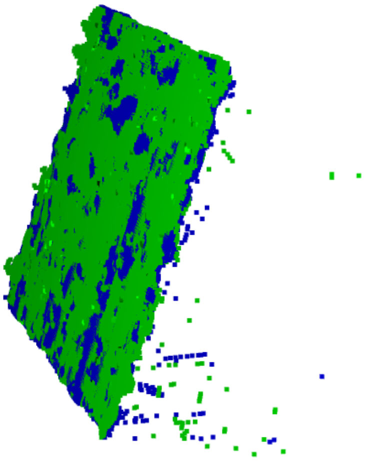

Fig. 6. Pose is calculated using Flow-CPD and the point clouds were aligned based on the estimates of Flow-CPD. It can be seen that the model and target point clouds interact with each other, showing improved accuracy.

5. Subject & methods

We first tested the Flow-CPD algorithm to evaluate rotational angles and translations. The results have shown that the Flow-CPD algorithm computes the pose of the rigid object with higher accuracy than CPD and shows close alignment with the ground truth. Instead of using depth only measurements, Flow-CPD fuses depth and optical flow, resulting in improved pose matches when tested with the Vicon. Pose errors are compared and the CPD algorithm has a 3.32 degree mean square error (MSE), but Flow-CPD shows improved pose recovery and an MSE of 0.76 degrees. Flow-CPD provides *low-cost and high-accuracy* pose estimates by upgrading Microsoft Kinect II.

6. Conclusion

In this study, the proposed method demonstrates improved accuracy in relative pose estimation. The Flow-CPD algorithm is shown to be a reliable, high-precision tracking approach for indoor environments that overcomes the inherent limitations of the Kinect II sensor. In addition, a calibration framework is introduced that enables external calibration across multiple viewpoints. The Flow-CPD approach also shows potential for adaptation to multi-robot pose estimation and cooperative tasks in large-scale production lines, which will be further investigated in future work.

References

[1] Feng, B., Liu, Z., Zhang, H., Fan, H. (2024). Research on the measurement system and remote calibration technology of a dual linear array camera. *Measurement Science Review*, 24 (3), 105-112. https://doi.org/10.2478/msr-2024-0015

[2] Murphy-Chutorian, E., Trivedi, M. M. (2010). Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. *IEEE Transactions on Intelligent Transportation Systems*, 11 (2), 300-311. https://doi.org/10.1109/TITS.2010.2044241

[3] Yang, T., Zhao, Q., Wang, X., Zhou, Q. (2018). Sub-pixel chessboard corner localization for camera calibration and pose estimation. *Applied Sciences*, 8 (11), 2118. https://doi.org/10.3390/app8112118

[4] Zhao, R., Ali, H., van der Smagt, P. (2017). Two-stream RNN/CNN for action recognition in 3D videos. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. https://doi.org/10.1109/IROS.2017.8206288

[5] Andriluka, M., Roth, S., Schiele, B. (2010). Monocular 3D pose estimation and tracking by detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. https://doi.org/10.1109/CVPR.2010.5540156

[6] Kalaitzakis, M., Cain, B., Carroll, S., Ambrosi, A., Whitehead, C., Vitzilaios, N. (2021). Fiducial markers for pose estimation. *Journal of Intelligent & Robotic Systems*, 101, 71. https://doi.org/10.1007/s10846-020-01307-9

[7] Besl, P. J., McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 (2), 239-256. https://doi.org/10.1109/34.121791

[8] Myronenko, A., Song, X. (2010). Point set registration: Coherent point drifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (12), 2262-2275. https://doi.org/10.1109/TPAMI.2010.46

[9] Delavari, M., Foruzan, A. H., Chen, Y.-W. (2019). Accurate point correspondences using a modified coherent point drift algorithm. *Biomedical Signal Processing and Control*, 52, 429-444. https://doi.org/10.1016/j.bspc.2017.02.009

[10] Biber, P., Strasser, W. (2003). The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*. IEEE, 3, 2743-2748. https://doi.org/10.1109/IROS.2003.1249285

[11] Opromolla, R., Fasano, G., Rufino, G., Grassi, M. (2015). A model-based 3D template matching technique for pose acquisition of an uncooperative space object. *Sensors*, 16 (3), 6360-6382. https://doi.org/10.3390/s150306360

[12] Picos, K., Diaz-Ramirez, V. H., Kober, V., Montemayor, A. S., Pantrigo, J. J. (2016). Accurate three-dimensional pose recognition from monocular images using template matched filtering. *Optical Engineering*, 55 (6), 063102. https://doi.org/10.1117/1.OE.55.6.063102

[13] Chen, S., Liang, L., Liang, W., Foroosh, H. (2016). 3D pose tracking with multitemplate warping and SIFT correspondences. *IEEE Transactions on Circuits and Systems for Video Technology*, 26 (11), 2043-2055. https://doi.org/10.1109/TCSVT.2015.2452782

[14] Leng, D. W., Sun, W. D. (2011). Contour-based iterative pose estimation of 3D rigid object. *IET Computer Vision*, 5 (5), 291-300. https://doi.org/10.1049/iet-cvi.2010.0098

[15] Schlobohm, J., Pösch, A., Reithmeier, E., Rosenhahn, B. (2016). Improving contour based pose estimation for fast 3D measurement of free form objects. *Measurement*, 92, 79-82. https://doi.org/10.1016/j.measurement.2016.05.093

[16] Zhang, X., Jiang, Z., Zhang, H., Wei, Q. (2018). Vision-based pose estimation for textureless space objects by contour points matching. *IEEE Transactions on Aerospace and Electronic Systems*, 54 (5), 2342-2355. https://doi.org/10.1109/TAES.2018.2815879

[17] Wang, B., Zhong, F., Qin, X. (2019). Robust edge-based 3D object tracking with direction-based pose validation. *Multimedia Tools and Applications*, 78 (9), 12307-12331. https://doi.org/10.1007/s11042-018-6727-5

[18] He, Z., Jiang, Z., Zhao, X., Zhang, S., Wu, C. (2020). Sparse template-based 6-D pose estimation of metal parts using a monocular camera. *IEEE Transactions on Industrial Electronics*, 67 (1), 390-401. https://doi.org/10.1109/TIE.2019.2897539

[19] Tsai, C.-Y., Hsu, K.-J., Nisar, H. (2018). Efficient model-based object pose estimation based on multi-template tracking and PnP algorithms. *Algorithms*, 11 (8), 122. https://doi.org/10.3390/a11080122

[20] Song, K.-T., Wu, C.-H., Jiang, S.-Y. (2017). CAD-based pose estimation design for random bin picking using a RGB-D camera. *Journal of Intelligent & Robotic Systems*, 87, 455-470. https://doi.org/10.1007/s10846-017-0501-1

[21] Zeng, A., Yu, K.-T., Song, S., Suo, D., Walker, E., Rodriguez, A. (2017). Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. https://doi.org/10.1109/ICRA.2017.7989165

[22] Le, T., Hamilton, L., Torralba, A. (2017). Benchmarking convolutional neural networks for object segmentation and pose estimation. In *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE. https://doi.org/10.1109/AIPR.2017.8457943

[23] Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C. (2014). Learning 6D object pose estimation using 3D object coordinates. In *Computer Vision - ECCV 2014*. Springer, LNIP 8690, 536-551. https://doi.org/10.1007/978-3-319-10605-2_35

[24] Su, Y., Rambach, J., Pagani, A., Stricker, D. (2021). SynPo-Net—Accurate and fast CNN-based 6DoF object pose estimation using synthetic training. *Sensors*, 21 (1), 300. https://doi.org/10.3390/s21010300

[25] Deng, J., Pan, Y., Yao, T., Zhou, W., Li, H., Mei, T. (2020). Single shot video object detector. *IEEE Transactions on Multimedia*, *23*, 846-858. https://doi.org/10.1109/TMM.2020.2990070

[26] Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2758-2766. https://doi.org/10.1109/ICCV.2015.316

Received September 29, 2024
Accepted September 16, 2025