A Benchmark Dataset for 6DoF Object Pose Tracking

Po-Chen Wu*
Media IC & System Lab
National Taiwan University

Yueh-Ying Lee†
Media IC & System Lab
National Taiwan University

Hung-Yu Tseng‡
Media IC & System Lab
National Taiwan University

Hsuan-I Ho§
Media IC & System Lab
National Taiwan University

Ming-Hsuan Yang¶
Vision and Learning Lab
University of California at Merced

Shao-Yi Chien‖
Media IC & System Lab
National Taiwan University

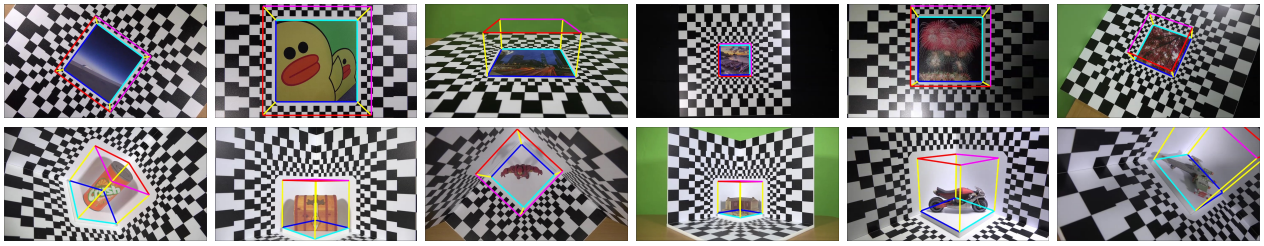


Figure 1: Images of 2D (top row) and 3D objects (bottom row) in our benchmark dataset with 6DoF pose ground-truth notation. The proposed benchmark dataset contains 690 color and depth videos of various textured and geometric objects with over 100,000 frames. The recorded sequences also contain image distortions for performance evaluation in real-world scenarios.

Abstract

Accurately tracking the six degree-of-freedom pose of an object in real scenes is an important task in computer vision and augmented reality with numerous applications. Although a variety of algorithms for this task have been proposed, it remains difficult to evaluate existing methods in the literature as oftentimes different sequences are used and no large benchmark datasets close to real-world scenarios are available. In this paper, we present a large object pose tracking benchmark dataset consisting of RGB-D video sequences of 2D and 3D targets with ground-truth information. The videos are recorded under various lighting conditions, different motion patterns and speeds with the help of a programmable robotic arm. We present extensive quantitative evaluation results of the state-of-the-art methods on this benchmark dataset and discuss the potential research directions in this field. The proposed benchmark dataset is available online at media.ee.ntu.edu.tw/research/OPT.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Benchmarking I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, Augmented, and Virtual Realities

1 Introduction

In recent years, numerous methods for six degree-of-freedom (DoF) object pose recovering have been developed and applied to a wide range of problems including robotic manipulation, augmented reality (AR), and human-computer interaction. Existing algorithms for recovering the 6DoF pose (i.e., rotation and translation) of an object can be broadly categorized into three main approaches:

Direct approaches [9, 21]. These approaches address the problem by finding the best fit from numerous candidates based on a holistic template or appearance matching. The corresponding pose of the best candidate is considered as the estimation result.

Feature-based approaches [5]. The core idea is to first establish a set of feature correspondences between the target object and projected camera frame [12, 23]. Outliers are then removed to obtain reliable feature pairs [7], and the final pose is computed with the Perspective-*n*-Point (P*n*P) algorithms [24, 6]. In contrast to direct methods, the performance of feature-based methods depends on whether both features can be extracted and matched well.

Learning-based approaches [20, 2, 3]. These methods learn an abstract representation of an object from a set of images captured from different viewpoints, from which the pose of the target in a new input frame is determined. While feature-based and direct methods are more effective for textured and non-occluded objects respectively, learning-based approaches have shown the potential to track poses of objects with diverse textures under partial occlusion.

Real-time pose tracking can be accomplished by leveraging the information obtained from previous frames [15]. In addition, the pose estimation task can be accelerated by exploiting a small search range within the camera viewpoint or reducing the number of pose candidates. To prevent pose jittering during the tracking process, which is indispensable especially in AR applications, further pose refinement should be performed.

To evaluate existing pose estimation algorithms, a number of benchmark datasets have been proposed [9, 4, 20, 2, 10, 16]. However, there are two main issues that need to be addressed. First, while the datasets are mainly designed for single-frame based pose estimation, most images do not contain distortions (e.g., motion blur caused by different object motions) that are crucial for performance evaluation for real-world scenarios. Second, the camera

*e-mail: pcwu@media.ee.ntu.edu.tw
†e-mail: yylee@media.ee.ntu.edu.tw
‡e-mail: hytseng@media.ee.ntu.edu.tw
§e-mail: hiho@media.ee.ntu.edu.tw
¶e-mail: mhyang@ucmerced.edu
‖e-mail: sychien@ntu.edu.tw

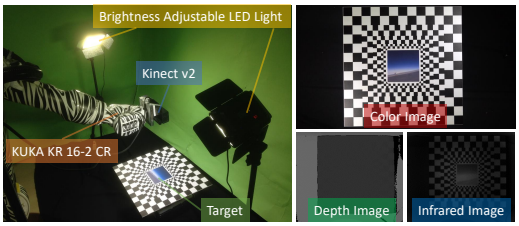


Figure 2: Sequences are recorded with a Kinect v2 sensor mounted on a programmable robotic arm. Note that we normalize the intensity of the depth image in this figure for clarity.

trajectories in most datasets are not carefully designed (i.e., free-style motion), which do not allow detailed analysis for specific situations. Most importantly, it is of great interest for fields of computer vision and augmented reality to develop an extensive benchmark dataset for thorough performance evaluation of 6DoF pose tracking in real-world scenarios.

In this work, we propose a large-scale benchmark dataset of RGB-D video sequences for both 2D and 3D objects with ground-truth information, as shown in Figure 1. The proposed benchmark dataset contains 690 color and depth videos of varying degrees of textured and geometric objects with over 100,000 frames. These videos are annotated with different imaging conditions (i.e., translation, forward and backward, in-plane rotation, out-of-plane rotation, flashing light, moving light, and arbitrary motion) and speed recorded with a Kinect v2 sensor mounted on a programmable robotic arm. A 3D printer renders the objects in the benchmark dataset with distinct textures. The ground-truth poses are computed using a designed checkerboard and checkerbox for 2D and 3D objects. Due to the global-shutter infrared camera with fast shutter speed from the Kinect v2 sensor, we can annotate the ground-truth poses by leveraging the clear infrared images under fast motions.

The contributions of this work are summarized below:

Benchmark dataset. We design a benchmark dataset for 6DoF object pose tracking. It consists of 690 videos under seven varying conditions with five speeds. It is a large dataset where images are acquired from a moving camera for performance evaluation of both 2D and 3D object pose tracking algorithms. Furthermore, the proposed dataset can also be used in other computer vision tasks such as 3D feature tracking and matching.

Performance evaluation. Each pose tracking method is extensively evaluated and analyzed using more than 100,000 frames including both 2D and 3D objects. Since the state-of-the-art simultaneous localization and mapping (SLAM) methods [14, 22] are able to track and relocalize camera pose in real time, we also evaluate these approaches by adapting them to object pose tracking scenarios. We present the extensive performance evaluation of the state-of-the-art methods using the proposed benchmark dataset.

2 Related Work

Numerous datasets have been developed to evaluate algorithms in areas related to 3D pose estimation and tracking. The dataset presented by Lieberknecht *et al.* [11] contains 40 sequences of eight different textured 2D objects and five unconstrained motions (e.g., zoom-in and translation). A dataset with 96 videos from six textured planar targets and varying geometric distortions as well as lighting conditions is constructed by Gugglitz *et al.* [8]. The homography transformation parameters are provided in this dataset. Since a rolling-shutter camera is used, it may be difficult to obtain the exact 6DoF pose from the homography parameters when the relative motion is significant.

Hinterstoisser *et al.* [9] construct a dataset of 18,000 images with 15 texture-less 3D objects, which is further extended for multi-

Low Normal Rich

Figure 3: 2D objects with low (*wing*, *duck*), normal (*city*, *beach*), and rich (*maple*, *firework*) texture.

Simple Normal Complex

Figure 4: 3D objects with simple (*soda*, *chest*), normal (*ironman*, *house*), and complex (*bike*, *jet*) geometry.

instance 3D object detection and pose estimation [20]. Similarly, a dataset with 20 textured and textureless objects is proposed [2] where each one is recorded under three different lighting conditions. For the above-mentioned datasets, both color and depth images are recorded using handheld Kinect v1 cameras. The target objects are attached to a planar board surrounded with fiducial markers, which provide the corresponding poses. Since markers cannot be accurately localized in a blurry image, the recorded targets need to be static in front of the camera, and thus these datasets do not contain distortions that are crucial for performance evaluation of pose tracking in real-world scenarios. The real pose is also arduous to compute because of the rolling-shutter effect which will change the appearance of markers whenever there exists some camera movement. Different from using fiducial markers, the ground-truth object poses in the datasets [16, 10] are manually labeled and less accurate. Even the poses in [10] are further refined by the iterative closest point (ICP) method, the estimates are not accurate due to noisy measurements of depth values. The dataset proposed by Choi and Christensen [4] consists of four synthetically generated sequences of four models. The main drawback of this synthetic dataset is the lack of distortions in both RGB-D images and motion blurs. We summarize the characteristics of existing benchmark datasets for pose recovering in Table 1.

3 Proposed Benchmark Dataset

3.1 Acquiring Images

The color, depth, and infrared images of each sequence are obtained from a Kinect v2 sensor mounted on a programmable KUKA KR 16-2 CR robot arm, as illustrated in Figure 2. The robotic arm, which has six axes and repeatability of 0.05 mm, can be programmed to move in complex trajectories precisely. Each 2D object shown in Figure 3 is a printed pattern with size 133.6×133.6 mm^2 surrounded by a checkerboard glued on an acrylic plate. Each 3D object shown in Figure 4 is generated by a 3D printer with resolution 300×450 dpi and 0.1 mm layer thickness. The length, width, and height of 3D objects illustrated in Figure 4 are in the ranges of (57.0, 103.6), (57.0, 103.6), and (23.6, 109.5), respectively in mm.

The object motions in the proposed benchmark dataset are (regarded as moving object rather than the camera):

Translation. An object moves along a circle parallel to the camera sensor plane with motion blur in all directions.

Forward and Backward. An object moves forward first and then backward.

In-plane Rotation. An object rotates along an axis perpendicular to the camera sensor plane.

Table 1: Benchmark datasets for object pose estimation. Using a programmable robotic arm, we can record images under different motion patterns and different speed. The recorded sequences hence contain different distortions that are crucial for performance evaluation of pose tracking algorithms for real-world scenarios. The proposed object pose dataset is also the only one where color and depth image sequences are recorded by a Microsoft Kinect v2 sensor.

Benchmark	Device	Mechanism	Pose Establishment	Video Clips	# 2D Targets	# 3D Targets	# Motion Patterns	# Frames
Lieberknecht [11]	Marlin F-080C	Handheld	Marker-based	Yes	8	-	5	48,000
Gauglitz [8]	Fire-i	Manually Operated Contraption	Direct Alignment	Yes	6	-	16	6,889
Hinterstoisser [9]	Kinect v1	Handheld	Marker-based	No	-	15	-	18,000
Tejani [20]	Kinect v1	Handheld	Marker-based	No	-	3	-	5,229
Brachmann [2]	Kinect v1	Handheld	Marker-based	No	-	20	3	10,000
Rennie [16]	Kinect v1	Robotic Arm	Manual	No	-	24	-	10,368
Krull [10]	Kinect v1	Handheld	ICP	Yes	-	3	-	1,100
Choi [4]	Synthetic	-	Synthetic	Yes	-	4	-	4,000
Proposed	**Kinect v2**	**Programmable Robotic Arm**	**Checkerboard-based**	Yes	6	6	23	**100,956**

Out-of-plane Rotation. An object rotates along an axis parallel to the camera sensor plane.

Flashing Light. The light source is turned on and off repeatedly, and the object moves slightly.

Moving Light. The light source moves and results in illumination variations while the object moves slightly.

Free Motion. An object moves in arbitrary directions.

The objects move at five speeds in translation, forward and backward, in-plane and out-of-plane rotations such that the videos are close to real-world scenarios with different image distortions (e.g., motion blurs). For each 3D object, videos from four camera perspectives are recorded. The motion patterns are detailed in the supplementary material.

3.2 Obtaining Ground-truth Object Pose

We estimate the intrinsic camera parameters using the calibration toolbox [1]. It is worth noting that depth and infrared images, as shown in Figure 2, are obtained from the same sensor (i.e., depth camera). Therefore, we calibrate depth camera using infrared images. Next, we conduct an extrinsic calibration [1] resulting in the transformation matrix T_{d2c} from the depth camera coordinate system to color camera coordinate system. The estimated intrinsic parameters are shown in the supplementary material.

After rectifying the images, we obtain the ground-truth pose using the camera parameters and the checkerboard (or checkerbox) around an object as follows. The positions of a few crossed points are initialized with known 2D-to-3D correspondences in the first frame of each sequence and updated by the nearest corners using [19]. Other crossed points can then be obtained with an initial pose $\mathbf{p}_0$ estimated according to the correspondences with the OPnP method [24]. The location of each point is refined with a sub-pixel corner detection method [1]. A point may be discarded if it is close to another crossed point for robust pose estimation. We compute the object pose $\mathbf{p}$ according to the refined points with the OPnP method [24] again and refine $\mathbf{p}$ with the Newton method. Figure 5 shows an example of object pose estimation in the first frame of each sequence. We determine the corresponding points in the current and following frames with the KLT tracker [13], and estimate poses according to these points with the scheme mentioned above. As such, the object pose in each frame can be obtained sequentially. The checkerboard (or checkerbox) is designed with increasing block size from center to border. This pattern facilitates detecting a sufficient number of corner points when the target object is either near or far from the sensor as illustrated in Figure 1.

Table 2: Evaluated algorithms. Run time is measured in seconds. In the code column, C: C/C++, M: Matlab, CU: CUDA.

Algorithm	Description	Code	Run Time
SIFT [12]	Feature detector	C, M	5.287
ASIFT [23]	Feature detector	C, M	50.995
OPnP [24]	PnP algorithm	M	0.008
IPPE [6]	PnP algorithm	M	0.001
DPE [21]	Pose estimator (2D)	C, M, CU	4.811
UDP [3]	Pose estimator (3D)	C	9.262
PWP3D [15]	Pose tracker (3D)	C, CU	0.066
OS2 [14]	SLAM approach (sparse)	C	0.067
EF [22]	SLAM approach (dense)	C, CU	0.078

Because of the symmetric form, the crossed corner points can also be better localized than the corners of fiducial markers used in existing datasets [11, 9, 20, 2]. The exact 3D target position related to the base is calibrated manually.

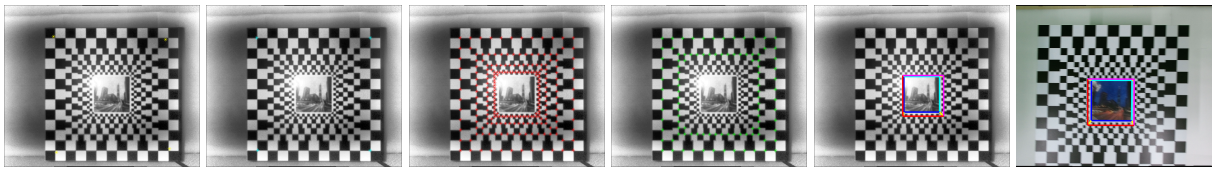
We use the infrared images to obtain the ground-truth object pose $\mathbf{p}$ instead of using color images which can be distorted due to the rolling-shutter effect, as the skewed image illustrated in Figure 5(f). Furthermore, the exposure time of the infrared camera is much shorter such that infrared images contain less motion blur. The object pose in the color images of the first and following frames are obtained by transforming $\mathbf{p}$ according to the transformation matrix $\mathbf{T}_{d2c}$. As the intensity contrast of the original infrared image is relatively low (as shown in Figure 2), the images shown in Figure 5(a)–(e) are processed with a tone mapping algorithm for presentation purpose. Finally, we generate the mask image related to each frame according to the ground-truth pose, as illustrated in Figure 6. These mask images are used for cropping target templates for the training purpose.

4 Evaluation Methodology

In this work, we evaluate pose tracking algorithms for both 2D and 3D target objects. Table 2 lists the main characteristics of the evaluated algorithms. We further explain our evaluation metrics in 4.2.

4.1 Evaluated Algorithms

To estimate the pose of a planar target, we look into feature-based approaches, and evaluate algorithms with a combination of two feature detectors [12, 23] and two PnP algorithms [24, 6] for pose es-



(a) (b) (c) (d) (e) (f)

Figure 5: Ground-truth object pose annotation. (a) We first initialize a few points with known 2D-to-3D correspondences. (b) The nearest corner points of the initialized points are detected. (c) The other corner points are computed with an initial pose $\mathbf{p}_0$ according to the initial correspondences. (d) We later refine these points and discard non-robust ones. (e) The final pose $\mathbf{p}$ is estimated according to the remaining points. (f) The object pose in the related color image is computed according to the estimated transformation matrix.

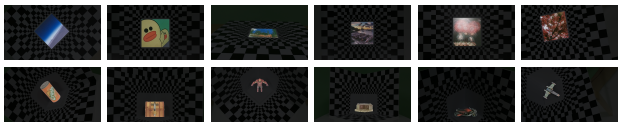


Figure 6: Camera frames for 2D (top row) and 3D (bottom row) target objects blended with masks. The mask is generated using the related pose and its geometric model of the target.

timation. The RANSAC-based schemes [7] are applied to these methods to remove incorrect feature correspondences. We implement a CUDA-based direct pose estimator, which is ten times faster than the recent method [21] with equivalent accuracy.

To recover 3D object poses, we evaluate two state-of-the-art approaches for pose estimation and pose tracking, i.e., UDP [3] and PWP3D [15]. We note numerous camera pose trackers have been released recently that achieve real-time performance by leveraging the reconstructed environment maps. Two state-of-the-art approaches in this field, i.e., ORB-SLAM2 [14] and ElasticFusion [22], are used for evaluation. The ORB-SLAM2 method tracks camera poses based on sparse features, and the ElasticFusion scheme solves a minimization problem based on intensity and depth values. These camera pose trackers are evaluated by deactivating them within background regions of a video sequence. Foreground and background regions are separated according to the geometric model and related ground-truth pose, as illustrated in Figure 6. For SLAM-based approaches, we construct a 3D map of the target object for evaluation. Each map is constructed with synthetic frames created by rendering a mesh from 341 viewpoints on one-half of a recursively divided icosahedron. Since the UDP method [3] does not perform well if it is trained on synthetic images (as discussed in [3] and confirmed in our comparative study), we select about 10% of the images from the proposed dataset as the training data for UDP. For the iterative energy minimization schemes (i.e., PWP3D and ElasticFusion), the ground-truth pose in the first frame is provided and object pose tracking is performed subsequently. The mask image of the first camera frame is also used for the PWP3D scheme to set the color distribution of foreground and background regions. To fairly compare different approaches, the result of the first frame in each video sequence is not considered. As the SLAM approaches are able to deal with 2D cases, we also evaluate these methods with 2D objects. More details on experimental settings can be found in the supplementary material.

4.2 Evaluation Metrics

Given the ground-truth rotation matrix $\hat{\mathbf{R}}$ and translation vector $\hat{\mathbf{t}}$, we compute the error of the estimated pose $(\mathbf{R}, \mathbf{t})$ by $e = \text{avg}_{\mathbf{x}\in\mathscr{M}} \|\mathbf{R}\mathbf{x}+\mathbf{t}-(\hat{\mathbf{R}}\mathbf{x}+\hat{\mathbf{t}})\|$, where $\mathbf{x}$ is a 3D point of model $\mathscr{M}$ [9]. For a 2D object, we define the model points as vertices of a bounding box, whose height is half of its side length, as illustrated in Figure 1. The pose is considered to be successfully estimated if e is less than $k_e d$ where d is the diameter (i.e., the largest distance between vertices) of $\mathscr{M}$ and k_e is a pre-defined threshold. We evaluate a method by the percentage of frames with correct estimations under different values of k_e in a precision plot. A method with higher area-under-a-curve (AUC) scores achieves better pose estimation results.

5 Evaluation Results

All the experiments are carried out on a machine with an Intel Core i7-6700K processor, 32 GB RAM, and a NVIDIA GTX 960 GPU. The RGB-D video frame size is 1920×1080. Each approach for 2D and 3D target objects is evaluated on 20,988 images and 79,968 images, respectively. The iterative energy minimization approaches (e.g., ElasticFusion and PWP3D) tend to lose track of all frames once the matching baseline is too wide. We thus evaluate the ElasticFusion+ and PWP3D+ methods (variants of ElasticFusion and PWP3D) by feeding the ground-truth pose in the previous frame for re-initialization when a failure occurs which is determined by visual inspection. We report the main results of the comparative study on pose tracking in this manuscript, and present more details in the supplementary material.

5.1 Overall Performance

The experimental results are shown in Figure 7. The maximum coefficient k_e is set to 0.2 in the plots, with AUC scores ranging from 0 to 20.

2D objects. The average score of tracking the *wing* sequence is lower than the others since the target object contains less texture or structure. There exist many ambiguous pose candidates that cannot be distinguished by all evaluated approaches as the corresponding cost values are similar. In contrast, although the object in the *duck* sequence does not contain much texture, the DPE method is able to estimate poses well based on the distinct contour. The feature-based schemes outperform direct methods when a sufficient number of features can be extracted from a target object, as shown in the other four cases.

Despite the IPPE algorithm is designed for pose estimation of planar objects, it does not perform as well as the OPnP algorithm that is able to estimate pose in more general scenarios. As the FAST-based detector [17], which is used in the ORB-SLAM2 method, is designed for efficiently detecting corner points in an image, it does not localize features well. Therefore the AUC scores of the ORB-SLAM2 method are lower than those of SIFT-based methods in most cases. It is worth noticing that the ORB-SLAM2 method performs well based on the feature-based scheme as it achieves wide baseline matching, which prevents the tracker from getting stuck in a local minimum. In contrast, the ElasticFusion method tends to lose track of the target object when the initial pose is not accurate since the energy minimization scheme is sensitive to perturbation caused by the introduced distortion in this work.

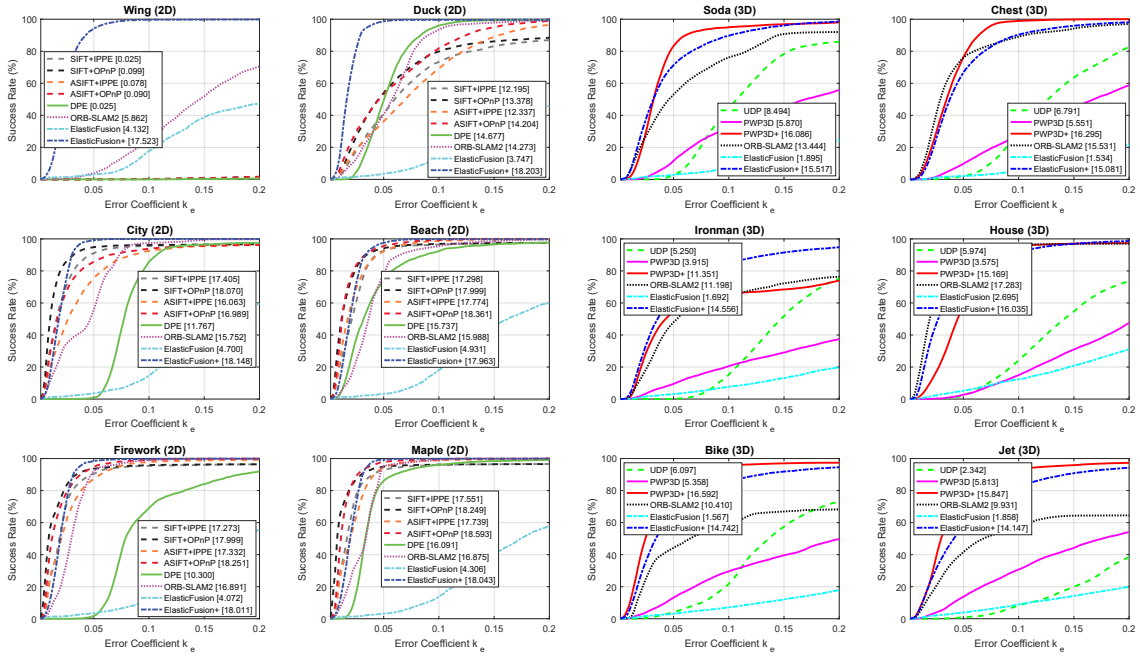


Figure 7: Overall performance on proposed benchmark dataset. The AUC score for each approach is shown in the legend.

3D objects. Since the tracking accuracy and area of an object within one frame are in positive correlation, most approaches achieve better performance on tracking the *soda*, *chest*, and *house* sequences. Similar to tracking 2D objects, methods with energy minimization scheme do not perform well on the 3D dataset. However, they also show the ability to refine poses under the short-baseline conditions. We note that although the AUC scores of the ElasticFusion+ and PWP3D+ methods seem to be higher than the other approaches, it does not mean that they outperform others because their tasks are significantly simplified as the ground truth of the previous pose is given when a failure occurs. As the UDP algorithm does not have any further pose refinement scheme, the estimated pose accuracy is not as high as the other approaches. Both PWP3D and ElasticFusion methods are prone to losing track of the target when its appearance changes drastically.

5.2 Performance Analysis by Attributes

In this section, we show experimental results for each method with respect to different lighting and movement conditions.

2D objects. We present the pose tracking results under two different lighting conditions and freestyle condition movements in Table 3. As both ORB [18] and SIFT [12] are less sensitive to illumination change, the feature-based methods perform well in sequences under lighting variations. In contrast, the DPE algorithm does not track object poses well under different lighting conditions as the direct methods operate on the pixel values without extracting features that are designed to handle illumination changes.

The pose tracking results of target objects in different motion patterns and speeds are shown in Figure 8. Due to fast camera speeds, the recorded images in the translation case contain significant motion blur. As the feature-based approaches are not able to determine useful correspondences in blurry images, these methods do not track poses well. On the other hand, the DPE algorithm performs well with different camera speeds as it can handle objects with less texture.

The ASIFT algorithm outperforms other feature-based approaches in the sequences with out-of-plane rotation since it is designed to account for the affine transformation. We note the ElasticFusion method performs better at higher camera speed. This may be attributed to the fact that the decreased frame number of high-speed sequences also reduces the changes that iterative minimization approaches lose track. As in-depth analysis of this issue requires different experimental setups which are beyond the scope of this work, we will address it in future work.

Table 3: AUC scores of evaluated approaches in the dynamic lighting conditions and the freestyle motion conditions.

Approach	Flashing Light	Moving Light	Free Motion
SIFT+IPPE	14.194	13.902	13.904
SIFT+OPnP	15.380	15.183	14.408
ASIFT+IPPE	13.996	13.584	12.808
ASIFT+OPnP	15.312	14.902	13.461
DPE	12.996	7.516	9.793
ORB-SLAM2	14.879	14.128	14.986
ElasticFusion	1.974	7.479	2.948
ElasticFusion+	16.981	18.173	18.107
UDP	5.170	7.245	3.857
PWP3D	5.084	4.907	2.890
PWP3D+	13.071	14.434	16.041
ORB-SLAM2	15.906	15.987	9.104
ElasticFusion	1.444	2.005	0.278
ElasticFusion+	14.598	12.299	10.871

3D objects. Since we only change the visible light in the above-mentioned experiments with illumination variations, the depth images are not significantly affected. Compared to the pose tracking results of most approaches under normal light, the performance difference on 3D objects is not significant. In contrast, as the PWP3D method recovers the object pose using color frames only, the pose tracking results are worse than those under normal light.

We note all approaches perform worse when the target object moves forward and backward in front of the camera. One reason is the size change of a target object in two consecutive frames. For

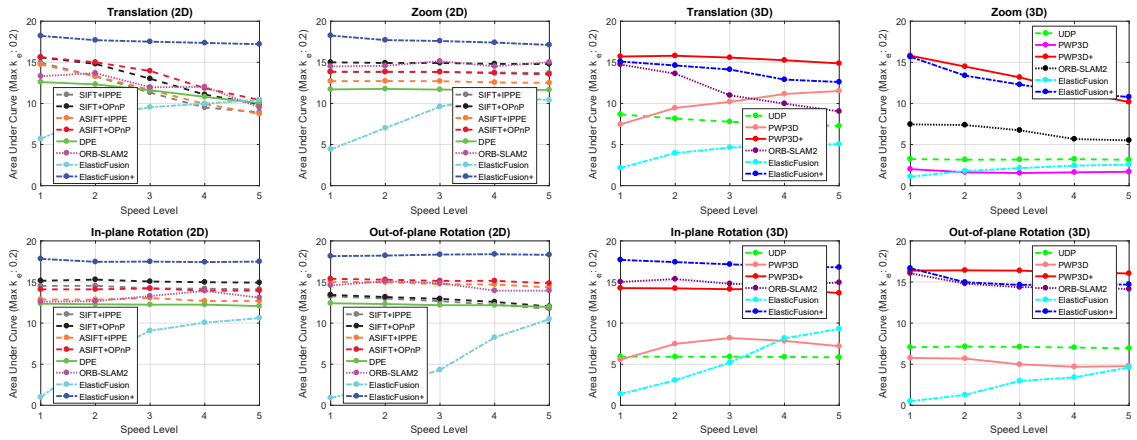


Figure 8: Performance by attributes with different speeds on proposed benchmark dataset. Level 5 stands for the fastest speed.

the ICP-based approaches, e.g., ElasticFusion, it is difficult to align two point sets of different sizes. For the segmentation-based approaches, e.g., PWP3D, it is crucial to set a gradient step in the z-direction.We also notice that the depth values captured by Kinect v2 occasionally change significantly even under the static conditions. As such, the evaluated approaches may occasionally lose track of objects when the camera is not moving.

6 Conclusion

In this work, we propose a large benchmark dataset and perform thorough performance evaluation under various conditions close to real-world scenarios. The proposed benchmark dataset contains 690 color and depth videos with over 100,000 frames. These videos are recorded under seven different movement and lighting conditions with five speeds. We select six 2D target planes with three different texture levels, and six 3D target objects with three different geometric level. The ground-truth poses are annotated by leveraging the clear infrared images recorded by the global-shutter infrared camera with fast shutter speed from the Kinect v2 sensor, which enables us to record sequence even under fast motions. Based on the benchmark experiments, we discuss some tracking components that are essential for improving the tracking performance. This large-scale performance evaluation facilitates a better understanding of the state-of-the-art object pose tracking approaches, and provide a platform for gauging new algorithms.

Acknowledgements

The authors wish to thank Professor Shih-Chung Kang and Ci-Jyun Liang from NTUCE for providing their programmable robotic arm and the fruitful discussions. We would also like to show our gratitude to Po-Hao Hsu for sharing his photos used in this work.

References

[1] J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. *MATLAB*, 2004. 3

[2] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In *ECCV*, 2014. 1, 2, 3

[3] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In *CVPR*, 2016. 1, 3, 4

[4] C. Choi and H. I. Christensen. RGB-D Object Tracking: A Particle Filter Approach on GPU. In *IROS*, 2013. 1, 2, 3

[5] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson. Object Recognition and Full Pose Registration from a Single Image for Robotic Manipulation. In *ICRA*, 2009. 1

[6] T. Collins and A. Bartoli. Infinitesimal Plane-Based Pose Estimation. *IJCV*, 109(3):252–286, 2014. 1, 3

[7] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography. *CACM*, 24(6):381–395, 1981. 1, 4

[8] S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking. *IJCV*, 94(3):335–360, 2011. 2, 3

[9] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In *ACCV*, 2012. 1, 2, 3, 4

[10] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother. 6-DOF Model Based Tracking via Object Coordinate Regression. In *ACCV*, 2014. 1, 2, 3

[11] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab. A Dataset and Evaluation Methodology for Template-based Tracking. In *ISMAR*, 2009. 2, 3

[12] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004. 1, 3, 5

[13] B. D. Lucas, T. Kanade, et al. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, 1981. 3

[14] R. Mur-Artal and J. D. Tardos. Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras. *arXiv*, 2016. 2, 3, 4

[15] V. A. Prisacariu and I. D. Reid. PWP3D: Real-Time Segmentation and Tracking of 3D Objects. *IJCV*, 98(3):335–354, 2012. 1, 3, 4

[16] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza. A Dataset for Improved RGBD-based Object Detection and Pose Estimation for Warehouse Pick-and-Place. *RAL*, 1(2):1179–1185, 2016. 1, 2, 3

[17] E. Rosten and T. Drummond. Machine Learning for High-Speed Corner Detection. In *ECCV*, 2006. 4

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *ICCV*, 2011. 5

[19] J. Shi and C. Tomasi. Good Features to Track. In *CVPR*, 1994. 3

[20] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim. Latent-Class Hough Forests for Object Detection and Pose Estimation. In *ECCV*, 2014. 1, 2, 3

[21] H.-Y. Tseng, P.-C. Wu, M.-H. Yang, and S.-Y. Chien. Direct 3D Pose Estimation of a Planar Target. In *WACV*, 2016. 1, 3, 4

[22] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison. Elasticfusion: Dense slam without a pose graph. In *RSS*, 2015. 2, 3, 4

[23] G. Yu and J.-M. Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *IPOL*, 1:11–38, 2011. 1, 3

[24] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi. Revisiting the PnP Problem: A Fast, General and Optimal Solution. In *ICCV*, 2013. 1, 3