ELSEVIER

Computer Vision and Image Understanding
journal homepage: www.elsevier.com

Direct pose estimation for planar objects

Po-Chen Wu[a], Hung-Yu Tseng[a], Ming-Hsuan Yang[b], Shao-Yi Chien[a,**]

[a]*Media IC & System Lab, Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan*
[b]*Vision and Learning Lab, Electrical Engineering and Computer Science, University of California, Merced, CA 95344, USA*

ABSTRACT

Estimating six degrees of freedom poses of a planar object from images is an important problem with numerous applications ranging from robotics to augmented reality. While the state-of-the-art Perspective-n-Point algorithms perform well in pose estimation, the success hinges on whether feature points can be extracted and matched correctly on target objects with rich texture. In this work, we propose a two-step robust direct method for six-dimensional pose estimation that performs accurately on both textured and textureless planar target objects. First, the pose of a planar target object with respect to a calibrated camera is approximately estimated by posing it as a template matching problem. Second, each object pose is refined and disambiguated using a dense alignment scheme. Extensive experiments on both synthetic and real datasets demonstrate that the proposed direct pose estimation algorithm performs favorably against state-of-the-art feature-based approaches in terms of robustness and accuracy under varying conditions. Furthermore, we show that the proposed dense alignment scheme can also be used for accurate pose tracking in video sequences.

1. Introduction

Determining the six degrees of freedom (6-DoF) pose of a target object from a calibrated camera is a classical problem in computer vision that finds numerous applications such as robotics and augmented reality (AR). While much progress has been made in the past decade, it remains a challenging task to develop a fast and accurate pose estimation algorithm, especially for planar target objects lacking textured surfaces.

Existing pose estimation methods can be broadly categorized into two groups. The approaches in the first category are based on features extracted from target objects with rich textures. The core idea behind feature-based methods is to compute a set of n correspondences between 3D points and their 2D projections from where the relative positions and orientations between the camera and target can be estimated. In recent years, numerous feature detection and tracking schemes (Lowe, 2004; Bay et al., 2008; Leutenegger et al., 2011; Rublee et al., 2011; Alahi et al., 2012) have been developed and applied to a wide range of applications including simultaneous localization and mapping applications (SLAM) (Klein and Murray, 2007; Lim et al., 2012; Mur-Artal and Tardós, 2014). In order to match features robustly, variants of RANSAC algorithms (Fischler and Bolles, 1981; Chum and Matas, 2005) have been used to eliminate outliers before object pose is estimated from a set of feature correspondences. After this step, typically the perspective-n-point (PnP) algorithms (Schweighofer and Pinz, 2006; Lepetit et al., 2009; Zheng et al., 2013) are applied to the feature correspondences for estimating the 6-DoF object pose. We note that feature-based methods are less effective in pose estimation when the tilt angle between the camera and the planar target is large. While the affine-SIFT (ASIFT) (Yu and Morel, 2011) approach matches feature points well when there are large viewpoint changes, it is computationally more expensive than others. Since the performance of feature-based pose estimation methods hinges on whether or not point correspondences can be correctly established, these approaches are less effective when the target images contain less textured surfaces or motion blurs.

The second category consists of direct methods that do not depend heavily on features or textures. Since the seminal work by Lucas and Kanade (1981), numerous algorithms for template matching based on global, iterative, nonlinear optimization have been proposed (Hager and Belhumeur, 1998; Shum and Szeliski, 2001; Baker and Matthews, 2001; Malis, 2004; Xiong and De la Torre, 2015; Lin and Lucey, 2017). As the pose estimation problem can be formulated as the template match-

[**]Corresponding author.
e-mail: pcwu@media.ee.ntu.edu.tw (Po-Chen Wu), hytseng@media.ee.ntu.edu.tw (Hung-Yu Tseng), mhyang@ucmerced.edu (Ming-Hsuan Yang), sychien@ntu.edu.tw (Shao-Yi Chien)

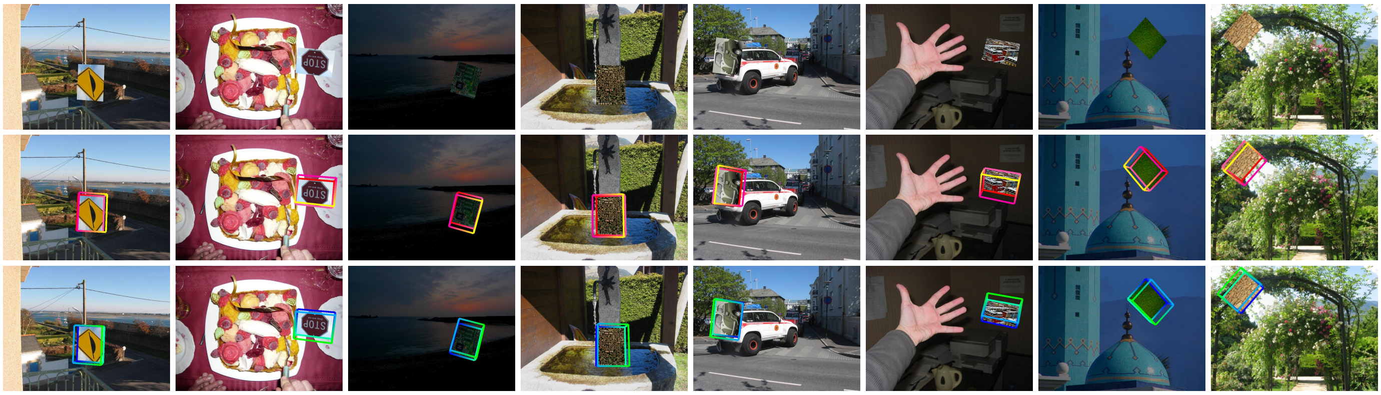


Figure 1. Pose estimation results on synthetic images. The pose ambiguity problem occurs when the objective function has several local minima for a given configuration, which is the primary cause of flipping estimated poses. First row: original images. Second row: images rendered with a box model according to the ambiguous pose obtained from proposed algorithm without refinement approach. Third row: pose estimation results from the proposed algorithm, which can disambiguate plausible poses effectively.

ing problem with the reference frame, poses can be estimated through optimizing the parameters to account for rigid transformations of observed target images (Crivellaro and Lepetit, 2014; Engel et al., 2014). However, these methods rely on initial reference parameters and may be trapped in a local minimum. To alleviate the limitations of nonlinear optimization problems, non-iterative approaches (Chi et al., 2011; Korman et al., 2017; Henriques et al., 2014) have recently been proposed. Nonetheless, these template matching approaches are limited by the misalignment problem between affine or homography transformation in the pose space. It may result in the additional pose error from transformation matrix decomposition while estimating the 6-DoF pose.

In this paper, we propose a direct method to estimate the 6-DoF poses of a planar target from a calibrated camera by measuring the similarity between the projected planar target object image and observed 2D frame based on appearance. As the proposed method is based on a planar object rather than a 3D model, the pose ambiguity problem as discussed in prior arts (Oberkampf et al., 1993; Schweighofer and Pinz, 2006; Li and Xu, 2011; Wu et al., 2014), is inevitably bound to occur. Pose ambiguity is related to situations where the error function has several local minima for a given configuration, which is the main cause of flipping estimated poses in an image sequence. Based on image observations, one of the ambiguous poses with local minima, according to an error function, is the correct pose. Therefore, after obtaining an initial rough pose using an approximated pose estimation scheme, we determine all ambiguous poses and refine the estimates until they converge to local minima. The final pose is chosen as the one with the lowest error among these refined ambiguous poses. We show some pose estimation results by the proposed method in Figure 1. Extensive experiments are conducted to validate the proposed algorithm in this work. In particular, we evaluate the proposed algorithm on different types of templates with different levels of degraded images caused by blur, intensity, tilt angle, and compression noise. Furthermore, we evaluate the proposed algorithm on the datasets by Gauglitz et al. (2011) and Wu et al. (2017) against the state-of-the-art pose estimation methods.

The main contributions of this work are summarized as follows. First, we propose an efficient direct pose estimation algorithm for planar targets undergoing arbitrary 3D perspective transformations. Second, we show the proposed pose estimation algorithm performs favorably against the state-of-the-art feature-based approaches in terms of robustness and accuracy. Third, we demonstrate the proposed pose refinement method not only improves the accuracy of estimated results but also alleviates the pose ambiguity problem effectively.

Based on our prior work in Tseng et al. (2016), in this paper, we extend and construct an image pyramid for the APE method as described in Section 4.1, and we apply a new PR approach based on the Lucas & Kanade (LK) algorithm as described in Section 4.2. We show experimental results with significant improvements regarding accuracy and efficiency compared to the previous work in Section 5. The remainder of this paper is organized as follows. In Section 2, we discuss related work on object pose estimation. We formulate the pose estimation problem in Section 3, and then describe the proposed method, including the approximated pose estimation (APE) and pose refinement (PR) approaches, thoroughly in Section 4. Extensive experimental results are presented in Section 5. We conclude this paper with discussions on future work in Section 6.

2. Related Work

In this section, we first discuss methods for planar object 6-DoF pose estimation in two categories, i.e., feature-based as well as direct approaches, and then introduce techniques for pose disambiguation.

2.1. Feature-based Methods

Establishing feature correspondences across different images typically involves three distinct steps. First, features with rich visual information are detected in both images. The SIFT detector (Lowe, 2004) leverages difference of Gaussians (DoG) to accelerate the detection process in different scales, while the SURF (Bay et al., 2008) detector uses a Haar wavelet approximation of the determinant of the Hessian matrix. As these detectors are computationally expensive, several methods including FAST (Rosten and Drummond, 2006) and AGAST (Mair et al., 2010) have been developed for improvement of execution speed. Second, a feature representation based on a local patch centered at a detected feature is constructed. Although the SIFT descriptor (Lowe, 2004) have been shown to perform robustly in numerous tasks, the incurred computational cost is high as the feature dimensionality is high. Subsequently, binary descriptors, such as BRIEF (Calonder et al., 2010), BRISK (Leutenegger et al., 2011), ORB (Rublee et al., 2011), and FREAK (Alahi et al., 2012), are designed for improvement of execution speed. Third, a feature point is associated with another in the other image. While a method is expected to detect plenty of distinct features accurately in one image and match most of them across different views of the same object, some correspondences are incorrectly determined in practice and most PnP methods do not handle these outliers well. Outliers are typically rejected at a preliminary stage using projective transformation models or P3P algorithms (Gao et al., 2003; Kneip et al., 2011; Ke and Roumeliotis, 2017) in combination with RANSAC-based schemes (Fischler and Bolles, 1981; Chum and Matas, 2005; Fragoso et al., 2013).

After removing outliers, PnP algorithms, e.g., LM (Lu et al., 2000b) and RPP (Schweighofer and Pinz, 2006), can be applied to all the remaining inlier matches by minimizing an appropriate objective function. These methods perform well when reliable initial estimates are provided although at the expense of execution time. Recently, several non-iterative methods without requiring good initial estimates have been proposed. The EPnP method (Lepetit et al., 2009) uses four virtual control points to represent the 3D reference points and performs at the linear computational complexity. This problem formulation and use of linearization strategies facilitate the PnP methods perform efficiently. Numerous approaches have since been developed to improve the accuracy by replacing the linear formulation with polynomial solvers, e.g., , DLS (Hesch and Roumeliotis, 2011), RPnP (Li et al., 2012), UPnP (Kneip et al., 2014), OPnP (Zheng et al., 2013), REPPnP (Ferraz et al., 2014b), CEPPnP (Ferraz et al., 2014a), and IPPE Collins and Bartoli (2014).

2.2. *Direct Methods*

The template matching problem has been widely studied in computer vision, and one critical issue for pose estimation is how to efficiently obtain accurate results while evaluating only a subset of the possible transformations. Since the appearance distances between a template and two sliding windows shifted by a few pixels (e.g., one or two pixels) are usually close due to the nature of image smoothness, Pele and Werman (2007) exploit this property to reduce the time complexity of pattern matching. Alexe et al. (2011) derive an upper bound of the Euclidean distance (based on pixel values) according to the spatial overlap of two windows in an image, and use it for efficient pattern matching. Korman et al. (2017) show that 2D affine transformations of a template can be approximated by samples of a density function based on smoothness of a given image, and propose a fast matching method.

To refine pose estimates, a dense image alignment approach based on the LK algorithm (Lucas and Kanade, 1981) is proposed in this work to improve accuracy. In general, direct image alignment methods estimate the transformation parameters to align a given target image to a camera image. The parameter set which minimizes an objective function (i.e., appearance difference between a transformed target image and a camera image) is regarded as the final estimated pose. The crux of the LK-based algorithm is that an approximately linear relationship exists between object appearance and geometric displacement. As such a relationship is seldom exactly linear, a linearization process is typically repeated until convergence. However, as this process does not always converge within a fixed step size, a line search method is performed every time when we find a descent direction. Among existing methods, the backtracking line search algorithm has been demonstrated to be effective for efficient convergence with the presence of image noise (Orozco et al., 2013).

2.3. *Pose Disambiguation*

The pose ambiguity problem occurs not only under orthographic projection but also for perspective transformation, especially when the target planar object is significantly tilted with respect to camera views. A typical approach for pose disambiguation is first to find all possible poses which are stationary points with local minima of a designed objective function, and then the one with smallest objective values is considered as the estimated pose. Empirically, the number of ambiguous poses is two in general. In Schweighofer and Pinz (2006), it has been shown that two local minima exist for cases with images of a planar target object viewed by a perspective camera, and a method is developed to determine a unique solution based on iterative pose estimation (Lu et al., 2000a). The PnP problem can be posed as a minimization problem (Zheng et al., 2013) and all the stationary points can be determined by using the Gröbner basis method (Kukelova et al., 2008). In addition, given a pose solution, the other ambiguous pose can also be generated by reflecting the first pose with respect to a plane whose normal vector is the line-of-sight from the camera image center to the planar target center (Collins and Bartoli, 2014).

3. Problem Formulation

Given a target image $\mathcal{I}_t$ and an observed camera image $\mathcal{I}_c$ with pixel values normalized in the range $[0, 1]$, the task is to determine the object pose of $\mathcal{I}_t$ in six degrees of freedom parameterized based on the orientation and position of the target object with respect to a calibrated camera. With a set of reference points $\mathbf{x_i} = [x_i, y_i, 0]^\top, i = 1, \ldots, n, n \geq 3$ in the object-space coordinate of $\mathcal{I}_t$, and a set of camera-image coordinates $\mathbf{u}_i = [u_i, v_i]^\top$ in $\mathcal{I}_c$, the transformation between them can be formulated as:

$$\begin{bmatrix} hu_i \\ hv_i \\ h \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix}, \tag{1}$$

where

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \in SO(3), \; \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \in \mathbb{R}^3, \tag{2}$$

are the rotation matrix and translation vector, respectively. In (1), (f_x, f_y) and (x_0, y_0) are the focal length and the principal point of the camera, respectively, and h is the scale factor representing the depth value in the camera coordinate system.

Given the observed camera-image points $\hat{\mathbf{u}}_i = [\hat{u}_i, \hat{v}_i]^\top$, the pose estimation algorithm needs to determine values for pose $\mathbf{p} \equiv (\mathbf{R}, \mathbf{t})$ that minimize an appropriate error function. The rotation of the pose $\mathbf{p}$ can be parameterized in numerous ways (Grassia, 1998) including Euler angles (see Section 4.1) and axis-angle representation (see Section 4.2).

There are two types of error functions commonly used for pose estimation. The first one is based on projection error and used in the PnP algorithms:

$$E_r(\mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n}\left((\hat{u}_i - u_i)^2 + (\hat{v}_i - v_i)^2\right). \tag{3}$$

The second type of error function is based on appearance distance and used in direct methods including this work:

$$E_{a_1}(\mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n}|\mathcal{I}_c(\mathbf{u}_i) - \mathcal{I}_t(\mathbf{x}_i)|, \tag{4}$$

or

$$E_{a_2}(\mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{I}_c(\mathbf{u}_i) - \mathcal{I}_t(\mathbf{x}_i))^2. \tag{5}$$

The error functions in (4) and (5) are the normalized Sum-of-Absolute-Differences (SAD) and Sum-of-Squared-Difference (SSD) errors, respectively.

4. Proposed Algorithm

The proposed algorithm consists of two steps. First, the 6-DoF pose of a planar target object with respect to a calibrated camera is estimated. Second, the object pose is refined and disambiguated.

4.1. Approximate Pose Estimation

Let $T_\mathbf{p}$ be the transformation at pose $\mathbf{p}$ in (1). Assume a reference point $\mathbf{x}_i$ in a target image is transformed separately to $\mathbf{u}_{i1}$ and $\mathbf{u}_{i2}$ in a camera image with two different poses $\mathbf{p}_1$ and $\mathbf{p}_2$. It has been shown (Korman et al., 2017) that if any distance between $\mathbf{u}_{i1}$ and $\mathbf{u}_{i2}$ is smaller than a positive value ε, with upper bound in the Big-O notation (Cormen et al., 2009),

$$\forall \mathbf{x}_i \in \mathcal{I}_t : d(T_{\mathbf{p}_1}(\mathbf{x}_i), T_{\mathbf{p}_2}(\mathbf{x}_i)) = O(\varepsilon), \tag{6}$$

then the following equation holds

$$|E_{a_1}(\mathbf{p}_1) - E_{a_1}(\mathbf{p}_2)| = O(\varepsilon\bar{\mathcal{V}}), \tag{7}$$

where $\bar{\mathcal{V}}$ denotes the mean variation of $\mathcal{I}_t$, which represents the mean value over the entire target image of the maximal difference between each pixel and any of its neighbors. The mean variation $\bar{\mathcal{V}}$ can be constrained by filtering $\mathcal{I}_t$. The main result is that the difference between $E_{a_1}(\mathbf{p}_1)$ and $E_{a_1}(\mathbf{p}_2)$ is bounded in terms of ε. In the proposed direct method, we only need to consider a limited number of poses by constructing a ε-covering pose set $\mathcal{S}$ (Wikipedia, 2018) based on (6) and (7).

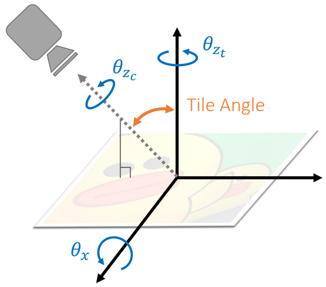


Figure 2. Illustration of rotation angle: θ_x indicates the tilt angle between the camera and the target image when the rotation is factored as $\mathbf{R} = \mathbf{R}_z(\theta_{z_c})\mathbf{R}_x(\theta_x)\mathbf{R}_z(\theta_{z_t})$.

Constructing the ε-Covering Set. By factoring the rotation as $\mathbf{R} = \mathbf{R}_z(\theta_{z_c})\mathbf{R}_x(\theta_x)\mathbf{R}_z(\theta_{z_t})$ (Eberly, 2008) as shown in Figure 2, the pose then can be parameterized as $\mathbf{p} = [\theta_{z_c}, \theta_x, \theta_{z_t}, t_x, t_y, t_z]^\top$. These Euler angles θ_{z_c}, θ_x, and θ_{z_t} are in the range $[-180°, 180°]$, $[0°, 90°]$, and $[-180°, 180°]$, respectively. In addition, the translation parameters t_x, t_y, and t_z are bounded such that the whole target image would be within the camera image, and the bounds depend on the camera intrinsic parameters. Furthermore, we set an upper bound for t_z since it is not practical to detect an extreme tiny target image in the camera image. A pose set $\mathcal{S}$ is constructed such that any two consecutive poses, $\mathbf{p}_k$ and $\mathbf{p}_k + \Delta\mathbf{p}_k$ on each dimension satisfy (6) in $\mathcal{S}$. To construct the set, the coordinates of $\mathbf{x}_i \in \mathcal{I}_t$ are normalized to the range $[-1, 1]$. Starting with t_z, we derive the following equation by using (1) for each $\mathbf{x}_i$:

$$\begin{aligned} &d(T_{\mathbf{p}_{t_z}}(\mathbf{x}_i), T_{\mathbf{p}_{t_z+\Delta t_z}}(\mathbf{x}_i)) \\ &= \sqrt{[(\frac{f_x x_i}{t_z}) - (\frac{f_x x_i}{t_z + \Delta t_z})]^2 + [(\frac{f_y y_i}{t_z}) - (\frac{f_y y_i}{t_z + \Delta t_z})]^2} \\ &= O(\frac{1}{t_z} - \frac{1}{t_z + \Delta t_z}). \end{aligned} \tag{8}$$

To satisfy the constraint in (6), we use the step size with tight bound in the Big-Theta notation (Cormen et al., 2009):

$$\Delta t_z = \Theta(\frac{\varepsilon t_z^2}{1 - \varepsilon t_z}), \tag{9}$$

which means that (8) can be bounded if we construct $\mathcal{S}$ using (9) on dimension t_z.

Since θ_x describes the tilt angle between camera and target image as shown in Figure 2, we obtain the following equation based on t_z:

$$\begin{aligned} &d(T_{\mathbf{p}_{\theta_x}}(\mathbf{x}_i), T_{\mathbf{p}_{\theta_x+\Delta\theta_x}}(\mathbf{x}_i)) \\ &= \sqrt{d_{\mathbf{u_i}}^2 + d_{\mathbf{v_i}}^2} \\ &= O(\frac{1}{t_z - \sin(\theta_x + \Delta\theta_x)} - \frac{1}{t_z - \sin(\theta_x)}), \end{aligned} \tag{10}$$

for each $\mathbf{x}_i$, where

$$\begin{aligned} d_{\mathbf{u_i}} &= (\frac{f_x x_i}{y_i \sin\theta_x + t_z}) - (\frac{f_x x_i}{y_i \sin(\theta_x + \Delta\theta_x) + t_z}), \\ d_{\mathbf{v_i}} &= (\frac{f_y y_i \cos\theta_x}{y_i \sin\theta_x + t_z}) - (\frac{f_y y_i \cos(\theta_x + \Delta\theta_x)}{y_i \sin(\theta_x + \Delta\theta_x) + t_z}). \end{aligned} \tag{11}$$

In addition, to satisfy the constraint in (6), we set the step size

Table 1. Bounded step size on each dimension in the pose domain for constructing the ε-covering pose set.

Dimension	Step Size
θ_{z_c}	$\Theta(\varepsilon t_z)$
θ_x	$\Theta(\sin^{-1}(t_z - \frac{1}{\varepsilon + \frac{1}{t_z - \sin(\theta_x)}}) - \theta_x)$
θ_{z_t}	$\Theta(\varepsilon t_z)$
t_x	$\Theta(\varepsilon(t_z - \sqrt{2}\sin(\theta_x)))$
t_y	$\Theta(\varepsilon(t_z - \sqrt{2}\sin(\theta_x)))$
t_z	$\Theta(\frac{\varepsilon t_z^2}{1 - \varepsilon t_z})$

when using (10):

$$\Delta\theta_x = \Theta(\sin^{-1}(t_z - \frac{1}{\varepsilon + \frac{1}{t_z - \sin(\theta_x)}}) - \theta_x). \tag{12}$$

Similarly, we derive the steps for the other dimensions based on t_z and θ_x. Table 1 summarizes the bounded step size on each dimension for the ε-covering pose set, and the derivation details are presented in Appendix A.

Finally, the pose set is constructed recursively starting from t_z based on the bounded step shown in Table 1. We then determine values of θ_x based on its bounded step which is influenced by t_z. The remaining pose parameters θ_{z_c}, θ_{z_t}, t_x, and t_y are determined based on each of their bounded steps, which are afftected only by t_z and θ_x and independent of each other.

Coarse-to-Fine Estimation. As the parameter space is large, the computational and memory costs are prohibitively high if the ε-covering set is used straightforwardly for pose estimation. In this work, we develop a coarse-to-fine approach for fast and accurate pose estimation. The pose set $\mathcal{S}$ is first constructed with a coarse ε. After obtaining the best pose $\mathbf{p}_b$ and the associated error measure $E_{a_1}(\mathbf{p}_b)$, we select the poses within a threshold:

$$\mathcal{S}_L = \{\mathbf{p}_L \;\mid E_{a_1}(\mathbf{p}_L) < E_{a_1}(\mathbf{p}_b) + L\}, \tag{13}$$

to be considered in the next step. Here the constant L is a threshold empirically determined. Based on $\mathcal{S}_L$, we create sets with finer ε':

$$\mathcal{S}' = \{\mathbf{p}' \;\mid \exists \mathbf{p}_L \in \mathcal{S}_L : (6) \text{ holds for } \mathbf{p}', \mathbf{p}_L \text{ and } \varepsilon'\}, \tag{14}$$

and repeat this process until we obtain the desired precision parameter ε^*. In our implementation, the initial ε is set to be 0.25 and is diminished by multiplying a scale factor of 0.662 in each iteration. The precision parameter ε^* is set to meet the condition that for each point in the target image, the maximum distance between neighboring points in the camera image transformed by poses in the ε-covering pose set is less than 1 pixel. Empirically, ε^* would be around 0.01. The best pose in the last set is considered as the approximated estimate.

Approximate Error Measure. If we approximate the error measure E'_{a_1} with random sampling only a portion of pixels instead of computing E_{a_1} with sampling all pixels in $\mathcal{I}_t$, according to Hoeffding's inequality (Abu-Mostafa et al., 2012), E'_{a_1} is close to E_{a_1} within a precision parameter δ if the number of sampling pixels m is sufficiently large:

$$P(|E'_{a_1} - E_{a_1}| > \delta) \leq 2e^{-2\delta^2 m}, \tag{15}$$

where $P(\cdot)$ represents the probability measure. This inequality suggests that if m is properly selected, the approximation error between E'_{a_1} and E_{a_1} can be bounded with high probability. In other words, E'_{a_1} is a close approximation of E_{a_1} within the probably approximately correct (PAC) framework (Kearns and Vazirani, 1994). With this approximation, the runtime of estimating the error measure can be significantly reduced by inspecting only a small fraction of pixels in a target image. We normalize the intensity term and add the chroma components to the appearance distance measure to account for lighting variation.

Pyramidal Implementation. To constrain the mean variation $\bar{\mathcal{V}}$ in (7), it is common to blur $\mathcal{I}_t$ (and $\mathcal{I}_c$) before carrying out the proposed approximated pose estimation method. Since a blurry image has a texture similar to that of a lower resolution image, we construct an image pyramid instead of directly blurring images. It is worth using a lower resolution image for pose estimation from some perspectives. First, when we sample pixels on a smaller image, the cache miss rate will be lower and thus reduce memory traffic. Second, we can also sample a smaller amount of pixels in (15) when using low-resolution images. Starting from the lowest resolution image, we proceed to the next level (i.e., higher resolution image) when the distance in (6) is smaller than one pixel for all transformations. Empirically, the pyramid implementation can increase the runtime performance significantly while achieving similar or even higher accuracy and robustness for pose estimation.

4.2. *Pose Refinement*

We obtain a coarse pose $\mathbf{p}' \equiv (\mathbf{R}', \mathbf{t}')$ using the proposed approximate pose estimation scheme. However, this estimate is bounded based on the distance in the appearance space rather than the pose space. Thus the estimated and actual poses may be significantly different even when the appearance distance is small, particularly when the tilt angle of a target image is large. In the meanwhile, the pose ambiguity problem is likely to occur as illustrated in Figure 1. As such, we propose a pose refinement method to improve accuracy and address the ambiguity problem of estimates.

Determining Candidate Poses. In order to address the pose ambiguity problem, we first transform four corner points $\mathbf{x}_{c1}$, $\mathbf{x}_{c2}$, $\mathbf{x}_{c3}$, and $\mathbf{x}_{c4}$ in the target image $\mathcal{I}_t$ to $\mathbf{u}_{c1}$, $\mathbf{u}_{c2}$, $\mathbf{u}_{c3}$, and $\mathbf{u}_{c4}$ in the observed camera image $\mathcal{I}_c$ with $\mathbf{p}'$, respectively. We then compute all stationary points of the error function (3) based on the Gröbner basis method (Kukelova et al., 2008). Only the stationary points with the two smallest objective values in (3) are plausible poses, and these two ambiguous poses $\mathbf{p}'_1$ and $\mathbf{p}'_2$ are both chosen as the candidate poses.

Refining Candidate Poses. After obtaining the two candidate poses, we further refine the estimates using a dense image alignment method which minimizes the SSD error in (5) (instead of the SAD error in (4) as it is not continuously differentiable) by the LK-based approach. For each candidate pose $\mathbf{p}_c$, we solve the nonlinear least squares problem using the Gauss-Newton iteration method. To approximate how the image changes with respect to pose, we use the first-order Taylor series as follows:

$$\begin{aligned} \Delta\mathbf{p}^* &= \underset{\Delta\mathbf{p}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(\mathcal{I}_c\left(\mathbf{u}_i\left(\mathbf{p}_c + \Delta\mathbf{p}\right)\right) - \mathcal{I}_t\left(\mathbf{x}_i\right)\right)^2 \\ &\approx \underset{\Delta\mathbf{p}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathcal{I}_c\left(\mathbf{u}_i\left(\mathbf{p}_c\right)\right) + \left.\frac{\partial \mathcal{I}_c}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}_c} \Delta\mathbf{p} - \mathcal{I}_t\left(\mathbf{x}_i\right)\right)^2 . \end{aligned} \tag{16}$$

Different from the method described in Section 4.1, here the pose $\mathbf{p}$ is parameterized as a 6D vector consisting of the 3D axis angles of the rotation matrix and the 3D translation vector:

$$\mathbf{p} = \begin{bmatrix} \mathbf{r} \\ \mathbf{t} \end{bmatrix}, \ \mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \in \mathbb{R}^3, \ \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \in \mathbb{R}^3. \tag{17}$$

To compute $\Delta\mathbf{p}$ in each iteration, we set the first derivative of (16) to zero and solve the resulting system of linear equations:

$$\mathbf{J}_c \Delta\mathbf{p} = \mathbf{I}_t - \mathbf{I}_c, \tag{18}$$

where $\mathbf{I}_t$ and $\mathbf{I}_c$ are vector forms of $\mathcal{I}_t\left(\mathbf{x}_i\right)$ and $\mathcal{I}_c\left(\mathbf{u}_i\right)$, respectively. In (18), $\mathbf{J}_c$ is the Jacobian matrix of $\mathbf{I}_c$ with respect to $\mathbf{p}$ at the pose $\mathbf{p} = \mathbf{p}_c$ and computed by the chain rule (in the numerator-layout notation):

$$\mathbf{J}_c = \left.\frac{\partial \mathbf{I}_c}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}_c} = \begin{bmatrix} \frac{\partial \mathcal{I}_c(\mathbf{u}_1)}{\partial \mathbf{p}} \\ \frac{\partial \mathcal{I}_c(\mathbf{u}_2)}{\partial \mathbf{p}} \\ \vdots \\ \frac{\partial \mathcal{I}_c(\mathbf{u}_n)}{\partial \mathbf{p}} \end{bmatrix}, \tag{19}$$

$$\frac{\partial \mathcal{I}_c}{\partial \mathbf{p}} = \frac{\partial \mathcal{I}_c}{\partial \mathbf{u}} \left[\frac{\partial \mathbf{u}}{\partial \mathbf{r}}, \frac{\partial \mathbf{u}}{\partial \mathbf{t}}\right] = \left[\frac{\partial \mathcal{I}_c}{\partial u}, \frac{\partial \mathcal{I}_c}{\partial v}\right] \left[\frac{\partial \mathbf{u}}{\partial \hat{\mathbf{x}}} \frac{\partial \hat{\mathbf{x}}}{\partial \hat{\mathbf{R}}} \frac{\partial \hat{\mathbf{R}}}{\partial \mathbf{r}}, \frac{\partial \mathbf{u}}{\partial \hat{\mathbf{x}}}\right], \tag{20}$$

$$\frac{\partial \mathbf{u}}{\partial \hat{\mathbf{x}}} = \begin{bmatrix} \frac{f_x}{\hat{z}} & 0 & -\frac{f_x \hat{x}}{\hat{z}^2} \\ 0 & \frac{f_y}{\hat{z}} & -\frac{f_y \hat{y}}{\hat{z}^2} \end{bmatrix}, \frac{\partial \hat{\mathbf{x}}}{\partial \hat{\mathbf{R}}} = \begin{bmatrix} x & y & 0 & 0 & 0 & 0 \\ 0 & 0 & x & y & 0 & 0 \\ 0 & 0 & 0 & 0 & x & y \end{bmatrix}, \tag{21}$$

where $\hat{\mathbf{R}} = [R_{11}, R_{12}, R_{21}, R_{22}, R_{31}, R_{32}]^\top$ denotes the vector with elements in the left two columns of the rotation matrix $\mathbf{R}$, and

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & t_x \\ R_{21} & R_{22} & t_y \\ R_{31} & R_{32} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{22}$$

is the camera-space coordinate transformed from the object-space coordinate $\mathbf{x} = [x, y, 0]^\top$.

In addition, the derivative of $\hat{\mathbf{R}}$ with respect to $\mathbf{r}$ can be obtained using the following formula (Gallego and Yezzi, 2015):

$$\frac{\partial \mathbf{R}}{\partial r_i} = \frac{r_i \left[\mathbf{r}\right]_\times + \left[\mathbf{r} \times \left(\mathbf{I} - \mathbf{R}\right) \mathbf{e}_i\right]_\times}{\|\mathbf{r}\|^2} \mathbf{R}, \tag{23}$$

where $\mathbf{I}$ and $\mathbf{e}_i$ are the identity matrix and the i-th vector of the standard basis in $\mathbb{R}^3$, respectively. In (23), $[\mathbf{r}]_\times$ is defined by:

$$[\mathbf{r}]_\times = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}, \tag{24}$$

which represents the cross product (skew-symmetric) matrix for the vector $\mathbf{r}$.

A closed form solution of (18) is:

$$\Delta\mathbf{p} = \left(\mathbf{J}_c^\top \mathbf{J}_c\right)^{-1} \mathbf{J}_c^\top \left(\mathbf{I}_t - \mathbf{I}_c\right). \tag{25}$$

As the least squares problem is nonlinear, the Gauss-Newton iteration method does not always converge with a fixed step size. We thus perform a backtracking line search to scale the step size after each iteration of computing (25). We shrink $\Delta\mathbf{p}$ by $\Delta\mathbf{p} \leftarrow \alpha\Delta\mathbf{p}$ until it meets the Armijo-Goldstein condition:

$$E_{a_2}(\mathbf{p}_c + \Delta\mathbf{p}) \leq E_{a_2}(\mathbf{p}_c) + c\nabla E_{a_2}(\mathbf{p}_c)^\top \Delta\mathbf{p}, \tag{26}$$

where $\nabla E_{a_2}(\mathbf{p}_c)$ is the local function gradient. We set $\alpha = 0.5$ and $c = 10^{-4}$ empirically in this work. The candidate pose $\mathbf{p}_c$ is refined by $\mathbf{p}_c \leftarrow \mathbf{p}_c + \Delta\mathbf{p}$ until the vector norm $\|\Delta\mathbf{p}\|$ is less than a predefined threshold $\varepsilon_{\Delta\mathbf{p}}$.

Finally, the pose corresponding to the smaller E_{a_2} is selected from the two refined candidate poses. The main steps of the proposed pose estimation method are summarized in Algorithm 1. It should be noted that we also perform the pyramid implementation for the refinement process to increase both the accuracy and efficiency.

5. Experimental Results

We evaluate the proposed algorithm for the 6-DoF pose estimation problem using a synthetic image dataset that we develop and two real image benchmark datasets (Gauglitz et al., 2011; Wu et al., 2017). As the color of each template in the real image benchmark datasets is slightly changed after being generated by a printer and then viewed by a camera, we calibrate each template in the two real image benchmark datasets before carrying out performance evaluation.

We compare the proposed algorithm with feature-based pose estimation methods. The proposed direct pose estimation (DPE) algorithm is constructed with the approximated pose estimation (APE) and pose refinement (PR) approaches. Based on preliminary experiments, we determine the SIFT (Lowe, 2004) representation performs better than other alternative features in terms of repeatability and accuracy. Similar observations have also be reported in the literature (Gauglitz et al., 2011). As the ASIFT (Yu and Morel, 2011) method is considered the state-of-the-art affine-invariant method to determine correspondences under large view changes, we use both the SIFT and ASIFT representations in the evaluation against feature-based schemes. The RANSAC-based method (Fischler and Bolles, 1981) is then used to eliminate outliers before object pose is estimated by the PnP algorithm. It has been shown that, among the PnP algorithms (Schweighofer and Pinz, 2006; Lepetit et al., 2009; Zheng et al., 2013; Kneip et al., 2014;

Algorithm 1: Direct 6-DoF Pose Estimation

```
Input: Target image $\mathcal{I}_t$, camera image $\mathcal{I}_c$, intrinsic
    parameters, and parameters $\varepsilon^*$, $\varepsilon_{\Delta\mathbf{p}}$;
Output: Estimated pose result $\mathbf{p}^*$;
Build image pyramids for $\mathcal{I}_t$ and $\mathcal{I}_c$;
Start from images with lowest resolution;
Create an $\varepsilon$-covering pose set $\mathcal{S}$;
Find $\mathbf{p}_b$ from $\mathcal{S}$ with $E'_{a_1}$ according to (15);
while $\varepsilon > \varepsilon^*$ do
    Obtain the set $\mathcal{S}_L$ according to (13);
    Diminish $\varepsilon$;
    if $d < 1$ according to (6) then
        Change to the next image resolution;
    end if
    Replace $\mathcal{S}$ according to (14);
    Find $\mathbf{p}_b$ from $\mathcal{S}$ with $E'_{a_1}$ according to (15);
end while
Determine the candidate poses $\mathbf{p}_1$ and $\mathbf{p}_2$ with $\mathbf{p}_b$;
for $i = 1 \rightarrow 2$ do
    Let $\mathbf{p}_c = \mathbf{p}_i$;
    repeat
        Compute $\mathbf{J}_c$ according to (19);
        Compute $\Delta\mathbf{p}$ according to (25);
        while Condition according to (26) is not met do
            $\Delta\mathbf{p} \leftarrow \alpha\Delta\mathbf{p}$
        end while
        $\mathbf{p}_c \leftarrow \mathbf{p}_c + \Delta\mathbf{p}$
    until $\|\Delta\mathbf{p}\| < \varepsilon_{\Delta\mathbf{p}}$
    Let $\mathbf{p}_i = \mathbf{p}_c$;
end for
Return the pose $\mathbf{p}^*$ with smaller $E_{a_2}$ from $\mathbf{p}_1$ and $\mathbf{p}_2$;
```

Collins and Bartoli, 2014), the OPnP (Zheng et al., 2013) and IPPE (Collins and Bartoli, 2014) algorithms achieve the state-of-the-art results in terms of efficiency and precision for planar targets. Thus, we use these two algorithms as the pose estimator in the feature-based methods.

All the experiments are carried out using MATLAB on a machine with an Intel Core i7-6700K 4.0 GHz processor and 32 GB RAM. In addition, we implement the proposed direct method on an NVIDIA GTX 970 GPU using CUDA based on Tseng et al. (2017). Table 2 shows average runtime for different algorithms. The source code and datasets are available on our project website at media.ee.ntu.edu.tw/research/DPE.

Given the true rotation matrix $\tilde{\mathbf{R}}$ and translation vector $\tilde{\mathbf{t}}$, we compute the rotation error of the estimated rotation matrix $\mathbf{R}$ by $E_r(degree) = \arccos((\mathrm{Tr}(\mathbf{R}^\top \cdot \tilde{\mathbf{R}}) - 1)/2)$, where $\arccos(\cdot)$ represents the inverse cosine operation in degrees and $\mathrm{Tr}(\cdot)$ is the trace of a matrix. The translation error of the estimated translation vector $\mathbf{t}$ is measured by the relative difference between $\tilde{\mathbf{t}}$ and $\mathbf{t}$ defined by $E_t(\%) = \|\tilde{\mathbf{t}} - \mathbf{t}\|/\|\tilde{\mathbf{t}}\| \times 100$. We define a pose to be successfully estimated if its both errors are under predefined thresholds. We use $\delta_r = 20°$ and $\delta_t = 10\%$ as the thresholds on rotation error and translation error empirically, as shown in Figure 3. The success rate (SR) is defined as the percentage of the successfully estimated poses within each test condition. In the following sections, the average rotation and translation errors are computed only for successfully estimated poses.

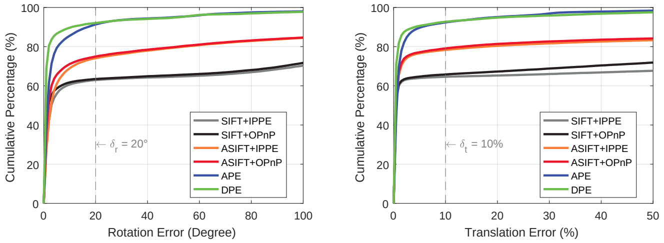


Figure 3. Cumulative percentage of poses whose rotation or translation errors are under values specified in the x-axis over experiments. The vertical dashed lines correspond to the thresholds used to detect unsuccessfully estimated poses. There is a total of 36,277 poses estimated by each pose estimation approach.

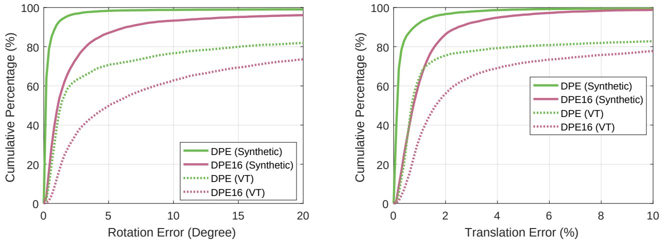


Figure 4. Cumulative percentage of poses whose rotation or translation errors are under thresholds specified in the x-axis over experiments on the same datasets used by (Tseng et al., 2016) (i.e., the synthetic dataset and the visual tracking dataset (Gauglitz et al., 2011)).

We compare the DPE algorithm proposed in this work with the algorithm proposed in the previous work (i.e., DPE16) (Tseng et al., 2016) on the same datasets Tseng et al. (2016). Figure 4 shows that the proposed DPE algorithm performs accurately and robustly against the DPE16 method. For presentation clarify, we do not show the evaluation results of the DPE16 method in the following sections.

5.1. Synthetic Image Dataset

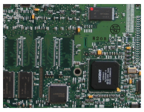
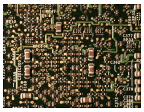
For our experiments we use a set of synthetic images consisting of 8,400 test images covering 21 different test conditions. Each test image is generated from warping a template image according to the randomly generated pose with the tilt angle in the range [0°, 75°] with a randomly chosen background image as shown in Figure 5. The template image size is 640×480 pixels. These templates are classified into four different classes, namely "Low Texture", "Repetitive Texture", "Normal Texture", and "High Texture" (Lieberknecht et al., 2009) as shown from top to bottom in Figure 5. Each class is represented by two targets. The background images are from the database (Jegou et al., 2008) and resized to 800×600 pixels.

Undistorted Images. The pose estimation results of the SIFT-based, ASIFT-based, and proposed direct methods on the undistorted test images are shown in Table 3. For each image, the average rotation error E_r, translation error E_t, and success rate

Table 2. Average runtime (measured in seconds) for approaches on different datasets. Although SIFT-based Approach is the fastest method among these three different schemes, its performance is quite limited. Numbers in parentheses denote the average runtime of the CUDA implementation of the proposed method, which can be executed more efficiently on a GPGPU platform as it can be easily parallelized.

Dataset	SIFT-based Approach				ASIFT-based Approach				DPE		
	SIFT	RANSAC	IPPE/OPnP	Total	ASIFT	RANSAC	IPPE/OPnP	Total	APE	PR	Total
Synthetic	7.431	0.010	0.001/0.009	*7.446*	10.903	0.004	0.001/0.009	*10.912*	10.549 (**1.505**)	0.571 (**0.117**)	*11.120* (***1.622***)
VT	3.608	0.005	0.001/0.008	*3.618*	15.806	0.003	0.001/0.008	*15.814*	17.920 (**1.217**)	0.694 (**0.180**)	*18.615* (***1.397***)
OPT	11.261	0.098	0.001/0.008	*11.364*	38.884	0.055	0.001/0.008	*38.944*	18.545 (**0.994**)	0.214 (**0.088**)	*18.759* (***1.082***)

Table 3. Evaluation results for feature-based approaches and the proposed direct methods with undistorted test images in terms of average numbers of rotation error E_r, translation error E_t, and success rate in each test condition. The best values are highlighted in bold.

	Bump Sign			Stop Sign			Lucent			MacMini Board			Isetta			Philadelphia			Grass			Wall		
																								
Method	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)
SIFT+IPPE	0.85	0.34	*40.0*	1.90	0.54	*86.0*	0.23	0.25	*28.0*	0.32	0.24	*86.0*	0.74	0.35	*92.0*	0.56	0.40	*98.0*	1.15	0.50	*30.0*	0.28	0.37	*96.0*
SIFT+OPnP	0.76	0.40	*40.0*	1.18	0.46	*86.0*	0.20	0.24	*28.0*	0.25	0.24	*86.0*	0.56	0.32	*92.0*	0.55	0.43	*98.0*	1.48	0.47	*30.0*	0.25	0.36	*96.0*
ASIFT+IPPE	9.70	2.92	*20.0*	2.96	0.81	*94.0*	1.48	0.43	***100***	1.65	0.51	*94.0*	1.59	0.57	***100***	1.29	0.34	*98.0*	2.17	0.52	*52.0*	1.96	0.36	*90.0*
ASIFT+OPnP	8.20	2.22	*22.0*	2.72	0.74	***100***	1.38	0.41	***100***	1.53	0.45	*96.0*	1.40	0.50	*98.0*	1.26	0.35	***100***	1.33	0.37	*52.0*	1.80	0.36	*94.0*
APE	1.10	0.33	***100***	1.44	0.42	***100***	0.90	0.47	*98.0*	2.56	1.23	*94.0*	1.03	0.35	***100***	1.63	0.49	***100***	1.96	0.91	***100***	1.57	0.68	*98.0*
DPE	**0.39**	**0.17**	***100***	**0.42**	**0.24**	***100***	**0.16**	**0.14**	***100***	**0.16**	**0.12**	***98.0***	**0.21**	**0.16**	***100***	**0.21**	**0.11**	***100***	**0.15**	**0.14**	***100***	**0.17**	**0.13**	***100***

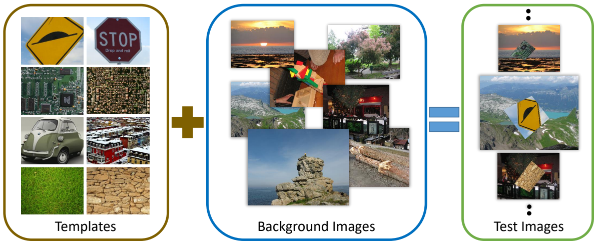


Figure 5. A synthetic test image was generated from a warping template image according to a randomly generated pose on a randomly chosen background image.

are presented. The evaluation results show that the proposed DPE method performs accurately and robustly against feature-based approaches on various template images. In addition, the proposed refinement approach can effectively improve accuracy that is first estimated by the APE method.

In most cases, the feature-based approaches do not estimate pose accurately on textureless template images or template images with feature points that are similar to each other. Although 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 more accurately in general scenarios.

Degraded Images. We evaluate these approaches using all templates with different types of image degradation: 1) Gaussian blur with kernel width of {1, 2, 3, 4, 5} pixels, 2) JPEG compression with the quality parameter set to {90, 80, 70, 60, 50}, 3) intensity change with pixel intensity scale factor set to {0.9,0.8,0.7,0.6,0.5}, and 4) tilt angle in the range of {[0°, 15°), [15°, 30°), [30°, 45°), [45°, 60°), and [60°, 75°)}. Figure 6 shows the evaluation results. The proposed DPE algorithm performs favorably against the other feature-based methods on blurry images. Although the translation errors of the proposed method appear to be larger than those of feature-based methods, these errors are computed only on successfully estimated poses. As the proposed method can estimate template poses successfully even under blur conditions, the errors are larger due to slightly inaccurate pose estimates in blurry images.

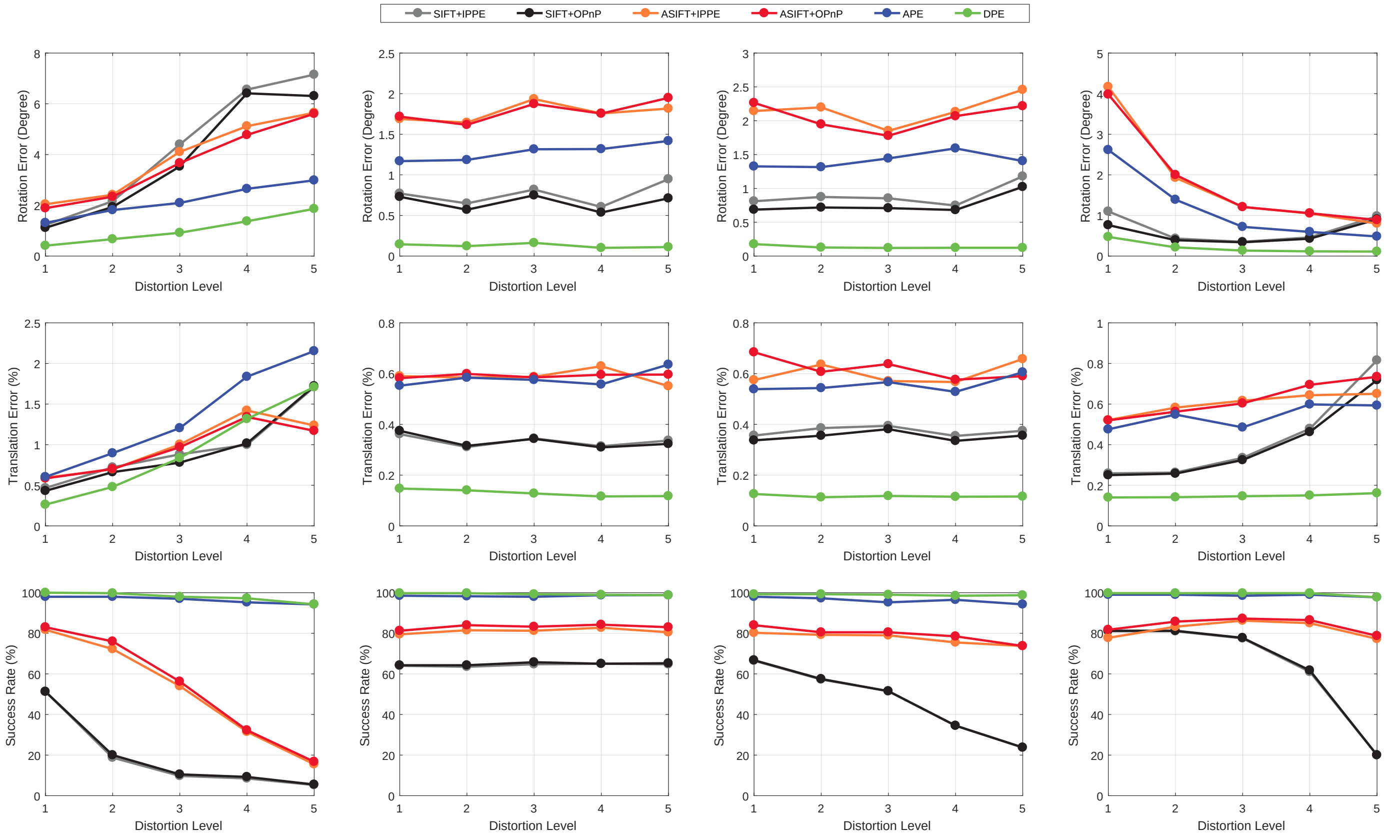
All approaches are able to deal with certain levels of distortion with JPEG compression noise.

For images with intensity changes, the SIFT-based methods perform worse than other approaches as fewer features are detected in low contrast images by the SIFT detector. We note that the SIFT-based methods can still perform well under low-intensity conditions when we adjust the feature detection threshold to extract more features.

Although the SIFT-based approaches can detect and match features accurately under small tilt angles, these methods frequently fail when the tilt angles are larger. In contrast, the proposed algorithm and the ASIFT-based methods are able to estimate 6-DoF poses relatively well even the template images are perspectively distorted in the camera images.

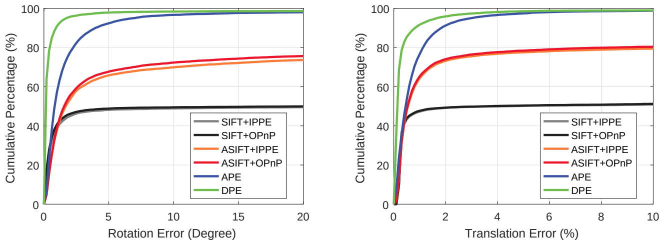
We show the overall evaluation results on the proposed synthetic image dataset in Figure 7. Overall, the proposed direct method performs favorably against the feature-based approaches with the success rate of 98.90%. The success rate of the SIFT-based and ASIFT-based approaches are 49.65% and 74.26%, respectively.

Refinement Analysis. To improve pose estimation accuracy, we propose a refinement method that minimizes the appearance distance between the template and camera images using an LK-



(a) Gaussian blur (b) JPEG compression (c) Intensity change (d) Tilt angle

Figure 6. Experimental results on synthetic data under varying conditions.



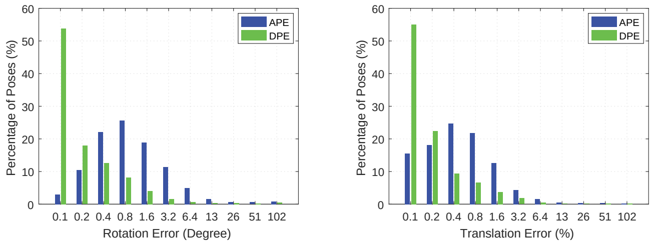


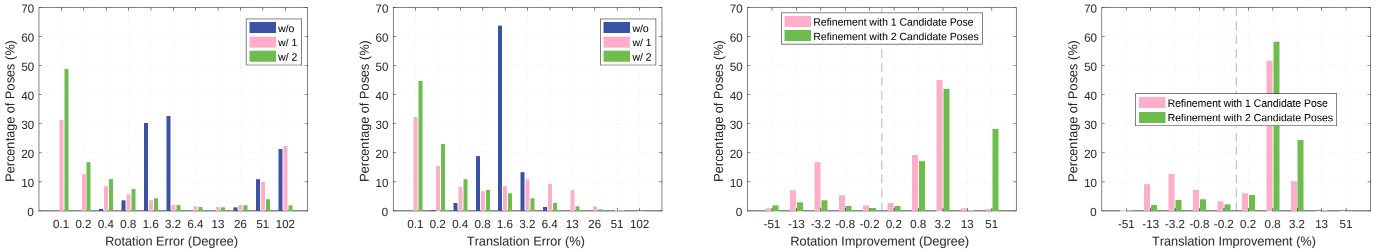
Figure 7. Cumulative percentage of poses whose rotation or translation errors are under thresholds specified in the x-axis over experiments on the proposed synthetic image dataset. There is a total of 8,400 poses estimated by each pose estimation approach.

Figure 8. Pose estimation results with refinement approach (DPE) and without refinement approach (APE). The average value of rotation and translation errors are both reduced by the proposed refinement approach.

based scheme as described in Section 4.2. Figure 8 shows pose estimation results with and without the refinement approach on the synthetic dataset. The rotation and translation errors of estimated poses are smaller after the proposed refinement process. The rotation and translation errors can be reduced by 1.951° and 0.670% respectively with proposed refinement scheme. Sample images rendered with poses estimated by the proposed algorithm with and without the refinement scheme on the synthetic image dataset are shown in Figure 1.

We design another experiment to demonstrate the proposed algorithm is able to disambiguate plausible poses. A template image from the synthetic dataset is warped according to pose $\mathbf{p}_t$. Two ambiguous pose, $\mathbf{p}_{a_1}$ and $\mathbf{p}_{a_2}$, can be obtained from $\mathbf{p}_t$ using the functional minimization method (Zheng et al., 2013). One of the two plausible poses $\mathbf{p}'_a$ is randomly chosen and added with some Gaussian noise. The refinement approach is then applied to $\mathbf{p}'_a$ for estimating the pose of the warped template image. Finally, we compute E_r and E_t of both the initial noisy pose $\mathbf{p}'_a$ and the refined pose $\mathbf{p}_r$ according to $\mathbf{p}_t$. Thus, if the proposed refinement approach can disambiguate the plausible pose $\mathbf{p}'_a$, the rotation error can be reduced significantly. All images in the synthetic dataset are used for the experiment.

We compare the proposed refinement method with the refinement approach with only one candidate pose in Algorithm 1,



(a) Distribution of errors

(b) Improvements

Figure 9. Results of the proposed method without refinement (w/o), refinement with one candidate (w/ 1), and refinement with two candidates (w/ 2). (a) The rotation errors are reduced significantly in the ambiguous cases, but the translation errors are relatively not because the translation terms of ambiguous poses are quite similar in most cases. (b) The difference of pose errors before and after applying two kinds of refinement approaches. While the proposed refinement approach can disambiguate the object pose effectively, approach with only one candidate pose suffers from the risk of getting trapped into a local minimum.

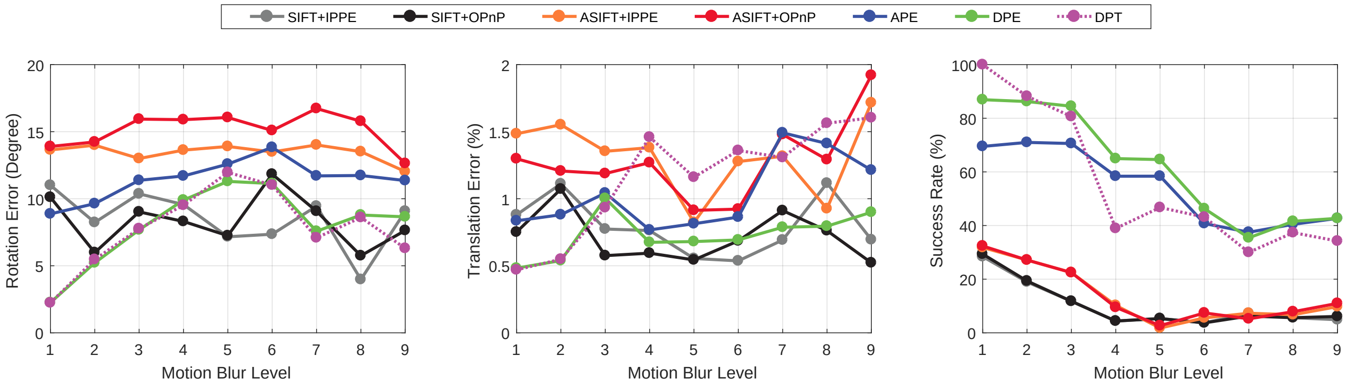


Figure 10. Experimental results on the visual tracking dataset (Gauglitz et al., 2011) under varying motion blur levels, where level 9 stands for the strongest motion blur.

Table 4. Evaluation results for different pose refinement approaches on the synthetic image dataset in the refinement analysis experiment.

Approach	$E_r(°)$	$E_t(\%)$	SR(%)
Without Refinement	2.235	1.369	*66.82*
Refinement with 1 Candidate Pose	0.734	0.461	*65.49*
Refinement with 2 Candidate Poses	**0.558**	**0.416**	***92.05***

and present the results in Figure 9. While the rotation errors of ambiguous poses are usually large (which causes the pose flipping), the proposed refinement approach can disambiguate the object pose effectively and reduce the rotation errors significantly (which result in smoother pose estimations throughout an image sequence). Table 4 shows that the proposed refinement method can help improve estimation accuracy in terms of rotation and translation and address the pose ambiguity problem effectively.

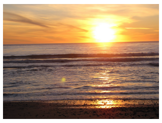
5.2. *Visual Tracking Dataset*

We analyze the performance of the proposed algorithm and state-of-the-art methods on the visual tracking (VT) dataset (Gauglitz et al., 2011) which contains 96 videos and 6,889 frames with 6 templates. These videos are recorded under different moving and lighting conditions with motion-blurs. The camera image size in this dataset is 640×480 pixels. And since the templates have different primary resolutions, we resize each template to 570×420 pixels uniformly. It is a challenging database for pose estimation due to significant viewpoint changes, drastic illumination differences, and noisy camera images.

The evaluation results of the proposed and feature-based methods on six templates under different conditions are shown in Table 5. Different from synthetic images, the color appearance of a template image may change significantly within a video sequence in this real image dataset. The DPE algorithm performs favorably against the feature-based methods under most conditions, especially when distinguishable features cannot be found on a template image.

While PnP algorithms perform well in pose estimation, the success hinges on whether the feature can be well matched. As shown in Figure 10, feature-based approaches do not perform well when motion blurs occur. Similarly, feature-based methods do not estimate pose well on videos listed in Table 5 due to motion blurs. On the other hand, the proposed algorithm can estimate poses well under blur conditions. As motion blurs are likely to occur in AR applications, the proposed algorithm can be better applied to estimate 6-DoF pose than feature-based approaches. However, if the target object appears an extremely flat color in a camera image, the proposed method is likely to

Table 5. Experimental results on the visual tracking dataset (Gauglitz et al., 2011) under different conditions. The best results (excluding the proposed direct pose tracking method) for each condition are highlighted in bold.

		Bricks			Building			Mission			Paris			Sunset			Wood		
																			
Condition	Method	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)	$E_r(°)$	$E_t(\%)$	SR(%)
Unconstrained	SIFT+IPPE	2.98	1.07	*0.40*	2.60	0.89	*6.60*	1.64	0.72	*60.6*	1.61	0.66	*44.0*	3.22	**1.32**	*26.6*	2.04	0.53	*5.00*
	SIFT+OPnP	2.37	0.98	*0.40*	2.60	0.88	*6.80*	1.48	0.73	*61.6*	1.44	0.65	*43.8*	**2.81**	1.43	*28.4*	1.43	**0.41**	*5.00*
	ASIFT+IPPE	2.67	1.05	*37.0*	2.80	0.92	*31.4*	3.03	1.11	*58.0*	1.57	0.72	*91.0*	3.40	2.10	*44.4*	2.54	1.03	*33.0*
	ASIFT+OPnP	1.92	0.93	*37.6*	2.48	0.88	*31.4*	2.35	0.91	*57.4*	1.31	0.85	*91.0*	3.03	2.11	*46.0*	2.20	1.01	*34.8*
	APE	2.12	1.45	***52.0***	1.11	0.92	*88.0*	1.57	1.14	*98.0*	0.75	0.76	*99.0*	4.24	3.89	*43.4*	1.41	2.24	*53.0*
	DPE	**1.00**	**0.72**	***52.0***	**0.93**	**0.66**	***88.4***	**1.17**	**0.70**	***98.2***	**0.47**	**0.54**	***99.8***	3.49	4.12	***46.8***	**0.67**	1.53	***56.0***
	DPT	1.11	0.94	*90.2*	1.23	0.81	*92.2*	1.17	0.70	*98.4*	0.81	0.65	*96.0*	3.69	4.46	*49.9*	0.88	1.43	*91.2*
Panning	SIFT+IPPE	–	–	*0.00*	1.29	0.55	*10.0*	2.18	0.65	*96.0*	3.34	1.04	*40.0*	5.49	0.75	*20.0*	–	–	*0.00*
	SIFT+OPnP	–	–	*0.00*	0.81	**0.54**	*10.0*	2.39	**0.61**	***100***	4.45	1.12	*50.0*	6.79	1.27	*24.0*	–	–	*0.00*
	ASIFT+IPPE	5.91	1.52	*80.0*	7.89	0.98	*2.00*	4.95	1.15	*82.0*	5.85	**0.91**	*44.0*	13.8	2.56	*10.0*	9.94	1.42	*4.00*
	ASIFT+OPnP	5.80	1.40	*80.0*	19.2	5.59	*2.00*	4.95	0.98	*88.0*	5.93	1.16	*62.0*	15.5	3.34	*10.0*	16.0	2.02	*4.00*
	APE	4.27	0.50	***96.0***	1.56	1.08	***100***	1.79	0.96	***100***	3.94	1.05	*74.0*	4.53	**0.56**	***100***	6.03	1.24	*56.0*
	DPE	**1.04**	**0.29**	***96.0***	**0.38**	0.63	***100***	**0.90**	0.89	***100***	**1.52**	0.97	***86.0***	**2.75**	0.68	***100***	**1.05**	**0.93**	***60.0***
	DPT	1.64	0.36	*95.9*	0.38	0.64	*100*	0.95	0.89	*100*	1.51	0.98	*100*	2.68	0.71	*100*	1.33	0.92	*79.6*
Rotation	SIFT+IPPE	1.65	0.34	*44.0*	2.79	**0.50**	*56.0*	1.17	0.42	***100***	1.71	0.37	*98.0*	5.97	0.57	*74.0*	3.76	0.40	*62.0*
	SIFT+OPnP	1.74	0.37	*46.0*	2.69	0.52	*56.0*	1.05	0.41	***100***	1.61	0.33	***100***	5.61	0.66	*84.0*	2.51	**0.39**	*70.0*
	ASIFT+IPPE	2.83	0.39	***100***	6.15	1.24	*76.0*	2.35	**0.36**	***100***	1.35	0.33	***100***	6.68	0.91	*72.0*	3.39	0.47	*94.0*
	ASIFT+OPnP	1.78	0.39	***100***	5.09	1.11	*74.0*	1.66	0.37	***100***	1.23	0.36	***100***	5.69	0.90	*78.0*	2.88	0.49	*98.0*
	APE	1.20	0.25	***100***	2.00	0.66	***100***	1.11	0.42	***100***	0.71	**0.25**	***100***	2.18	0.66	***100***	1.64	0.66	***100***
	DPE	**0.84**	**0.24**	***100***	**1.50**	0.59	***100***	**0.31**	0.46	***100***	**0.56**	0.29	***100***	**0.90**	**0.52**	***100***	**0.98**	0.55	***100***
	DPT	0.84	0.24	*100*	1.50	0.59	*100*	0.32	0.46	*100*	0.55	0.29	*100*	0.88	0.51	*100*	0.99	0.54	*100*
Perspective Distortion	SIFT+IPPE	2.99	0.46	*58.0*	4.38	**0.40**	*34.0*	2.77	0.43	*76.0*	3.98	0.40	*76.0*	6.56	0.87	*58.0*	4.70	0.59	*20.0*
	SIFT+OPnP	1.45	0.30	*58.0*	2.62	0.45	*34.0*	**0.68**	0.53	*76.0*	1.53	0.45	*76.0*	4.79	0.74	*62.0*	6.23	**0.43**	*24.0*
	ASIFT+IPPE	3.01	**0.25**	***72.0***	4.99	0.43	*68.0*	3.74	**0.35**	*80.0*	3.07	**0.34**	*84.0*	4.96	**0.64**	*58.0*	3.69	0.75	*66.0*
	ASIFT+OPnP	1.55	0.29	***72.0***	3.51	0.54	*68.0*	1.95	0.39	*80.0*	1.78	0.51	*84.0*	3.73	0.87	*62.0*	2.07	0.82	*66.0*
	APE	1.81	0.94	*56.0*	0.97	0.77	***92.0***	1.35	0.56	***86.0***	0.69	0.42	***90.0***	2.44	2.32	*68.0*	1.74	1.34	***68.0***
	DPE	**0.89**	0.29	*56.0*	**0.74**	0.51	***92.0***	0.81	0.52	***86.0***	**0.43**	0.46	***90.0***	**1.47**	1.96	***78.0***	**0.56**	0.86	***68.0***
	DPT	0.72	0.34	*93.9*	0.71	0.51	*100*	0.84	0.61	*95.9*	0.57	0.68	*98.0*	1.61	1.63	*75.5*	0.62	1.13	*87.8*
Zoom	SIFT+IPPE	2.51	0.53	*6.00*	3.28	0.34	*26.0*	4.01	0.42	***100***	3.09	0.40	***100***	9.75	0.94	*60.0*	4.23	0.45	*40.0*
	SIFT+OPnP	1.15	0.38	*6.00*	3.14	0.30	*28.0*	2.30	**0.40**	*98.0*	2.73	0.43	***100***	7.42	0.91	*60.0*	2.83	0.46	*42.0*
	ASIFT+IPPE	4.91	0.76	*64.0*	4.60	0.56	*58.0*	5.24	0.67	*76.0*	2.54	**0.20**	*74.0*	10.5	1.05	*50.0*	4.10	**0.43**	*48.0*
	ASIFT+OPnP	3.32	0.65	*64.0*	3.95	0.52	*58.0*	3.36	0.48	*80.0*	1.67	0.36	*76.0*	6.47	1.18	*56.0*	4.33	0.50	*54.0*
	APE	3.37	0.77	***94.0***	1.73	0.33	***100***	3.13	0.63	***100***	1.22	0.55	***100***	5.58	**0.74**	***100***	3.79	1.06	***100***
	DPE	**1.14**	**0.33**	***94.0***	**0.86**	**0.27**	***100***	**1.94**	0.51	***100***	**0.50**	0.45	***100***	**2.50**	0.80	***100***	**0.87**	0.61	***100***
	DPT	1.16	0.33	*100*	0.87	0.27	*100*	1.98	0.51	*100*	0.52	0.45	*100*	2.43	0.80	*100*	0.93	0.58	*100*
Static Lighting	SIFT+IPPE	1.51	0.83	*27.5*	2.75	0.98	*20.0*	1.09	0.48	*81.3*	1.56	0.79	*72.5*	2.28	**0.87**	*57.5*	1.01	**0.50**	*21.3*
	SIFT+OPnP	1.49	0.91	*28.7*	2.42	1.18	*20.0*	0.77	**0.43**	*81.3*	1.58	0.86	*72.5*	**1.94**	0.91	*60.0*	1.00	0.52	*21.3*
	ASIFT+IPPE	1.20	**0.81**	***75.0***	2.77	0.88	*42.5*	1.43	0.48	***100***	1.28	**0.65**	***100***	2.66	1.73	*47.5*	1.80	0.58	*52.5*
	ASIFT+OPnP	**1.09**	0.82	***75.0***	2.41	0.82	*42.5*	1.27	0.45	***100***	1.23	0.76	***100***	2.45	1.59	*62.5*	1.46	0.58	*52.5*
	APE	1.75	1.44	*71.3*	0.90	0.50	***100***	0.95	0.60	***100***	1.24	0.72	***100***	2.97	3.59	*81.3*	1.61	1.85	***85.0***
	DPE	1.20	1.06	*71.3*	**0.85**	**0.40**	***100***	**0.61**	0.51	***100***	**1.03**	0.68	***100***	2.24	2.44	***82.5***	**0.94**	0.78	***85.0***
	DPT	1.20	1.05	*100*	0.85	0.39	*100*	0.61	0.51	*100*	1.02	0.68	*100*	2.85	3.13	*100*	0.91	0.72	*100*
Dynamic Lighting	SIFT+IPPE	1.38	0.41	*13.0*	1.81	0.89	*17.0*	1.16	0.55	*78.0*	1.12	0.47	*38.0*	1.45	**0.67**	*44.0*	1.08	**0.42**	*28.0*
	SIFT+OPnP	1.37	0.43	*13.0*	1.59	0.90	*17.0*	0.98	0.58	*77.0*	1.13	0.52	*38.0*	**1.29**	0.70	*48.0*	1.01	0.43	*28.0*
	ASIFT+IPPE	1.22	**0.36**	*62.0*	2.81	1.10	*38.0*	1.53	0.54	***100***	0.95	0.48	***100***	3.31	1.33	*47.0*	1.79	0.56	*48.0*
	ASIFT+OPnP	1.14	0.38	***63.0***	3.01	1.15	*37.0*	1.42	0.55	***100***	0.92	0.53	***100***	2.60	1.33	*48.0*	1.47	0.59	*51.0*
	APE	1.25	0.71	*40.0*	**1.06**	0.68	***98.0***	0.99	0.70	***100***	0.65	**0.33**	*84.0*	3.26	3.10	*72.0*	1.26	1.31	***52.0***
	DPE	**1.00**	0.47	*40.0*	1.20	**0.65**	***98.0***	**0.47**	**0.52**	***100***	**0.63**	0.41	*84.0*	2.75	3.19	***77.0***	**0.82**	0.72	***52.0***
	DPT	1.00	0.45	*100*	1.20	0.66	*100*	0.46	0.52	*100*	0.63	0.42	*100*	3.29	3.67	*100*	0.81	0.63	*100*

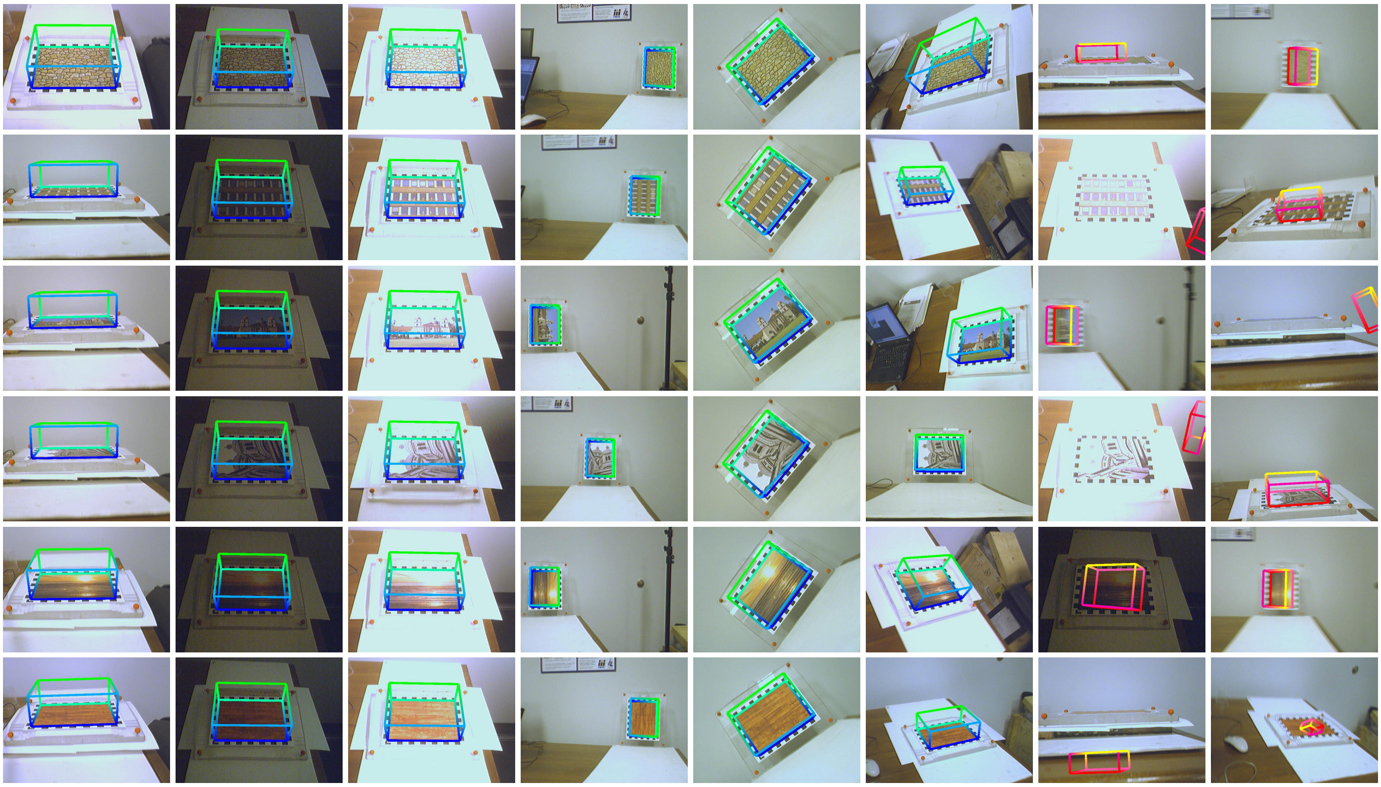


Figure 11. Estimation results by the proposed DPE method on the visual tracking dataset (Gauglitz et al., 2011) under different conditions. The success cases are represented with rendered cyan boxes, and the failure cases are represented with rendered magenta boxes.

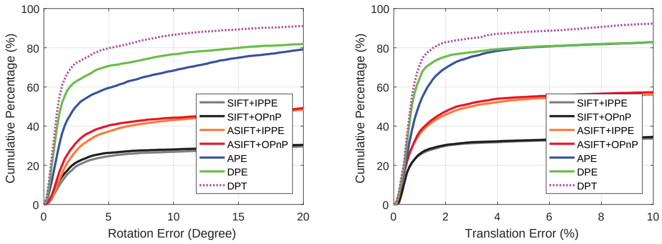


Figure 12. Cumulative percentage of poses whose rotation or translation errors are under thresholds specified in the x-axis over experiments on the visual tracking dataset (Gauglitz et al., 2011). There is a total of 6,889 poses estimated by each pose estimation approach.

fail because the appearance between the template and its local patches are almost indistinguishable.

Sample pose estimation results from the proposed DPE method are shown in Figure 11. The cumulative percentage of estimated poses according to different translation and rotation errors are shown in Figure 12. Overall, the proposed direct method performs favorably against the feature-based approaches within the success rate of 77.76%. The success rate of the SIFT-based and ASIFT-based approaches are 29.98% and 48.52% respectively.

Note that the proposed pose refinement approach can also be regarded as a direct pose tracking (DPT) algorithm. The evaluation results of the DPT method on the VT dataset are shown in Table 5, Figure 10, and Figure 12. If the DPT method loses track of the object pose (i.e., the rotation or translation error is larger than the pre-defined threshold, i.e., δ_r and δ_t), we reset the initial object pose in the current frame as the object pose in the previous frame. Overall, the proposed DPT method can track object poses well. The DPT algorithm can be integrated with the DPE method for more robust performance with certain re-initialization schemes (e.g., periodic restarts).

5.3. *Object Pose Tracking Dataset*

We evaluate the proposed algorithm and feature-based methods on the object pose tracking (OPT) benchmark dataset (Wu et al., 2017). For 2D objects, it contains 138 videos with 20,988 frames. These videos are recorded under four designed motion patterns and five camera speeds controlled by a programmable robotic arm. Furthermore, these videos contain two different lighting conditions and a free-motion case. The frame size in this dataset is 1920×1080 pixels, and we resize each template to 300×300 pixels. Sample images rendered according to the pose estimated by the proposed DPE method on this OPT dataset are shown in Figure 13.

The pose tracking results of all evaluated algorithms under *Flashing Light*, *Moving Light*, and *Free Motion* conditions with six templates and different texture levels are shown in Table 6. Similar to the results in Section 5.1 and Section 5.2, feature-based methods do not perform well on the template images with less texture or structure. In contrast, the proposed DPE method is able to track object poses well except the *Wing* image. When

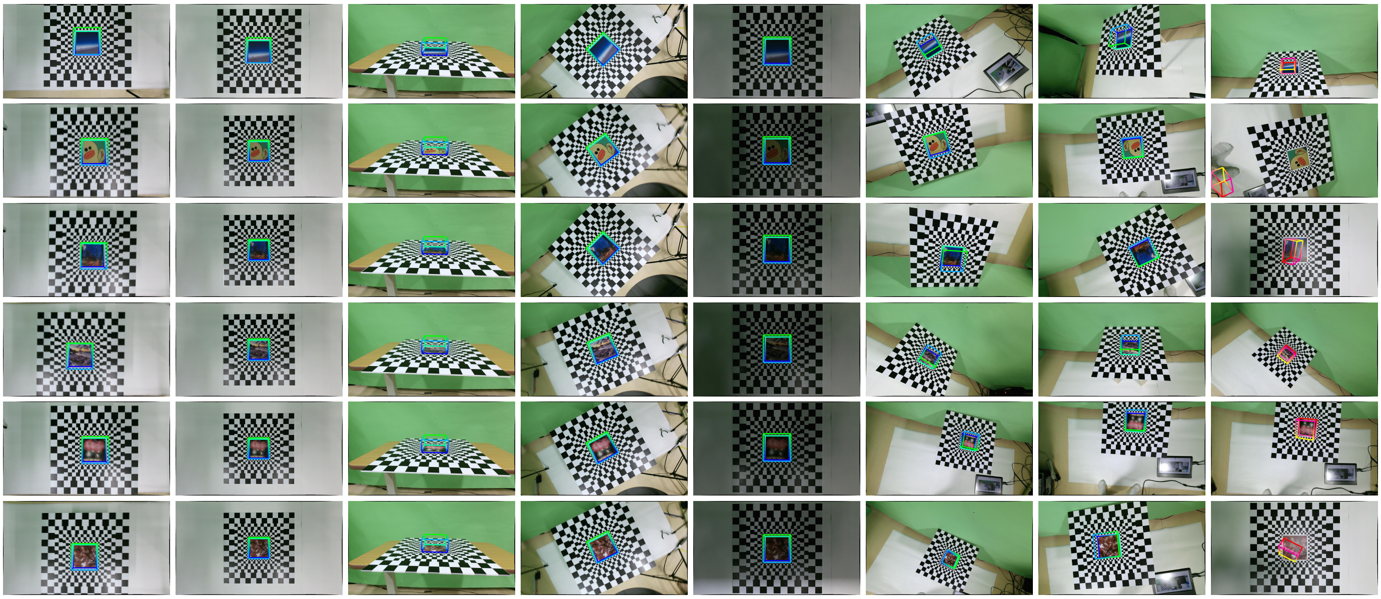
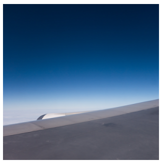
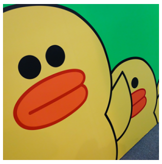
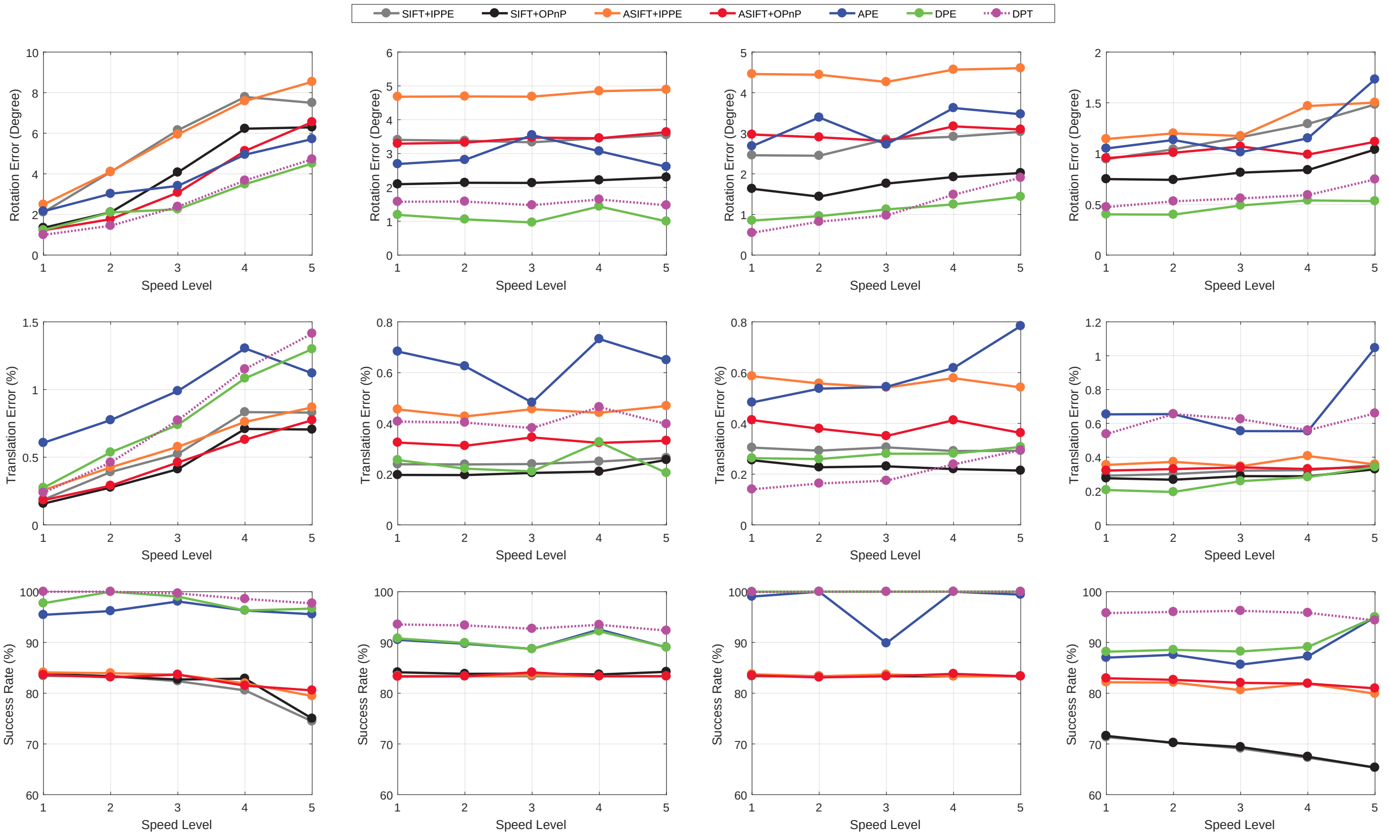


Figure 13. Estimation results by the proposed DPE method on the object pose tracking dataset (Wu et al., 2017) under different conditions. The success cases are represented with rendered cyan boxes, and the failure cases are represented with rendered magenta boxes.

Table 6. Experimental results of the object pose tracking dataset (Wu et al., 2017) under different conditions. The best results (excluding the proposed direct pose tracking method) for each condition are highlighted in bold.

		Wing			Duck			City			Beach			Maple			Firework		
																			
Condition	Method	$E_r(^\circ)$	$E_t(\%)$	SR(%)	$E_r(^\circ)$	$E_t(\%)$	SR(%)	$E_r(^\circ)$	$E_t(\%)$	SR(%)	$E_r(^\circ)$	$E_t(\%)$	SR(%)	$E_r(^\circ)$	$E_t(\%)$	SR(%)	$E_r(^\circ)$	$E_t(\%)$	SR(%)
Flashing Light	SIFT+IPPE	10.6	2.96	*1.24*	6.99	0.43	***100***	2.11	0.20	***100***	2.80	0.16	***100***	1.80	0.15	***100***	1.63	0.12	***100***
	SIFT+OPnP	14.2	2.33	*9.32*	5.94	0.33	***100***	**0.86**	**0.10**	*99.4*	0.83	0.09	***100***	**0.35**	0.08	***100***	**0.23**	0.07	***100***
	ASIFT+IPPE	14.6	3.27	*4.35*	6.36	0.50	***100***	3.01	0.28	***100***	1.79	0.19	***100***	2.43	0.22	***100***	2.73	0.25	***100***
	ASIFT+OPnP	17.5	2.84	*3.11*	3.58	0.34	***100***	1.47	0.16	***100***	0.88	0.18	***100***	1.35	0.12	***100***	0.91	0.16	***100***
	APE	10.4	**1.51**	*36.0*	2.12	0.22	***100***	1.95	0.56	***100***	1.28	0.28	***100***	1.98	0.41	***100***	2.00	0.34	***100***
	DPE	**8.32**	1.52	***42.2***	**0.72**	**0.05**	***100***	1.08	0.19	***100***	**0.50**	**0.09**	*99.4*	0.50	**0.05**	*98.1*	0.38	**0.04**	***100***
	DPT	6.46	1.72	*86.3*	0.76	0.05	*99.4*	1.16	0.17	*100*	0.47	0.09	*100*	0.56	0.05	*97.5*	0.43	0.05	*100*
Moving Light	SIFT+IPPE	17.8	0.69	*0.61*	7.54	0.63	*94.5*	2.52	0.22	***100***	2.60	0.15	***100***	1.87	0.15	***100***	1.64	0.13	***100***
	SIFT+OPnP	15.2	2.75	*8.54*	5.95	0.50	*94.5*	**1.02**	**0.11**	***100***	**0.69**	**0.09**	***100***	**0.55**	**0.09**	***100***	**0.22**	0.07	***100***
	ASIFT+IPPE	15.6	2.97	*1.83*	7.13	0.61	***100***	4.71	0.41	*99.4*	1.74	0.20	***100***	2.42	0.20	***100***	2.68	0.27	***100***
	ASIFT+OPnP	19.4	**0.38**	*0.61*	5.10	0.46	***100***	2.73	0.29	*99.4*	0.84	0.15	***100***	0.98	0.12	***100***	0.80	0.18	***100***
	APE	11.3	4.98	*27.4*	4.24	0.37	*99.4*	5.43	0.71	*55.5*	3.64	0.35	*75.0*	6.09	1.03	*62.2*	3.26	0.54	*95.1*
	DPE	**8.41**	4.38	***45.1***	**2.14**	**0.12**	***100***	2.34	0.18	*56.7*	1.51	0.09	*77.4*	3.20	0.32	*59.8*	0.71	**0.04**	*94.5*
	DPT	9.22	1.92	*64.4*	1.96	0.11	*100*	2.59	0.19	*98.8*	1.42	0.10	*99.4*	4.08	0.37	*82.2*	0.76	0.05	*100*
Free Motion	SIFT+IPPE	7.55	3.95	*1.15*	5.80	0.59	*93.2*	1.00	0.28	***100***	0.61	0.42	*99.9*	0.73	0.39	***100***	1.38	0.39	***100***
	SIFT+OPnP	9.81	2.87	*2.04*	3.68	0.57	*96.8*	0.77	0.27	***100***	0.61	0.41	***100***	0.72	0.38	***100***	1.09	0.38	***100***
	ASIFT+IPPE	11.6	**2.54**	*0.38*	7.89	1.18	*90.6*	2.43	0.39	*99.4*	0.95	0.53	*99.9*	1.45	0.49	*96.4*	1.78	0.39	*98.7*
	ASIFT+OPnP	11.4	5.38	*1.15*	6.53	0.90	*96.7*	2.03	0.36	*99.7*	0.91	0.52	***100***	1.39	0.49	*99.9*	1.55	0.36	*99.7*
	APE	6.14	5.16	*56.1*	2.73	0.31	***98.7***	1.35	0.66	***100***	1.53	0.86	*83.7*	3.18	1.98	*98.3*	1.79	0.55	***100***
	DPE	**4.84**	4.41	***59.7***	**1.16**	**0.23**	***98.7***	**0.60**	**0.18**	***100***	**0.54**	**0.27**	*91.1*	**0.65**	**0.34**	*99.1*	**1.05**	**0.30**	***100***
	DPT	4.52	3.14	*69.5*	0.88	0.18	*100*	0.55	0.22	*100*	0.49	0.26	*99.6*	0.58	0.26	*100*	1.02	0.30	*100*



(a) Translation (b) Zoom (c) In-plane Rotation (d) Out-of-plane Rotation

Figure 14. Experimental results of the object pose tracking dataset Wu et al. (2017) in four designed motion patterns with different speeds.

a template image does not contain sufficient structural information, the proposed direct method may estimate erroneous poses which cover only parts of the template image, as shown in the failure cases in Figure 13. The proposed method does not perform well on images when drastic color distortion occurs, e.g., under *Moving Light* condition, as the appearance distance metric is less effective in such scenarios.

The pose tracking results of the template images in different motion patterns and speed are shown in Figure 14. Since the images in the *Translation* condition are more blurry than those in other motion patterns at higher speed, the plot trends of the evaluation results under this condition are similar as those under the *Gaussian Blur* conditions in Figure 6. In contrast, the other three motion patterns do not result in blurry images at the highest speed, the performance of all approaches under conditions at different speeds are similar. As all the evaluated approaches are scale and rotation invariant, they all perform favorably on template images with the *Zoom* and *In-plane Rotation* patterns. However, the success rates of SIFT-based methods are lower in the *Out-of-plane Rotation* motion pattern as they are not invariant under perspective distortion.

We evaluate the proposed DPT algorithm on the OPT dataset to analyze the tracking performance using the same experimental setting as that described in Section 5.2, Figure 14 and Table 6 show that the DPT algorithm can track object poses well on most template images except one. As discussed above, the proposed DPT method does not work well on images, e.g., *Wing*, without sufficient structure for pose estimation based on appearance. The curves of cumulative percentages of poses estimated by the evaluated algorithms on the OPT dataset are shown in Figure 15. Overall, the proposed direct method performs favorably against feature-based approaches with a success rate of 91.27%. The success rates of the SIFT-based and ASIFT-based approaches are 79.46% and 82.74%, respectively.

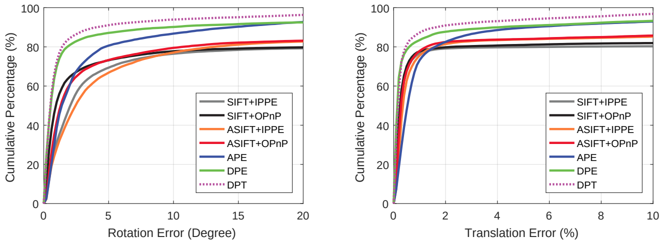


Figure 15. Cumulative percentage of poses whose rotation or translation errors are under thresholds specified in the x-axis over experiments on the object pose tracking dataset (Wu et al., 2017). There is a total of 20,988 poses estimated by each pose estimation approach.

6. Conclusions

In this paper, we propose a robust direct method for 6-DoF pose estimation based on two main steps. First, the pose of a planar target with respect to a calibrated camera is approximately estimated using an efficient coarse-to-fine scheme. Next, we use the LK-based method to further refine and disambiguate the object pose. Extensive experimental evaluations on both synthetic image and real image datasets demonstrate the proposed algorithm performs favorably against two state-of-the-art feature-based pose estimation approaches in terms of robustness and accuracy under several varying conditions. We have also implemented the proposed algorithm on a GPGPU platform as the algorithm can be easily parallelized.

Acknowledgments

This work is partially supported by Ministry of Science and Technology (MOST), Taiwan (R.O.C.), under Grant MOST103-2221-E-002 -268 -MY3 and MOST106-2221-E-002 -215 -MY3.

Appendix A. Derivation Details of Bounded Steps

For presentation clarity, we use the notation c_a for $\cos(\theta_a)$ and s_a for $\sin(\theta_a)$, where a stands for z_c, x, or z_t. As discussed in Section 4.1, the rotation can be factorized as:

$$\begin{aligned}\mathbf{R} &= \mathbf{R}_z(\theta_{z_c})\mathbf{R}_x(\theta_x)\mathbf{R}_z(\theta_{z_t}) \\ &= \begin{bmatrix} c_{z_c}c_{z_t} - c_x s_{z_c} s_{z_t} & -c_x c_{z_t} s_{z_c} - c_{z_c} s_{z_t} & s_x s_{z_c} \\ c_{z_t} s_{z_c} + c_x c_{z_c} s_{z_t} & c_x c_{z_c} c_{z_t} - s_{z_c} s_{z_t} & -s_x c_{z_c} \\ s_x s_{z_t} & s_x c_{z_t} & c_x \end{bmatrix}.\end{aligned} \tag{A.1}$$

Our objective is to construct an ε-covering pose set $\mathcal{S}$ based on (6) and (7). In this work, we construct $\mathcal{S}$ by first determining bounded steps for horizontal distance t_z and tilt angle θ_x. Next, the bounded steps for the other dimensions θ_{z_c}, θ_{z_t}, t_x, and t_y can be determined based on t_z and θ_x. Let $\theta_{z_t'} = \theta_{z_t} + \Delta\theta_{z_t}$, we obtain the following equation based on the current t_z and θ_x,

$$\begin{aligned} d(T_{\mathbf{p}_{\theta_{z_t}}}(\mathbf{x}_i), T_{\mathbf{p}_{\theta_{z_t}+\Delta\theta_{z_t}}}(\mathbf{x}_i)) &= \sqrt{f_x^2\alpha_{\theta_{z_t}}^2 + f_y^2 c_x^2 \beta_{\theta_{z_t}}^2} \\ &\leq \sqrt{f_x^2\alpha_{\theta_{z_t}}^2 + f_y^2 \beta_{\theta_{z_t}^2}} \\ &= O\left(\frac{\Delta\theta_{z_t}}{t_z + k\sin(\theta_x)}\right), \end{aligned} \tag{A.2}$$

$$\alpha_{\theta_{z_t}} = \frac{c_{z_t}x - s_{z_t}y}{s_x(s_{z_t}x + c_{z_t}y) + t_z} - \frac{c_{z_t'}x - s_{z_t'}y}{s_x(s_{z_t'}x + c_{z_t'}y) + t_z}, \tag{A.3}$$

$$\beta_{\theta_{z_t}} = \frac{s_{z_t}x + c_{z_t}y}{s_x(s_{z_t}x + c_{z_t}y) + t_z} - \frac{s_{z_t'}x + c_{z_t'}y}{s_x(s_{z_t'}x + c_{z_t'}y) + t_z}, \tag{A.4}$$

where k denotes any constant in the range of $[-\sqrt{2}, \sqrt{2}]$. An illustrative example of (A.2) is shown in Figure A.16. To make (A.2) satisfy the constraint in (6), we set the step size,

$$\Delta\theta_{z_t} = \Theta(\varepsilon(t_z + k\sin(\theta_x))), \tag{A.5}$$

where larger k means larger bounded steps for constructing $\mathcal{S}$. We set k to be 0 for $\Delta\theta_{z_t}$ in the proposed method.

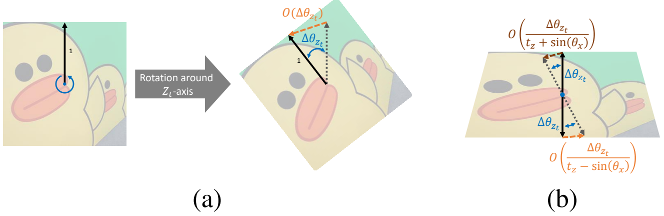

Rotation around Z_t-axis
1
1
$O(\Delta\theta_{z_t})$
$\Delta\theta_{z_t}$
$O\left(\frac{\Delta\theta_{z_t}}{t_z + \sin(\theta_x)}\right)$
$\Delta\theta_{z_t}$
$\Delta\theta_{z_t}$
$O\left(\frac{\Delta\theta_{z_t}}{t_z - \sin(\theta_x)}\right)$
(a)
(b)

Figure A.16. (a) 2D illustration of rotation around Z_t-axis. The linear distance (orange solid line) between points before and after applying rotation is bounded by the arc length (brown dotted line). (b) 3D illustration of rotation around Z_t-axis. The linear distance between points is a function of tilt angle θ_x.

As θ_{z_t} denotes 2D image rotation of the planar target, it does not influence the bounded steps for θ_{z_c}. Let $\theta_{z_c'} = \theta_{z_c} + \Delta\theta_{z_c}$, we obtain the following equation depending on the current t_z and θ_x:

$$\begin{aligned} d(T_{\mathbf{p}_{\theta_{z_c}}}(\mathbf{x}_i), T_{\mathbf{p}_{\theta_{z_c}+\Delta\theta_{z_c}}}(\mathbf{x}_i)) &= \sqrt{f_x^2\alpha_{\theta_{z_c}}^2 + f_y^2\beta_{\theta_{z_c}}^2} \\ &= O\left(\frac{\Delta\theta_{z_c}}{t_z + k\sin(\theta_x)}\right), \end{aligned} \tag{A.6}$$

$$\alpha_{\theta_{z_c}} = \frac{c_{z_c}x - c_x s_{z_c}y}{s_x y + t_z} - \frac{c_{z_c'}x - c_x s_{z_c'}y}{s_x y + t_z}, \tag{A.7}$$

$$\beta_{\theta_{z_c}} = \frac{s_{z_c}x + c_x c_{z_c}y}{s_x y + t_z} - \frac{s_{z_c'}x + c_x c_{z_c'}y}{s_x y + t_z}. \tag{A.8}$$

We can realize (A.6) in a similar way to (A.2). To make (A.6) satisfy the constraint in (6), we set the step size:

$$\Delta\theta_{z_c} = \Theta(\varepsilon(t_z + k\sin(\theta_x))) = \Theta(\varepsilon(t_z)), \tag{A.9}$$

which k is set to 0.

As the bounded steps for t_x and t_y are also influenced by horizontal distance t_z and tilt angle θ_x only, we have

$$\begin{aligned} d(T_{\mathbf{p}_{t_x}(\mathbf{x}_i)}, T_{\mathbf{p}_{t_x+\Delta t_x}}(\mathbf{x}_i)) &= \sqrt{f_x^2\alpha_{t_x}^2 + f_y^2\beta_{t_x}^2} \\ &= O\left(\frac{\Delta t_x}{t_z + k\sin(\theta_x)}\right), \end{aligned} \tag{A.10}$$

$$\alpha_{t_x} = \frac{x + t_x}{s_x y + t_z} - \frac{x + t_x + \Delta t_x}{s_x y + t_z}, \tag{A.11}$$

$$\beta_{t_x} = \frac{y}{s_x y + t_z} - \frac{y}{s_x y + t_z}, \tag{A.12}$$

and:

$$\begin{aligned} d(T_{\mathbf{p}_{t_y}(\mathbf{x}_i)}, T_{\mathbf{p}_{t_y+\Delta t_y}}(\mathbf{x}_i)) &= \sqrt{f_x^2\alpha_{t_y}^2 + f_y^2\beta_{t_y}^2} \\ &= O\left(\frac{\Delta t_y}{t_z + k\sin(\theta_x)}\right), \end{aligned} \tag{A.13}$$

$$\alpha_{t_y} = \frac{x}{s_x y + t_z} - \frac{x}{s_x y + t_z}, \tag{A.14}$$

$$\beta_{t_y} = \frac{y+t_y}{s_x y + t_z} - \frac{y+t_y+\Delta t_y}{s_x y + t_z}. \tag{A.15}$$

To make (A.10) and (A.13) satisfy the constraint in (6), we set these step sizes,

$$\Delta t_x = \Theta(\varepsilon(t_z + k\sin(\theta_x))) = \Theta(\varepsilon(t_z - \sqrt{2}\sin(\theta_x))), \tag{A.16}$$

$$\Delta t_y = \Theta(\varepsilon(t_z + k\sin(\theta_x))) = \Theta(\varepsilon(t_z - \sqrt{2}sin(\theta_x))). \tag{A.17}$$

as k is set to $-\sqrt{2}$ for pratical consideration.

References

Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.T., 2012. Learning from data. AMLBook.

Alahi, A., Ortiz, R., Vandergheynst, P., 2012. Freak: Fast retina keypoint, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Alexe, B., Petrescu, V., Ferrari, V., 2011. Exploiting spatial overlap to efficiently compute appearance distances between image windows.

Baker, S., Matthews, I., 2001. Equivalence and efficiency of image alignment algorithms, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (surf). Computer Vision and Image Understanding 110, 346–359.

Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. Brief: Binary robust independent elementary features, in: Proceedings of European Conference on Computer Vision.

Chi, Y.T., Ho, J., Yang, M.H., 2011. A direct method for estimating planar projective transform, in: Proceedings of Asian Conference on Computer Vision.

Chum, O., Matas, J., 2005. Matching with prosac-progressive sample consensus, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Collins, T., Bartoli, A., 2014. Infinitesimal Plane-Based Pose Estimation. International Journal of Computer Vision 109, 252–286.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. Introduction to algorithms. MIT Press.

Crivellaro, A., Lepetit, V., 2014. Robust 3d tracking with descriptor fields, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Eberly, D., 2008. Euler angle formulas. Geometric Tools, LLC, Technical Report .

Engel, J., Schöps, T., Cremers, D., 2014. Lsd-slam: Large-scale direct monocular slam, in: Proceedings of European Conference on Computer Vision.

Ferraz, L., Binefa, X., Moreno-Noguer, F., 2014a. Leveraging feature uncertainty in the pnp problem, in: Proceedings of British Machine Vision Conference.

Ferraz, L., Binefa, X., Moreno-Noguer, F., 2014b. Very fast solution to the pnp problem with algebraic outlier rejection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395.

Fragoso, V., Sen, P., Rodriguez, S., Turk, M., 2013. Evsac: accelerating hypotheses generation by modeling matching scores with extreme value theory, in: Proceedings of IEEE International Conference on Computer Vision.

Gallego, G., Yezzi, A., 2015. A compact formula for the derivative of a 3-d rotation in exponential coordinates. Journal of Mathematical Imaging and Vision 51, 378–384.

Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F., 2003. Complete solution classification for the perspective-three-point problem. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 930–943.

Gauglitz, S., Höllerer, T., Turk, M., 2011. Evaluation of interest point detectors and feature descriptors for visual tracking. International Journal of Computer Vision 94, 335–360.

Grassia, F.S., 1998. Practical parameterization of rotations using the exponential map. Journal of Graphics Tools 3, 29–48.

Hager, G.D., Belhumeur, P.N., 1998. Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 1025–1039.

Henriques, J.F., Martins, P., Caseiro, R.F., Batista, J., 2014. Fast training of pose detectors in the fourier domain.

Hesch, J.A., Roumeliotis, S.I., 2011. A direct least-squares (dls) method for pnp, in: Proceedings of IEEE International Conference on Computer Vision.

Jegou, H., Douze, M., Schmid, C., 2008. Hamming embedding and weak geometric consistency for large scale image search, in: Proceedings of European Conference on Computer Vision.

Ke, T., Roumeliotis, S.I., 2017. An efficient algebraic solution to the perspective-three-point problem, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Kearns, M.J., Vazirani, U.V., 1994. An introduction to computational learning theory. MIT Press.

Klein, G., Murray, D., 2007. Parallel tracking and mapping for small ar workspaces, in: Proceedings of IEEE International Symposium on Mixed and Augmented Reality.

Kneip, L., Li, H., Seo, Y., 2014. Upnp: An optimal o (n) solution to the absolute pose problem with universal applicability, in: Proceedings of European Conference on Computer Vision.

Kneip, L., Scaramuzza, D., Siegwart, R., 2011. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Korman, S., Reichman, D., Tsur, G., Avidan, S., 2017. Fast-match: Fast affine template matching. International Journal of Computer Vision 121, 111–125.

Kukelova, Z., Bujnak, M., Pajdla, T., 2008. Automatic generator of minimal problem solvers, in: Proceedings of European Conference on Computer Vision.

Lepetit, V., Moreno-Noguer, F., Fua, P., 2009. Epnp: An accurate O(n) solution to the pnp problem. International Journal of Computer Vision 81, 155–166.

Leutenegger, S., Chli, M., Siegwart, R.Y., 2011. BRISK: Binary Robust Invariant Scalable Keypoints, in: Proceedings of IEEE International Conference on Computer Vision.

Li, S., Xu, C., 2011. Efficient lookup table based camera pose estimation for augmented reality 22, 47–58.

Li, S., Xu, C., Xie, M., 2012. A robust o(n) solution to the perspective-n-point problem. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 1444–1450.

Lieberknecht, S., Benhimane, S., Meier, P., Navab, N., 2009. A dataset and evaluation methodology for template-based tracking algorithms, in: Proceedings of IEEE International Symposium on Mixed and Augmented Reality.

Lim, H., Sinha, S.N., Cohen, M.F., Uyttendaele, M., 2012. Real-time image-based 6-dof localization in large-scale environments, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Lin, C.H., Lucey, S., 2017. Inverse compositional spatial transformer networks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60.

Lu, C., Hager, G., Mjolsness, E., 2000a. Fast and globally convergent pose estimation from video images. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 610–622.

Lu, C.P., Hager, G.D., Mjolsness, E., 2000b. Fast and globally convergent pose estimation from video images. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 610–622.

Lucas, B.D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision, pp. 674–679.

Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G., 2010. Adaptive and generic corner detection based on the accelerated segment test, in: Proceedings of European Conference on Computer Vision.

Malis, E., 2004. Improving vision-based control using efficient second-order minimization techniques, in: Proceedings of IEEE International Conference on Robotics and Automation.

Mur-Artal, R., Tardós, J.D., 2014. Fast relocalisation and loop closing in keyframe-based slam, in: Proceedings of IEEE International Conference on Robotics and Automation.

Oberkampf, D., DeMenthon, D.F., Davis, L.S., 1993. Iterative pose estimation using coplanar points, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Orozco, J., Rudovic, O., Gonzàlez, J., Pantic, M., 2013. Hierarchical on-line appearance-based tracking for 3d head pose, eyebrows, lips, eyelids and irises. Image and Vision Computing 31, 322–340.

Pele, O., Werman, M., 2007. Accelerating pattern matching or how much can you slide?, in: Proceedings of Asian Conference on Computer Vision.

Rosten, E., Drummond, T., 2006. Machine learning for high-speed corner detection, in: Proceedings of European Conference on Computer Vision.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. Orb: an efficient alternative to sift or surf, in: Proceedings of IEEE International Conference on Computer Vision.

Schweighofer, G., Pinz, A., 2006. Robust Pose Estimation from a Planar Target. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 2024–2030.

Shum, H.Y., Szeliski, R., 2001. Construction of panoramic image mosaics with global and local alignment, in: Panoramic Vision, pp. 227–268.

Tseng, H.Y., Wu, P.C., Lin, Y.S., Chien, S.Y., 2017. D-pet: A direct 6 dof pose estimation and tracking system on graphics processing units, in: Proceedings of IEEE International Symposium on Circuits and Systems.

Tseng, H.Y., Wu, P.C., Yang, M.H., Chien, S.Y., 2016. Direct 3d pose estimation of a planar target, in: Proceedings of IEEE Winter Conference on Applications of Computer Vision.

Wikipedia, 2018. Delone set — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Delone%20set&oldid=795315991. [Online; accessed 09-February-2018].

Wu, P.C., Lee, Y.Y., Tseng, H.Y., Ho, H.I., Yang, M.H., Chien, S.Y., 2017. A benchmark dataset for 6dof object pose tracking, in: Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct).

Wu, P.C., Tsai, Y.H., Chien, S.Y., 2014. Stable pose tracking from a planar target with an analytical motion model in real-time applications, in: Proceedings of IEEE International Workshop on Multimedia Signal Processing.

Xiong, X., De la Torre, F., 2015. Global supervised descent method, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Yu, G., Morel, J.M., 2011. Asift: A new framework for fully affine invariant image comparison. Image Processing On Line .

Zheng, Y., Kuang, Y., Sugimoto, S., Astrom, K., Okutomi, M., 2013. Revisiting the PnP Problem: A Fast, General and Optimal Solution, in: Proceedings of IEEE International Conference on Computer Vision.