國立臺灣大學電機資訊學院電子工程學研究所

碩士論文

Graduate Institute of Electronics Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

針對單一平面目標物之三維姿態的直接分析：演算法和系統實作

Direct 3D Pose Estimation of a Planar Target: Algorithm and Implementation

曾泓諭

Hung-Yu Tseng

指導教授：簡韶逸 博士

Advisor: Shao-Yi Chien, Ph.D.

中華民國 105 年 6 月

June 2016

doi:10.6342/NTU201600474

國立臺灣大學碩士學位論文
口試委員會審定書

針對單一平面目標物之三維姿態的直接分析：
演算法和系統實作

Direct 3D Pose Estimation of a Planar Target:
Algorithm and Implementation

本論文係曾泓諭君（R03943005）在國立臺灣大學電子工程學研究所完成之碩士學位論文，於民國一百零五年六月十三日承下列考試委員審查通過及口試及格，特此證明

口試委員：

簡韶逸
（指導教授）

楊明玄　　黃朝宗

莊永裕　　傅立成

系主任、所長　劉深淵

Direct 3D Pose Estimation of a Planar Target: Algorithm and Implementation

By

Hung-Yu Tseng

THESIS

Submitted in partial fulfillment of the requirement
for the degree of Master of Science in Electronics Engineering
at National Taiwan University
Taipei, Taiwan, R.O.C.

June. 2016

Approved by：

Chao-Tsung Huang

Advised by：

Approved by Director： Shen-Iuan

致謝

在完成這篇碩士論文的當下，心中只有滿滿的感謝。在研究的這條路上遇到了許多貴人，給了我許多的幫助，其中我要特別感謝三個人。第一位是我的指導老師簡韶逸教授，從大三修習老師的專題研究開始，便很敬佩老師的研究態度，特別是對於吸收新知的熱情。老師給予我很大的研究空間，並沒有特別限制我研究的走向。除了指導之外，老師更會盡其所能的尋找相關資源，甚至是實驗室外的專業人士來幫助我們。對於能擁有一位為學生著想的指導老師，我滿懷感恩。

第二位是我的共同指導老師楊明玄老師，在 2015 年春天第一次與老師 Skype 通話以來，就深深的感受到老師對於台灣學生們的照顧，包括投稿 Conference 時帶著我們日以繼夜的修改 paper、每周 meeting 時給予許多寶貴的意見、幫助我們找實習、還有在我們低潮時鼓勵我們，特別是 how to survive in this business。回想起來，真的很感謝老師如此的照顧與關心我們。

第三位是我的實驗室學長吳柏辰學長，學長是我研究生涯中最重要的貴人。從專題以來，無論是基礎知識、看 paper 的能力、做投影片、上台報告、做研究甚至是寫程式，學長都是不厭其煩的給予我指導和幫助。除此之外，學長做事情嚴謹的態度也讓我印象深刻。當然這兩年一起待在博理 421 的日子，投稿、寫程式、討論問題、看 LOL 比賽還有練肖話，我想我是不會忘記的。真的很謝謝學長！

最後要感謝我的家人。爸爸、媽媽、妹妹、恰恰、爺爺、奶奶、外公和外婆，始終默默地支持並鼓勵我。我的女友婷瑋，陪伴了我走過這段時光，你們是我能夠一直向前的最佳動力。以及我的好朋友們，人碩、明仁、昇勳、乃群、冠豪、令儀、國婷、昱廷、則安、宗緯、廷瑋、致睿等等族繁不及備載，為這段日子增添了許多色彩。最後是實驗室學長們，偉志、嘉洋、熊、柯楊、強強林、岳穎、明倫、培恒，以及陳凱、賓四，給予了我研究和課業上許多幫助。謝謝你們！

僅將本篇論文獻給一路上幫助我的人們。

2016.06.22 曾泓諭

doi:10.6342/NTU201600474

中文摘要

近年來，隨著擴充實境與機器人學的發展，如何即時且準確地分析已知的單一平面目標物其三維姿態成為了一個重要的議題。即使過去十幾年中，不少有效率的系統陸續被提出，但由於這些系統只能針對特定的平面目標物，如基準標記或含有簡易封閉曲線的平面目標物，此問題仍舊缺少一個適用於任意平面目標物的解決方法。目前針對此問題最好的解決方法為基於特徵點的方法，但是此方法必須在目標物與相機圖片中的特徵點能對應的前提下才能運作。

為了解決這個問題，在本篇碩士論文中，我們提出了一個表現穩定、能針對任意平面目標物的直接分析演算法。首先，我們採用模板匹配的概念求出一個近似的三維姿態。接下來針對此近似的三維姿態，我們提出了一個梯度下降尋找的演算法來求出更為精準的三維姿態。更進一步，基於所提出的演算法，我們在圖形處理器上實作了一套分析和追蹤平面目標物之三維姿態的系統。此系統包含了分析單元和追蹤單元兩部分。分析單元負責計算出起始的三維姿態，是基於我們提出的演算法所設計的；追蹤單元則負責追蹤三維姿態，其所使用的方法為我們提出的一種三階層搜尋法。在系統中，無論是分析單元還是追蹤單元都充分利用了圖形處理器中平行運算的優點，使得我們的系統能夠非常有效率的運作。我們透過大量的實驗，證明我們所提出的演算法和系統，其表現都比目前基於特徵點的方法要更精準而且穩定。並於實際應用中，我們的系統達到了每秒 11 幀的運算速度。

doi:10.6342/NTU201600474

doi:10.6342/NTU201600474

Direct 3D Pose Estimation of a Planar Target: Algorithm and Implementation

Hung-Yu Tseng

Advisor: Shao-Yi Chien

Co-Advisor: Ming-Hsuan Yang

Graduate Institute of Electronics Engineering

National Taiwan University

Taipei, Taiwan

June 2016

doi:10.6342/NTU201600474

doi:10.6342/NTU201600474

Abstract

Real-time estimating and tracking accurate 3D poses of a known planar target from a calibrated camera are essential for augmented reality and robotics. Although numerous efficient systems have been proposed in the past few decades, it remains a challenging task since the planar targets are limited to fiducial markers and targets with simple contours. The feature-based schemes are the state-of-the-art solutions for obtaining poses of arbitrary planar targets. However the success hinges on whether feature points can be extracted and matched correctly on targets with rich texture.

In this thesis, we propose a robust direct method for 3D pose estimation with high accuracy that performs well on both texture and textureless planar targets. First, the pose of a planar target with respect to a calibrated camera is approximated estimated by posing it as a template matching problem. Next, the object pose is further refined and disambiguated with a gradient descent search scheme. In order to make the proposed algorithm applicable, we also develop D-PET, a direct 3D pose estimation and tracking system implemented on graphics computing units (GPU) which is able to obtain poses in real-time. The system consists of a pose estimation unit and a pose tracker. The pose estimation unit is built based on the approximated pose estimation scheme in the proposed algorithm to find the initial pose. A 3-scale search scheme is proposed for the pose tracker to track the pose precisely. Both of them utilize the characteristics of GPU and accomplish the work efficiently. Extensive experiments on both synthetic and real datasets demonstrate that both the proposed algorithm and system perform favor-

doi:10.6342/NTU201600474

ably against state-of-the-art feature-based approaches in terms of accuracy and robustness. The proposed system achieves a processing speed of 11 fps on an embedded GPU.

doi:10.6342/NTU201600474

Contents

doi:10.6342/NTU201600474

List of Figures

doi:10.6342/NTU201600474

doi:10.6342/NTU201600474

doi:10.6342/NTU201600474

List of Tables

doi:10.6342/NTU201600474

doi:10.6342/NTU201600474

Chapter 1

Introduction

1.1 3D Pose Estimation

Estimating and tracking 3D poses is a classical problem that aims to find the 3D relationship between target objects and the calibrated camera, as as shown in Figure 2.1. With the rapid development of augmented reality [2] and robotics, the demand for obtaining accurate and stable 3D poses becomes increasingly vital. The problem is even more challenging for the case of arbitrary planar targets.

In the early years, several simple and efficient works have been proposed. Systems such as [3, 4, 5] are constructed for the planar target with binary pattern, which is called fiducial marker. Fiducial marker plays an important role that enables these methods to extract the region of the planar target in the camera image. To break the limit of fiducial marker, systems [6, 7] which are able to obtain poses of the planar target with simple contours are proposed. These methods apply training-based algorithms to recognize simple contours in the camera image and compute the pose efficiently.

Solutions for estimating and tracking the poses of arbitrary planar targets can be categorized into two categories. The first categories are based on features extracted from planar targets. The core idea behind feature-based method is to compute a set of *n* correspondences between 3D points and their 2D projection-

doi:10.6342/NTU201600474

s from which the relative position and orientation between the camera and target can be estimated. In recent years, numerous feature detection and tracking schemes [8, 9, 10, 11, 12] have been developed. In order to match features more robustly, variants of RANSAC algorithms [13, 14, 15] have been used to eliminate outliers before the object pose is estimated from a set of feature correspondences. Typically the Perspective-*n*-Point (P*n*P) [16, 17, 18] algorithms or the other related method [19] are applied to the correspondences after RANSAC for estimating the pose. Nonetheless, since the success of this method hinges on whether point correspondences can be corretly matched or not, these approaches are less effective when the target image is textureless or the camera image is blurry.

The second categories consists of direct methods that do not depend on features. Since the seminal work by Lucas and Kanade [20], several algorithms based on global, iterative and nonlinear optimization is proposed [21, 22, 23, 24]. As the 3D pose estimation problem can be reduced to 2D template matching, [25, 26] estimate the poses through optimizing the parameters which is account for rigid transformation of observed target image. However, these methods rely on initial reference and may be trapped in local minimum. To conquer the limitations, non-iterative approaches [27, 28, 29] are proposed in recent years. Nevertheless, these template matching methods have the shortcoming of misalignment between affine or homography transformation space and pose space. It causes the additional pose error produced by transformation matrix decomposition while estimation the 3D pose. Briefly speaking, although these methods are able to work normally in the cases which feature-based methods fail, a direct method suitable for pose estimation and tracking is still lacking in the literature.

1.2 Pose Ambiguity

Due to the lack of the information about the third dimension on the planar target, the pose ambiguity problem as discussed in previous work [30, 16, 31, 32]

doi:10.6342/NTU201600474

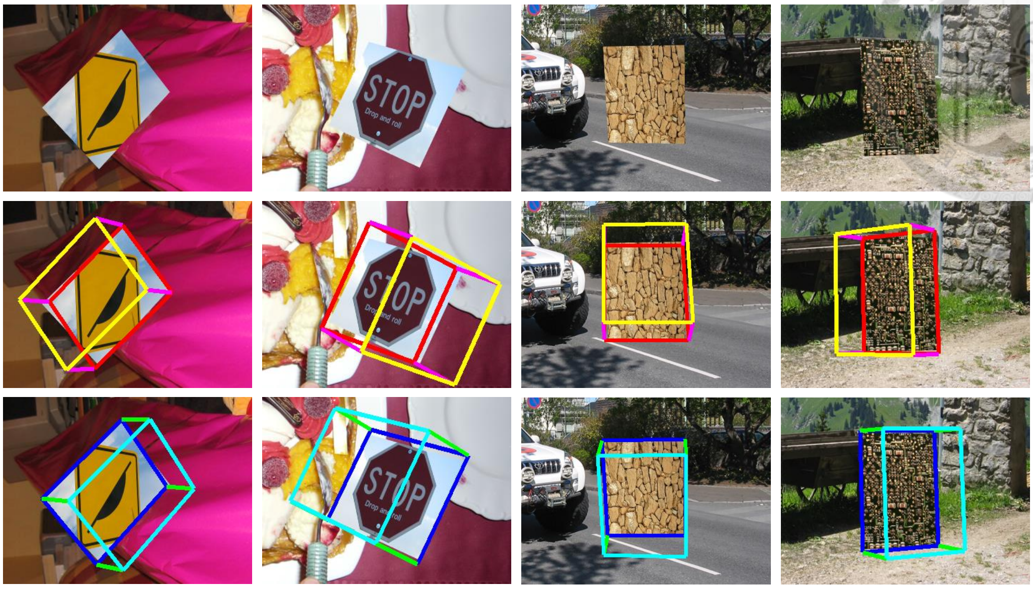


Figure 1.1: Examples of the pose ambiguity. First row: origin images. Second row: images are rendered magenta boxes according to wrong ambiguous poses. Third row: images are rendered cyan boxes according to correct poses.

is inevitably bound to occur. Pose ambiguity is related to the situation that the according error function has multiple local minima for a given configure. Based on the observations, one of the ambiguous poses with local minimum will be the correct pose. Figure 1.1 gives several examples. The wrong ambiguous poses and correct poses are represented with magenta and cyan boxes respectively. Since the pose ambiguity is the main cause of jumping pose estimation results in an image sequence, it is an important issue in the problem of 3D pose estimation of a planar target.

1.3 Contribution

In this thesis, we propose a direct 3D pose estimation algorithm which do not depend on features. We also utilize the parallel characteristic of the graphics computing unit (GPU) and implement D-PET, an embedded real-time direct 3D pose

doi:10.6342/NTU201600474

estimation and tracking system on NVIDIA Jetson TX1 board.

The proposed algorithm estimates the 3D poses of the planar target from a calibrated camera by measuring the similarity between the projected planar target and the 2D image based on appearance. After obtaining an initial pose using an approximated pose estimation scheme, we determine all ambiguous poses and refine them until they converge to local minima. The final pose is chosen as the one with the lowest error among these ambiguous poses. Extensive experiments are conducted to validate the proposed algorithm. We evaluate the performance of proposed algorithm on a synthetic dataset with different types of planar targets and different levels of degraded camera images. Further more, we also evaluate the proposed system on the real dataset by Gauglitz *et al.* [1] against the state-of-the-art feature-based algorithms.

On the other hand, the proposed system consists of a pose estimation unit and a pose tracker. We reference the approximated pose estimation scheme in the proposed algorithm and build the pose estimation unit. The pose estimation unit is in charge of finding the initial pose for tracking. The proposed pose tracker then takes the initial pose and applies a 3-scale search with a pose search pattern to track the poses. We verify that the pose estimation unit has similar performance compared to the approximated pose estimation scheme through the experiment on synthetic dataset. The performance of the pose tracker is investigated using the real dataset. Finally, we conduct several practical tests to demonstrate the performance of the proposed system in the real world. The main contribution of this thesis can be summarized as follows.

- We propose a non-feature based algorithm to estimate 3D pose of a planar target.

- We develop an efficient direct 3D pose estimation and tracking system implemented on NVIDIA embedded GPU.

- Extensive experiments show that both the proposed algorithm and system

doi:10.6342/NTU201600474

perform favorably in terms of accuracy and robustness against the state-of-the-art feature-based methods, and the proposed system achieves the processing speed of 11 fps.

1.4 Thesis organization

In this chapter, we introduce 3D pose estimation and present the main contribution of this thesis. The remainder of the thesis is organized as follows. The background knowledge and related works are mentioned in Chapter 2. In Chapter 3, we describe the proposed direct 3D pose estimation algorithm. The datasets and the evaluation results for the proposed algorithm is provided in Chapter 4. Chapter 5 introduces the proposed system and its performance under several experiments. Finally, the conclusion is given in Chapter 6.

doi:10.6342/NTU201600474

Chapter 2

Background Knowledge and Related Works

2.1 Problem Definition

The goal of 3D pose estimation is to find the 3D relationship between the target object and the calibrated camera. The coordinate system between the target object and the camera image is shown in Figure 2.1. Given a planar target image I_t, a camera image I_c, a set of 3D coordinates of points $\mathbf{x}_i = [x_i, y_i, 0]^\top, i = 1\ldots., n, n \geq 3$ in target object coordinate and a set of image coordinates $\mathbf{u}_i = [u_i, v_i]^\top$ in camera image coordinate, the transformation between these two sets of points can be formulated as

$$\begin{bmatrix} hu_i \\ hv_i \\ h \end{bmatrix} = \mathbf{K} \cdot \mathbf{T}_{cm} \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & x_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} | \mathbf{t} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix}, \tag{2.1}$$

where $\mathbf{K}$ is the intrinsic matrix and $\mathbf{T}_{cm}$ is the extrinsic matrix. In the intrinsic matrix $\mathbf{K}$, (f_x, f_y) and (x_0, y_0) refer to focal length and principle point which are

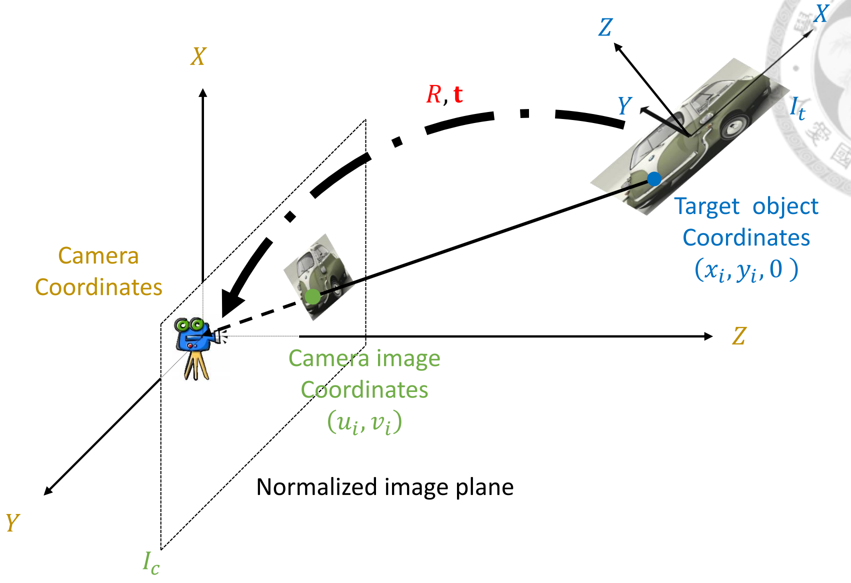


Figure 2.1: Coordinate system between the planar target and the camera image.

treated as known factors. In the extrinsic matrix $\mathbf{T}_{cm}$ on the other hand,

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \in SO(3), \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \in R(3), \tag{2.2}$$

are the rotation matrix and translation vector, respectively. The physical meaning in (2.1) can be described in two steps, which are explained in detail as follows.

First, the 3D coordinates $\mathbf{x}_i$ in target object coordinate are transformed to 3D coordinates $\mathbf{x}_{ci} = [x_{ci}, y_{ci}, z_{ci}]$ in camera coordinate through the extrinsic matrix $\mathbf{T}_{cm}$, namely

$$\begin{bmatrix} x_{ci} \\ y_{ci} \\ z_{ci} \end{bmatrix} = \mathbf{T}_{cm} \begin{bmatrix} x_i \\ y_i \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R} | \mathbf{t} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 0 \end{bmatrix} . \tag{2.3}$$

The rotation matrix $\mathbf{R}$ is determined by the three degrees of freedom parameterized based on the orientation, and the translation vector $\mathbf{t}$ is parameterized based on position of the target object with respect to the calibrated camera. The second step is to project the 3D coordinates $\mathbf{x}_{ci}$ in camera coordinate to 2D coordinates $\mathbf{u}_i$

doi:10.6342/NTU201600474

in camera image coordinate through the intrinsic matrix **K**. From (2.1) and (2.3) we know that

$$u_i = f_x \frac{x_{ci}}{z_{ci}} + x_0, \quad v_i = f_y \frac{y_{ci}}{z_{ci}} + y_0. \tag{2.4}$$

The operations behind this formulation are projection and translation. After $\mathbf{x}_{ci}$ is projected to the normalized camera image plane, the origin point on the normalized camera image plane is changed to the top-left corner of the camera image, which results in $\mathbf{u}_i$ in camera image coordinate.

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 $\mathbf{p} = (\mathbf{R}, \mathbf{t})$ that minimize an appropriate error function. In principle, there are two possible error functions. One is the reprojection error, which is mostly used in the P*n*P algorithms,

$$E_r(\mathbf{p}) = \frac{1}{n} \sum_{i=1}^{n} \left[(\hat{u}_i - u_i)^2 + (\hat{v}_i - v_i)^2 \right]. \tag{2.5}$$

Another error function is based on the sum of absolute differences (also known as appearance distance) and is mostly used in direct methods and this thesis,

$$E_a(\mathbf{p}) = \frac{1}{n_t} \sum_{i=1}^{n_t} |I_c(\mathbf{u}_i) - I_t(\mathbf{x}_i)|, \tag{2.6}$$

where n_t represents the total number of pixels in I_t.

2.2 State-of-the-Art Pose Estimation

2.2.1 Marker-based Pose Estimation

The approaches in this category require predefined planar targets with simple binary pattern, which is called fiducial markers. To recognize the planar target, these approaches threshold the camera image, extract the rectangular region of the fiducial marker and perform pattern checking. After the recognition, these method finds the four corners of the target on the camera image to proceed to the estimation stage. A lot of works based on this category are proposed in the past few decades. According to [33], these works can be classified based on the geometry of the fiducial marker such as square [34, 35, 36, 37, 38], circular [39, 40] and dots [41, 42]. Since these methods do not need complicated computation such as feature extraction or template matching, the computation time is relatively small.

2.2.2 Feature-based Pose Estimation

Since the feature-based pose estimation relies on nature features, it does not require predefined planar targets. The algorithm flow is shown in Figure 2.2. The features in the target image and the camera image are detected and matched in the first stage. There are numerous feature detection, description and matching methods proposed recently. The most classical one must be SIFT [8] which is a multi-scale detection and scale invariant description scheme. SURF [9] applies the basic idea of SIFT and uses integral image for acceleration. To make the process even faster, fast detector and binary descriptor are proposed in numerous works such as BRISK [10], ORB [11] and FREAK [12]. Since all the methods described above are not affine or perspective invariant, they are less effective in pose estimation when the tilt angle between the planar target and the camera is large. Affine-SIFT (ASIFT) [43] matches feature points well because it simulates all obtainable viewpoints for description. However, there is no existing system applying ASIFT since it is more computationally expensive than others.

The pose is estimated after the quality of feature matching is further enhanced by RANSAC algorithms. Typically there are two kinds of methods used in the estimation stage to obtain poses. The first one is the perspective-*n*-points (P*n*P) algorithms. Numerous P*n*P algorithms such as OI [44], RPP [16], EPnP [17], RPnP [45] and OPnP [18] are proposed to optimize the pose using the reprojection error function which is described in (2.5). Among these P*n*P algorithms, OPnP shows the dominant performance in terms of efficiency and accuracy. It formulates the the P*n*P problem as an unconstrained optimization problem and applies Gröbner bases solver to estimate the pose.

The second kind of method is to decompose the homography transformation

doi:10.6342/NTU201600474

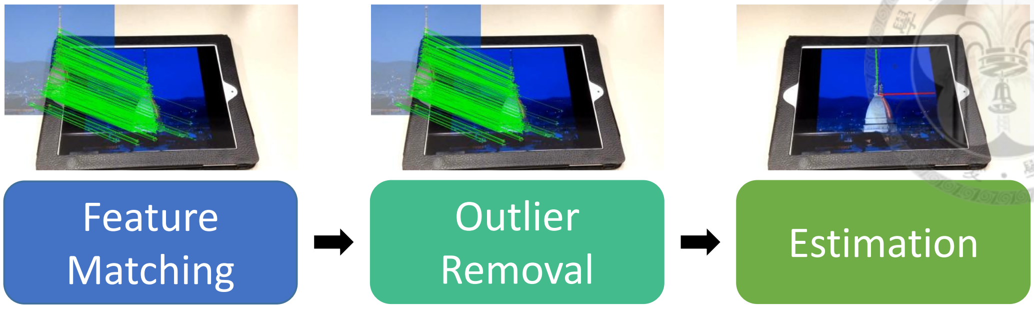


Figure 2.2: The algorithm flow of feature-based pose estimation methods.

which is estimated from the planar target to camera image. Since the transformation between planar target and camera image can be viewed as a 2D-2D homography transformation, the 3D pose can be obtained by decompose the homography. Although the homography decomposition methods proposed in the early years [46, 47, 48] are less effective than P*n*P algorithms, Collins and Bartoli propose a new analytic solution [19] which have a comparable performance against P*n*P algorithms with shorter computation time.

2.3 Template Matching

The template matching problem has been widely studied in the literatures, and one important issue is how to efficiently obtain accurate results with evaluating only a subset of the possible transformationes. Since the appearance distances between a template and two sliding windows shifted by a few pixels are usually close due to the nature of image smoothness, Pele and Werman [49] exploit this fact to reduce the time complexity of pattern matching. Alex *et al.* [50] 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 a efficient pattern matching. In [28], Korman *et al.*show that the 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, which inspires us to propose the direct

doi:10.6342/NTU201600474

3D pose estimation algorithm.

2.4 Motion Estimation

The refinement method in proposed algorithm and the 3-scale search in D-PET system are motivated by fast motion estimation methods in video coding. Liu and Feig [51] propose the Gradient Descent Search (GDS) algorithm that evaluates the values of a given objective function from a centralized search neighborhood for motion estimation. When the minimum within a neighborhood is found, it is used to determine the position for the next search until it converges. Compared with the full search method, the GDS algorithm achieves similar performance but with much lower computational complexity. Zhu and Ma [52] develop an algorithm for block-based motion estimation based on two designed diamond-shaped search patterns, and it further reduced the required number of search points. A motion estimation method that exploits more elaborated coarse-to-fine search patterns is subsequently developed by Zhu *et al.* [53].

doi:10.6342/NTU201600474

Chapter 3

Direct Pose Estimation and Tracking

The proposed algorithm consists of two steps, as shown in figure 3.1. First, the 3D pose of a planar target with respect to a calibrated camera is estimated. Second, the object pose is further refined and disambiguated. We describe these steps as follows.

3.1 Approximated Pose Estimation

Let $T_{\mathbf{p}}$ be the transformation at pose $\mathbf{p}$. Assume a reference point $\mathbf{x}_i$ in target image transformed is transformed to two points $\mathbf{u}_{i1}$ and $\mathbf{u}_{i2}$ with two different poses $\mathbf{p}_1$ and $\mathbf{2}$, respectively. It is shown in [28] that if the spatial distance between $\mathbf{u}_{i1}$ and $\mathbf{u}_{i2}$ is bounded by a positive value ε,

$$\forall \mathbf{x}_i \in I_t : d(T_{\mathbf{p}_1}(\mathbf{x}_i), T_{\mathbf{p}_2}(\mathbf{x}_i)) = O(\varepsilon), \tag{3.1}$$

then the following equation holds,

$$|E_a(\mathbf{p}_1) - E_a(\mathbf{p}_2)| = O(\varepsilon \bar{\mathcal{V}}), \tag{3.2}$$

where $\bar{\mathcal{V}}$ denotes the mean variation of target image I_t, which represents the mean value over the entire target image of the maximal difference between each pixel and any of its neighborhood. Since the mean variation $\bar{\mathcal{V}}$ can be constrained by

doi:10.6342/NTU201600474

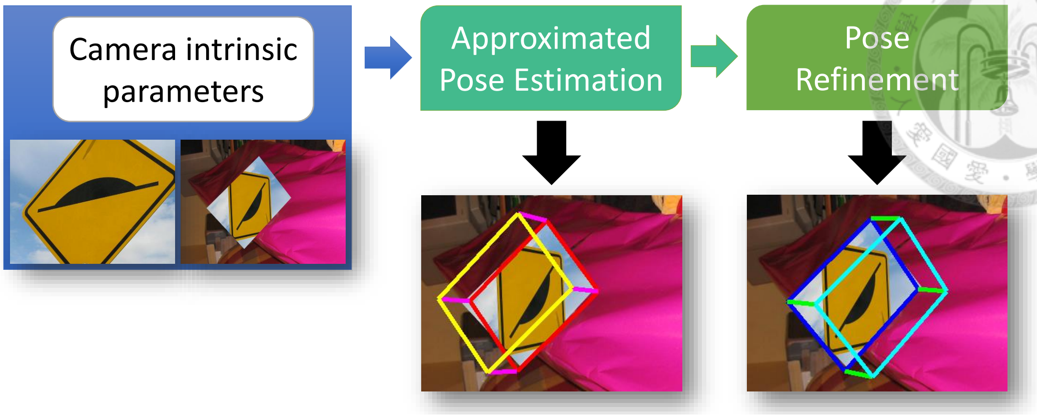


Figure 3.1: The proposed algorithm consists of two steps. The pose is estimated in the first stage and refined in the second stage.

filtering I_t, the difference between $E_a(\mathbf{p}_1)$ and $E_a(\mathbf{p}_2)$ is bounded in terms of ε. In the proposed algorithm, we only need to consider a limited number of poses by constructing a ε-covering pose set $\mathcal{S}$ based on (3.1) and (3.2).

3.1.1 ε-Covering Set Construction

In this thesis, we factorize the rotation matrix as $\mathbf{R} = \mathbf{R}_z(\theta_{z_c})\mathbf{R}_x(\theta_x)\mathbf{R}_z(\theta_{z_t})$ [54] as shown in Figure 3.2(a). The pose is 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. According to the factorization and (2.1), the transformation between a reference point $\mathbf{x}_i = [x_i, y_i, 0]$ on I_t and the corresponding image coordinate $\mathbf{u}_i = [u_i, v_i]$ can be formulated as

$$\begin{bmatrix} hu_i \\ hv_i \\ h \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & x_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} | \mathbf{t} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 0 \end{bmatrix}, \tag{3.3}$$

where

$$\mathbf{R} = \begin{bmatrix} c_{z_c} c_{z_t} - c_x s_{z_c} s_{z_t} & -c_x c_{z_t} s_{z_c} - c_{z_c} s_{z_t} & s_x s_{z_c} \\ c_{z_t} s_{z_c} + c_x c_{z_c} s_{z_t} & c_x c_{z_c} c_{z_t} - s_{z_c} s_{z_t} & -s_x c_{z_c} \\ s_x s_{z_t} & s_x c_{z_t} & c_x \end{bmatrix}, \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \tag{3.4}$$

The notation *s* and *c* indicates *sin* and *cos* operation, respectively.

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 (3.1) in $\mathcal{S}$. To construct the set favorably, the coordinates of $\mathbf{x}_i \in I_t$ are pre-normalized to the range $[-1,1]$. Starting with t_z, we derive the equation below by using (3.3) for each $\mathbf{x}_i$,

$$\begin{aligned} d(T_{\mathbf{p}_{t_z}}(\mathbf{x}_i), T_{\mathbf{p}_{t_z+\Delta t_z}}(\mathbf{x}_i)) &= \sqrt{[(\frac{f_x x_i}{t_z}) - (\frac{f_x x_i}{t_z + \Delta t_z})]^2 + [(\frac{f_y y_i}{t_z}) - (\frac{f_y y_i}{t_z + \Delta t_z})]^2} \\ &= O(\frac{1}{t_z} - \frac{1}{t_z + \Delta t_z}). \end{aligned} \tag{3.5}$$

To make (3.5) satisfy the constraint in (3.1), we use the step size, with tight bound in Big-Theta notation,

$$\Delta t_z = \Theta(\frac{\varepsilon t_z^2}{1 - \varepsilon t_z}), \tag{3.6}$$

which means that (3.5) can be bounded if we construct $\mathcal{S}$ with the step (3.6) on dimension t_z.

Since θ_x describes the tilt angle between camera and target image as shown in Figure 3.2(a), we obtain the following equation depending on the current t_z,

$$\begin{aligned} d(T_{\mathbf{p}_{\theta_x}}(\mathbf{x}_i), T_{\mathbf{p}_{\theta_x+\Delta\theta_x}}(\mathbf{x}_i)) &= \sqrt{d_{\mathbf{u_i}}^2 + d_{\mathbf{v_i}}^2} \\ &= O(\frac{1}{t_z - \sin(\theta_x + \Delta\theta_x)} - \frac{1}{t_z - \sin(\theta_x)}), \end{aligned} \tag{3.7}$$

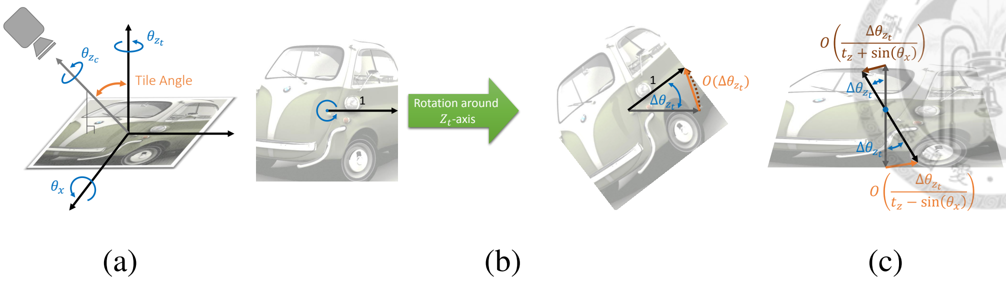
for each $\mathbf{x}_i$, where

$$\begin{aligned} d_{\mathbf{u_i}} &= (\frac{f_x x_i}{y_i \sin\theta_x + t_z}) - (\frac{f_x x_i}{y_i \sin(\theta_x + \Delta\theta_x) + t_z}), \\ d_{\mathbf{v_i}} &= (\frac{f_y y_i \cos\theta_x}{y_i \sin\theta_x + t_z}) - (\frac{f_y y_i \cos(\theta_x + \Delta\theta_x)}{y_i \sin(\theta_x + \Delta\theta_x) + t_z}). \end{aligned} \tag{3.8}$$

In addition, to make (3.7) satisfy the constraint in (3.1), we set the step size,

$$\Delta\theta_x = \Theta(\sin^{-1}(t_z - \frac{1}{\varepsilon + \frac{1}{t_z - \sin(\theta_x)}}) - \theta_x). \tag{3.9}$$

Let $\theta_{z_t'} = \theta_{z_t} + \Delta\theta_{z_t}$, we obtain the following equation based on the current t_z


θ_{z_c}
θ_{z_t}
Tile Angle
θ_x
1
Rotation around Z_t-axis
1
$\Delta\theta_{z_t}$
$O(\Delta\theta_{z_t})$
$O\left(\frac{\Delta\theta_{z_t}}{t_z+\sin(\theta_x)}\right)$
$\Delta\theta_{z_t}$
$\Delta\theta_{z_t}$
$O\left(\frac{\Delta\theta_{z_t}}{t_z-\sin(\theta_x)}\right)$

(a) (b) (c)

Figure 3.2: (a) Illustration of rotation matrix factorization. θ_x indicates the tilt angle between the camera and the target. (b) 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). (c) 3D illustration of rotation around Z_t-axis. The linear distance between points is a function of tilt angle θ_x.

and θ_x,

$$\begin{aligned} d(T_{\mathbf{p}_{\theta_{z_t}}}(\mathbf{x}_i), T_{\mathbf{p}_{\theta_{z_t}+\Delta\theta_{z_t}}}(\mathbf{x}_i)) = \sqrt{f_x^2C_1^2 + f_y^2C_2^2} &\leq \sqrt{f_x^2C_1^2 + f_y^2\left[\frac{C_2}{c_x}\right]^2} \\ &= O\left(\frac{\Delta\theta_{z_t}}{t_z + k\sin(\theta_x)}\right), \end{aligned} \tag{3.10}$$

, where

$$\begin{aligned} C_1 &= \frac{c_{z_t}x - s_{z_t}y}{s_x(s_{z_t}x + c_{z_t}y) + t_z} - \frac{c_{z_t'}x - s_{z_t'}y}{s_x(s_{z_t'}x + c_{z_t'}y) + t_z}, \\ C_2 &= \frac{c_x(s_{z_t}x + c_{z_t}y)}{s_x(s_{z_t}x + c_{z_t}y) + t_z} - \frac{c_x(s_{z_t'}x + c_{z_t'}y)}{s_x(s_{z_t'}x + c_{z_t'}y) + t_z} \end{aligned} \tag{3.11}$$

The constant k denotes any constant in the range of $[-\sqrt{2}, \sqrt{2}]$. An illustrative example of (3.10) is shown in Figure 3.2(b)(c). To make (3.10) satisfy the constraint in (3.1), we set the step size

$$\Delta\theta_{z_t} = \Theta(\varepsilon(t_z + k\sin(\theta_x))), \tag{3.12}$$

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.

Since θ_{z_t} denotes 2D image rotation of the planar target, it does not affect 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,

$$\begin{aligned}
&d(T_{\mathbf{p}_{\theta_{z_c}}}(\mathbf{x}_i), T_{\mathbf{p}_{\theta_{z_c}+\Delta\theta_{z_c}}}(\mathbf{x}_i)) \\
&= \sqrt{f_x^2\left[\frac{c_{z_c}x - c_x s_{z_c} y}{s_x y + t_z} - \frac{c_{z_c'}x - c_x s_{z_c'} y}{s_x y + t_z}\right]^2 + f_y^2\left[\frac{s_{z_c}x + c_x c_{z_c} y}{s_x y + t_z} - \frac{s_{z_c'}x + c_x c_{z_c'} y}{s_x y + t_z}\right]^2} \\
&= O\left(\frac{\Delta\theta_{z_c}}{t_z + k\sin(\theta_x)}\right).
\end{aligned} \tag{3.13}$$

We can realize (3.13) in a similar way to (3.10). To make (3.13) satisfy the constraint in (3.1), we set the step size,

$$\Delta\theta_{z_c} = \Theta(\varepsilon(t_z + k\sin(\theta_x))) = \Theta(\varepsilon(t_z)), \tag{3.14}$$

which k is set to 0.

Finally, as the bounded steps for t_x and t_y are also affected by horizontal distance t_z and tilt angle θ_x only, the equations below can be derived,

$$\begin{aligned}
&d(T_{\mathbf{p}_{t_x}(\mathbf{x}_i)}, T_{\mathbf{p}_{t_x+\Delta t_x}}(\mathbf{x}_i)) \\
&= \sqrt{f_x^2\left[\frac{x + t_x}{s_x y + t_z} - \frac{x + t_x + \Delta t_x}{s_x y + t_z}\right]^2 + f_y^2\left[\frac{y}{s_x y + t_z} - \frac{y}{s_x y + t_z}\right]^2} \\
&= O\left(\frac{\Delta t_x}{t_z + k\sin(\theta_x)}\right),
\end{aligned} \tag{3.15}$$

$$\begin{aligned}
&d(T_{\mathbf{p}_{t_y}(\mathbf{x}_i)}, T_{\mathbf{p}_{t_y+\Delta t_y}}(\mathbf{x}_i)) \\
&= \sqrt{f_x^2\left[\frac{x}{s_x y + t_z} - \frac{x}{s_x y + t_z}\right]^2 + f_y^2\left[\frac{y + t_y}{s_x y + t_z} - \frac{y + t_y + \Delta t_y}{s_x y + t_z}\right]^2} \\
&= O\left(\frac{\Delta t_y}{t_z + k\sin(\theta_x)}\right),
\end{aligned} \tag{3.16}$$

To make (3.15) and (3.16) satisfy the constraint in (3.1), we set these step sizes,

$$\Delta t_x = \Theta(\varepsilon(t_z + k\sin(\theta_x))) = \Theta(\varepsilon(t_z - \sqrt{2}\sin(\theta_x))), \tag{3.17}$$

$$\Delta t_y = \Theta(\varepsilon(t_z + k\sin(\theta_x))) = \Theta(\varepsilon(t_z - \sqrt{2}\sin(\theta_x))), \tag{3.18}$$

Table 3.1: Step size on each dimension 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	$\Theta(\frac{\varepsilon t_z^2}{1-\varepsilon t_z})$

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

Table 3.1 summarizes the bounded step size on each domain for constructing ε-covering pose set. From the table we know that the step size of θ_{z_c}, θ_{z_t} and t_z are function of t_z. The step size of θ_x, t_x and t_y are function of t_z and θ_x. Based on these observations, the construction of the ε-covering set is designed to be nested. First, we construct the set of t_z using (3.6) to satisfy the constraint for approximated pose estimation. We then construct the set of θ_x according to (3.9) within each t_z. Finally the set of θ_{z_c}, θ_{z_t}, t_x and t_y are constructed sequetially by considering each value of t_z and θ_x.

3.1.2 Coarse-to-Fine Estimation

Due to the large parameter space, the computational and memory costs are prohibitively high if the ε-covering pose set is used directly for pose estimation. For fast and accurate pose estimation, a coarse-to-fine approach is employed. As illustrated in Figure 3.3, 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 $E_a(\mathbf{p}_b)$, we select the poses within a threshold,

$$\mathcal{S}_L = \{\mathbf{p}_L \mid E_a(\mathbf{p}_L) < E_a(\mathbf{p}_b) + L\}, \tag{3.19}$$

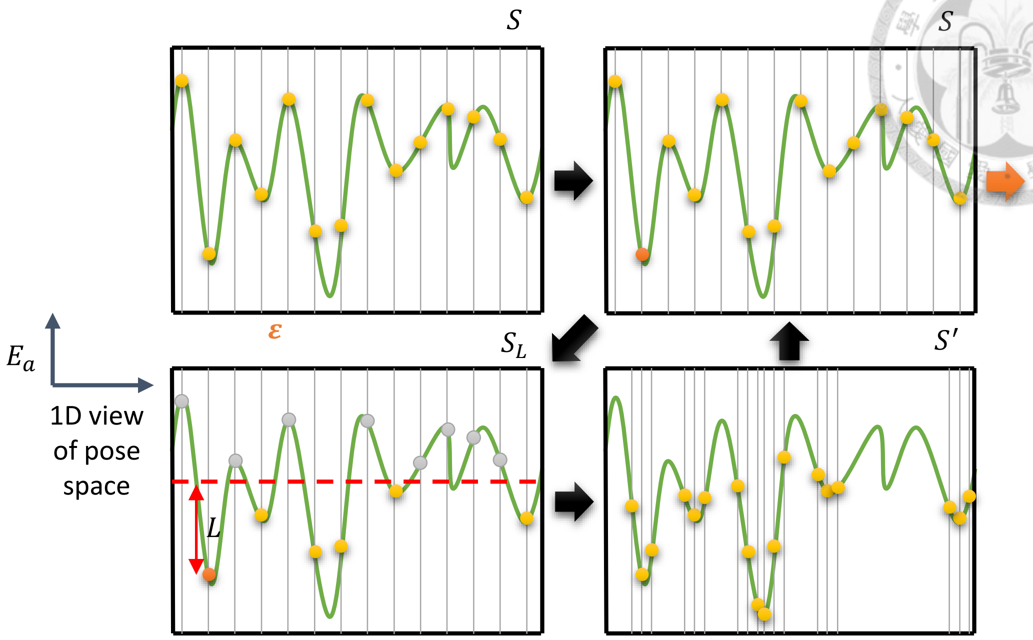


Figure 3.3: An illustration of proposed coarse-to-fine estimation.

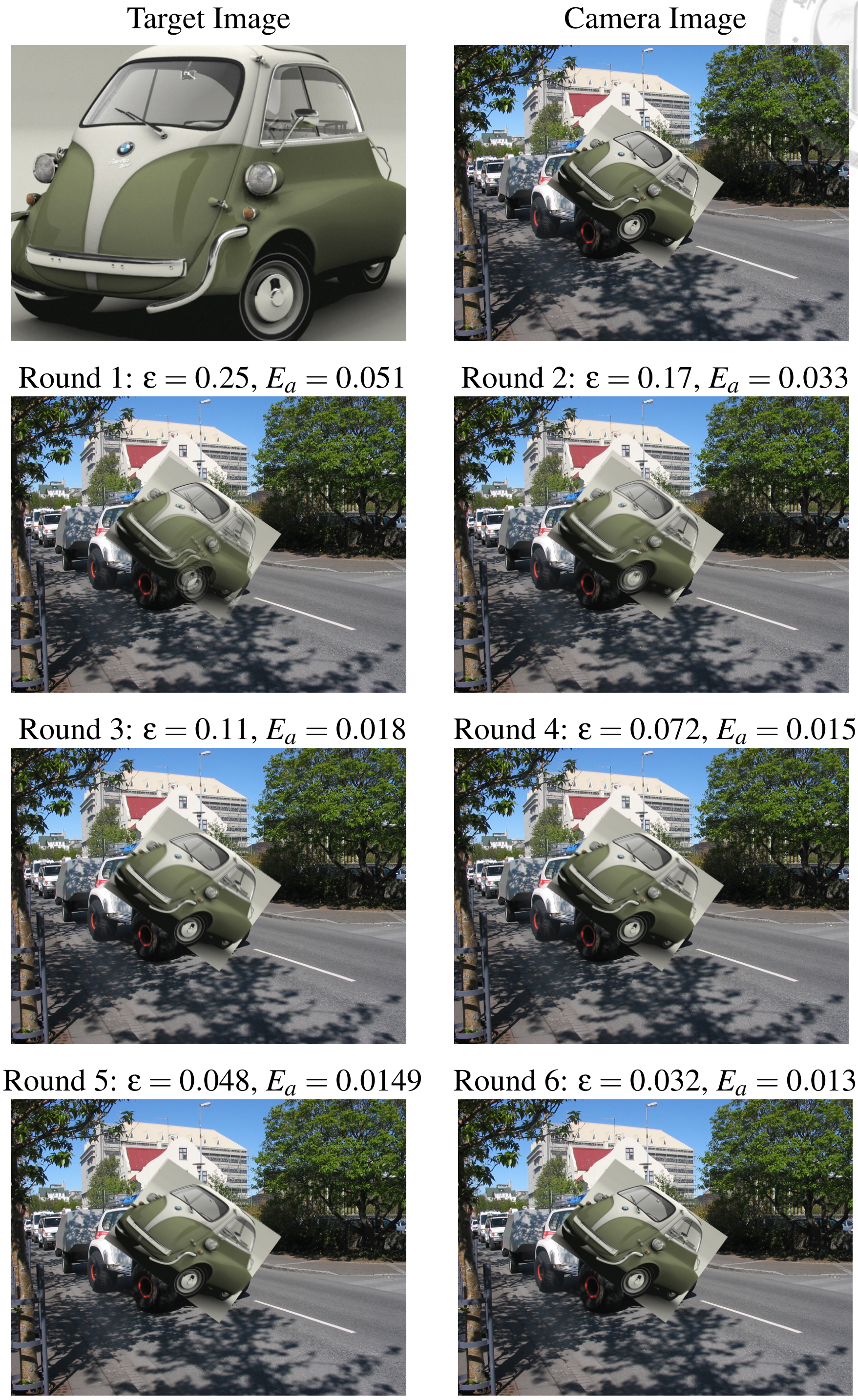
to be considered in the next round. Here the constant L is a threshold set empirically. Based on E_a, we create sets with finer $\varepsilon^{'}$,

$$\mathcal{S}^{'} = \{\mathbf{p}^{'} \mid \exists \mathbf{p}_L \in \mathcal{S}_L : (3.1) \text{ holds for } \mathbf{p}^{'}, \mathbf{p}_L \text{ and } \varepsilon^{'}\}, \tag{3.20}$$

and repeat search until we obtain the desired precision parameter ε^*. The best pose in the last set is used as the approximated estimate. Figure 3.4 shows an example of the coarse-to-fine estimation. According to the pose $\mathbf{p}_b$ with minimum E_a estimated in each round of the coarse-to-fine estimation, the projected target image is rendered on the camera image. It is obvious that the projected target image is approximately aligned to the target in the camera image as the precision parameter ε becomes finer.

The algorithm is accelerated by random sampling a potion of pixels to approximated the error measure $E_a^{'}$ instead of using all pixels in I_t to compute E_a in (2.6). According to Hoeffding's inequality [55], $E_a^{'}$ is probably close to E_a within a pre-

doi:10.6342/NTU201600474



Target Image

Camera Image

Round 1: $\varepsilon = 0.25$, $E_a = 0.051$

Round 2: $\varepsilon = 0.17$, $E_a = 0.033$

Round 3: $\varepsilon = 0.11$, $E_a = 0.018$

Round 4: $\varepsilon = 0.072$, $E_a = 0.015$

Round 5: $\varepsilon = 0.048$, $E_a = 0.0149$

Round 6: $\varepsilon = 0.032$, $E_a = 0.013$

Figure 3.4: An example of the coarse-to-fine estimation. According to the pose $\mathbf{p}_b$ with minimum E_a estimated in each round of the coarse-to-fine estimation, the projected target image is rendered on the camera image.

doi:10.6342/NTU201600474

cision parameter δ if the number of sampling pixels m is large enough,

$$P(|E_a^{'} - E_a| > \delta) \leq 2e^{-2\delta^2 m}, \tag{3.21}$$

where $P(\cdot)$ represents the probability measure. This equation suggests that if m is properly selected, the approximation error between $E_a^{'}$ and E_a can be bounded with high probability. In other words, $E_a^{'}$ is a approximation of E_a within the probably approximately correct (PAC) framework [55]. With this approximation, the runtime of the proposed algorithm is dramatically reduced.

Moreover, in order to be invariant to different lighting conditions, we normalizes the intensity term and adds chroma terms to the appearance distance measure. In implementation, we convert the RGB values of sampling pixels to the YCbCr color space. The appearance distance is then computed as follows,

$$E_a = 0.5E_{a_Y} + 0.25E_{a_{Cb}} + 0.25E_{a_{Cr}}. \tag{3.22}$$

We determine the weight for each term by experiments and find that the proposed algorithm is more invariant to intensity change when the chroma terms are added than the normalized intensity values are only used.

3.2 Pose Refinement

We obtain $(\mathbf{R}^{'}, \mathbf{t}^{'})$ after the proposed approximated pose estimation scheme. However, this result is bounded based on the distance in the appearance space rather than the pose space. Therefore the estimated pose and actual pose may be significantly different even when the appearance distance is small, which is often the case when the tilt angle of a target image is large. In the meanwhile, the pose ambiguity problem is likely occur as illustrated in Figure 1.1. Consequently, a pose refinement scheme is proposed to further improve the accuracy and address the ambiguity problem.

3.2.1 Candidate Poses Exploration

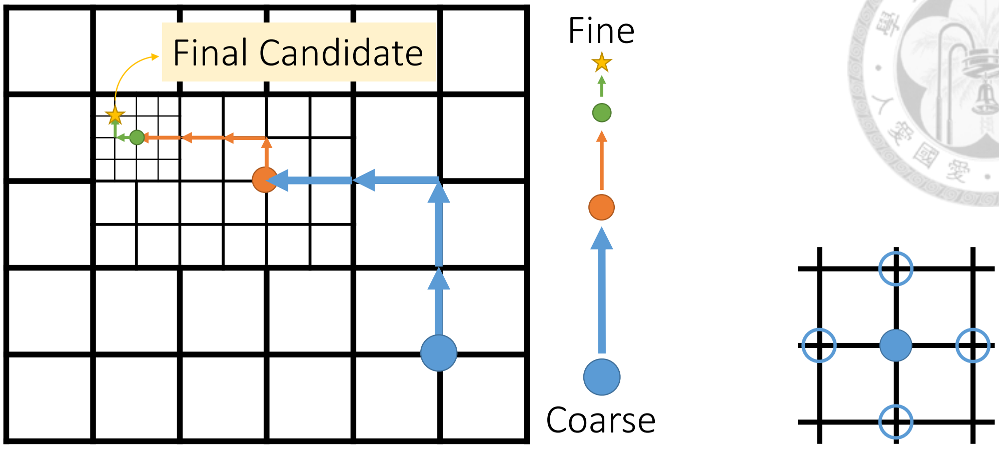
In order to address the problem of pose ambiguity, we first transform four corner points $\mathbf{x}_{c1}$, $\mathbf{x}_{c2}$, $\mathbf{x}_{c3}$, and $\mathbf{x}_{c4}$ in the target image I_t to $\mathbf{u}_{c1}$, $\mathbf{u}_{c2}$, $\mathbf{u}_{c3}$, and $\mathbf{u}_{c4}$ in the camera image I_c with $(\mathbf{R}^{'}, \mathbf{t}^{'})$, respectively. We then compute all local minima of the error function (2.5). Finally, only the local minima with the two smallest objective values in (2.6) are plausible poses, and these two ambiguous poses are both chosen as the candidate poses. In practice, IPPE [19] and OPnP [18] are the two methods we consider to apply to get the candidate poses. Through the experiment we describe in Section 4.1, we choose to apply OPnP to accomplish the work.

3.2.2 Refining Pose Estimation

After obtaining the two candidate poses, we can further improve the accuracy using a coarse-to-fine gradient descent search scheme. In contrast to the 2D motion estimation in video coding, we consider a 6D pose motion with infinity resolution in this work. A 2D view of the coarse-to-fine gradient descent search is shown in Figure 3.5(a). The largest blue circle denotes the approximate pose estimated in Section 3.1, and the smaller one (orange) represents the local minimum found by the search pattern at the starting ε-precision. As the minimum under the current precision level is found, we diminish the precision parameter ε and perform gradient descent search again on the next level. This process is repeated until we obtain the local minimum under the desired precision parameter ε^*. Finally, the pose with smaller E_a is chosen from the two refined candidate poses.

The 2D view of the checking pattern in the coarse-to-fine GDS scheme [51] are shown in Figure 3.5(b). It is formed by 13 checking points, including the center point and its 12 neighbors. These 12 neighbors are ε-away from the center separately in the 6D pose space. Let $\mathbf{p}_c = [\theta_{z_c}, \theta_x, \theta_{z_t}, t_x, t_y, t_z]^\top$ be the center point of the checking pattern in the pose space and $\mathbf{P}_c$ be the 6×13 matrix with repeating $\mathbf{p}_c$ in a row. Also let $\mathbf{s}_\varepsilon = [s_{\theta_{z_c}}, s_{\theta_x}, s_{\theta_{z_t}}, s_{t_x}, s_{t_y}, s_{t_z}]^\top$ be the step size listed

doi:10.6342/NTU201600474



(a) (b)

Figure 3.5: (a) 2D illustration of coarse-to-fine gradient descent search. We carry out the GDS on the coarse level in the beginning. After reaching the local minimum, we move to the finer level and repeat this coarse-to-fine GDS approach until we obtain minimum within the desired precision. (b) The checking pattern in 2D view, including 1 center checking point and its 4 neighbnors (2 neighbors per dimension).

in Table 3.1 with precision parameter ε and $\mathbf{S}_\varepsilon$ be the 6×13 matrix with repeating $\mathbf{s}_\varepsilon$ in a row. The mathematical description of the checking pattern $\mathbf{M}$ can then be written as

$$\mathbf{M} = \mathbf{P}_c + \mathbf{D} \circ \mathbf{S}_\varepsilon, \tag{3.23}$$

where

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \tag{3.24}$$

and $\circ$ represents the Hadamard product. Each column in $\mathbf{M}$ represents one of the checking points within the checking pattern. The main steps of the proposed pose estimation method are summarized in Algorithm 1.

doi:10.6342/NTU201600474

Algorithm 1 Proposed Direct 3D Pose Estimation

```
Input: Target image $I_t$, camera image $I_c$, intrinsic parameters, and precision parameters $\varepsilon_c^*, \varepsilon_f^*$.
Output: Estimated pose result $\mathbf{p}^*$.
Create an $\varepsilon$-covering pose set $\mathcal{S}$.
Find $\mathbf{p}_b$ from $\mathcal{S}$ with $E_a^{'}$ according to (3.21).
while $\varepsilon > \varepsilon_c^*$ do
    Obtain the set $\mathcal{S}_L$ according to (3.19);
    Diminish $\varepsilon$;
    Replace $\mathcal{S}$ according to (3.20);
    Find $\mathbf{p}_b$ from $\mathcal{S}$ with $E_a^{'}$ according to (3.21);
end while
Explore the candidate poses $\mathbf{p}_1$ and $\mathbf{p}_2$ with $\mathbf{p}_b$.
for $i = 1 \rightarrow 2$ do
    Let $\mathbf{p}_c = \mathbf{p}_i$ and $\varepsilon_i = \varepsilon$
    while $\varepsilon_i > \varepsilon_f^*$ do
        Find $\mathbf{p}_b$ from (3.23) with $E_a^{'}$ according to (3.21).
        if $\mathbf{p}_c \neq \mathbf{p}_b$ then
            $\mathbf{p}_c = \mathbf{p}_b$
        else
            Diminish $\varepsilon_i$;
        end if
    end while
    Let $\mathbf{p}_i = \mathbf{p}_c$
end for
Return the pose $\mathbf{p}^*$ with smaller $E_a$ from $\mathbf{p}_1$ and $\mathbf{p}_2$
```

Chapter 4

Experiments

We experimentally evaluate the proposed algorithm for the 3D pose estimation problem using both synthetic and real datasets, and compare it with the feature-based schemes. Through some preliminary experiments, we find SIFT [8] method performs better than other alternative features in terms of repeatability and accuracy. Similar observations can also be found in [1]. As the ASIFT [43] method is considered the state-of-the-art affine-invariant method to find correspondences under large view change, we use both the SIFT and ASIFT methods in the compared feature-based schemes. The RANSAC-based method [13] is then used to eliminate outliers before object pose is estimated by the P*n*P algorithms. It has been shown that, among the P*n*P algorithms [16, 17, 18, 56], the OP*n*P [18] algorithm achieves the state-of-the-art results in terms of efficiency and accuracy. Therefore we use the OP*n*P algorithm as the pose estimator in the feature-based schemes.

Given the true rotation matrix $\hat{\mathbf{R}}$ and translation vector $\hat{\mathbf{t}}$, we compute the rotation error of the estimated rotation matrix $\mathbf{R}$ by $E_{\mathbf{R}}(degree) = acosd((\mathrm{Tr}(\mathbf{R}^{\top} \cdot \hat{\mathbf{R}}) - 1)/2)$, where $acosd(\cdot)$ represents the arc-cosine operation in degrees. The translation error of the estimated translation vector $\mathbf{t}$ is measured by the relative difference between $\hat{\mathbf{t}}$ and $\mathbf{t}$ defined as $E_{\mathbf{t}}(\%) = \|\hat{\mathbf{t}} - \mathbf{t}\| / \|\hat{\mathbf{t}}\| \times 100$. We define a pose to be successfully estimated if its both errors are under pre-defined thresholds.

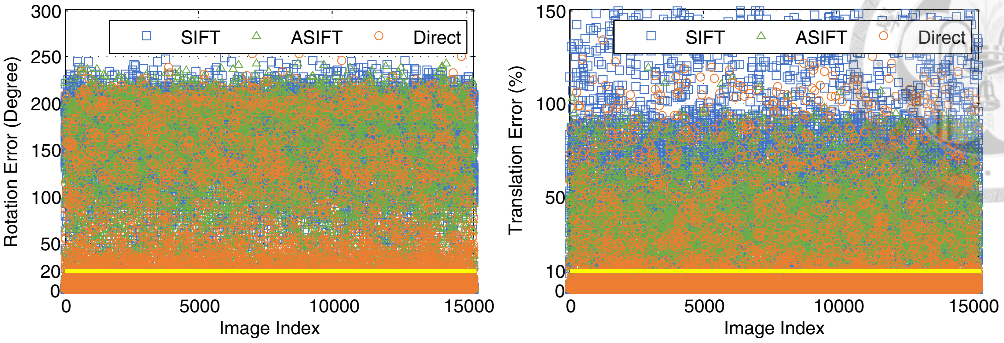


Figure 4.1: Distributions of rotation and translation errors over experiments. The horizontal lines correspond to the thresholds used to detect unsuccessfully estimated poses. There is a total of $15,289$ poses estimated by each pose estimation approach.

We use $\delta_{\mathbf{R}} = 20°$ and $\delta_{\mathbf{t}} = 10\%$ as the threshold on rotation error and translation error empirically, as shown in Figure 4.1. The success rate (SR) is defined as the percentage of the successfully estimated poses within each test condition.

4.1 Synthetic Dataset

We use a set of synthetic images consisting of 8400 test images for experiments, including 21 different test conditions. Each test image is generated from a warping target image according to the randomly generated pose with tilt angle in the range $[0°, 75°]$ in a randomly chosen background image, as shown in Figure 4.2. The target image size is 640×480. These images are classified into four different classes, namely “Low Texture”, “Repetitive Texture”, “Normal Texture”, and “High Texture” [57] as shown from top to bottom in Figure 4.2. Each class is represented by two targets. The background images are acquired from the database [58] and resized to 800×600 pixels. As mentioned in Section 3.2.1, IPPE [19] and OPn-P [18] are the candidate methods for us to obtain the second pose. To judge which one is better, we evaluate each performance using this dataset.

doi:10.6342/NTU201600474

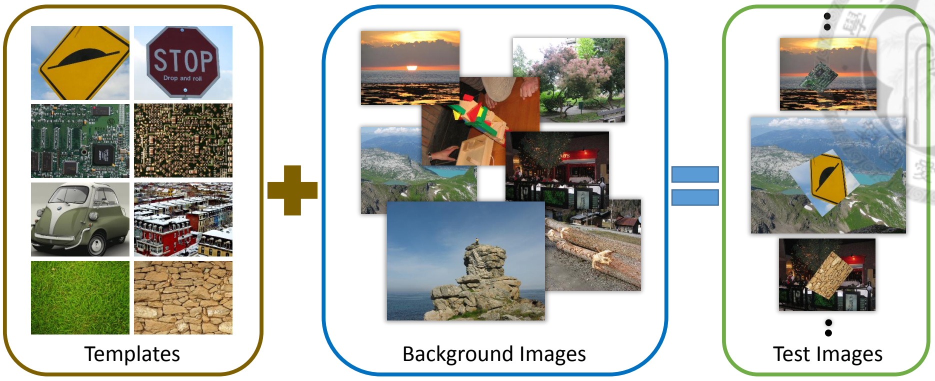


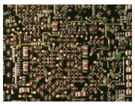
Figure 4.2: The test image was generated from a warping target image according to the randomly generated pose on randomly chosen background image.

4.1.1 Normal Condition

The pose estimation results of the SIFT-based, ASIFT-based, the proposed direct method without refinement, with refinement using IPPE and with refinement using OPnP under undistorted condition are shown in Table 4.1. Each test condition contains the average rotation error $E_{\mathbf{R}}$, translation error $E_{\mathbf{t}}$, and success rate. The evaluation results show that although the proposed method is sometimes slightly less accurate than the feature-based approaches, it performs more robustly with different target images. Although the SIFT-based approach can detect and match the features accurately under small tilt angle, it frequently fails in the experiments when the target undergoes large pose change. In the case of targets Philadelphia, the ASIFT-based method obtains accurate results due to abundant extinctive features in the target image. However it fails to correctly estimate the pose of the targets Grass (repetitive features), Bump Sign and Stop Sign (lacking features). It is obvious that the direct method with refinement obtains more accurate results than without refinement. Moreover, the direct method with refinement using OPnP performs slightly better than using IPPE most of the time.

doi:10.6342/NTU201600474

Table 4.1: Evaluation results under undistorted test images. D0, D1, D2 indicates the proposed direct method without refinement, refinement with IPPE and refinement with OPnP, respectively. The best values are highlighted in bold.

		SIFT	ASIFT	D0	D1	D2
Bump Sign 	$E_{\mathbf{R}}(^\circ)$	100	72.1	1.82	1.71	**1.65**
	$E_{\mathbf{t}}(\%)$	54.8	24.3	0.55	0.54	**0.53**
	SR(%)	10.0	22.0	**100**	**100**	**100**
Stop Sign 	$E_{\mathbf{R}}(^\circ)$	69.2	5.07	2.40	2.11	**1.95**
	$E_{\mathbf{t}}(\%)$	35.3	**0.74**	1.01	0.85	0.88
	SR(%)	38.0	96.0	**100**	**100**	**100**
Lucent 	$E_{\mathbf{R}}(^\circ)$	30.1	1.90	3.07	**1.01**	1.07
	$E_{\mathbf{t}}(\%)$	16.5	**0.38**	0.84	0.82	0.82
	SR(%)	72.0	**100**	96.0	98.0	98.0
MacMini Board 	$E_{\mathbf{R}}(^\circ)$	28.0	6.37	5.56	**2.85**	5.27
	$E_{\mathbf{t}}(\%)$	13.4	2.59	2.91	2.52	**2.23**
	SR(%)	78.0	96.0	96.0	**98.0**	**98.0**
Isetta 	$E_{\mathbf{R}}(^\circ)$	20.6	2.08	1.21	1.12	**1.04**
	$E_{\mathbf{t}}(\%)$	16.2	**0.49**	0.61	0.59	0.62
	SR(%)	82.0	98.0	**100**	**100**	**100**
Philadelphia 	$E_{\mathbf{R}}(^\circ)$	13.5	**1.16**	2.31	1.75	1.95
	$E_{\mathbf{t}}(\%)$	4.57	**0.35**	0.69	0.47	0.48
	SR(%)	90.0	**100**	**100**	**100**	**100**
Grass 	$E_{\mathbf{R}}(^\circ)$	97.3	51.2	2.5	1.93	**1.80**
	$E_{\mathbf{t}}(\%)$	212	16.7	2.16	2.12	**1.88**
	SR(%)	24.0	52.0	92.0	96.0	**98.0**
Wall 	$E_{\mathbf{R}}(^\circ)$	17.1	2.76	3.14	1.50	**1.39**
	$E_{\mathbf{t}}(\%)$	27.3	0.36	1.26	1.04	**1.01**
	SR(%)	86.0	96.0	96.0	**100**	**100**

4.1.2 Varying Conditions

We further evaluate the proposed methods using all target images with five degradation levels: Gaussian blur with kernel width of $\{1, 2, 3, 4, 5\}$ pixels, JPEG compression with the quality parameter set to $\{90, 80, 70, 60, 50\}$, intensity change with pixel intensity scalar parameter set to $\{0.9,0.8,0.7,0.6,0.5\}$, and tilt angle in the range of $\{[0°15°), [15°30°), [30°45°), [45°60°), \text{and } [60°75°)\}$.

The results are shown in Figure 4.3 and Figure 4.4. The proposed algorithm outperforms the other two feature-based methods with blurry images. All three approaches are able to deal with certain levels of distortion in intensity or JPEG compression noise. The SIFT-based approach performs well when the tilt angle is small since the marker images are not perspective distorted in the camera images. In the other conditions, however, the proposed algorithm and the ASIFT method are able to estimate 3D poses relatively well.

Detailed results are shown in Table 4.2 and Table 4.3. We note that the success of feature-based methods hinges on the number of correctly matching features. The ASIFT-based method sometimes performs equally well or better than the proposed algorithm in the case of targets Lucent, Philadelphia and Wall since there are numerous distinctive features in the target images. However in general, the proposed algorithm is more robust and accurate according to the tables.

In this synthetic image experiment, the proposed direct method without refinement achieves an overall success rate of 96.80%, while the proposed direct method with refinement using IPPE and OPnP achieve success rates of 97.23% and 97.52% respectively. SIFT-based approach achives 47.62% and ASIFT-based approach achieve 74.74%. Due to the slightly better performance, refinement using OPnP is applied in the proposed algorithm.

4.1.3 Refinement Analysis

To improve the accuracy of our pose estimation algorithm, we propose a refinement approach as described in Section 3.2.2. Pose estimation results (i.e., rotation

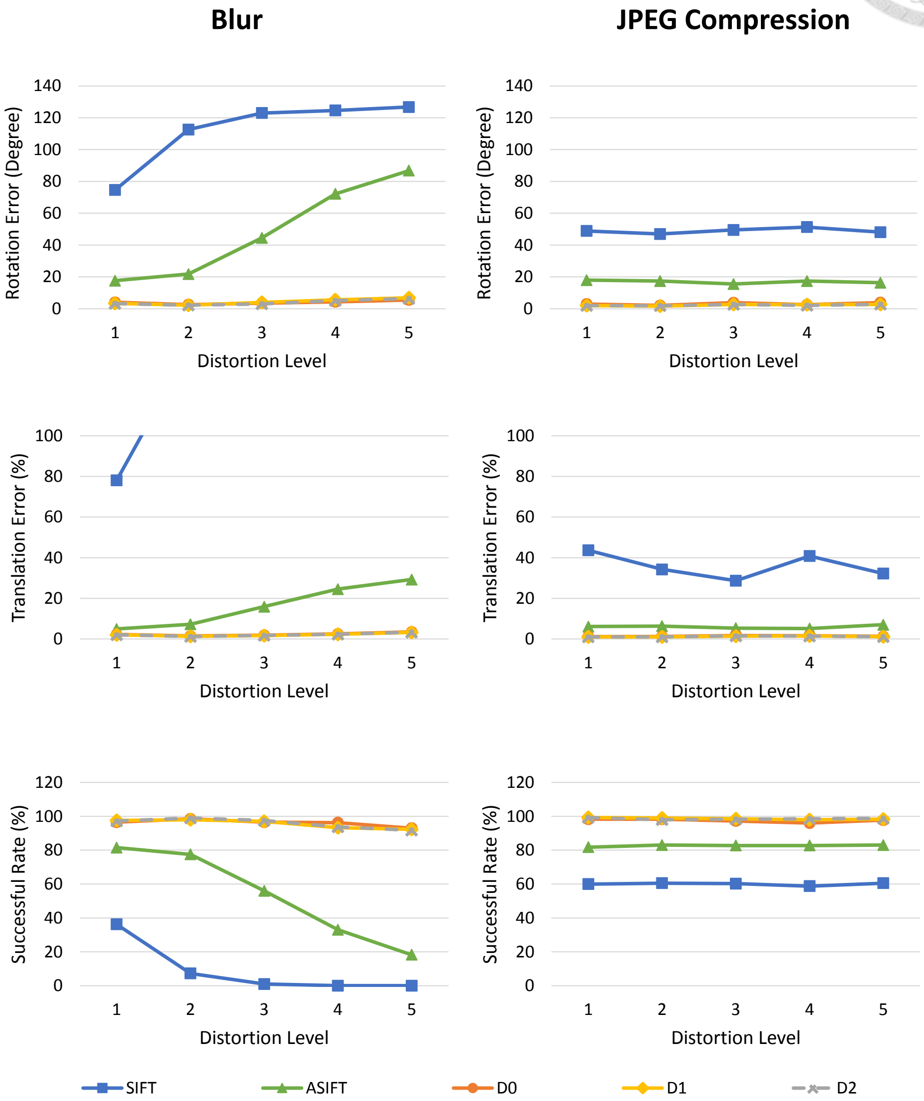


Figure 4.3: Experimental results on synthetic data under conditions blur and JPEG compression. D0, D1, D2 indicates the proposed direct method without refinement, refinement with IPPE and refinement with OPnP, respectively.

doi:10.6342/NTU201600474

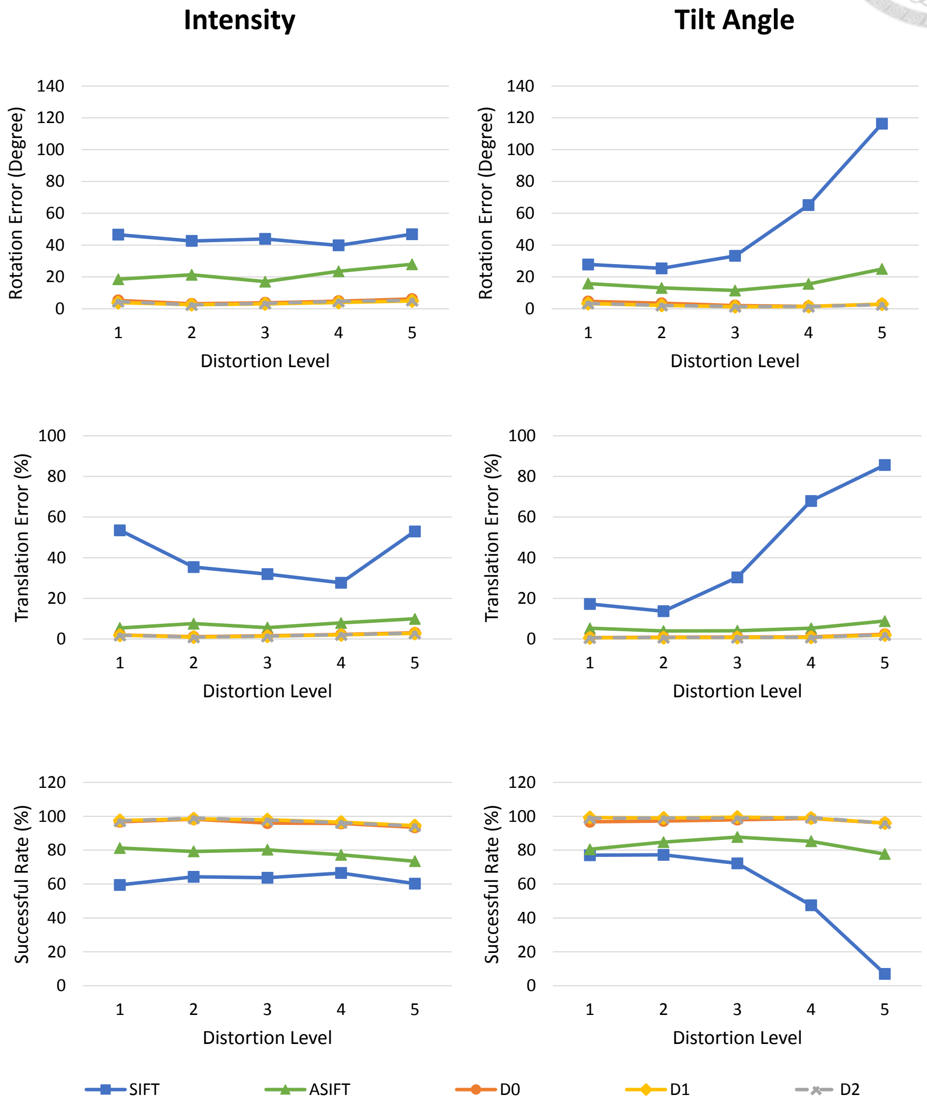


Figure 4.4: Experimental results on synthetic data under conditions intensity and tilt angle. D0, D1, D2 indicates the proposed direct method without refinement, refinement with IPPE and refinement with OPnP, respectively.

Table 4.2: Evaluation results under varying conditions. S, A, D0, D1 and D2 indicates SIFT-based, ASIFT-based, the proposed direct method without refinement, refinement with IPPE and refinement with OPnP, respectively.

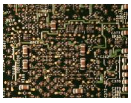
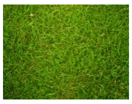
		Bump Sign 			Stop Sign 			Lucent 			MacMini Board 		
		$E_{\mathbf{R}}$(°)	$E_{\mathbf{t}}$(%)	SR(%)	$E_{\mathbf{R}}$(°)	$E_{\mathbf{t}}$(%)	SR(%)	$E_{\mathbf{R}}$(°)	$E_{\mathbf{t}}$(%)	SR(%)	$E_{\mathbf{R}}$(°)	$E_{\mathbf{t}}$(%)	SR(%)
B1	S	142	49.8	4.00	134.8	60.2	10.0	139	99.8	16.0	91.1	69.9	34.0
	A	69.3	19.5	28.0	6.04	0.79	94.0	7.80	0.93	92.0	6.57	1.17	94.0
	D 0	1.56	0.54	100	2.00	0.84	100	2.11	0.73	100	12.3	9.78	88.0
	D 1	1.34	0.58	100	1.68	0.91	100	1.41	0.58	100	12.5	9.47	88.0
	D 2	1.49	0.55	100	1.66	0.86	100	1.68	0.61	100	10.6	9.15	88.0
B2	S	153.7	56.0	0.00	156	60.6	2.00	171	114	0.00	165	181	2.00
	A	85.5	24.7	18.0	6.40	0.94	96.0	4.41	0.99	94.0	8.90	1.18	88.0
	D 0	2.78	0.87	98.0	1.96	0.89	100	1.09	1.40	98.0	5.21	2.80	96.0
	D 1	2.66	0.77	98.0	1.61	0.96	100	1.60	1.19	96.0	4.30	2.20	94.0
	D 2	2.60	0.78	98.0	1.64	0.93	100	1.03	1.10	98.0	3.85	2.26	98.0
B3	S	150	64.1	0.00	154	72.8	0.00	161	180	0.00	163	669	0.00
	A	101	29.5	10.0	34.5	6.26	68.0	39.8	9.24	72.0	61.0	20.0	50.0
	D 0	2.51	0.91	100	2.65	1.09	98.0	2.52	1.79	96.0	8.40	4.28	82.0
	D 1	2.96	0.97	98.0	3.46	1.02	98.0	5.87	1.67	96.0	8.23	3.92	88.0
	D 2	2.65	0.98	98.0	2.78	1.08	98.0	2.11	1.64	98.0	5.48	4.23	90.0
B4	S	158	53.4	0.00	164	76.0	0.00	166	398	0.00	169	459	0.00
	A	125	36.1	4.00	61.8	15.1	50.0	61.4	18.5	46.0	120	29.7	20.0
	D 0	3.04	1.72	98.0	4.39	1.40	94.0	3.26	1.74	100	6.92	3.42	90.0
	D 1	4.11	1.64	96.0	3.37	1.42	98.0	3.31	1.73	98.0	9.34	2.97	88.0
	D 2	4.23	1.56	96.0	3.34	1.48	98.0	3.09	1.68	98.0	9.78	3.18	88.0
B5	S	165	72.4	0.00	160	87.5	0.00	166	764	0.00	166	894	0.00
	A	124	38.1	0.00	114	27.6	28.0	96.6	26.9	24.0	119	27.3	18.0
	D 0	3.09	1.53	100	3.03	1.73	98.0	5.53	2.69	94.0	7.35	2.88	88.0
	D 1	6.23	1.44	94.0	3.16	1.68	96.0	4.10	2.50	100	6.01	2.29	92.0
	D 2	3.40	1.48	98.0	3.61	1.61	94.0	6.45	2.47	94.0	8.26	2.59	90.0
J1	S	148	60.5	2.00	70.5	29.2	52.0	58.8	36.7	66.0	40.9	50.8	74.0
	A	76.3	22.6	26.0	5.50	0.74	96.0	2.23	0.40	98.0	4.48	1.32	94.0
	D 0	1.44	0.68	100	2.80	0.84	96.0	1.13	0.49	100	8.65	3.63	92.0
	D 1	1.38	0.67	100	1.40	0.81	100	0.97	0.67	100	5.89	3.42	94.0
	D 2	1.39	0.68	100	1.41	0.81	100	0.97	0.41	100	6.10	3.50	94.0
J2	S	144	61.1	2.00	77.3	26.7	46.0	42.8	25.1	76.0	29.5	9.73	82.0
	A	93.7	27.9	28.0	11.1	0.81	94.0	1.64	0.36	100	4.18	1.59	96.0
	D 0	1.46	0.72	100	1.83	0.85	100	1.09	0.57	100	2.98	1.48	96.0
	D 1	1.46	0.64	100	1.10	0.73	100	1.09	0.51	100	1.96	1.17	98.0
	D 2	1.17	0.67	100	1.38	0.72	100	0.91	0.48	100	3.93	1.24	94.0
J3	S	143	52.4	6.00	76.0	27.6	52.0	65.2	28.0	62.0	33.8	11.3	80.0
	A	64.2	18.8	28.0	8.38	0.82	94.0	2.93	0.29	98.0	8.42	0.46	94.0
	D 0	1.87	0.57	100	2.58	0.92	98.0	1.10	0.64	100	12.2	6.63	88.0
	D 1	1.78	0.63	100	1.45	0.79	100	1.00	0.52	100	6.50	5.85	90.0
	D 2	1.68	0.56	100	1.30	0.76	100	0.94	0.56	100	9.26	6.00	92.0
J4	S	153	60.9	4.00	61.3	19.8	54.0	53.6	21.1	66.0	40.0	29.7	76.0
	A	91.5	24.0	22.0	4.37	0.64	96.0	3.26	0.42	98.0	6.12	0.50	96.0
	D 0	2.04	0.62	98.0	2.64	0.75	96.0	1.67	0.87	98.0	5.49	4.95	88.0
	D 1	1.97	0.66	98.0	1.92	0.73	98.0	1.25	0.69	100	8.09	4.43	94.0
	D 2	1.80	0.56	98.0	1.55	0.82	100	1.35	0.64	100	5.29	4.40	96.0
J5	S	153	57.8	2.00	58.9	27.4	60.0	48.1	16.9	68.0	32.1	24.9	80.0
	A	61.6	21.3	36.0	9.04	7.33	94.0	1.68	0.38	100	5.84	1.37	98.0
	D 0	1.45	0.70	100	1.80	0.75	100	1.43	0.61	100	12.6	3.64	90.0
	D 1	2.09	0.64	98.0	1.64	0.71	100	1.17	0.68	100	7.88	3.29	94.0
	D 2	1.40	0.64	100	1.68	0.65	100	1.13	0.69	100	11.2	3.53	92.0
I1	S	137	48.1	4.00	58.7	22.6	56.0	53.6	85.1	64.0	39.3	59.1	74.0
	A	72.7	17.5	28.0	9.26	0.84	90.0	3.18	0.42	96.0	10.2	1.48	92.0
	D 0	2.54	0.73	100	5.30	0.99	98.0	9.15	2.21	92.0	14.3	7.06	86.0
	D 1	2.59	0.61	98.0	2.10	0.88	100	7.43	2.14	96.0	9.26	6.71	88.0
	D 2	2.35	0.65	100	5.46	0.88	98.0	7.60	2.15	96.0	11.3	6.68	86.0
I2	S	139	53.4	12.0	60.2	23.9	56.0	33.2	34.7	80.0	30.4	20.6	80.0
	A	82.6	27.6	22.0	15.6	2.59	92.0	1.66	0.43	100	5.40	0.52	90.0
	D 0	1.43	0.53	100	1.79	0.80	100	1.81	0.53	98.0	8.67	2.13	94.0
	D 1	1.35	0.49	100	1.40	0.85	100	1.24	0.46	98.0	7.92	1.68	94.0
	D 2	1.24	0.50	100	1.53	0.86	100	1.28	0.51	100	5.98	1.75	96.0
I3	S	151	64.7	2.00	69.4	28.7	48.0	38.6	25.9	76.0	25.6	31.3	84.0
	A	79.2	20.6	18.0	5.14	0.80	98.0	4.13	0.40	96.0	2.50	0.47	98.0
	D 0	1.54	0.59	100	3.06	1.04	98.0	3.46	1.54	98.0	8.42	2.60	92.0
	D 1	1.51	0.57	100	1.90	0.96	100	3.18	1.60	98.0	2.78	1.97	98.0
	D 2	1.37	0.56	100	2.71	0.99	98.0	3.24	1.53	98.0	4.81	2.23	98.0
I4	S	149	59.3	6.00	47.8	9.80	64.0	50.6	29.9	66.0	16.9	15.4	90.0
	A	97.4	32.5	14.0	6.81	0.97	94.0	6.25	1.00	96.0	7.55	1.34	96.0
	D 0	2.20	0.61	98.0	2.00	0.63	100	14.6	6.68	90.0	5.70	1.49	96.0
	D 1	2.30	0.61	98.0	2.11	0.62	98.0	10.5	6.46	92.0	4.45	1.25	98.0
	D 2	1.90	0.67	100	1.70	0.62	100	12.1	6.68	92.0	4.59	1.26	98.0
I5	S	153	61.7	2.00	79.7	31.6	44.0	79.6	184	48.0	44.9	32.4	74.0
	A	107	37.3	8.00	10.75	1.57	88.0	25.0	4.69	82.0	8.88	1.06	96.0
	D 0	1.67	0.61	100	2.27	0.68	100	8.02	3.37	92.0	7.45	2.41	94.0
	D 1	1.65	0.53	100	1.68	0.61	100	5.15	3.12	94.0	7.88	2.10	92.0
	D 2	1.59	0.53	100	1.69	0.89	100	6.91	3.12	94.0	5.03	2.24	92.0
T1	S	136	50.7	12.0	30.7	9.43	68.0	6.95	6.58	96.0	0.62	0.17	100
	A	55.9	15.7	22.0	13.1	0.63	82.0	4.14	0.34	98.0	6.60	0.42	92.0
	D 0	3.84	0.51	100	4.90	0.65	98.0	2.09	0.38	100	7.34	0.98	92.0
	D 1	2.93	0.79	100	3.71	0.65	100	1.97	0.37	100	5.10	0.68	98.0
	D 2	3.30	0.46	100	3.84	0.68	100	1.83	0.37	100	5.24	0.70	98.0
T2	S	131	56.1	6.00	56.4	24.0	60.0	7.65	5.56	94.0	0.50	0.18	100
	A	58.8	14.9	32.0	6.70	0.66	92.0	3.48	0.33	96.0	3.67	0.42	98.0
	D 0	1.68	0.73	100	3.29	0.84	98.0	1.25	0.40	100	8.30	0.92	90.0
	D 1	1.63	0.67	100	1.94	0.78	100	1.18	0.45	100	4.81	0.36	94.0
	D 2	1.60	0.71	100	2.11	0.82	100	1.18	0.44	100	3.93	0.75	96.0
T3	S	141	64.8	6.00	73.4	24.7	56.0	20.3	11.6	86.0	0.34	0.20	100
	A	62.5	19.4	44.0	3.63	0.67	98.0	0.75	0.29	100	5.79	0.42	96.0
	D 0	1.03	0.66	100	1.21	0.88	100	0.74	0.46	100	2.33	1.66	98.0
	D 1	0.90	0.47	100	1.07	0.85	100	0.69	0.42	100	1.88	1.11	100
	D 2	0.91	0.50	100	1.01	0.77	100	0.68	0.43	100	1.94	1.16	98.0
T4	S	154	88.5	0.00	123	52.5	18.0	99.4	207	40.0	58.8	34.8	66.0
	A	77.4	23.2	36.0	6.81	0.81	96.0	0.85	0.43	100	6.60	0.47	96.0
	D 0	0.67	0.64	100	0.74	0.74	100	0.54	0.64	100	3.11	2.63	92.0
	D 1	0.55	0.61	100	0.66	0.67	100	0.50	0.57	100	2.56	1.73	94.0
	D 2	0.59	0.65	100	0.67	0.67	100	0.49	0.54	100	4.52	1.88	96.0
T5	S	150	51.7	0.00	165	57.2	0.00	169	97.0	0.00	168	145	0.00
	A	113	39.5	12.0	8.97	1.71	96.0	0.86	0.65	100	18.5	3.92	90.0
	D 0	0.53	0.58	100	0.57	0.74	100	0.53	0.67	100	17.0	9.60	80.0
	D 1	0.51	0.67	100	0.50	0.78	100	0.52	0.63	100	17.8	8.46	80.0
	D 2	0.54	0.66	100	0.45	0.78	100	0.53	0.65	100	15.0	8.67	80.0

Table 4.3: Evaluation results under varying conditions. S, A, D0, D1 and D2 indicates SIFT-based, ASIFT-based, the proposed direct method without refinement, refinement with IPPE and refinement with OPnP, respectively.

		Isetta			Philadelphia			Grass			Wall		
													
		E_R(°)	E_t(%)	SR(%)	E_R(°)	E_t(%)	SR(%)	E_R(°)	E_t(%)	SR(%)	E_R(°)	E_t(%)	SR(%)
B1	S	71.1	24.5	50.0	25.9	9.64	82.0	147	280	4.00	31.8	30.3	84.0
	A	7.37	0.88	92.0	2.93	0.40	98.0	87.5	15.5	46.0	2.00	0.37	98.0
	D 0	1.23	0.55	100	4.98	0.88	94.0	5.57	3.46	92.0	2.33	0.99	98.0
	D 1	0.99	0.54	100	1.79	0.61	100	5.52	3.23	92.0	1.54	0.90	100
	D 2	1.02	0.60	100	1.87	0.70	100	5.27	3.06	92.0	2.04	0.87	98.0
B2	S	125	53.9	14.0	116	46.3	22.0	161	285	0.00	129	351	16.0
	A	5.56	0.74	96.0	2.75	0.50	98.0	101	26.1	30.0	17.3	3.13	90.0
	D 0	1.85	0.72	100	3.07	2.09	98.0	2.51	1.94	98.0	2.15	1.29	100
	D 1	1.48	0.64	100	2.48	1.79	98.0	2.24	1.75	98.0	1.66	1.13	100
	D 2	1.69	0.72	100	2.53	1.75	98.0	2.25	1.61	100	1.64	1.07	100
B3	S	167	71.3	0.00	162	89.9	2.00	162	669	0.00	155	968	4.00
	A	31.4	11.2	74.0	20.1	9.98	78.0	137	35.1	6.00	53.2	12.0	62.0
	D 0	1.72	1.11	100	3.16	1.25	100	3.88	2.64	98.0	3.11	1.91	98.0
	D 1	1.78	0.97	100	2.41	0.92	100	4.31	2.52	96.0	2.48	1.62	100
	D 2	1.55	1.08	100	2.65	1.02	100	4.43	2.56	96.0	2.53	1.64	100
B4	S	159	62.6	0.00	167	163	0.00	152	893	0.00	167	715	0.00
	A	63.9	14.9	54.0	78.3	20.2	44.0	148	35.1	0.00	110	27.1	26.0
	D 0	2.02	1.53	100	4.49	2.20	98.0	6.24	4.96	94.0	4.33	2.90	96.0
	D 1	2.31	1.53	98.0	6.59	2.05	92.0	9.14	4.96	88.0	6.00	2.52	88.0
	D 2	1.99	1.55	100	7.08	2.06	90.0	6.25	4.93	88.0	4.65	2.48	92.0
B5	S	166	58.1	0.00	167	131	0.00	160	821	0.00	160	1316	0.00
	A	120	26.2	16.0	79.6	21.3	38.0	141	34.0	0.00	128	33.2	12.0
	D 0	0.06	0.59	100	5.45	1.54	96.0	4.96	5.85	86.0	12.2	10.4	82.0
	D 1	1.95	1.71	100	5.37	1.25	94.0	7.75	5.55	82.0	19.8	9.73	80.0
	D 2	3.03	1.73	96.0	4.40	0.33	96.0	6.77	5.38	88.0	14.9	10.1	78.0
J1	S	35.9	12.8	76.0	10.3	4.55	94.0	107	133	34.0	39.3	21.9	80.0
	A	4.76	0.52	96.0	1.25	0.29	100	93.1	22.4	36.0	5.31	0.71	98.0
	D 0	1.20	0.61	100	2.25	0.93	100	1.94	1.22	100	3.90	1.15	98.0
	D 1	1.10	0.59	100	1.75	0.80	100	1.30	1.10	100	1.58	0.77	100
	D 2	1.08	0.61	100	1.87	0.81	100	1.40	1.03	100	1.75	0.80	100
J2	S	52.7	15.8	64.0	19.7	15.6	90.0	98.7	109	36.0	28.1	11.7	86.0
	A	7.40	1.11	94.0	1.38	0.42	100	64.9	18.4	52.0	1.83	0.30	98.0
	D 0	0.95	0.78	100	2.46	0.92	100	2.25	3.34	94.0	3.30	1.40	96.0
	D 1	0.87	0.73	100	2.22	0.72	100	2.04	2.71	96.0	2.63	1.04	98.0
	D 2	0.87	0.72	100	2.04	0.72	100	1.86	2.88	94.0	3.08	1.05	96.0
J3	S	42.8	13.4	74.0	8.17	3.69	96.0	121	45.6	24.0	23.7	47.5	86.0
	A	5.04	0.53	92.0	2.52	0.33	98.0	74.8	21.2	48.0	1.86	0.35	100
	D 0	1.33	0.64	100	3.32	0.87	100	5.64	2.23	96.0	2.44	1.34	96.0
	D 1	1.22	0.55	100	2.24	0.64	100	4.74	1.85	98.0	2.18	0.89	100
	D 2	1.19	0.56	100	2.57	0.66	100	2.64	2.22	96.0	2.18	0.86	98.0
J4	S	51.7	16.2	70.0	23.2	8.91	86.0	118	115	28.0	31.7	55.1	84.0
	A	7.77	0.95	94.0	1.50	0.28	98.0	63.7	14.1	60.0	6.21	0.67	96.0
	D 0	1.55	0.68	98.0	1.63	0.71	100	2.22	2.92	96.0	2.84	1.76	94.0
	D 1	1.12	0.64	100	1.24	0.53	100	2.16	3.13	96.0	2.27	1.17	98.0
	D 2	1.00	0.61	100	1.27	0.53	100	2.03	3.02	96.0	2.21	1.09	98.0
J5	S	41.3	19.6	70.0	24.9	6.37	88.0	131	101	20.0	13.7	4.41	92.0
	A	5.05	0.63	92.0	1.37	0.29	100	88.8	24.7	36.0	2.47	0.37	98.0
	D 0	1.55	0.60	100	1.59	0.83	100	3.26	1.95	98.0	7.24	1.44	94.0
	D 1	1.43	0.51	100	1.25	0.61	100	2.94	1.63	96.0	2.76	0.93	98.0
	D 2	1.35	0.53	100	1.28	0.62	100	2.48	1.65	98.0	1.98	1.01	100
I1	S	32.4	12.4	76.0	21.2	6.72	88.0	130	170	20.0	14.8	23.2	90.0
	A	4.33	0.45	90.0	1.99	0.36	98.0	97.4	22.4	36.0	1.15	0.26	100.0
	D 0	1.69	0.62	100	2.80	0.83	100	1.83	1.27	100	4.48	2.16	98.0
	D 1	1.53	0.56	100	1.86	0.67	100	1.72	1.10	100	4.21	2.15	98.0
	D 2	1.60	0.55	100	2.29	0.68	100	1.66	1.06	100	4.34	2.20	98.0
I2	S	32.5	12.3	80.0	18.0	7.01	90.0	99.4	96.9	36.0	33.7	34.7	78.0
	A	7.70	0.71	92.0	1.36	0.32	100	111	28.1	28.0	2.64	0.43	98.0
	D 0	1.38	0.71	100	2.99	0.79	98.0	5.44	2.93	96.0	1.20	0.75	100
	D 1	1.25	0.65	100	1.62	0.69	100	4.95	2.41	96.0	1.14	0.70	100
	D 2	1.23	0.64	100	1.67	0.70	100	5.06	2.41	96.0	1.05	0.67	100
I3	S	31.6	8.00	80.0	27.4	7.27	84.0	107	80.8	36.0	6.68	9.08	94.0
	A	2.49	0.47	98.0	2.59	0.52	98.0	84.6	32.6	34.0	1.78	0.32	98.0
	D 0	2.33	0.58	98.0	2.96	0.80	98.0	6.42	5.46	86.0	1.61	0.68	98.0
	D 1	1.13	0.60	100	2.14	0.68	100	10.3	4.85	88.0	1.45	0.54	100
	D 2	1.09	0.51	100	2.09	0.68	100	10.8	4.66	88.0	1.25	0.57	100
I4	S	37.5	13.3	72.0	14.5	3.56	92.0	88.2	67.7	48.0	16.5	22.8	88.0
	A	8.58	0.61	90.0	3.36	0.39	98.0	110	24.9	28.0	11.9	2.11	94.0
	D 0	2.96	1.09	98.0	2.34	1.26	98.0	5.18	4.78	90.0	3.13	1.44	96.0
	D 1	2.67	1.01	98.0	2.31	1.06	96.0	5.80	3.97	94.0	1.69	1.44	98.0
	D 2	2.75	1.13	98.0	2.01	1.03	98.0	9.84	4.75	86.0	1.87	1.45	98.0
I5	S	41.6	14.0	74.0	18.0	10.4	90.0	56.1	69.9	58.0	20.4	19.1	88.0
	A	6.93	0.67	88.0	6.07	0.44	96.0	128	33.2	18.0	5.06	1.17	96.0
	D 0	3.07	0.61	98.0	8.59	2.99	90.0	16.6	12.7	76.0	1.32	1.54	98.0
	D 1	1.41	0.45	100	4.79	2.79	94.0	15.5	11.7	78.0	1.32	0.78	98.0
	D 2	1.52	0.55	100	4.98	2.88	94.0	18.3	12.2	76.0	1.27	0.75	98.0
T1	S	8.71	2.55	94.0	2.28	0.22	98.0	104	69.1	30.0	1.25	0.20	100
	A	7.33	0.46	90.0	5.53	0.37	94.0	75.4	24.3	40.0	4.11	0.30	96.0
	D 0	2.83	0.50	100	4.78	0.65	96.0	4.93	0.96	96.0	6.09	1.19	92.0
	D 1	2.57	0.45	100	2.83	0.42	100	3.14	0.91	100	4.21	0.91	96.0
	D 2	2.48	0.43	100	2.96	0.44	100	3.24	0.89	100	4.36	0.91	64.0
T2	S	11.5	2.12	92.0	0.49	0.20	100	57.1	21.5	60.0	0.48	0.20	100
	A	5.45	0.51	94.0	1.09	0.27	100	59.6	14.3	60.0	1.38	0.27	100
	D 0	1.48	0.77	100	2.53	1.01	98.0	2.64	1.16	98.0	6.36	1.15	94.0
	D 1	1.32	0.62	100	1.95	0.69	100	2.61	0.98	98.0	1.63	0.93	100
	D 2	1.39	0.65	100	2.19	0.63	98.0	2.52	1.08	98.0	1.74	0.98	100
T3	S	21.5	8.09	88.0	0.42	0.25	100	88.0	49.7	44.0	2.85	83.5	98.0
	A	3.08	0.38	98.0	0.82	0.31	100	46.5	11.1	66.0	0.90	0.34	100
	D 0	0.99	0.63	100	4.70	0.87	94.0	1.69	1.55	96.0	3.18	1.04	96.0
	D 1	0.84	0.56	100	2.78	0.65	98.0	1.48	1.42	98.0	1.01	0.75	100
	D 2	0.84	0.61	100	3.14	0.65	96.0	1.47	1.40	98.0	0.99	0.78	100
T4	S	67.5	22.8	60.0	9.33	3.76	92.0	147	120	18.0	20.3	14.5	86.0
	A	0.91	0.51	100	0.75	0.42	100	70.5	16.3	50.0	0.84	0.35	100
	D 0	0.55	0.41	100	1.18	0.91	100	1.32	0.97	100	3.83	1.89	98.0
	D 1	0.54	0.59	100	1.06	0.75	100	1.12	0.81	100	3.72	1.46	98.0
	D 2	0.55	0.52	100	1.05	0.66	100	1.10	0.77	100	2.13	1.43	98.0
T5	S	150	58.3	4.00	113	33.0	34.0	169	116	0.00	128	127	18.0
	A	5.27	0.84	98.0	0.63	0.48	100	115	23.0	28.0	3.54	0.89	98.0
	D 0	0.45	0.50	100	0.78	0.96	100	1.77	3.63	92.0	1.15	2.40	96.0
	D 1	0.40	0.59	100	0.68	0.61	100	1.54	2.93	90.0	1.07	1.47	98.0
	D 2	0.42	0.56	100	0.69	0.69	100	1.53	3.03	90.0	1.05	1.43	98.0

and translation error) with and without the refinement approach are shown in Figure 4.5. The rotation and translation error can be reduced averagely by $-0.517°$ and -0.169% respectively with proposed refinement scheme.

To demonstrate the proposed algorithm is able to disambiguate among plausible poses, we design another experiment conducted as follows: For each test, we choose a test image from the synthetic images. The target image in this test image is warped according to pose $\mathbf{p}_t$. An ambiguous pose $\mathbf{p}_a$ is then determined from $\mathbf{p}_t$ using the functional minimization method [18]. One of the two plausible poses $\mathbf{p}_a^{'}$ is randomly chosen and added some Gaussian noise. Later the refinement approach is applied to $\mathbf{p}_a^{'}$ for estimating the pose of the warped target image. Finally, we compute $E_{\mathbf{R}}$ and $E_{\mathbf{t}}$ 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 is able to disambiguate the plausible pose $\mathbf{p}_a^{'}$, the rotation error can be reduced significantly. We compare the proposed refinement method to the approach with only one candidate pose in Algorithm 1, and present the results in Figure 4.6. The success rate before refinement, refinement with one candidate pose, and refinement with two candidate poses are 51.26%, 51.19% and 90.34%, respectively. As shown in the figure, the rotation error are reduced drastically in the ambiguous cases, but the translation errors are relatively not because the translation terms of ambiguous poses are quite similar in most case. All the results show that the proposed refinement method can help improve estimation accuracy and address pose ambiguity problem effectively.

It can be observed in Figure 4.6(b) that there are few cases that the refinement with two candidate poses performs worse than the proposed method without refinement. We discuss the success and failure of the proposed refinement scheme in the rest of this subsection. Assume that the initial pose estimate is near an ambiguous pose, as the purple circle shown in Figure 4.7(a). If we try to refine this pose estimate directly, we cannot obtain the correct result denoted by the orange triangle as this pose estimate will be trapped at a local minimum denoted by the

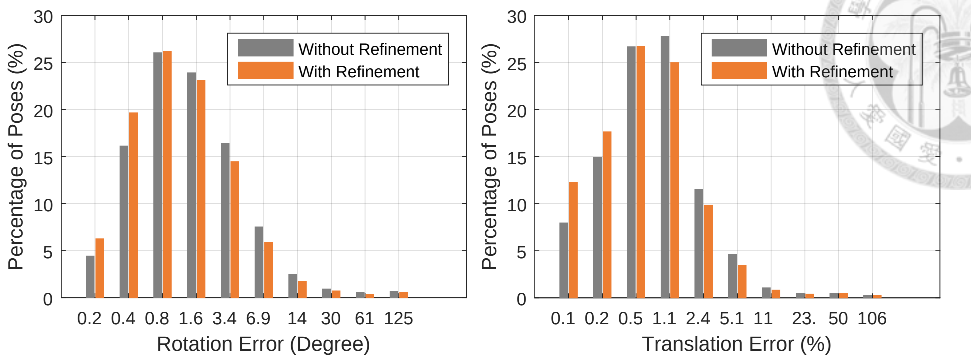
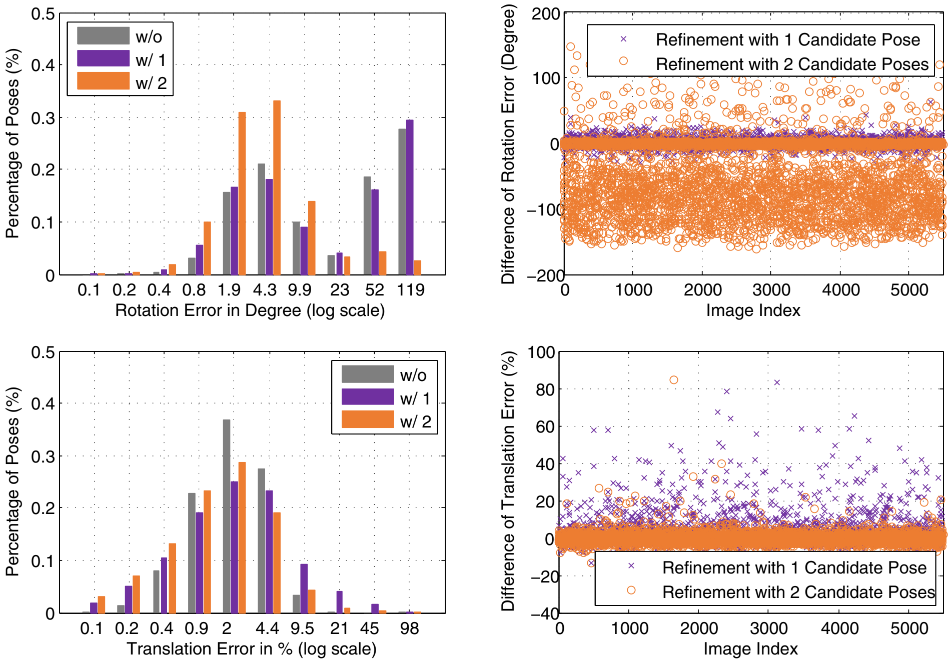


Figure 4.5: Pose estimation results with and without refinement approaches. The average value of rotation and translation error are both reduced by the refinement approach.



(a) Distribution of errors.

(b) Difference of errors.

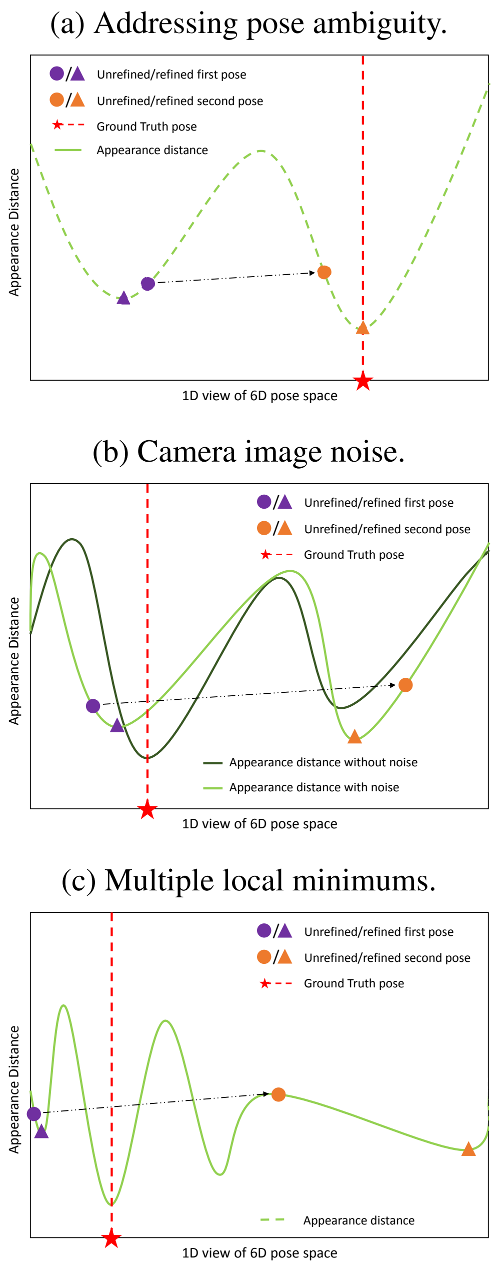
Figure 4.6: The proposed method without refinement (w/o), refinement with one candidate (w/ 1), and refinement with two candidates (w/ 2) are evaluated. (a) Distribution of errors. (b) The difference of pose errors before and after applying two kinds of refinement approaches.

purple triangle shown in Figure 4.7(a). On the other hand, it is feasible to obtain the correct result by considering both of the ambiguous poses in the refinement process. Even with a poor initial pose estimate, it is still workable to obtain the proper result.

There are some cases which the proposed refinement method generates worse result. For example, the rotation error can become larger when the appearance distance value is not consistent with the real one due to camera image noise. This case is illustrated in Figure 4.7(b) where the two initial pose estimates are denoted by the purple circle and the orange circle. The appearance distance function for pose estimation without noise is denoted by the dark green line, and the function with image noise is represented by the light green line. In addition, the unknown ground truth pose is indicated by star on the x-axis. Through refining each pose, the corresponding purple and orange triangles are reached. However, the second estimated pose denoted by orange triangle is selected based on the proposed refinement scheme. In this case, the resulting pose estimate after the refinement is worse.

With respect to the translation error, the difference between initial pose and refined pose is much smaller no matter the correct ambiguous pose is chosen or not. This is because the translation terms of ambiguous poses are usually similar with each other, as the examples shown in Figure 1.1. Nevertheless, some poses may be refined with increasing translation errors as shown in Figure 4.6(b). The reason is that the appearance distance function has multiple local minima and the initial pose estimate is refined to a local minimum which is not the location of each ambiguous pose, as illustrated in Figure 4.7(c). The final pose, denoted by orange triangle, is trapped at a local minimum unrelated to the ambiguous poses and results in larger rotation or translation error.

doi:10.6342/NTU201600474


Unrefined/refined first pose
Unrefined/refined second pose
Ground Truth pose
Appearance distance
Appearance Distance
1D view of 6D pose space
Appearance distance without noise
Appearance distance with noise

(a) Addressing pose ambiguity.

(b) Camera image noise.

(c) Multiple local minimums.

Figure 4.7: Illustration to addressing pose ambiguity problems. (b), (c) are two cases that the refinement process with two candidate poses performs worse than with one candidate pose. The red dash line indicates the location of the ground truth pose. The purple circle indicates the first initial pose which is the input of the refinement scheme, and the orange circle indicates the second initial pose. The purple and orange triangle represent the refined first and second poses, respectively.

4.2 Real Dataset

In this experiment we investigate the performance of proposed method on a benchmark dataset by Gauglitz *et al.* [1], originally used to evaluate the tracking-related algorithms. This dataset consists of 96 videos with a total of 6889 frames including 6 different targets with 16 different conditions. The frame size in this data set is 640×480 pixels, and we resize the target image to 570×420 pixels. It is definitely a challenging database for the pose estimation problem due to significant viewpoint change, drastic illumination difference, and noisy camera images.

As the benchmark dataset only provides homography parameters of the videos, we have to calculate the 6D poses from these homography parameters by ourselves. This process is conducted as follows:

1. Straighten the camera images with calibration coefficients.
2. Modify the texture margin from 1.5 to 0.75 (the original texture margin is incorrect).
3. Explore the 4 correspondences of target image corners with provides homography parameters.
4. Compute the candidate poses with the 4 correspondences using OP*n*P [18].
5. Select the correct pose from all candidate poses.

Similar to [1], these pose parameters are used as the ground truth for performance evaluation. In addition, some homography parameters of some video sequences (i.e., "br-m1", "pa-pn", etc.) are recorded in the wrong way, we manually correct all of them.

The complete comparison results of two feature-based methods and the proposed direct algorithm are shown in Table 4.4 and Table 4.5. While OP*n*P performs well in pose estimation, the success hinges on whether feature can be well matched. The difficulty of the feature-based approaches to cope with motion blur

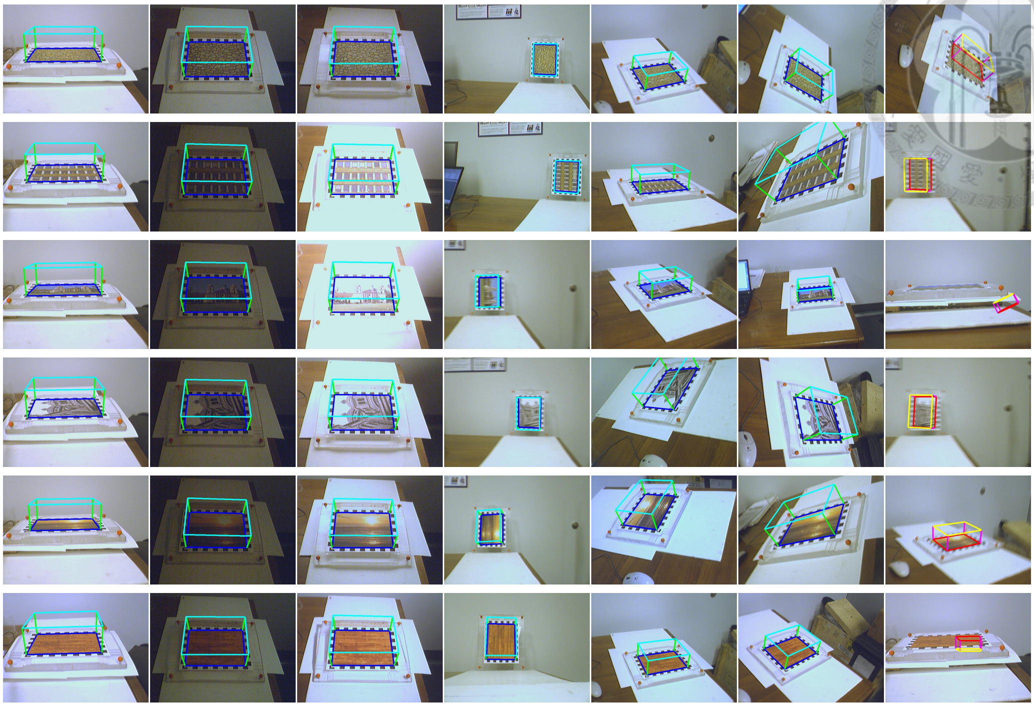


Figure 4.8: Estimation results by the proposed direct method on real images under different conditions. The success cases are represented with rendered cyan boxes, and the failure cases are represented with rendered magenta boxes.

is apparent. On the other hand, the proposed method can still estimate poses with low translation error and slightly higher rotation error under severe blur condition. As motion blurs are likely occur in AR applications, the proposed algorithm can be better applied to estimate 3D pose than feature-based approaches. However, if the target image appears a extremely flat color (target Wood) in the camera image, our proposed method still might fail because the appearance between the target image and its local patches are almost undistinguishable. Sample images rendered model with pose obtained from proposed algorithm are shown in Figure 4.8. Overall, the proposed direct method outperforms the feature-based approaches within an success rate 67.66%. The success rate of the SIFT-based and ASIFT-based approaches are 29.34% and 46.10% respectively.

doi:10.6342/NTU201600474

Table 4.4: Experimental results of the visual tracking dataset [1] under different conditions. The SIFT-based (S), ASIFT-based (A), and the proposed direct (D) methods are evaluated under different conditions (uc: unconstrained; pn: panning; rt: rotation; pd: perspective distortion; zm: zoom; mX: motion blur level X, X = 1...9; ls: static lighting; ld: dynamic lighting). The best results in each condition are highlighted in bold.

		Bricks			Building			Mission		
		E_{R}(°)	E_{t}(%)	SR(%)	E_{R}(°)	E_{t}(%)	SR(%)	E_{R}(°)	E_{t}(%)	SR(%)
uc	S	73.3	126	41.6	110	183	11.2	63.6	34.5	48.4
	A	75.1	**19.2**	35.8	82.6	25.5	29.4	49.3	15.2	57.0
	D	**61.4**	34.9	**42.6**	**17.9**	**8.63**	**83.4**	**15.3**	**6.94**	**85.4**
pn	S	14.8	3.96	90.0	124	83.4	0.00	33.2	26.2	70.0
	A	36.6	13.4	64.0	109	46.1	0.00	7.45	**1.06**	90.0
	D	**7.77**	1.98	**96.0**	**3.90**	**1.68**	**98.0**	**6.37**	2.02	**100**
rt	S	**1.08**	**0.28**	**100**	72.9	80.3	34.0	2.65	**0.39**	98.0
	A	3.76	0.61	98.0	29.6	10.0	52.0	**2.02**	0.44	**100**
	D	68.7	50.8	32.0	**7.71**	**2.41**	**94.0**	3.42	0.83	**100**
pd	S	41.1	138	66.0	82.0	104	34.0	37.3	15.4	70.0
	A	**40.7**	**13.1**	70.0	50.2	**16.8**	64.0	24.5	**7.21**	80.0
	D	53.8	22.8	**72.0**	**19.5**	18.0	**82.0**	**19.5**	14.4	**84.0**
zm	S	**1.18**	**0.30**	**100**	95.0	128	16.0	**5.30**	**0.56**	94.0
	A	27.9	8.56	58.0	50.0	15.7	50.0	8.73	0.85	78.0
	D	53.3	57.2	48.0	**0.95**	**0.45**	**100**	1.86	0.74	**100**
m1	S	**6.23**	**0.39**	**100**	127	100	1.12	**8.80**	**0.48**	90.9
	A	65.1	34.4	39.8	113	48.0	0.00	18.9	2.06	55.7
	D	79.0	15.6	27.3	**15.2**	**2.32**	**88.8**	13.0	1.53	**90.9**
m2	S	**68.6**	35.5	**31.1**	130	52.6	2.22	13.2	4.98	**95.6**
	A	106	46.8	6.67	104	43.2	0.00	18.6	2.62	57.8
	D	97.6	**16.4**	15.6	**19.4**	**1.41**	**84.4**	**14.4**	**0.94**	88.9
m3	S	123	88.1	9.38	141	67.0	0.00	89.9	46.2	18.8
	A	**99.4**	43.6	6.25	98.7	44.2	0.00	21.1	4.78	71.9
	D	103	**14.8**	**12.5**	**19.8**	**3.28**	**68.8**	**16.2**	**1.68**	**75.0**
m4	S	124	104	**8.70**	127	76.9	0.00	99.1	52.4	13.0
	A	106	42.1	4.35	111	43.3	0.00	96.7	38.8	8.70
	D	**88.2**	**17.5**	**8.70**	**36.7**	**8.96**	**73.9**	**17.6**	**1.42**	**60.9**
m5	S	115	104	**15.8**	146	87.5	0.00	91.2	537	15.8
	A	**93.7**	42.3	10.5	109	46.6	0.00	92.8	40.3	15.8
	D	108	**22.2**	0.00	**34.3**	**6.97**	**66.7**	**21.0**	**2.28**	**36.8**
m6	S	115	121	**16.7**	140	111	0.00	101	79.4	16.7
	A	105	51.2	0.00	102	43.9	0.00	90.3	36.8	11.1
	D	**105**	18.2	5.55	**53.7**	**12.3**	**38.9**	**21.4**	**10.1**	**44.4**
m7	S	**105**	85.0	**18.8**	131	120	0.00	102	107	18.8
	A	109	51.2	0.00	114	40.4	0.00	90.4	36.7	12.5
	D	111	**23.9**	0.00	**61.2**	**11.3**	**43.8**	**24.4**	**0.66**	**25.0**
m8	S	125	195	**13.3**	133	180	0.00	106	50.5	20.0
	A	104	35.0	6.67	98.4	45.2	0.00	70.8	33.8	13.3
	D	**106**	20.8	6.67	**78.9**	**17.4**	**42.9**	**23.8**	**1.51**	**46.7**
m9	S	108	70.9	**14.3**	130	98.2	0.00	99.6	36.5	15.4
	A	109	49.5	0.00	92.6	41.2	0.00	82.7	35.6	15.4
	D	**93.1**	16.4	**14.3**	**93.4**	**19.4**	**35.7**	**25.9**	**1.44**	**53.8**
ls	S	58.6	78.1	50.0	76.4	91.6	40.0	61.0	26.6	50.0
	A	**24.8**	**9.21**	**75.0**	69.5	**21.4**	37.5	**0.89**	**0.44**	**100**
	D	46.2	18.7	65.0	**53.3**	57.2	**48.0**	9.41	5.50	96.0
ld	S	86.3	317	29.0	111	122	14.0	94.4	40.4	26.0
	A	**45.9**	**14.1**	**58.0**	73.6	23.6	32.0	3.80	1.04	98.0
	D	87.0	52.0	25.0	**7.60**	**6.42**	**95.0**	**1.92**	**0.66**	**100**

Table 4.5: Experimental results of the visual tracking dataset [1] under different conditions. The SIFT-based (S), ASIFT-based (A), and the proposed direct (D) methods are evaluated under different conditions (uc: unconstrained; pn: panning; rt: rotation; pd: perspective distortion; zm: zoom; mX: motion blur level X, X = 1...9; ls: static lighting; ld: dynamic lighting). The best results in each condition are highlighted in bold.

		Paris			Sunset			Wood		
		E_R(°)	E_t(%)	SR(%)	E_R(°)	E_t(%)	SR(%)	E_R(°)	E_t(%)	SR(%)
uc	S	52.2	103	56.2	113	205	2.00	124	350	0.00
	A	11.1	3.80	91.0	63.1	19.5	40.4	80.0	**24.1**	**28.0**
	D	**2.68**	**0.93**	**99.0**	**5.96**	**10.2**	**60.6**	**77.5**	52.6	23.6
pn	S	15.1	13.2	74.0	116	87.2	0.00	124	138	0.00
	A	16.0	1.23	60.0	70.4	25.1	22.0	112	41.7	0.00
	D	**3.66**	**0.91**	**96.0**	**13.0**	**1.34**	**88.0**	**48.6**	**10.6**	**62.0**
rt	S	3.59	0.67	98.0	44.3	22.5	28.0	126	144	0.00
	A	**1.34**	**0.35**	**100**	15.1	1.58	58.0	**5.38**	**1.38**	**94.0**
	D	1.56	0.46	**100**	**3.65**	**1.38**	**100**	109	78.0	0.00
pd	S	32.1	30.8	74.0	102	86.2	2.00	120	428	0.00
	A	**24.6**	**7.45**	**84.0**	43.9	14.5	64.0	**51.0**	**13.5**	**62.0**
	D	27.5	22.7	80.0	**27.9**	**13.5**	**80.0**	**51.0**	63.6	42.0
zm	S	2.57	**0.42**	**100**	95.5	117	14.0	111	146	8.00
	A	7.51	0.43	74.0	21.7	4.11	54.0	**58.9**	**17.1**	**42.0**
	D	**2.14**	0.56	**100**	**5.02**	**1.51**	**100**	101	68	0.00
m1	S	16.1	1.29	69.0	118	75.6	0.00	119	80.0	0.00
	A	15.7	0.86	67.8	106	37.3	1.14	95.9	46.1	0.00
	D	**7.53**	**0.50**	**98.9**	**13.6**	**1.82**	**83.0**	**41.0**	**2.37**	**25.0**
m2	S	22.9	22.9	68.2	137	263	0.00	126	125	0.00
	A	17.1	1.33	63.6	125	47.9	0.00	102	43.7	0.00
	D	**6.70**	**0.62**	**100**	**16.4**	**3.60**	**75.6**	**51.1**	**3.32**	**17.8**
m3	S	93.2	429	16.1	128	87.0	0.00	130	221	0.00
	A	20.0	1.56	54.8	119	47.0	0.00	111	50.0	0.00
	D	**9.14**	**0.76**	**93.5**	**16.6**	**2.12**	**70.0**	**60.8**	**2.83**	**13.3**
m4	S	102	354	4.55	131	77.3	0.00	122	154	0.00
	A	37.5	9.72	54.5	112	58.1	0.00	100	42.3	0.00
	D	**9.16**	**0.59**	**90.9**	**16.9**	**2.85**	**65.2**	**68.8**	**2.77**	**13.0**
m5	S	111	216	11.1	139	74.9	0.00	140	104	0.00
	A	92.4	42.8	11.1	112	50.2	5.00	101	50.4	0.00
	D	**11.0**	**0.56**	**100**	**18.3**	**2.79**	**55.0**	**79.3**	**4.35**	0.00
m6	S	103	207	16.7	128	57.7	0.00	123	249	0.00
	A	80.7	31.9	11.1	126	42.9	0.00	128	42.2	0.00
	D	**12.0**	**1.09**	**94.4**	**18.6**	**3.51**	**61.1**	**77.8**	**4.04**	**11.1**
m7	S	111	157	18.8	122	148	0.00	119	163	0.00
	A	94.4	35.8	18.75	122	48.3	0.00	115	43.7	0.00
	D	**13.5**	**2.06**	**87.5**	**21.1**	**3.66**	**31.3**	**77.4**	**3.43**	**12.5**
m8	S	102	191	20.0	132	74.4	0.00	127	205	0.00
	A	71.9	36.5	20.0	119	54.8	0.00	102	46.4	0.00
	D	**14.2**	**2.01**	**80.0**	**20.2**	**3.51**	**53.3**	**72.0**	**4.61**	**13.3**
m9	S	93.6	183	14.3	135	160	0.00	122	91.6	0.00
	A	78.7	32.6	21.4	95.9	42.3	0.00	115	48.4	0.00
	D	**17.5**	**2.00**	**57.1**	**20.4**	**4.22**	**50.0**	**76.2**	**3.69**	**7.69**
ls	S	55.5	33.3	56.3	108	44.3	8.75	125	114	0.00
	A	**0.90**	**0.61**	**100**	39.7	10.4	61.3	52.5	18.3	51.3
	D	1.47	**0.61**	**100**	**7.74**	**7.78**	**72.5**	**5.27**	**5.17**	**96.3**
ld	S	84.3	76.8	27.0	125	78.6	2.00	126	182	0.00
	A	**0.88**	**0.40**	**100**	55.4	**19.2**	45.0	51.3	20.2	50.0
	D	14.2	14.4	84.0	**22.6**	26.5	**50.0**	**25.4**	**17.3**	**81.0**

doi:10.6342/NTU201600474

4.3 Runtime Comparison

We run all algorithms in MATLAB on a desktop computer with 3.6 GHz Core-i7 CPU and 16 GB RAM. Table 4.6 shows average runtimes for different algorithms. The SIFT-based method has the lowest average runtime, but it gives the worst results most of the time. The proposed direct method is slightly more efficient than ASIFT-based method for synthetic data but it is less efficient for real data compared to ASIFT-based method. The reason is that numerous images in the real dataset [1] are noisy and blurry, and thus the error measures of neighboring candidate poses are similar. In such cases, the pose pruning process is less effective as there is no distinct candidate pose. On the other hand, since fewer features can be detected in theses images, the computational time for feature matching and outlier removal in the ASIFT-based pose estimation is reduced, which makes the ASIFT-based method more efficient. However in such cases, the ASIFT-based method achieves worse results in terms of accuracy.

4.4 Textureless Images

As the target images provided by the synthetic and real datasets are not totally textureless, Figure 4.9 shows some pose estimation results with textureless images using the proposed direct method. Here we do not show the results using feature-based approaches because they are not able to estimate pose due to lack of sufficient feature correspondences. These results demonstrate that our method can perform pose estimation robustly not only on targets with texture but also on textureless targets.

doi:10.6342/NTU201600474

Table 4.6: Average runtimes for three approaches on synthetic and real test data. Although SIFT-based Approach is the fastest method among these three different schemes, its performance is quite limited.

<table>
<tr><th>Data Type</th><th colspan="4">SIFT-based Approach</th></tr>
<tr><td></td><td>SIFT</td><td>RANSAC</td><td>OPnP</td><td>Total</td></tr>
<tr><td>Synthetic</td><td>10.56 s</td><td>0.08 s</td><td>0.02 s</td><td>10.67 s</td></tr>
<tr><td>Real</td><td>5.09 s</td><td>0.08 s</td><td>0.02 s</td><td>5.19 s</td></tr>
<tr><td></td><td colspan="4">ASIFT-based Approach</td></tr>
<tr><td></td><td>ASIFT</td><td>RANSAC</td><td>OPnP</td><td>Total</td></tr>
<tr><td>Synthetic</td><td>46.45 s</td><td>0.07 s</td><td>0.02 s</td><td>46.58 s</td></tr>
<tr><td>Real</td><td>24.91 s</td><td>0.09 s</td><td>0.02 s</td><td>25.08 s</td></tr>
<tr><td></td><td colspan="4">Proposed Direct Method</td></tr>
<tr><td></td><td>Approximated</td><td colspan="2">Refinement</td><td>Total</td></tr>
<tr><td>Synthetic</td><td>38.35 s</td><td colspan="2">2.16 s</td><td>40.51 s</td></tr>
<tr><td>Real</td><td>35.13 s</td><td colspan="2">1.29 s</td><td>36.42 s</td></tr>
</table>

doi:10.6342/NTU201600474

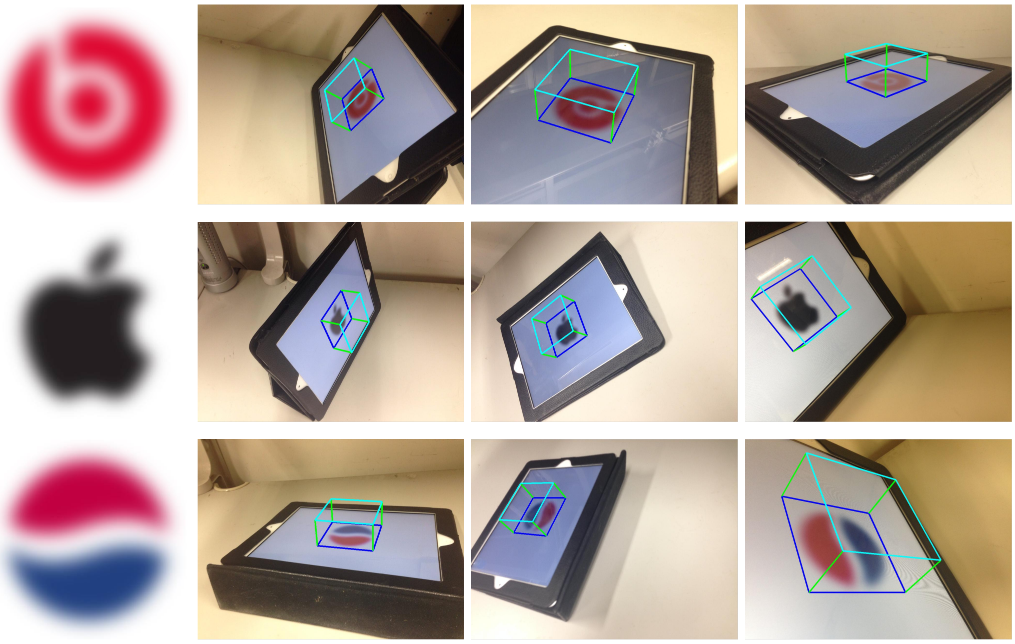


Figure 4.9: Pose estimation results on textureless images by proposed direct method. First column: pose estimation target. Other columns: pose estimation results with rendered box.

Chapter 5

Direct Pose Estimation and Tracking System

5.1 Proposed System

The proposed system consists of two parts, as shown in Figure 5.1. The pose estimation unit (PEU) is in responsible for finding the initial pose and re-computing the pose when the pose tracker loses the track. The pose tracker (PT) takes the initial estimated pose and performs tracking with the proposed 3-scale search scheme.

We reference the approximated pose estimation scheme in the proposed algorithm and consider the structure of GPU to design the pose estimation unit. The refinement scheme in the proposed algorithm is not considered. The main reason is that the approximated pose estimation performs the same computation with a large amount of poses, which is suitable for parallel computing on GPU. However the refinement scheme is not parallelizable since most of the processes must be executed in a sequential manner. To make the system more accurate, the final precision parameter ε_c^* in Algorithm 1 is set to be slightly finer than that in the proposed algorithm. We describe the detail of the system in the rest of this section.

doi:10.6342/NTU201600474

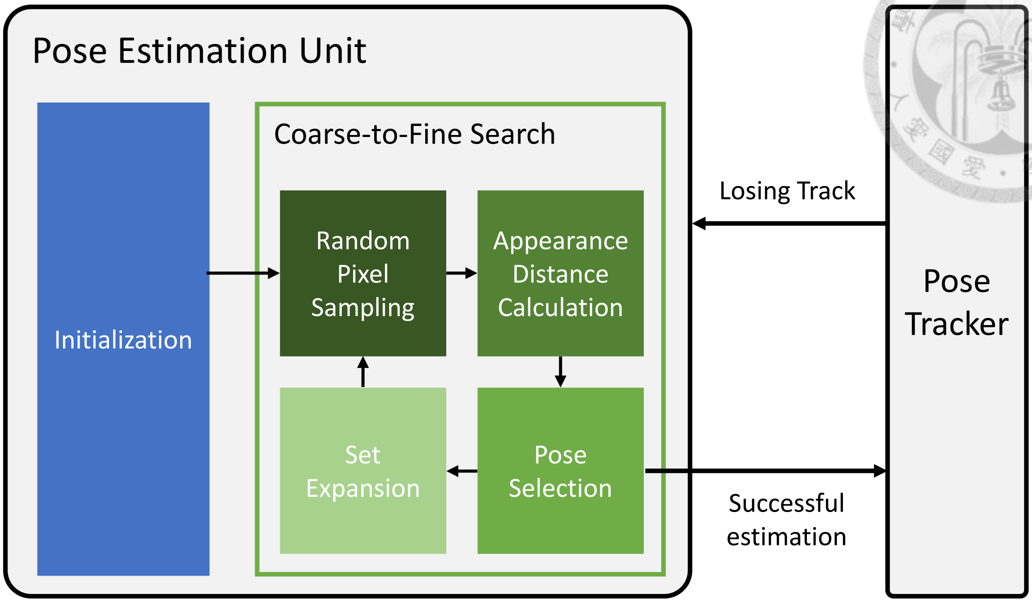


Figure 5.1: State chart of proposed D-PET system. The system consists of two main parts, which are the pose estimation unit (PEU) and the pose tracker (PT).

5.1.1 Initialization

The initialization system of PEU processes three main works before the coarse-to-fine estimation. First, the parameters such as step size on each pose dimension, camera intrinsic parameters are calculated according to the user input. Second, both the target image and the camera image are smoothed with Gaussian blur k-ernel. According to (3.2), if the mean variation $\bar{\mathcal{V}}$ is constrained, the difference between $E_a(\mathbf{p}_1)$ and $E_a(\mathbf{p}_2)$ is bounded only in terms of precision parameter ε, which makes the coarse-to-fine estimation more accurate. We accelerate Gaussian blur and the mean variation computing with shared memory in GPU which dramatically reduces the number of global memory reads during the process.

The final work is ε-covering set construction. The origin method is not parallelizable since it determines every parameters sequentially. However, according to Table 3.1, the step sizes of θ_{z_t}, θ_{z_c}, t_x and t_y are all constant given a pair of t_z and r_x. This observation gives us a chance to utilize parallel computing. The

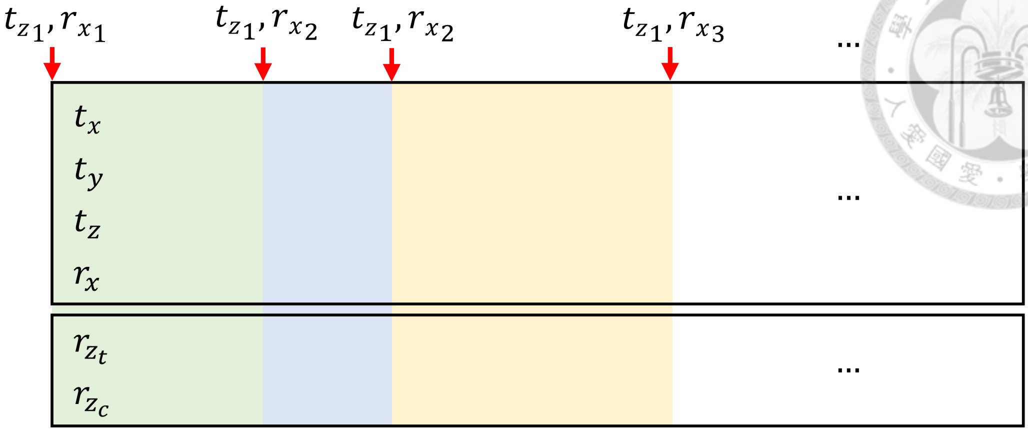


Figure 5.2: Initial ε-covering set construction. Two factors are determined sequentially while the others are calculated parallelly. The six pose parameters are stored in a grouped 4 floats and 2 floats structure in global memory.

proposed solution is to determine the combinations of t_z and r_x sequentially, and then construct the subset for each combination in parallel, as shown in Figure 5.2. Moreover, the six pose parameters are stored in a single grouped 4 floats and single grouped 2 floats structure since the global memory instructions in GPU support reading or writing with 2 floats or 4 floats. This memory allocation reduces the number of global memory instructions and makes the system more efficient.

5.1.2 Coarse-to-Fine Estimation.

The process is divided into four parts. The random pixel sampling and the appearance distance calculation are in charge of Step 2 and 7 in Algorithm 1. The pose Selection is responsible for Step 4 while the set expansion accomplish Step 5 and 6 in Algorithm 1.

Random Pixel Sampling. A set of points $\mathbf{x}_i = [x_i, y_i, 0]^\top, i = 1 \ldots, n$ on the target image I_t is randomly sampled. Since these points are used in calculating the appearance distances E_a for all poses during the process, the coordinates and the pixel values of the points are loaded into the uniform cache with LDU (load uniform) instructions in GPU. The uniform cache benefits the appearance distance calculation by reducing the memory reading time.

Appearance Distance Calculation. The appearance distance $E_a(\mathbf{p})$ for poses $\mathbf{p}$ in the set is calculated in parallel. For each pose $\mathbf{p}$, the system reads six pose parameters from the memory described in Figure 5.2. The coordinate $\mathbf{x}_i$ and the pixel value $I_t(\mathbf{x}_i)$ of each sample point are also loaded from the uniform cache. After that, the coordinate of corresponding camera image pixel $\mathbf{u}_{i\mathbf{p}}$ is then computed using (3.3) and the pixel value $I_c(\mathbf{u}_{i\mathbf{p}})$ is obtained from the global memory. Finally, $E_a(\mathbf{p})$ is calculated using (2.6) and written to the global memory.

Calculating appearance distance takes the longest processing time in the system because of the large amount of global memory reading for camera image pixel values. To improve the efficiency, we store the camera image pixel values in the texture memory, which is benefited by the on-chip texture cache designed for applications which the memory access patterns exhibit spatial locality. According to (3.1), the spatial distance between $\mathbf{u}_{i\mathbf{p}_1}$, $\mathbf{u}_{i\mathbf{p}_2}$ computed by two nearby poses $\mathbf{p}_1$, $\mathbf{p}_2$ is bounded, which makes the texture memory suitable for our system.

Pose Selection. We apply the min_element function in thrust library [59] to find the minimum appearance distance $E_a(\mathbf{p}_b)$. A threshold is then computed according to (3.19) for pose selection. Next, a simple function is implemented to validate whether each pose is within the threshold or not in parallel. According to the validation result, we apply remove_if function in thrust library to remove the threads of other poses. Finally the memory stored the remaining poses $\mathcal{S}_L$ is re-allocated to make it compact.

Set Expansion. In theory, there will be 3 directions to expand ($[-\Delta, 0, \Delta]$) for each pose dimension, which results in 729 (3^6) directions for a pose. If $n_{\mathcal{S}_L}$ remaining poses are obtained in the previous stage, there are $729n_{\mathcal{S}_L}$ poses in the expanded set. The number of expanded pose is too large for our system. Alternatively, we

doi:10.6342/NTU201600474

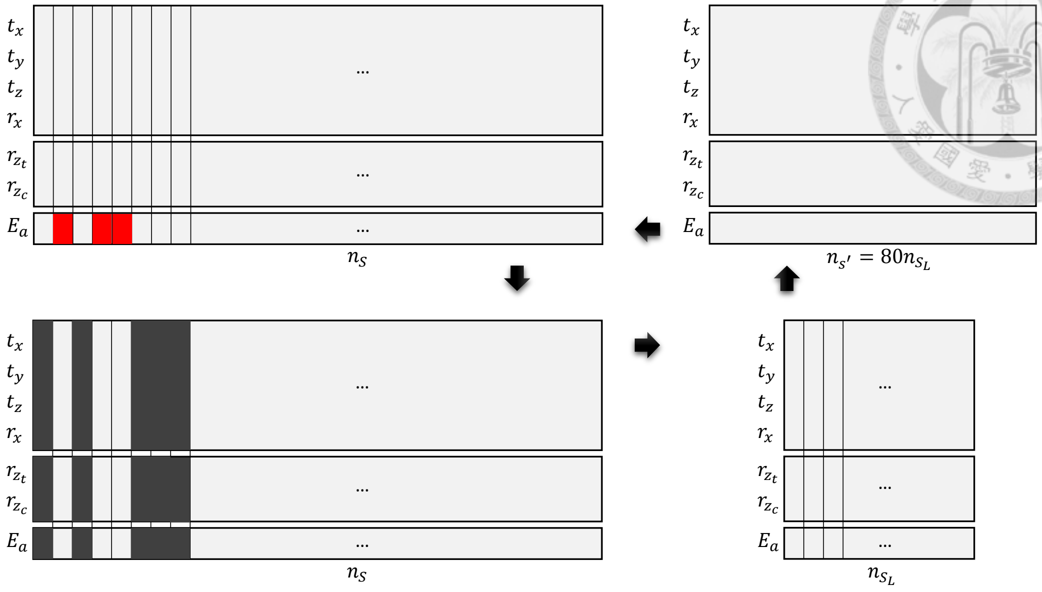


Figure 5.3: The global memory usage for storing poses and appearance distances in the coarse-to-fine estimation. The description below each memory block indicates the number of poses stored in the memory.

randomly generate 80 directions. We make 79 exact copies of the n_{S_L} poses and reallocate the memory. For each of the copied pose, we calculate the expanded pose in parallel. The global memory usage for storing poses and appearance distances in the coarse-to-fine estimation is summarized in Fig 5.3.

5.1.3 Pose Tracker

For consecutively captured frames, pose tracking can be employed instead of pose estimation to accelerate the pose deriving task. We propose to take the pose estimated in the previous frame as the initial pose and perform tracking with the 3-scale search, where a 6D search pattern is employed. In contrast to the checking pattern in the proposed algorithm described in Subsection 3.2.2, which has three search points in each dimension, we assign five points in each of the rotation dimensions and seven points in each of the translation dimensions. Such unequal assignment is a practical consideration since the variation in translation motion is

doi:10.6342/NTU201600474

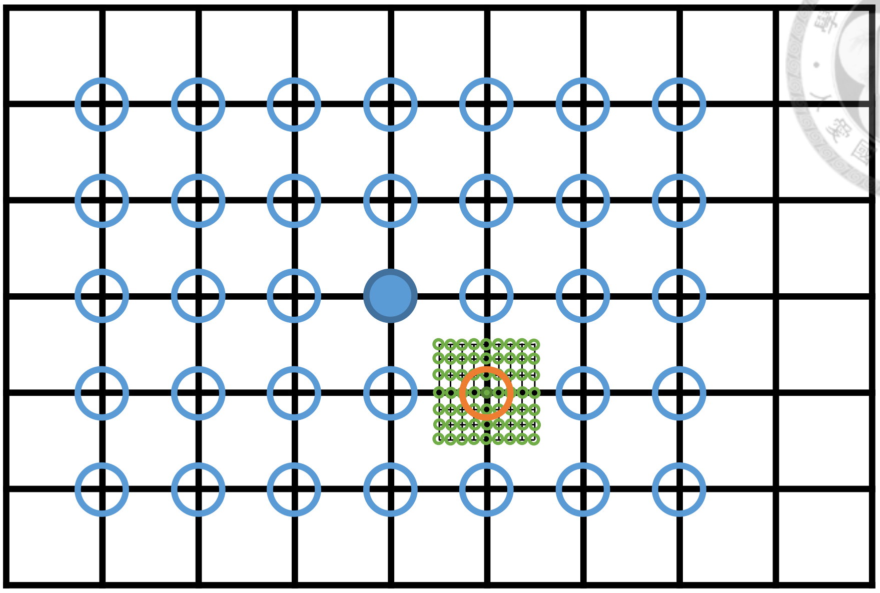


Figure 5.4: 2D view of the proposed 6D pose search pattern for the pose tracker. The search pattern has five and seven search points in each of the rotation and translation dimensions, respectively. After the point with minimum E_a, which is drawn in orange, is found, the size of the pattern is diminished to search in the finer scale. Totally 3 scales are used for the search in the pose tracker.

larger than that in rotation motion according to our statics.

There are two main reasons why we assign more than three points for each dimension. First, the overhead for passing parameters to GPU is hidden if the number of parallel computing units is large. The total number in the proposed search pattern is $5^3 \times 7^3 = 42875$ which is more suitable than that of the three-point pattern, which has $3^6 = 729$ points. The second reason is that the search range is larger for the proposed search pattern, which yields a faster search speed. In summary, deliberating the characteristic of GPU, we design a search pattern which makes the tracking more efficient.

The goal for the pose search pattern is to find the pose with minimum E_a. The process includes the random pixel sampling and the appearance distance calculation, which are described in Section 5.1.2. After the pose with minimum E_a is

doi:10.6342/NTU201600474

obtained, we move the pattern to the position of the pose and diminish the pattern size to perform the search in the finer scale. A brief illustration in 2D view is shown in Figure 5.4. The blue pattern indicates the coarser scale, the green pattern indicate the finer scale and the orange circle is the pose with minimum E_a found in the coarser scale. Considering the efficiency and accuracy, totally 3 scales are used for the search in the pose tracker to track the pose precisely.

In order to make the system invariant to different lighting conditions, we normalize the intensity terms for the appearance distance calculation, as described in Subsection 3.1.2. To do so, we need to read the camera image pixel values twice during the process. The first time is to perform the summation for calculating the intensity normalization factor, while the second time is to calculate the appearance distance with the normalization factor. The processing time for the appearance distance calculation is doubled in this situation. To conquer this problem, we propose a intensity normalization factor tracking scheme based on a observation that the lighting conditions between two nearby video frames are similar most of the time. The pose tracker computes the appearance distance with the normalization factor calculated in the previous frame; meanwhile, the pose tracker also performs the summation for calculating the factor for the next frame. Such scheme makes the tracking lighting-invariant and keeps it efficient since the pose tracker only needs to read pixel values once during the process.

5.2 Evaluation

Since the target applications are augmented reality or robotics, the proposed system is built on an embedded GPU—NVIDIA Jetson TX1 board. We evaluate it with the synthetic dataset described in Section 4 in order to show that the proposed pose estimation unit and the proposed APE algorithm has almost the same performance. As for the proposed pose tracker, we investigate the performance using the real videos in the real dataset. Since the hardware resources are limited

Table 5.1: Average runtime for APE algorithm on 3.4 GHz Core-i7 CPU and the proposed pose estimation unit on GTX770 and TX1 board. The first values in the third row are the runtimes for the pose estimation unit and the second values are the average runtimes for the pose tracker.

	Synthetic dataset	Real dataset
APE	38.35 s	35.13 s
System on desktop	2.65 s	2.26 s / 0.015 s
System on TX1	21.9 s	18.5 s / 0.09 s

on TX1 board, we also build the system on a desktop computer with NVIDIA GTX770 to have a fair runtime comparison between the proposed system and algorithm. The average runtime for each experiment is shown in Table 5.1. The proposed pose estimation unit processing on GTX770 is 10 times faster than APE algorithm running on 3.4 GHz Core-i7 CPU.

5.2.1 Pose Estimation Unit

The evaluation results for the proposed pose estimation unit (PEU) and the proposed approximated pose estimation (APE) algorithm in normal condition are shown in Table 5.2. According to the table, the performances of these two methods are almost the same. Both of them are able to estimate accurate poses of different types of target images.

Figure 5.5 shows the evaluation results under different conditions with five degradation levels. In this figure, PEU performs slightly better than APE since the final precision parameter ε_c^* PEU applies is finer. Combining the results under normal and varying conditions, we verify that our system is able to estimate accurate poses of various types of planar targets in different conditions.

doi:10.6342/NTU201600474

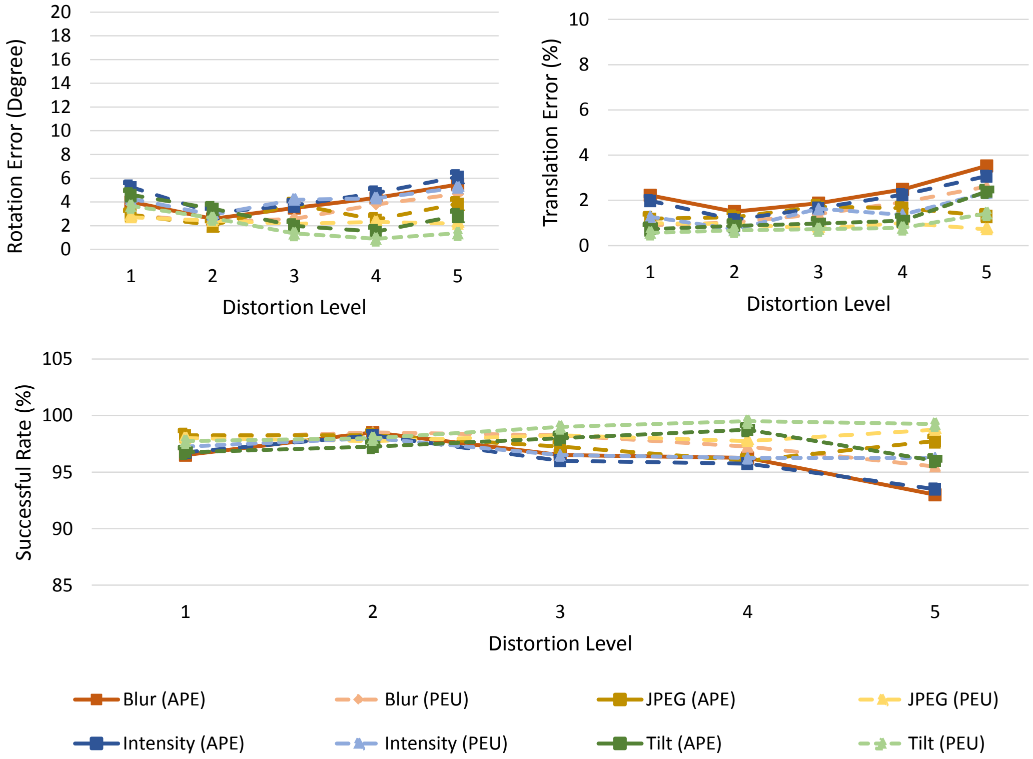
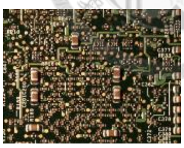


Figure 5.5: Evaluation results for PEU and APE under different conditions with different degradation levels.

doi:10.6342/NTU201600474

Table 5.2: Evaluation results for PEU and APE under undistorted condition.

	Bump Sign 			Stop Sign 			Lucent 			MacMini Board 		
	E_{R}(°)	E_{t}(%)	SR(%)	E_{R}(°)	E_{t}(%)	SR(%)	E_{R}(°)	E_{t}(%)	SR(%)	E_{R}(°)	E_{t}(%)	SR(%)
APE	1.82	0.55	100	**2.40**	1.01	**100**	**3.07**	0.84	96	5.56	2.91	96
PEU	**1.37**	**0.53**	100	3.45	**0.67**	96	3.70	**0.70**	96	**5.53**	**1.04**	96
	Isetta 			Philadelphia 			Grass 			Wall 		
	E_{R}(°)	E_{t}(%)	SR(%)	E_{R}(°)	E_{t}(%)	SR(%)	E_{R}(°)	E_{t}(%)	SR(%)	E_{R}(°)	E_{t}(%)	SR(%)
APE	**1.21**	0.61	100	2.31	0.69	100	2.51	2.16	92	3.14	1.26	96
PEU	1.29	**0.50**	100	**2.18**	**0.66**	100	**1.59**	**1.07**	**98**	**1.50**	**0.67**	**98**

5.2.2 Pose Tracker

We use all "unconstrained" videos in the dataset by Gauglitz *et al.* [1] in this experiment. The videos are extremely challenging for tracking due to significant view point change, drastic illumination change and serious image noise.

For each video, we use the pose estimation unit to get the initial pose. After that, we use the pose tracker to track the poses until it loses the track. We record the pose sequences and compute the error in each video frame. Figure 5.6 shows the results. The pose tracker tracks the poses of targets Building, Mission and Paris for more than 200 frames. As for the targets Sunset and Wood, the pose tracker fails to track since the target images appear a flat color. The pose motion of target Brick in the video is extremely large, which is often out of the tracking range and results in losing the track. However, the pose tracker still tracks the poses over fifty frames which shows the ability to track in severe conditions.

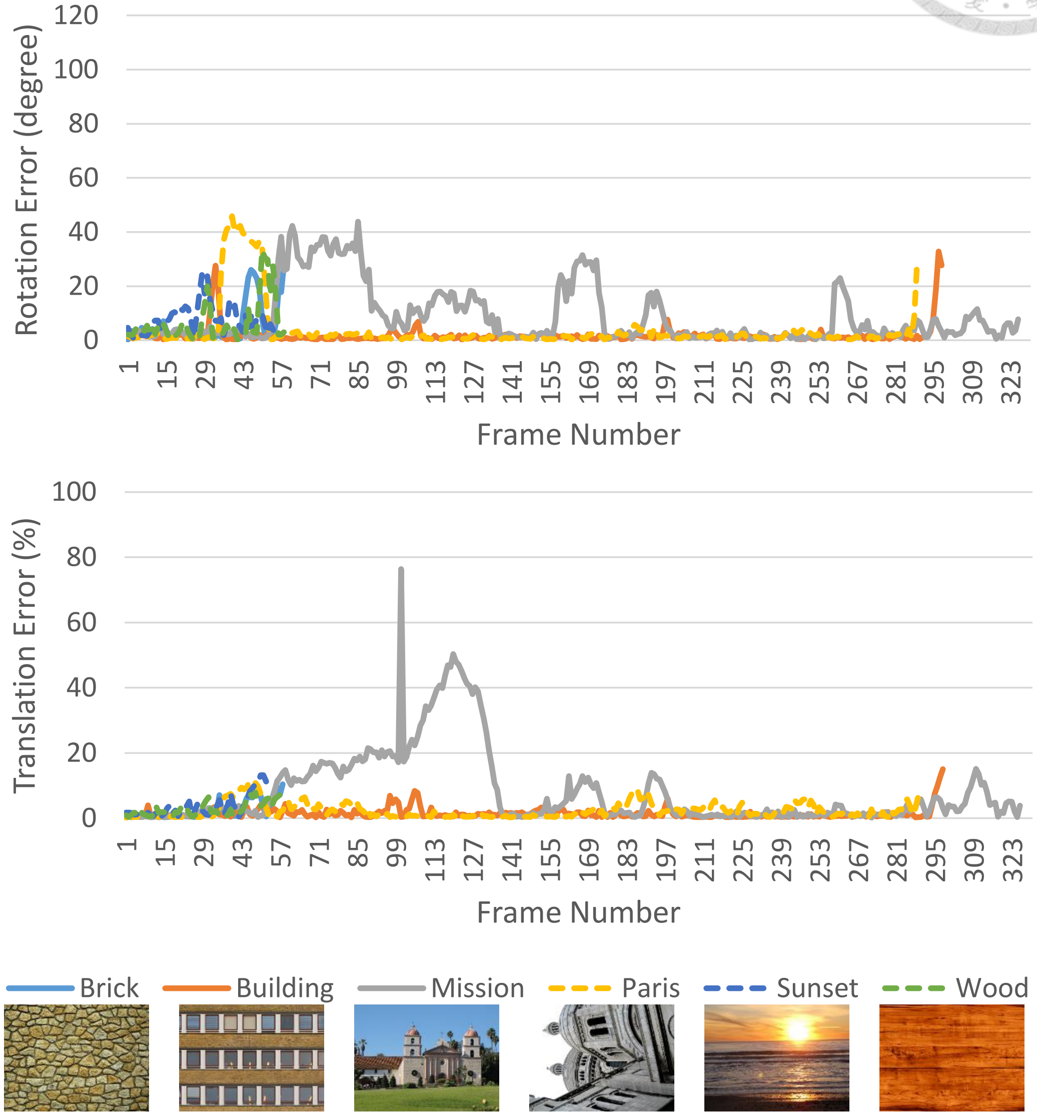


Figure 5.6: Experimental results by the proposed pose tracker. We use pose estimation unit to find the initial pose. After that, we use the pose tracker to track the pose until it loses the track.

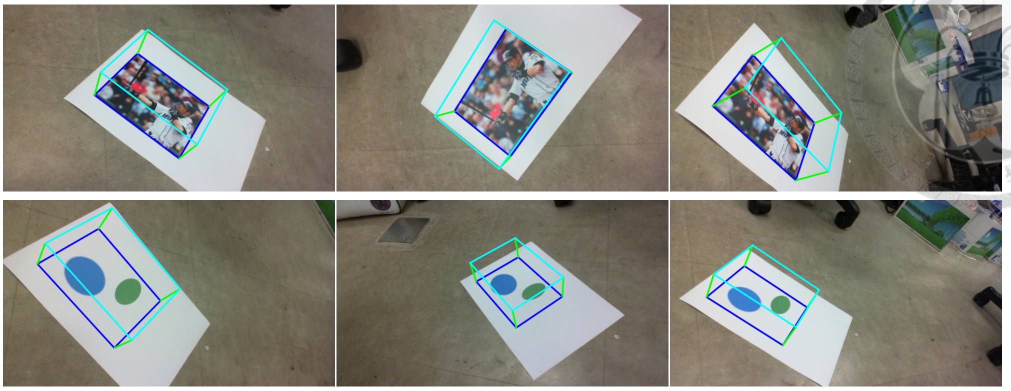
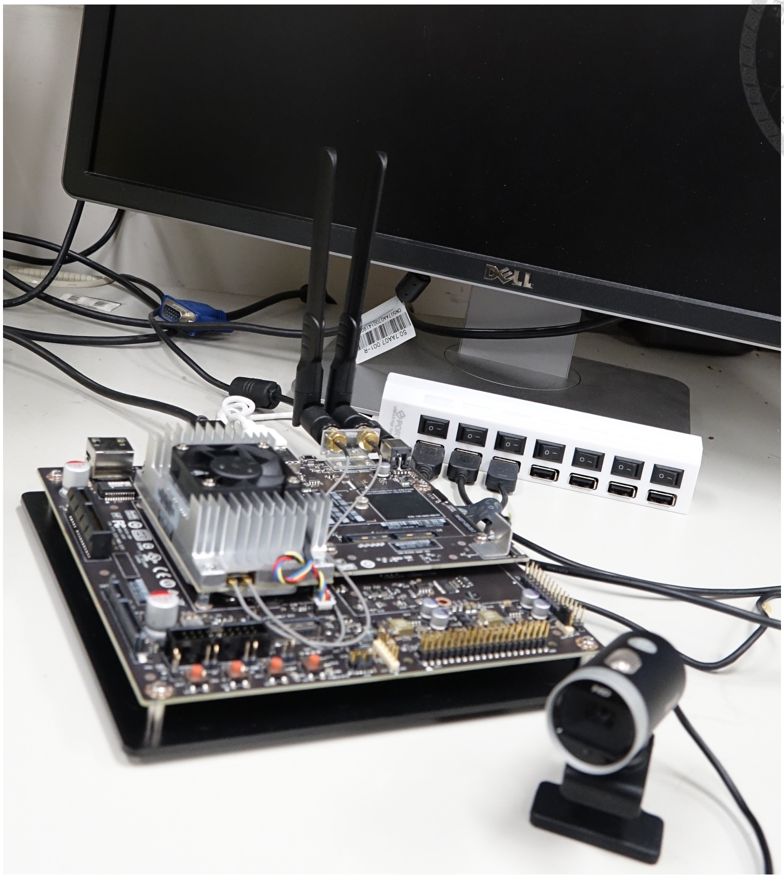


Figure 5.7: Results in the practical tests. Sample images are rendered cyan boxes with poses obtained from the proposed system.

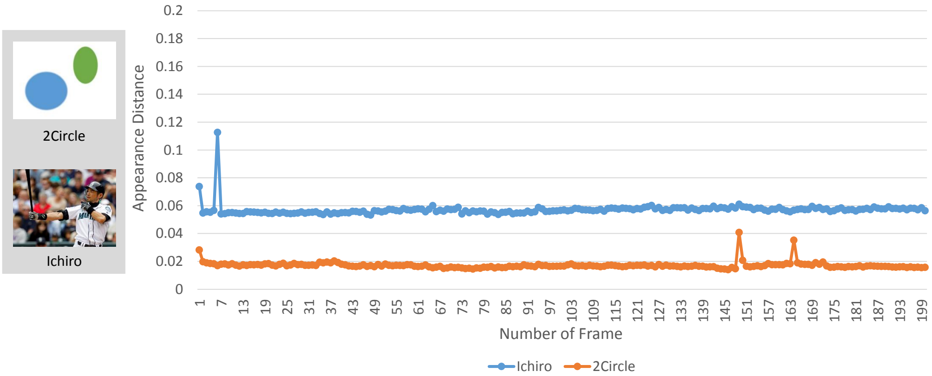
5.3 Practical Tests

In the end, we conduct a practical test to demonstrate the performance of the proposed system in the real world. As shown in Figure 5.8(a), the system is built on NVIDIA Jetson TX1 board with a Microsoft LifeCam Cinema webcam and a DELL display. The resolution of camera images captured by webcam are 640 × 360 and the resolution of the target image is 400 × 300. The area of the planar target in the real world is $16 \times 12\ cm^2$. We use a texture target image and a textureless target image for the test. Due to the lack of ground truth poses, we use appearance distance E_a defined in (2.6) to evaluate the performance. The target images and the results are shown in Figure 5.8(b) and sample images rendered model with poses obtained from the proposed system are shown in Figure 5.7. In this tests, the pose estimation unit spends 10 seconds to obtain the initial pose while the pose tracker achieves 11 fps for tracking. The proposed system is able to give the accurate and robust result for both texture and textureless target in the real world.

doi:10.6342/NTU201600474



(a)



(b)

Figure 5.8: (a) The picture of the proposed system for the practical tests. (b) The results of the practical tests for the proposed system. We use two planar targets in the tests, which are texture target Ichiro and textureless target 2Circle. The pixel values are normalized to $[0, 1]$ for calculating the appearance distance.

doi:10.6342/NTU201600474

doi:10.6342/NTU201600474

Chapter 6

Conclusion

In this thesis, we propose a robust direct 3D pose estimation algorithm and develop D-PET, a direct 3D pose estimation and tracking system for a planar target. The proposed algorithm is a two-step scheme. First, the pose of the target with respect to a calibrated camera is approximated estimated using a coarse-to-fine scheme. Next, we use a gradient descent search method to further refine and disambiguate the pose. Extensive experimental evaluations show that the proposed algorithm performs favorably against two state-of-the-art feature-based method-s in terms of accuracy and robustness. On the other hand, the proposed D-PET system which is implemented on an embedded GPU consists of a pose estimation unit and a pose tracker. The pose estimation unit is built based on the proposed algorithm and is responsible for finding the initial pose. In order to perform pose tracking, the pose tracker applies a 3-scale search with the proposed pose search pattern. Experimental results verify that the proposed pose estimation unit has similar performance compared to the proposed algorithm and the pose tracker are able to track the pose in severe conditions. The proposed D-PET system achieves the processing speed of 11 fps on an embedded GPU in practical. Our future work includes implementing the specific VLSI hardware to make the system available on wearable devices.

doi:10.6342/NTU201600474

doi:10.6342/NTU201600474

Reference

[1] S. Gauglitz, T. Höllerer, and M. Turk, “Evaluation of interest point detectors and feature descriptors for visual tracking,” *International Journal of Computer Vision*, vol. 94, no. 3, pp. 335–360, 2011.

[2] R. T. Azuma, “A survey of augmented reality,” *Presence: Teleoperators and virtual environments*, vol. 6, no. 4, pp. 355–385, 1997.

[3] H. Kato and M. Billinghurst, “Marker tracking and hmd calibration for a video-based augmented reality conferencing system,” in *Proc. IEEE and ACM International Workshop on Augmented Reality*, 1999.

[4] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, “Virtual object manipulation on a table-top ar environment,” in *Proc. IEEE and ACM International Symposium on Augmented Reality*. Ieee, 2000, pp. 111–119.

[5] G. A. Lee, C. Nelles, M. Billinghurst, and G. J. Kim, “Immersive authoring of tangible augmented reality applications,” in *Proc. IEEE International Symposium on Mixed and Augmented Reality*, 2004.

[6] N. Hagbi, O. Bergig, J. El-Sana, and M. Billinghurst, “Shape recognition and pose estimation for mobile augmented reality,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1369–1379, 2011.

[7] M. Donoser, P. Kontschieder, and H. Bischof, “Robust planar target tracking and pose estimation from a single concavity,” in *Proc. IEEE International Symposium on Mixed and Augmented Reality*, 2011.

[8] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, 2004.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Computer Vision and Image Understanding*, vol. 110, no. 3, 2008.

[10] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary Robust Invariant Scalable Keypoints,” in *Proc. IEEE International Conference on Computer Vision*, 2011.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: an efficient alternative to sift or surf,” in *Proc. IEEE International Conference on Computer Vision*, 2011.

[12] A. Alahi, R. Ortiz, and P. Vandergheynst, “Freak: Fast retina keypoint,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[13] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] O. Chum and J. Matas, “Matching with prosac-progressive sample consensus,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[15] T.-J. Chin, P. Purkait, A. Eriksson, and D. Suter, “Efficient globally optimal consensus maximisation with tree search,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

doi:10.6342/NTU201600474

[16] G. Schweighofer and A. Pinz, "Robust Pose Estimation from a Planar Target," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, 2006.

[17] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate O(n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, 2009.

[18] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the PnP Problem: A Fast, General and Optimal Solution," in *Proc. IEEE International Conference on Computer Vision*, 2013.

[19] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 252–286, 2014.

[20] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." vol. 81, 1981, pp. 674–679.

[21] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.

[22] H.-Y. Shum and R. Szeliski, "Construction of panoramic image mosaics with global and local alignment," 2001, pp. 227–268.

[23] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[24] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," 2004.

[25] A. Crivellaro and V. Lepetit, "Robust 3d tracking with descriptor fields," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[26] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Proc. European Conference on Computer Vision*, 2014.

[27] Y.-T. Chi, J. Ho, and M.-H. Yang, "A direct method for estimating planar projective transform," in *Proc. Asian Conference on Computer Vision*, 2011.

[28] S. Korman, D. Reichman, G. Tsur, and S. Avidan, "Fast-match: Fast affine template matching," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[29] J. F. Henriques, P. Martins, R. F. Caseiro, and J. Batista, "Fast training of pose detectors in the fourier domain," in *Proc. Annual Conference on Neural Information Processing Systems*, 2014.

[30] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, "Iterative pose estimation using coplanar points," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1993.

[31] S. Li and C. Xu, "Efficient lookup table based camera pose estimation for augmented reality," vol. 22, no. 1, pp. 47–58, 2011.

[32] P.-C. Wu, Y.-H. Tsai, and S.-Y. Chien, "Stable pose tracking from a planar target with an analytical motion model in real-time applications," 2014.

[33] J. Stork, "Camera pose estimation with circular markers," Ph.D. dissertation, Thesis, University of Amsterdam (UvA), 2012.

[34] E. Olson, "Apriltag: A robust and flexible visual fiducial system," 2011.

[35] D. Wagner and D. Schmalstieg, *Artoolkitplus for pose tracking on mobile devices*, 2007.

[36] J. Rekimoto and Y. Ayatsuka, "Cybercode: designing augmented reality environments with visual tags," in *Proceedings of DARE 2000 on Designing augmented reality environments*. ACM, 2000.

doi:10.6342/NTU201600474

[37] S. Lieberknecht, Q. Stierstorfer, G. Kuschk, D. Ulbricht, M. Langer, and S. Benhimane, "Evolution of a tracking system," in *Handbook of Augmented Reality*. Springer, 2011, pp. 355–377.

[38] D. Schmalstieg and D. Wagner, "Experiences with handheld augmented reality," in *Proc. IEEE International Symposium on Mixed and Augmented Reality*, 2007.

[39] S.-W. Shih and T.-Y. Yu, "On designing an isotropic fiducial mark," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1054–1066, 2003.

[40] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello, "Rune-tag: A high accuracy fiducial marker with strong occlusion resilience," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[41] P. Santos, A. Stork, A. Buaes, and J. Jorge, "Ptrack: introducing a novel iterative geometric pose estimation for a marker-based single camera tracking system," in *Proc. IEEE Virtual Reality*, 2006.

[42] H. Uchiyama and E. Marchand, "Deformable random dot markers," in *Proc. IEEE International Symposium on Mixed and Augmented Reality*, 2011.

[43] G. Yu and J.-M. Morel, "Asift: A new framework for fully affine invariant image comparison," *Image Processing On Line*, 2011.

[44] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, 2000.

[45] S. Li, C. Xu, and M. Xie, "A robust o (n) solution to the perspective-n-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1444–1450, 2012.

[46] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[47] P. Sturm, “Algorithms for plane-based pose estimation,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[48] T. Collins, J.-D. Durou, P. Gurdjos, and A. Bartoli, “Singleview perspective shape-from-texture with focal length estimation: A piecewise affine approach,” in *International Symposium 3D Data Processing, Visualization and Transmission*, 2010.

[49] O. Pele and M. Werman, “Accelerating pattern matching or how much can you slide?” in *Proc. Asian Conference on Computer Vision*, 2007.

[50] B. Alexe, V. Petrescu, and V. Ferrari, “Exploiting spatial overlap to efficiently compute appearance distances between image windows,” in *Proc. Annual Conference on Neural Information Processing Systems*, 2011, pp. 2735–2743.

[51] L.-K. Liu and E. Feig, “A block-based gradient descent search algorithm for block motion estimation in video coding,” *IEEE Transactions on Circuits and Systemsfor Video Technology*, vol. 6, no. 4, pp. 419–422, 1996.

[52] S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 287–290, 2000.

[53] C. Zhu, X. Lin, and L.-P. Chau, “Hexagon-based search pattern for fast block motion estimation,” *IEEE Transactions on Circuits and Systemsfor Video Technology*, vol. 12, no. 5, pp. 349–355, 2002.

[54] D. Eberly, “Euler angle formulas,” *Geometric Tools, LLC, Technical Report*, 2008.

doi:10.6342/NTU201600474

[55] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data*. AMLBook, 2012.

[56] L. Kneip, H. Li, and Y. Seo, "Upnp: An optimal o (n) solution to the absolute pose problem with universal applicability," in *Proc. European Conference on Computer Vision*, 2014.

[57] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab, "A dataset and evaluation methodology for template-based tracking algorithms," in *Proc. IEEE International Symposium on Mixed and Augmented Reality*, 2009.

[58] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. European Conference on Computer Vision*, 2008.

[59] N. Bell and J. Hoberock, "Thrust: a productivity-oriented library for cuda," *GPU Computing Gems: Jade Edition*, 2012.

doi:10.6342/NTU201600474