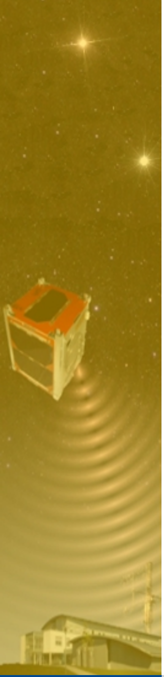


Institut für Informatik
Lehrstuhl für Robotik und Telematik
Prof. Dr. K. Schilling

Würzburger Forschungsberichte
in Robotik und Telematik

Uni Wuerzburg Research Notes
in Robotics and Telematics

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Band 10

Kaipeng Sun

Six Degrees of Freedom Object Pose Estimation with Fusion Data from a Time-of-Flight Camera and a Color Camera

Die Schriftenreihe

wird vom Lehrstuhl für Informatik VII: Robotik und Telematik der Universität Würzburg herausgegeben und präsentiert innovative Forschung aus den Bereichen der Robotik und der Telematik.

Die Kombination fortgeschrittener Informationsverarbeitungsmethoden mit Verfahren der Regelungstechnik eröffnet hier interessante Forschungs- und Anwendungsperspektiven. Es werden dabei folgende interdisziplinäre Aufgabenschwerpunkte bearbeitet:

- Robotik und Mechatronik: Kombination von Informatik, Elektronik, Mechanik, Sensorik, Regelungs- und Steuerungstechnik, um Roboter adaptiv und flexibel ihrer Arbeitsumgebung anzupassen.

- Telematik: Integration von Telekommunikation, Informatik und Steuerungstechnik, um Dienstleistungen an entfernten Standorten zu erbringen.

Anwendungsschwerpunkte sind u.a. mobile Roboter, Tele-Robotik, Raumfahrtsysteme und Medizin-Robotik.

Zitation dieser Publikation

SUN, K. (2014). Six Degrees of Freedom Object Pose Estimation with Fusion Data from a Time-of-Flight Camera and a Color Camera. Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 10. Würzburg: Universität Würzburg. URN: urn:nbn:de:bvb:20-opus-105089

Lehrstuhl Informatik VII
Robotik und Telematik
Am Hubland
D-97074 Wuerzburg

Tel.: +49 (0) 931 - 31 - 86678
Fax: +49 (0) 931 - 31 - 86679

schi@informatik.uni-wuerzburg.de
http://www7.informatik.uni-wuerzburg.de

Universitätsbibliothek Würzburg
Am Hubland
D-97074 Würzburg

Tel.: +49 (0) 931 - 31 - 85906

opus@bibliothek.uni-wuerzburg.de
http://opus.bibliothek.uni-wuerzburg.de

ISSN: 1868-7474 (online)
ISSN: 1868-7466 (print)
ISBN: 978-3-923959-97-6 (online)

Six Degrees of Freedom Object Pose Estimation with Fusion Data from a Time-of-Flight Camera and a Color Camera

Kaipeng Sun

Department of Computer Science VII: Robotics and Telematics

Julius-Maximilians-Universität Würzburg

Dissertation zur Erlangung des naturwissenschaftlichen Doktorgrades der Julius-Maximilians-Universität Würzburg

Würzburg, 2014

Eingereicht am: 06.03.2014
bei der Fakultät für Mathematik und Informatik
1. Gutachter: Prof. Dr. Klaus Schilling
2. Gutachter: Prof. Dr. Hubert Roth
Tag der mündlichen Prüfung: 21.10.2014

To my parents.

致我亲爱的爸爸妈妈.

Acknowledgements

First of all, I would like to thank my doctoral advisor Prof. Klaus Schilling for providing me the challenging topic to work with the PMD camera and the necessary supervisions for pursuing the PhD degree. I would also like to thank Prof. Hubert Roth for his careful review of this thesis and the valuable suggestions. I am also very grateful to Prof. Lei Ma and Prof. Xizhi Shi for many helpful discussions at those critical points.

Among my obliging collaborators, i.e. Daniel Eck, Stephan Busch, Philip Necsulescu, Heidi Frankenberger, Dieter Ziegler, Lakshminarasimhan Srinivasan, Dr. Christian Herrmann, Tristan Tzschichholz, Leonardo Regoli, Dr. Marco Schmidt, Florian Leutert, Zhongyang Wu, Pablo Baier, László Lemmer, Markus Sauer, etc., I owe special thanks to Robin Heß, Zhihao Xu and Martin Hess, who spent a lot of time working with me and helped me with a lot of scientific and practical problems.

I would like to express my gratitude to Lan Han, Dr. Muzheng Xiao, Hongmei Wu, Dr. Gang Li, Dr. Haixiang Zhao, Jun Ding and Dr. Yu Zhu for always being with me, encouraging me going on even during the hardest times. In particular, I want to dedicate this work to my parents. They are and always will be the ultimate power for me to overcome all the difficulties.

Abstract

Object six Degrees of Freedom (6DOF) pose estimation is a fundamental problem in many practical robotic applications, where the target or an obstacle with a simple or complex shape can move fast in cluttered environments. In this thesis, a 6DOF pose estimation algorithm is developed based on the fused data from a time-of-flight camera and a color camera. The algorithm is divided into two stages, an annealed particle filter based coarse pose estimation stage and a gradient decent based accurate pose optimization stage. In the first stage, each particle is evaluated with sparse representation. In this stage, the large inter-frame motion of the target can be well handled. In the second stage, the range data based conventional Iterative Closest Point is extended by incorporating the target appearance information and used for calculating the accurate pose by refining the coarse estimate from the first stage. For dealing with significant illumination variations during the tracking, spherical harmonic illumination modeling is investigated and integrated into both stages. The robustness and accuracy of the proposed algorithm are demonstrated through experiments on various objects in both indoor and outdoor environments. Moreover, real-time performance can be achieved with graphics processing unit acceleration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter discusses the object position and orientation estimation problem. This thesis concentrates on rigid objects, for which the position and orientation are often termed as the six Degrees of Freedom (6DOF) pose. An overview to the approaches to be presented in this thesis will be given, and an overall algorithm workflow will be described, which also leads to a brief introduction to the major chapters in this thesis. Then the sensors used for estimating the object pose are introduced with focus on the time-of-flight (TOF) camera for range data acquisition and the fusion methods for combining color information into the range measurements of the TOF camera. In the end, the structure of this monograph is explained.

1.1 Algorithm Overview

This section provides a general overview for the pose estimation algorithm, including the clarification of the problems to be solved, the descriptions for the major contributions of this work, and the introduction for the overall workflow of the proposed algorithm. The detailed information for each aspects of the individual algorithm components can be found in later chapters.

1.1.1 Problem Statement and Innovations

Robust object pose estimation is a fundamental problem for many robotic applications. For instance, the information of the relative pose between two robots is essential in mobile robot formation driving. In-plant automatic transportation requires the object pose information for grasping during robot maneuver. The relative pose of the robot to the scene in the SLAM (Simultaneous Localization And Mapping) can also be interpreted as a variant of the object pose estimation problem. In the outer space, the accurate pose estimation provides important information for spacecraft rendezvous and docking even for a cooperative satellite, just to name a few.

In real applications, the target object usually moves in a cluttered environment, sometimes with large motion between sensor frames. There can be signification illumination variations, which may cause remarkable appearance changes. Often real-time performance is of great importance as well. This composes the problems to be solved in this thesis. In short, this work will develop an efficient algorithm that can robustly estimate the 6DOF pose for a rigid object with high accuracy. A TOF camera and a commodity color camera are used for grabbing the object information. The method for fusing the two sensors will be introduced in Subsection 1.2.2.

The robustness in the above discussion refers to less restrictions. For example, the target is only required to have appropriate size with good surface reflectivity so that the TOF sensor can produce reliable measurements. Unlike purely range data based pose estimation methods, the proposed algorithm can also tackle targets with geometrically symmetric shapes, e.g. cylindrical or planar objects. The target is allowed to move fast in a cluttered background. The motion blur in the measurements of both color and TOF cameras can be well handled. The illumination variations which can significantly change the target appearance are also explicitly taken into account. The high accuracy means, despite of those harsh conditions, the estimated pose can always exhibit high quality and is competent for use in many robotic applications. The highly dynamic nature of a lot of robotic application scenarios requires a quick response to the changes in the scene, which indicates the estimation should be performed with real-time efficiency.

These three criteria for the tracking or pose estimation, i.e. robustness, accuracy and efficiency, may in fact contradict to each other. A method for track a fast moving object usually needs to search in a large area of the state space, which will normally reduce the ability to determine the detailed target position, especially for the high dimensional problems, e.g. 6DOF. Being able to estimate the target pose both robustly and accurately is expected to raise a lot of computation, which will decrease the processing frame rate the method can achieve.

In this work, a two-stage algorithm is proposed. The first stage is denoted as the coarse pose estimation stage, which aims at dealing with the large inter-frame motions. This stage adopts the particle based search scheme and can be used to explore a large volume of the state space. The resulting coarse pose estimate will be further refined by a second stage, the accurate pose estimation stage. This stage employs a gradient based optimization method and can achieve the optimal pose in an iterative manner. Under such a configuration, both the robustness and the accuracy requirements can be fulfilled. For the computational efficiency, the parallelism in both stages are investigated. Thanks to the development of modern many core Graphics Processing Units (GPUs), under a careful implementation, the real-time performance can be achieved with GPU acceleration.

Besides the proposed two-stage framework for the pose estimation and the GPU implementation, other major innovations includes:

- In the coarse stage, Sparse Representation (SR) is used for evaluating each particle in Annealed Particle Filter (APF). A new composition and update rules for the template matrix in SR is proposed, which yields a particle filter with high distinctive power.The multiresolution strategy is adopted to further harness the distinctive power. The coarse estimation will be detailed in Chapter 2.

- In the accurate stage, the conventional Iterative Closest Point (ICP) is extended to incorporate the target texture information into the cost function. The combination of range and texture data brings about several advantages: the ability for dealing with symmetric geometries; better convergence; higher tolerance to the noisy range measurements. Chapter 3 will give further information for the proposed Textured-ICP.

- Target appearance changes caused by illumination variations is handled by incorporating Spherical Harmonic (SH) illumination model into both the coarse and the accurate estimation stages. As shown by the experiments, the proposed algorithm can work robustly even under severe lighting changes in a video sequence. The use of SH modeling will be introduced in Chapter 4.

1.1.2 Algorithm Workflow

The overall workflow for the propose algorithm is depicted in Fig. 1.1. At the very beginning, the target to track is selected manually and initialized for use in different algorithm components. The target selection can be done by either drawing a 2D box containing the target points on the image, or masking out the pixels that belong to the target. The target information grabbed in the initialization module includes the Cartesian coordinates, the surface normals, the appearance values (i.e. intensities in the image), and the surface reflectances (see Chapter 4) for all target points. The Cartesian coordinate system attached to the color camera (with the principle point located at the focus of the color camera lens, the X and Y axes go along horizontal and vertical axes respectively on image plane, the Z axis approximately goes along the optical axis of the color camera) as the primary coordinate system used in the algorithm. This choice is made out of two major considerations: firstly, both the coarse and the accurate stages need the 2D image for the texture information; secondly, the color image has higher resolution and better quality compared to current TOF cameras. Therefore, working primarily in the color camera's coordinate system is more convenient as can be observed in Chapter 2 and 3. The initial target pose uses the mass center of all selected points as the translation component and zero Euler angle for the rotation. Regarding the choice of the state space, it is referred to Subsection 2.4.1 for more details.

The initialized target model with S surface points can be expressed as

$$\mathcal{M} \equiv \{(\mathbf{u}_1^{init}, v_1, \mathbf{N}_1^{init}), \cdots, (\mathbf{u}_\mathrm{S}^{init}, v_\mathrm{S}, \mathbf{N}_\mathrm{S}^{init})\}, \tag{1.1}$$

where $\mathbf{u}_s^{init}$, $\mathbf{N}_s^{init}$ are the s-th initial 3D coordinate and surface normal in the color camera's coordinate system, and v_s is the corresponding intensity value.

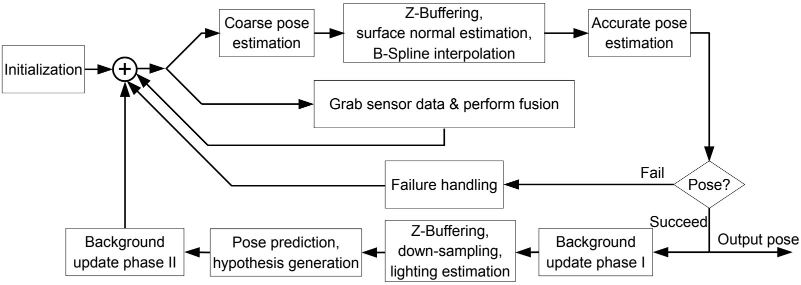

Initialization
+
Coarse pose estimation
Z-Buffering, surface normal estimation, B-Spline interpolation
Accurate pose estimation
Grab sensor data & perform fusion
Fail
Pose?
Failure handling
Succeed
Output pose
Background update phase II
Pose prediction, hypothesis generation
Z-Buffering, down-sampling, lighting estimation
Background update phase I

Figure 1.1: Workflow for the proposed pose estimation algorithm.

After the target is initialized, a new frame will be grabbed from the fused cameras, then it will come to the coarse pose estimation stage. In a nutshell, the coarse pose is obtained through comparing the target model data (3D colored point cloud) with the 2D observation image. Chapter 2 will detail this stage. After a coarse pose is estimated, all the 3D model points will be transformed with the coarse pose and then projected onto the image plane to check their visibility with z-buffering. A 2D upright bounding rectangle is obtained for all projected points, where the target is expected to reside. To count in the small variation between the coarse and the accurate poses, the bounding box is enlarged a bit, and the surface normal estimation and the B-Spline coefficient calculation will be carried out for all the points in this rectangle on the observation image. Then the accurate pose estimation will take place with the Textured-ICP proposed in Chapter 3.

If an accurate pose can be successfully determined, the model data will be under a series of updates to accommodate for the changes happened during tracking, e.g. the illumination variations, the background changes and the visibility changes of the target surface points. Since one of the major tasks in the coarse estimation stage is to distinguish the target region from the background, two background update phases are implemented for modeling and adapting to the background changes. The Background update phase I (Algorithm 2.3) incorporates some grabbed background templates for current down-sampled points into the template matrix in the sparse representation. This step takes into account those background regions that cannot be well handled by current model. Then after z-buffering to get visible target points under the predicted pose (the estimated pose for the current frame is used as the predict pose for next frame) and down-sampling to get the points used for the coarse estimation in the next frame, the Background update phase II (Algorithm 2.4) is employed to model the background with the new down-sampled points. Details with respect to the background update and the down-sampling of the target points can be found in Chapter 2. Alongside z-buffering, if the target reflectance has been estimated in the initialization module, the current light condition can be estimated with spherical harmonic modeling. The reflectance and the lighting estimation are detailed in Chapter 4. The predicted pose, the updated model and the grabbed new frame will be fed into the coarse estimation stage for the next frame. To make most of the current multi-core CPUs, the new frame grabbing and pose estimation algorithm are running in parallel.

If pose estimation has failed (either weights are too low in the coarse stage or the matched point pairs are too less in the accurate estimation stage), no updates will be performed. Instead, some failure handling can be considered. In this work, the error handling is simplified to keep searching around the last estimated target location.

1.2 Sensor Setup

This section introduces the sensors used for the pose estimation in this thesis, including a TOF camera and a color camera. The working principle of TOF cameras is described, the major error sources and the methods for the measurement enhancement are discussed, which is followed by a short review for some applications with TOF cameras. Then the method adopted in this thesis for fusing a TOF camera and a commodity color camera is introduced in detail, which includes the relative transformation between the two cameras, the derivation for converting the radial range measurement of the TOF camera to Cartesian coordinate that can be directly used in fusion, some pre-calculation revealed in this work that can reduce the computation cost during run time and the z-buffering technique for removing the pixels visible to one sensor but hidden for another. Since the TOF camera used in this thesis is based on PMD (Photonic Mixture Device) sensor, where no confusion will arise, the term TOF camera and PMD camera are used interchangeably, likewise for the RGB camera and the color camera.

1.2.1 TOF Sensor and Its Applications

Time-of-flight (TOF) sensor provides depth data for each pixel on its image. This is consistent to the 2D cameras, therefore, it is often denoted as TOF camera. This subsection gives a comprehensive introduction to TOF cameras, including some products appeared in the literature, the TOF principle, the major error sources and the methods for enhancing the quality of the range measurements, as well as some applications with current TOF cameras (please refer to [78] for more applications).

Introduction to TOF Sensor and TOF Principle

TOF sensor is a relatively new technology compared to the traditional CCD color camera. It is still in maturating stage yet having already drawn substantial attention in many fields. In a nutshell, it provides depth information for the scene by measuring the time-of-flight of the light emitted from an active light source in addition to the amplitude information. To be less influenced by the environmental visible lighting, it usually adopts light, modulated with an internal reference signal, in the near infrared spectrum emitted by the LED arrays mounted on the sensor. In this way, it can achieve resistance or can Suppress Background Illuminations (SBI), and thus has less restrictions on the application scenarios.[1]

Some TOF cameras produced by different manufactures are listed in Table 1.1. Current TOF cameras bear the advantages including reliable 3D measurement at high frame rate and low power consumption with a compact design, which make it an ideal sensor for robotic applications. In this paper, most of the works are done

[1] One exception is the scenario under strong sunshine, where the chip can be dramatically oversaturated and most of the range measurements will be invalid.

with a CamCube2.0, which has a lateral resolution 204×204 at the frame rate 25 Hz.

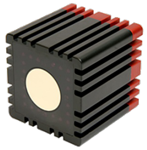
	 SR4000	FOTONIC C70	CamCube3.0
Manufacturer	MESA Imaging AG	FOTONIC	PMDTec GmbH
Lateral resolution	176 × 144	160 × 120	200 × 200
Depth range	0.1 to 10.0 m	0.1 to 7.0 m	0.3 to 7.0 m
Repeatability	(typ.) 4 mm	± 5 mm at 0.1 –1.5 m	<3 mm
Frame rate [fps]	50	75	40
Input voltage	12 V (-2%; +10%)	12 –24 V	12 V ±10%
Power	(typ.) 9.6 W	max 15 W	—
SBI	Yes	Yes	Yes

Table 1.1: Some commercial TOF cameras.

Due to the extremely high speed of light, it is impractical to directly measure the flight time of the light for a near range (e.g. within 10 m). Therefore, current TOF sensors adopt another strategy of measuring the phase difference between the received modulated signal and the internal reference signal. The working principle of the readout circuit [109] on PMD-based TOF sensor can be interpreted as sampling the autocorrelation function between the received signal and the reference signal. For calculating the desired phase difference, four samples - each shifted by $\pi/2$ - are used. When sinusoidal signals are assumed, this process can be illustrated in Fig. 1.2, where Fig. 1.2 (a) shows the four samples (A_i at $(i-1)\pi/2$) on the autocorrelation function, and Fig. 1.2 (b) shows the corresponding schematic circular form for the samples. Since A_1A_3 is perpendicular to A_2A_4, the span $A_3B_{13} = A_4B_{24}$, the calculation for the phase difference φ is then given as:

$$\varphi = arctan\frac{A_1B_{13}}{A_3B_{13}} = arctan\frac{A_1B_{13}}{A_4B_{24}} = arctan\frac{A_{1y} - A_{3y}}{A_{2y} - A_{4y}}, \tag{1.2}$$

where A_{iy} is the y value for point A_i in Fig. 1.2 (b).[1]

[1] Because of the influence from background illumination, the real sample values on the autocorrelation function will be attached with a constant bias [93] instead of only A_{iy} shown in Fig. 1.2 (b). Therefore, φ cannot be simply calculated from two samples A_1 and A_2 as $\varphi = arctan(A_{1y}/A_{2y})$

Correspondingly the distance can be obtained from [109]:

$$d = \frac{c\varphi}{4\pi f_{mod}},$$

where $c \approx 3{\cdot}10^8$ m/s represents the light speed, f_{mod} is the modulation frequency. The phase difference φ is in range $[0, 2\pi)$. For a commonly adopted frequency $f_{mod} = 20$ MHz, the unambiguous distance measurement will be within 7.5 m.[1]

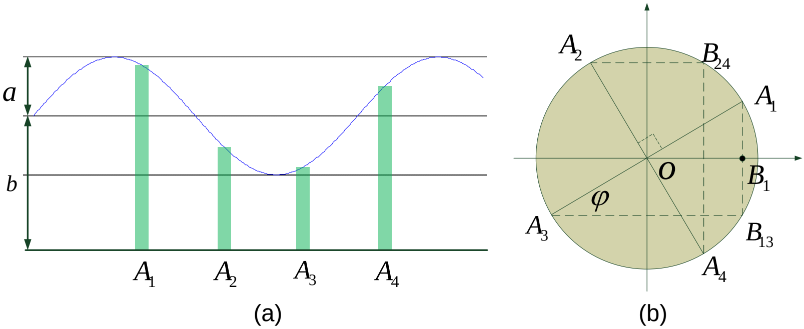


Figure 1.2: Autocorrelation function and phase offset for TOF measurement. (a) shows the sinusoidal form of the autocorrelation function and the four samples used for calculating the phase difference. (b) is the schematic circular demonstration for the phase calculation.

Along with a range measurement, the amplitude value of the received signal, which can be taken as a fidelity measure for the range data, is calculated as:

$$a = \frac{\sqrt{(A_{1y} - A_{3y})^2 + (A_{2y} - A_{4y})^2}}{2} = \frac{\sqrt{(A_1B_{13})^2 + (A_4B_{24})^2}}{2}.$$

And the intensity value, which contains the constant bias from background illumination, is obtained by:

$$b = \frac{A_{1y} + A_{2y} + A_{3y} + A_{4y}}{4}.$$

Error Sources and Enhancement for the TOF Sensor

[96] analyzed the major sources for the range measurement error. For example, the violation to the assumption of the sinusoidal form of the reference signal adopted

[1] If the TOF camera can be configured to use multiple modulation frequencies, the unambiguous range limit can be extended with the method presented in [154].

on current TOF cameras raises the systematic wiggling error. Based on the observation for distance related errors, [91] proposed to fit the error with a B-Spline curve, which worked effectively as a dense look-up table. [94], in comparison, empirically assumed a box-shaped reference signal instead of sinusoidal. Besides, the integration time plays an important role on data acquisition of a TOF camera [166]. An inappropriate setting can result in significant errors. A low received signal intensity value implies high signal-to-noise ratio and a high intensity can cause oversaturation for near objects. Such errors can usually be determined by checking the amplitude values [57]. Pixels with low amplitudes are recommended to be discarded or interpolated with its surrounding measurements.

Another important error source comes from intensity related errors, which often occurs on surfaces with too high or too low reflectivity. For instance, dark surfaces will make the measurement remarkably drifting towards the camera, which can clearly be visible if the TOF camera is observing a black-and-white paper chessboard pattern, where the range data for the black blocks can be 4 cm closer than the white blocks. On the other hand, a high reflectivity can cause oversaturation. For the intensity-related errors, [92] proposed to incorporate the intensity data into the aforementioned B-Spline fitting for calibrating the errors. Such a calibration process can be a non-trivial task, because a large set of measurements on objects with varying intensity at different distances as well as the ground truth range data are required. Furthermore, since individual sensor may have different characteristics, the calibration needs to be carried out for each sensor, and maybe also under different integration time settings. Making use of the connection between the scene geometry (underlying range data) and intensity image, [17] presented a framework to model the posterior probabilistic distribution of the range data with a synthetic intensity image. However, the use of the maximum likelihood for estimating the true range data was presumably time consuming and only applicable on offline scene reconstruction. Besides improving the range measurement, the quality of the intensity data can be standardized by compensating the influence caused by the integration time setting and the distance of the scene [145]. [132] presented a hardware level method to tackle the reflective surfaces. They placed infra-red emitters at various places and the improved measurement was obtained by combining multiple images captured by using one emitter at one time. The disadvantage of their approach is the loss of the compactness of the TOF camera and is incompetent for highly dynamic scenarios.

PMD-based TOF cameras take four phase images to produce a range image. If the relative motion occurs during the integration period, the phase images will be captured from the same object but at different distances. Calculating the range data with these inconsistent phase images will cause motion artefacts. Since the motion artefacts are most severe on geometrical edges of an object, [98] argued that the edge pixels detected in the fused 2D color image can be used for determining potential erroneous range measurements. That is, if an edge pixel is considered as an artefact, it can be replaced by the weighted sum from the range measurements in its neighborhood. However, this approach is more like an edge preserving filtering. [93] proposed a motion compensation algorithm for PMD-based TOF camera. They aligned the four phase images for a dynamic scene with the optical flow. In this way, the distance of a real world 3D point is calculated with the true phase values.

Current TOF cameras still suffer from lack of color data and limited lateral resolution, e.g. 200×200 for CamCube3.0. Many approaches were presented to improve the lateral resolution and refine the range measurement with the a fused high resolution color image. The fusion can be done with the method proposed in [89], which will also be detailed in Subsection 1.2.2. The aligned depth and color images can be modeled into Markov Random Fields (MRF) [40], with a depth measurement potential and a depth smoothness prior weighted by the color consistency in the neighborhood. The refinement of the range image with MRF can be optimized through conjugate gradient. To avoid over smoothing on edges, [60] extended the MRF approach by incorporating a depth discontinuity term and the object contour information into depth measurement potential and depth smoothness prior respectively, so that on object contour or strong range discontinuities, the related constraints can be switched off. Bilateral filter [152] is another widely adopted method for range image refinement. [176] proposed to apply bilateral filter calculated with the color image on each slice of the cost volume estimated with the range image for smoothing the cost volume with preserved edges. However, the color consistency does not always correlates with depth continuities. In comparison, [25] proposed to use joint bilateral upsampling. In addition to the intensity smoothness from the color image, they also integrated the smoothness from the intensity image of the TOF camera to guarantee edges being correctly determined. By GPU acceleration, their method was reported to be capable of real-time applications. Besides fusion with a single color camera, [8] presented an approach to refine the TOF depth image with the stereo matching of multiple color cameras. The stereo color image pairs, the pre-aligned depth data from TOF camera and the smoothness in the neighborhood were used to calculate the cost function and optimize the final depth map.

Applications with TOF Sensors

The high quality depth information, high frame rate data acquisition, low power consumption and compact design make TOF camera an ideal sensor for a lot of applications, ranging from 3D map building [127], industrial automation, home entertainment, etc. [117] compared the performance of the PMD camera with the stereo imaging on mobile robot self-motion estimation. The robust depth measurement of the PMD camera enables superior estimation in translation than stereo imaging. However, they also pointed out even better performance can be achieved if the two sensors were combined, because the limited lateral resolution of current TOF camera can be compensated with the help of the high resolution stereo images.

A straight forward application with the range image is foreground segmentation. [34] fused a TOF camera with a high resolution color camera to generate the trimap (mask image for foreground, background and under-determined pixels) of the scene. Then the trimap was further processed with a bilateral filter to yield the alpha-matte (the transparency map for foreground pixels). Instead of bilateral filter, [183] fused the stereo and the TOF sensors and used MRF for optimizing the alpha-matte. The output alpha-matte can be applied in film production or at-home interactive gaming.

The 3D range image also provides a new possibility for object tracking and pose estimation. [77] presented a method for estimating the pose of an articulated human body with fused TOF and color cameras. ICP was extended by incorporating 2D matched features into the correspondence pairs. They tested a human body model with 10 cylinders and 9 joints and achieved robust estimate under frame rate of 20∼25 Hz. Outer space is another potential field for employment of TOF sensors. Motivated by SIFT [99] and SURF [10] features, [155] developed a new feature descriptor which was applicable with the data from current TOF cameras. They showed the effectiveness and the efficiency of such a monocular approach on a test cube covered with solar panels. For a satellite with planar surfaces, [134] extracted target contours from the range image for pose estimation. Such an edge detection based method has very low demand on computing resources, which can be a critical factor in spacecraft applications, but it is not applicable on targets with more complex shapes.

Object shape reconstruction can also be performed with the range measurement from the TOF camera. Typically, 3D reconstruction requires high quality range data as input. The low resolution, the system bias and the high noise level of current TOF sensor may deteriorate its use on these applications at the first glance. The work of [35], however, showed that by integrating a number of subsequent depth images to form a superresolution depth map with motion parameters between scans found by optical flow, it was realistic to achieve satisfying reconstruction quality under a probabilistic framework, despite of the low quality of individual raw inputs. But their approach should be classified as a post-processing method due to its high computational cost.

[76] presented another interesting application with the TOF camera on automatic phenotype evaluation for outdoor plant. Measurements of the plant characteristics, like the size of the plant, the number of leaves and the leaf orientations, have great significance on optimizing fertilizer or water supply and controlling plant growth process. These data are usually collected manually by experts, which is time and cost consuming. [76] evaluated the TOF camera in plant supervision scenario. Their concluded, thanks to its ability on suppressing the background illumination, TOF camera is capable of collecting plant information in the outdoor environment. And due to the potential low cost if TOF camera is manufactured in mass production, it can be a promising sensor for field applications.

1.2.2 Fusion of TOF and Color Cameras

This section introduces the details for fusing data of a TOF range camera and a CCD color camera. This includes deriving the relative transformation between both cameras, converting a point measured by the TOF camera from pixel and radial range to Cartesian coordinates, and some pre-calculations for computational consideration. Since the cameras used in fusion observe the scene from different perspectives, it is inevitable that some points perceived by one camera are not visible to another. The technique for removing the hidden surfaces is also addressed. In the end, some trade-offs between the quality of the fused data and the computational costs are discussed.

The fused sensor setup is illustrated in Fig. 1.3. Besides CamCube2.0, for some experiments in this work, PMD 19K was also used as the TOF range camera.

Relative Transformation between Two Cameras

The TOF camera provides a radial range measurement for each of its pixels, for which the corresponding Cartesian coordinates can be obtained. When the relative transformation between the TOF camera and the RGB camera is available, the 3D points in TOF's Cartesian coordinates can be transformed to the color camera's coordinates. Such a transformation can be used to assign the range information to a pixel in the color camera, or to grab the RGB data for a pixel in the TOF image. Either way, a RGBD image is obtained, where "D" represents the depth.

For a mechanically fixed camera fusion setup as shown in Fig. 1.3, the relative transformation only needs to be determined once in a calibration preprocessing step

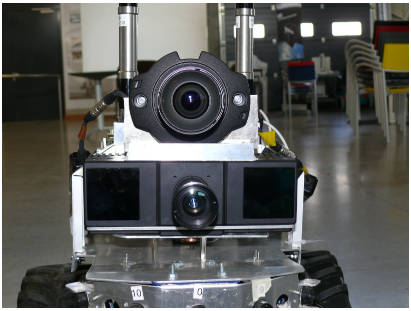


Figure 1.3: Setup of the fused cameras. An AXIS color camera is mechanically mounted on the top of a CamCube2.0. The data from both cameras are fused with the fusion algorithm.

and will remain unchanged as long as the configurations for both cameras (e.g. the zoom-in factor) are kept unchanged and the mechanical frame does not deform much.

When using a pin-hole camera model, the relative transformation can be derived by a purely geometric manipulation with the help of a calibration pattern with known physical size. A chessboard pattern was used in this work due to its convenience in the calculation [90]. The schematic view for calculating the relative transformation is depicted in Fig. 1.4, where the 3×4 matrix $\mathbf{T}_{A,B}$ transforms a 3D point from B's coordinate system to A's. For example, $\mathbf{T}_{PMD,ch}$ transforms a 3D point in the chessboard coordinates to the PMD camera's coordinate system.

Suppose $\mathbf{P}_{ch} = [P_x, P_y, 0]^\top$ is a point on the chessboard expressed in chessboard's 3D Cartesian coordinates, and $\dot{\mathbf{P}}_{ch}$ is the corresponding homogeneous coordinate. Denoting $\dot{\mathbf{P}}_P$ and $\dot{\mathbf{P}}_R$ as the corresponding point of $\dot{\mathbf{P}}_{ch}$ in PMD's and RGB's coordinates respectively, the relation

$$\begin{cases} \mathbf{P}_P = \mathbf{T}_{PMD,ch} \dot{\mathbf{P}}_{ch} \\ \mathbf{P}_R = \mathbf{T}_{RGB,ch} \dot{\mathbf{P}}_{ch} \end{cases}$$

can be converted to

$$\mathbf{P}_R = \mathbf{R}_R \mathbf{R}_P^\top \mathbf{P}_P - \mathbf{R}_R \mathbf{R}_P^\top \mathbf{t}_P + \mathbf{t}_R,$$

where the transformation matrix is decomposed to a rotation matrix and a translation vector as $\mathbf{T} = [\mathbf{R}, \mathbf{t}]$. The transformation $\mathbf{T}_{PMD,ch} = [\mathbf{R}_P, \mathbf{t}_P]$ and $\mathbf{T}_{RGB,ch} = [\mathbf{R}_R, \mathbf{t}_R]$ from the chessboard coordinate to the PMD and RGB coordinates can easily be obtained with the routine *cvFindExtrinsicCameraParams2*() from OpenCV library

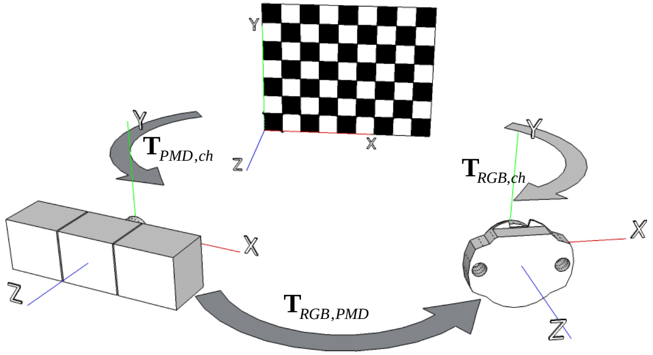


Figure 1.4: Calculating the relative transformation between two cameras. The chessboard is used to determine the relative transformation matrix between two cameras. The cameras are placed apart instead of being stacked for convenient illustration.

[18]. Above equation yields the relative transformation matrix

$$\mathbf{T}_{RGB,PMD} = [\mathbf{R}_R\mathbf{R}_P^\top, \mathbf{t}_R - \mathbf{R}_R\mathbf{R}_P^\top\mathbf{t}_P] = [\mathbf{R}_{RP}, \mathbf{t}_{RP}] \tag{1.3}$$

with $\mathbf{R}_{RP} = \mathbf{R}_R\mathbf{R}_P^\top$ and $\mathbf{t}_{RP} = \mathbf{t}_R - \mathbf{R}_R\mathbf{R}_P^\top\mathbf{t}_P$ representing the rotation matrix and the translation vector for transforming a point in the PMD camera's coordinates to the RGB camera's coordinates.

[98] proposed another strategy for camera fusion. They placed a beam slitter behind the lens, by which the reflected light from the scene was then split up into two parts. The light in the visible spectrum was forwarded to a color sensor and the light in the infra-red spectrum was fed into the TOF sensor. Since the same beam of light was used, the two images were inherently aligned. The drawback of this scheme is the requirement for a dedicated hardware with a careful mechanical calibration.

Convert the Radial Range Measurement to Z in Cartesian

The direct range measurements provided by the TOF camera are radial range values for all pixels. To apply previously derived relative transformation, the direct measurements need to be converted to 3D points in the Cartesian coordinates. For this purpose, the intrinsic parameters of the TOF camera are required. With Zhang's method [180], during a calibration procedure the 3 × 3 intrinsic matrix $\mathbf{M}$ ($\mathbf{M}_{PMD}$

for the TOF camera or $\mathbf{M}_{RGB}$ for the RGB camera) can be obtained:

$$\mathbf{M} = \begin{bmatrix} F_x & 0 & l_x \\ 0 & F_y & l_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1.4}$$

where $F_x = f \cdot s_x$ and $F_y = f \cdot s_y$ are the effective focal lengths in horizontal and vertical directions respectively with f the focal length (unit: mm) and s_x and s_y the physical pixel size (unit: pixel/mm). (l_x, l_y) is the pixel position of the principle point on the optical axis in the image. M can be directly used to project a 3D Cartesian point onto the image.

The geometric relation for converting a direct measurement (x, y, q_r) from the TOF camera to its Cartesian coordinate $[q_x, q_y, q_z]^\top$ is schematically illustrated in Fig. 1.5, where Q is a point in space, Q' is its projected point on the image plane with its the pixel position $[x, y]^\top$, and $OQ = q_r$ is the radial range value. As above, here the pin-hole camera model is assumed, and the optical axis OP of the lens is assumed perpendicular to the image plane.

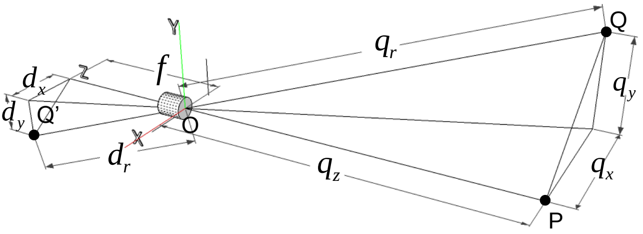


Figure 1.5: Converting the radial range to Z in Cartesian.

For the derivation convenience, the position Q' on the image plane scaled by the reciprocal of the focal length $1/f$ is calculated, it can be expressed with the geometric transformation [89]:

$$\begin{cases} d'_x = \frac{1}{f} d_x = \frac{x - l_x}{f s_x} = \frac{x - l_x}{F_x} \\ d'_y = \frac{1}{f} d_y = \frac{y - l_y}{f s_y} = \frac{y - l_y}{F_y} \\ d'_r = \frac{1}{f} d_r = \sqrt{1 + (\frac{d_x}{f})^2 + (\frac{d_y}{f})^2} = \sqrt{1 + d'^2_x + d'^2_y} \end{cases}.$$

Then correspondingly the world coordinates of Q can be obtained as

$$\begin{cases} q_x = \dfrac{d_x}{f} q_z = \dfrac{d'_x}{d'_r} q_r = q'_x q_r \\ q_y = \dfrac{d_y}{f} q_z = \dfrac{d'_y}{d'_r} q_r = q'_y q_r \\ q_z = \dfrac{f}{d_r} q_r = \dfrac{1}{d'_r} q_r = q'_z q_r \end{cases} . \tag{1.5}$$

Pre-Calculation for Better Efficiency

Combining the intrinsic matrix $\mathbf{M}_{RGB}$ of the RGB camera and the relative transformation matrix $\mathbf{T}_{RGB,PMD}$, the PMD's Cartesian coordinates from Eq. (1.5) can be used to obtain its pixel position $[u_{RGB}, v_{RGB}]^\top$ in the RGB image:

$$\begin{bmatrix} u_{RGB} \cdot Z_{wRGB} \\ v_{RGB} \cdot Z_{wRGB} \\ Z_{wRGB} \end{bmatrix} = \mathbf{M}_{RGB} \mathbf{T}_{RGB,PMD} \begin{bmatrix} q_x \\ q_y \\ q_z \\ 1 \end{bmatrix},$$

where Z_{wRGB} is the Z value in the RGB camera's Cartesian system. Substituting Eq. (1.3) and Eq. (1.5) into the above equation, the following formulation can be obtained

$$\begin{bmatrix} u_{RGB} \cdot Z_{wRGB} \\ v_{RGB} \cdot Z_{wRGB} \\ Z_{wRGB} \end{bmatrix} = \mathbf{M}_{RGB} \mathbf{R}_{RP} \begin{bmatrix} q'_x \\ q'_y \\ q'_z \end{bmatrix} q_r + \mathbf{M}_{RGB} \mathbf{t}_{RP} = q_r \mathbf{q}'_M + \mathbf{t}'_M, \tag{1.6}$$

where only q_r needs to be determined by the online observation data. The 3×1 vectors $\mathbf{q}'_M$ and $\mathbf{t}'_M$ for each point can be pre-calculated and stored. When the range measurements q_r for each pixel in the PMD camera are available, the calculation of its corresponding pixel position $[u_{RGB}, v_{RGB}]^\top$ in the RGB image can be performed quite efficiently. Also the calculation can be carried out in parallel, for which hardware acceleration can be utilized, e.g. with GPU or FPGA [89].

Hidden Surface Removal

The data fusion scheme discussed above projects the 3D data perceived by the TOF camera onto the image of the color camera, by which sub-pixel level mapping between the TOF camera and the color camera is established. Such a strategy works efficiently because most of the required information can be pre-calculated. However, it has a major drawback that the mapping for some points in the scene sensed by the TOF camera but not visible for the color camera due to different viewing angles from the two cameras will still be established and used in the complete RGBD data set. This is termed as the *hidden surface* artefacts [95].

Fig. 1.6 gives a clear schematic illustration for the hidden surface artefacts. Two boards are placed in front of the fused camera setup. A point P on the far board can be viewed by the TOF camera but is occluded to the color camera by the near board. With aforementioned fusion algorithm, P will be transformed from the TOF camera's coordinates to the color camera's Cartesian system, and then projected onto the color camera's image plane to build up a pixel level mapping. In this case, the pixel for P in the TOF image is mapped to the pixel of P' in the color image. Therefore, either the pixel of P in the TOF image will be assigned with a false color from P', or the pixel of P' in the color image will be (possibly) assigned with a false depth value from P, which can also be seen in the real captured RGBD image in Fig. 1.7 (a), where some points on the far board are assigned with the color from the near board.

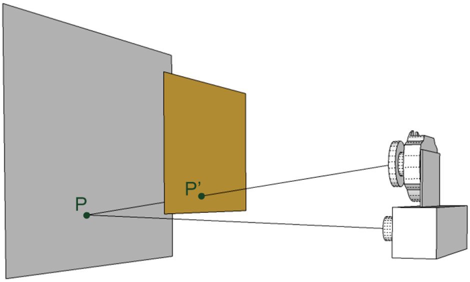
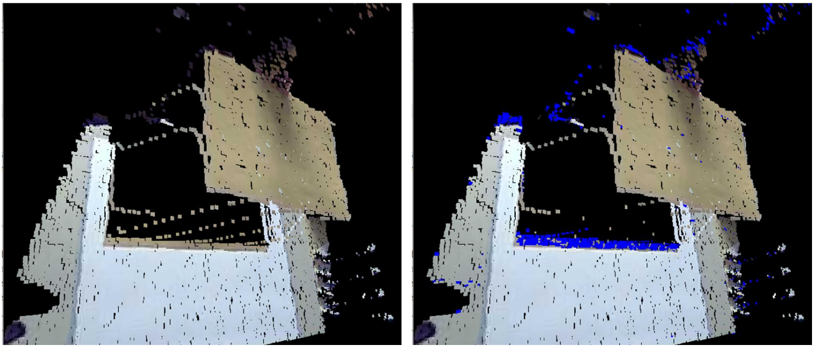


Figure 1.6: Schematic illustration for hidden surface artefacts.

One commonly adopted solution for the hidden surface removal is to perform *z-buffering* [90]. After the points from the TOF camera are projected onto the image of the color camera, if multiple range values are available on a pixel in the color image, only the closest range value will be recorded and the corresponding mapping will be established. While the other further points will be marked as invalid (no fusion mapping found). For example, in Fig. 1.6, the pixel in the TOF image corresponding to P will be marked as invalid by z-buffering. Fig. 1.7 (b) also shows the result of the hidden surface removal, where the invalid pixels determined by z-buffering on the far board are marked in blue for demonstration.



(a) with hidden pixels

(b) hidden pixels marked in blue

Figure 1.7: Hidden surface removal. The fused image is captured by placing a plate in front of a large plane. (a) shows the fused RGBD image without hidden surface removal. (b) shows the fused image with the hidden surface points determined by z-buffering marked in blue.

Z-buffering can effectively remove hidden pixels, however, it breaks the parallelism in the fusion calculation, because the depth value for a pixel in the color image depends on the comparison result from the currently and the previously projected depth values on the same point, if any. This means the transformation and the projection for all points have to be performed in serial and cannot make most of the modern many-core computing units. For some applications, where the influence of the hidden surface is not severe and real-time performance is of crucial importance, non-z-buffered fusion will be applied.

Fusion Result

The fusion result is illustrated in Fig. 1.8. On the left panel, the fused RGBD data is displayed, the intensity image from CamCube2.0 is on the top right panel, and the bottom right part shows the color image from the AXIS color camera. In the scene, a doll dwarf, a wall calendar and a paper box were put on a cupboard. The fusion error were within 1 pixel, which can clearly be seen on the upper boarder of the box. No points on the box were projected on the background board, and no points on the background board were mapped on the box.

There are some points that should be noted. First, if the reflected infra-red light cannot correctly represent the distance between the camera and the reflection surface, inaccurate range measurements may be present in the form of floating points. This

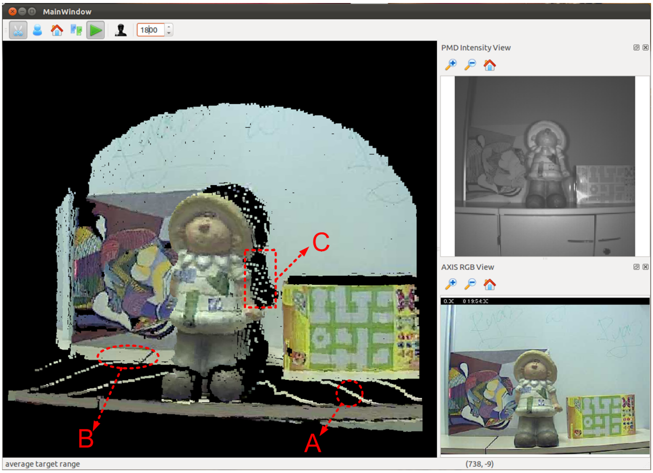


Figure 1.8: Fused RGBD image.

happens mostly on an object boarder as shown in the image area C in Fig. 1.8. This is because the received infra-red signal is a mixture of light reflected from the object and the background. Another error source is the less reliable measurements on the surfaces almost parallel to the optical axis of the PMD camera, e.g. the image region A in Fig. 1.8. It is obvious from the PMD intensity image on the top right panel in Fig. 1.8 that the upper surface of the cupboard is almost parallel to the viewing direction. Thus the strength of the reflected light will be too low to yield good range measurement. The produced range data are either floating points or simply invalid. Most of the floating points can be filtered out by checking the angle between the surface normal on a pixel and the optical axis. If the angle is close to 90°, the pixel can be marked as invalid. However, such filtering requires surface normal estimation for all points in the PMD image, which is a time consuming procedure.

Another commonly encountered artefact comes from multi-reflection. Above discussed parallel surfaces cannot reflect sufficient infra-red signal for the range data calculation. However, in some cases, multi-reflection from other surfaces will provide an extra amount of infra-red light and help producing data on parallel surfaces. For example, the image region B in Fig. 1.8 lies on the cupboard surface and should not have dense reliable range measurements. But some lights hit on the picture behind and are reflected on the cupboard surface then back into the camera lens. Such an inter-reflection between multiple objects produces dense range data in region B. Not only on the parallel surfaces, multi-reflection will also influence other surfaces in the scene. Due to the fact that the multi-reflected lights used for range calculation travel longer distance than the true range, these data also should not be used. But the filtering technique for tackling the errors caused by multi-reflection still remains an open problem.

Applying some dedicated filtering techniques can remove some particular artefacts. Meanwhile, after the pixel level mapping between the TOF and the color images is obtained, the quality and the lateral resolution can be improved by upsampling filter or Markov Random Field [182]. This can help applications like offline 3D reconstruction as the processing time is of secondary consideration and the accuracy is the most demanding characteristic. However, in robotic applications, real-time performance is of great significance. For the algorithm presented in this thesis, as illustrated in Fig. 1.1, the range data is only required for the accurate estimation stage in a relatively small image region where the target is expected to reside. Some filters, e.g. the surface normal filter, are only employed on this region, as presented in Chapter 3. This scheme improves the fusion data quality without increasing much computation.

1.3 Outline

The remainder of the thesis is organized as follows. Chapter 2, 3 and 4 describe the core algorithms proposed in this work. Each chapter contains a problem statement and a related works part, where the specific problem to be solved in this chapter is discussed and the corresponding state-of-the-art researches are reviewed. The related works are introduced in each chapter rather than in a dedicated separate chapter because the literature researches are more closely related to the topics discussed in the individual chapter. Besides, each chapter also describes some theoretical background knowledge that is involved in the description of the algorithm to be presented. Some of the equations or derivations described in the theoretical background section will also be used in the derivation of the proposed algorithm in the same chapter. In this way, the discussions in each chapter are self-contained with a clear boundary drawn between what is the part done in the literature and what is the contribution of this work.

More specifically, Chapter 2 presents the coarse pose estimation stage, which aims at dealing with the large motion between frames and providing a coarse 6DOF pose estimate which can be efficiently refined by further processing. The choice on the state vector used for describing the 6DOF pose is also discussed. Some implementation details regarding the GPU acceleration and the parameter settings are also described.

Chapter 3 describes the accurate pose estimation stage, which takes the coarse pose estimate as an initial guess and calculates the accurate pose with the gradient based iterative optimization. The Textured-ICP is proposed which incorporates the target visual appearance into the range data based ICP framework. The convergence of the iterative approach is analysed through experiments. Some implementation issues with respect to the surface normal estimation and the parallelism with GPU acceleration are discussed.

Chapter 4 introduces the approaches for handling the ambient illumination changes with Spherical Harmonics (SH). SH modeling requires surface reflectance. Therefore, the methods for estimating the reflectance are investigated. First, with the idea that the reflectance estimation in the visible spectrum may benefit from the reflectance information in the near infra-red spectrum, attempts are made to estimate the near infra-red reflectance through modeling the LED arrays on the PMD camera. However, results showed that the real LED arrays on the PMD camera cannot be accurately approximated by the theoretical LED array models provided in the literature. Then another method by using a calibration object is experimented and evaluated with a homogeneous test object. With the estimated reflectance, SH illumination model is integrated into both coarse and accurate estimation stages. Experiments demonstrated that the proposed method can work robustly even under severe lighting variations.

Although each algorithmic chapter gives some experimental results, more evaluations are given in Chapter 5. The reference evaluation is performed by comparing the estimated pose with the pose measured by iSpace, a high precision measurement system. A calibration method for unifying the estimated pose in the camera system and the measurement in the iSpace coordinates is introduced. Besides evaluation with the reference data, tests on various targets in both indoor and outdoor environments are presented as well. Moreover, the pose estimation algorithm is applied on non-cooperative mobile robot leader-follower formation. In the end, the thesis is summarized in Chapter 6 and some prospective works are suggested.

Chapter 2

Coarse Estimation with Sparse Representation

This chapter presents the coarse pose estimation algorithm that is capable of tracking an object moving fast in the cluttered background with real-time efficiency. The coarsely estimated pose will be used as the initial pose for a gradient based accurate pose estimation algorithm that is to be presented in the next chapter. The coarse pose estimation algorithm is developed based on the annealed particle filter framework, where each particle is evaluated with sparse representation. The problems to be solved are defined in the first section, which is followed by a review of the theories and approaches in the literature that are closely related to the proposed algorithm. Then details will be given regarding the composition and update rules for the template matrix in sparse representation and the use of multiresolution annealed particle filter for the high dimensional (6DOF) problem. Some implementation issues for the proposed algorithm will be discussed, and in the end the major points in this chapter will be summarized.

2.1 Problem Statement and Related Works

The problems to be addressed and the major contributions made in this chapter are clarified in this section. Meanwhile, the state-of-the-art works that are closely related to the proposed coarse pose estimation algorithm are discussed, including some recent developments of sparse representation, some methods for ℓ_1-regularized optimization and some filtering techniques for a high dimensional state space. More details regarding the theoretical formulation or derivations for the related works will be introduced in next section.

2.1.1 Problem Statement and Contributions

Although the visual tracking and pose estimation problems of a rigid object have been vastly investigated, the 6DOF pose estimation for a fast moving object in the cluttered background still largely remains unsolved. In this thesis, the problem is divided into two stages, a coarse pose estimation stage, which aims at locating the target object and providing a coarse pose estimate, and an accurate pose estimation stage, which deals with optimizing the accurate pose with the gradient based iterative method.

This chapter focuses on the coarse pose estimation with the 2D appearance information. The 2D appearance is adopted instead of the 3D range data, because a non-floating target usually moves on the ground, or is mounted on another object or placed on a table, for which the simple range based segmentation cannot be applied. Moreover, as also pointed out in later chapters, in-plane motion[1] can be well handled by using 2D appearance data, while out-of-plane movement can be efficiently determined with 3D range data. The coarse pose is more related to in-plane movement. Therefore, this stage takes the 2D appearance information for the pose estimation.

The main problem to be addressed in this chapter can be briefly stated as tracking a rigid object with mostly convex but arbitrary shape that can have large inter-frame motion in the cluttered background (Significant illumination variations can also be handled as will be addressed in Chapter 4). A coarse 6DOF pose will be provided from this stage that can be used as an initial pose to be further efficiently refined by the range data based pose estimation algorithms.

Recent developments of Sparse Representation (SR) have drawn substantial attention in fields like signal processing and computer vision. More specifically, [103] showed the ability of SR in object visual tracking. Furthermore, [83] demonstrated the real-time capability of SR in 2D tracking. Inspired by the performance of SR on tracking, this thesis investigated SR for providing a coarse 6DOF pose estimate. A new composition of the template matrix is proposed, with which the image patches grabbed from both target and background regions can be expressed sparsely and distinguished efficiently. For handling the high dimensional state space of the 6DOF problem, Annealed Particle Filter (APF) is adopted with multiresolution strategy for harnessing the distinctive power of SR. The parallelism of the proposed algorithm is

[1] For Cartesian coordinates with X and Y axes lying horizontally and vertically in the image plane and Z axis perpendicular to the image plane, the translation along X and Y axes, as well as the rotation around Z axis are denoted as in-plane motion, with the rest considered as out-of-plane motion.

studied and real-time performance is achieved under GPU acceleration. The major contributions of this chapter can be summarized as follows:

- A new composition of the template matrix in SR is proposed, with which a target image patch can be better distinguished from a background patch.
- Multiresolution strategy is adopted in APF, which can further harnessing the distinctive power of the proposed SR for the 6DOF tracking problem.
- Several rules for updating the model information is discussed, which can accommodate changes in video sequences without accumulating the inaccuracies during tracking.
- GPU acceleration is investigated and implemented for the proposed algorithm to achieve real-time performance.

2.1.2 Sparse Representation and the ℓ_1 Regularized Optimization

Sparse Representation (SR) or Compressed Sensing (CS) are not new concepts, they can be dated back to World War II [153], when the syphilis tests for soldiers were conducted in large groups instead of individually due to high costs. Only the positive groups will be further investigated. Such a sampling strategy was rest on the assumption that the infected subjects were sparse in total. Here sparsity refers to the number of non-zero elements compared to the size of the complete set, which can be measured by the ℓ_0 norm.[1]

Many real signals exhibit sparse nature or can be cast into a sparse approximation when expressed under an appropriate basis (so called *transform sparsity*), for instance, the correlation between a certain disease and a large number of medical indices [149], the real number of illumination with respect to the illumination from all possible directions [104], the number of infected soldiers in an army, the images with bounded variation in the neighborhood [42], the images or audios that are compressible under Fourier or wavelet basis,[2] just to name a few.

Although we can be aware that some signals are sparse, we are less likely to know a priori the magnitude and the location of the nonzero elements. And in a lot of cases, the recovery of these information is required, which is an optimization problem

[1] The ℓ_0 norm of a vector counts the number of nonzero entries in the vector [167].

[2] Although all transform coefficients can be nonzero, most of them are supposed to have negligible values

based on observation. As stated above, the sparsity of a signal is modeled by its ℓ_0 norm. Solving problems regularized with ℓ_0 norm requires combinatorial optimization [42] and is NP-hard [167]. SR has been quite dormant until recent decades. As the researches on the equivalence between ℓ_0 and ℓ_1[1] minimization [142], the theoretical analysis regarding the robustness of ℓ_1-based recovery or reconstruction [42], and the advances on ℓ_1 constrained convex optimization, SR drew a substantial attention in information theory, signal processing, image reconstruction fields, etc.[65].

Recent researches on SR raise three important arguments: 1. ℓ_1 minimization can exactly or stably recover the underlying sparse signal, i.e. the equivalence between ℓ_1 and ℓ_0 regularizations; 2. the number of measurements (rows in the measurement matrix) can be significantly less than what *Nyquist rate* suggests [7];[2] 3. even a random matrix can be used as the measurement matrix for obtaining an exact recovery (nonadaptive sensing [23]). These arguments have significant implications not only on signal recovery, but also on data acquisition process.

Most data are perceived by sensors through a *acquire – compress – transmit – decompress* procedure. The development of modern sensors usually enables massive data acquisition, e.g. even higher resolution CCD/CMOS chips for digital cameras. However, most of the data are redundant and are just compressed or dropped, due to difficulties in storage and transmission, without much human perceptual losses [42]. The second argument above provides a new way of sensing, which integrates acquisition and compression into one measurement process, that is how the name *Compressed Sensing* came from. Such a process can be interpreted as just sampling the important information. Less measurements also have significance on the scenarios where the sensing process is costly (MRI), or hazardous (X-Ray).

In the conventional sensing procedure, the compression is performed with the processing units on the sensor. The third argument above saves the limited memory resources because each required element in the measurement matrix can be generated with pseudo random generator. After it is used, the elements can be discarded rather than being stored. Moreover, most of the computation will lie in the recovery system, which usually has more powerful computing resources. Thus Compressed Sensing (CS) is asymmetrical [47]. Unlike Fourier or wavelet coefficients, CS gives equal

[1] The ℓ_1 norm of a vector $\mathbf{x}$ sums the absolute values of all entries in $\mathbf{x}$, i.e. $\|\mathbf{x}\|_1 = \sum_{i=1}^{N} |x_i|$.

[2] Given $\mathbf{x}$ is a sampled discrete signal of length N as required by Nyquist rate for an analog signal, if it has a sparse representation $\boldsymbol{\theta}$ under an orthonormal basis $\boldsymbol{\Psi}$, i.e. $\mathbf{x} = \boldsymbol{\Psi}\boldsymbol{\theta}$ where the nonzero entries in $\boldsymbol{\theta}$ is significantly less than N, a $M \times N$ measurement matrix $\boldsymbol{\Phi}$ can yield an observation vector $\mathbf{y}$ that contains all the information for recovery $\mathbf{x}$ with $M < N$, i.e. $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x}$. This means the sampling can be done with a sub-Nyquist rate [46]. See the single pixel camera in [47] for a good example.

importance on all measurements. This implies that the recovery can be performed robustly even when some of the measurements are contaminated.

The advances on optimization, especially the ℓ_1 minimization, plays an important role on the recent burst of SR researches. In the following, some often-mentioned optimization methods for SR as well as for ℓ_1-constrained problems in general are briefly introduced. This is followed by a short description for some computer vision applications with SR. More detailed theoretical formulation will be given in the next section.

Optimization Methods for Sparse Representation

Least Absolute Shrinkage and Selection Operator (LASSO) [148] formulates the ℓ_1 optimization problem as a bounded ℓ_1 norm constrained quadratic programming. Starting from a non-constrained least square solution $\boldsymbol{\beta}_0$, the signs of the elements in $\boldsymbol{\beta}_0$ are gathered and used to construct the inequality constraint. Then a new estimate $\boldsymbol{\beta}_i$ for the inequality constrained problem is acquired. When $\boldsymbol{\beta}_i$ satisfies the original bounded ℓ_1 norm constraint, it is adopted as the final solution; otherwise collect its signs and insert into the existing inequality constraints and continue the iteration. Because the $\boldsymbol{\beta}_i$ in each iteration forms a feasible solution that satisfies Karush-Kuhn-Tucker (KKT) condition, the final output will be a solution to the original problem. Although the convergence can be guaranteed, the size of the constraints increases as the iteration continues, which implies an increasing amount of computation load. Thus it can be inefficient for the large-scaled problems.

Basis Pursuit (BP) [28] converts the ℓ_1 minimization to Linear Programming (LP) by separating the nonnegative and negative components in the solution vector into two vectors. Then the simplex method or the interior-point methods [31] for solving LP can be applied. When the data is corrupted with noise, the problem can be formulated as a Quadratic Programming (QP) problem (also dubbed as Basis Pursuit DeNoising (BPDN) in [28]), where similar strategies as BP can be adopted, which iterates among the feasible solutions and gradually minimizes the cost function.

[74] transforms the ℓ_1-regularized optimization to a convex quadratic problem with linear inequality constraints, which is solved by the interior-point method (matlab code available from `http://www.stanford.edu/~boyd/l1_ls/`). They customize the primal barrier method and truncate the Newton system with the Preconditioned Conjugate Gradient (PCG). Such an approximation yields a memory and computation efficient ℓ_1 optimization solver which can be applied on the large-scale problems, e.g. $\approx 10^6$ variables in a few minutes. However, there are two iteration levels, one in the interior-point level, one in the PCG level, which made the implementation and

the parameter tuning quite complicated.

[54] cast the ℓ_1 regularized convex unconstrained optimization problem into the Bound-Constrained Quadratic Program (BCQP), which is solved by their proposed gradient projection algorithm (GPSR). In fact, the GPSR method is not restricted to the sparse problems, rather it can be used for solving general BCQP optimization problems. Their algorithm is reported to significantly outperform all the other ℓ_1 solvers compared in [54] with respect to the computational efficiency, especially when the problem is not very sparse. Another advantage is its simplification for implementation.

The greedy method, Matching Pursuit (MP) [102], is proposed to solve SR problem in a different manner compared to ℓ_1 minimization. MP selects an atom in the basis/measurement matrix in each iteration that most correlates with the residual vector. Each selected basis vector is augmented into a solution matrix. When the iteration terminates, the final sparse solution vector is obtained by solving the least squares with the selected solution matrix. Orthogonal Matching Pursuit (OMP) [153] adds an orthogonalization step during augmenting the solution matrix, so that it forms an orthonomal basis. OMP runs extremely fast when the problem is very sparse. However, as the number of non-zero elements increases, aforementioned GPSR is reported to outperform OMP.

The methods discussed above are generic in that they assume no prior knowledge on the test signal except for sparsity. [149] worked on problems where the coefficients were assumed locally constant, i.e. most of the neighboring coefficients were expected to have the similar magnitudes. LASSO was extended to incorporate another ℓ_1 regularized term into the constraint, which penalized big variations between neighboring coefficients. The new formulation, dubbed as fussed LASSO, can be efficiently solved with the Least Angle Regression (LAR) procedure [50].

Local constancy in the coefficient vector can be closely related to the dictionary design. [136] introduced a training stage. By extending the K-SVD procedure from [52], a dictionary can be built, with which the test signal can be expressed with coefficients from only a few column blocks in the dictionary. Besides local constancy of the coefficients, piecewise smoothness of the sparse signal was studied in [48]. From the observation, they exploited that the wavelet coefficients of a piecewise smooth signal bear not only sparsity, but also congregation around a connected subtree. This knowledge was fused with the reweighted ℓ_1 minimization, by which the difference between the ℓ_1 and ℓ_0 norms can be mitigated. The wavelet coefficient cluster was characterized with the Hidden Markov Tree (HMT) model. The magnitude of each wavelet coefficient was then estimated by the HMT model, which provided a weighting scheme for the reweighted ℓ_1 minimization. Simulation results showed that under such schemes, a wavelet piecewise smooth signal can be reconstructed with fewer measurements than conventional CS methods.

Application with Sparse Representation

As previous stated, Sparse Representation (SR) or Compressive Sensing (CS) has drawn a substantial attention in many fields. For instance, [21] recently applied CS on the Matrix Completion (aka Netflix problem) for recovery the entries in a matrix with only a few measurements; [56] investigated ℓ_1 minimization in curve and surface fitting problems, where conventional least squares solution is known sensitive to the outliers. [100] proposed a hybrid-CS to improve the throughput in wireless sensor networks, etc. Despite the promising trials in various realms, the following contents will only briefly review some vision related applications.

CS has direct implication on the data acquisition process. In CS framework, much less measurements can fulfill high dimensional signal sensing task. [47] built a single pixel imaging architecture. In stead of a high resolution photon detector, a Digital Micromirror Device (DMD) was used to mimic the behavior of the random measurement matrix. The DMD contained as much micromirrors as the required resolution. Although the results from the prototype cannot provide an image quality as good as the CCD camera, it exposed a new way of thinking for the sensor design.

Identity determination with facial appearance in a large database is intrinsically a sparse problem. With the assumption that the target appearance in question lies in the subspace spanned by the training set of the same subject, [167] built a template matrix, which contained the training face templates from many subjects captured under different illuminations and expressions. After solving the coefficients under the template matrix for an observation face image, the test image was associated with the subject whose templates captured the most portion of the non-zero coefficients. Thanks to SR theory, the optimization can be performed in a much lower down-sampled dimension, which greatly relieved the computation cost of the global recognition approach. Compare to the local approaches, e.g. the nearest neighbor and the nearest subspace, the method with SR yielded better or comparable recognition results.

Since the expressions can significantly influence the appearance of a human face, [85] proposed to use the 3D features subtracted from human faces to conduct robust identity recognition. They put the 3D facial features that have passed through an expression-invariant ranking process into the basis matrix instead of using the pixel intensities. However, it involved 3D mesh of human faces in both basis generation and test stages. If these information are not available, the 2D image based methods are more intriguing.

SR has been applied on the facial pose estimation as well. [110] collected a set of features into the basis matrix. These features were selected for their capability on detecting the head poses and the facial poses. Each atom in the basis matrix corresponded to a pose configuration. Due to the high dimension of the pose estimation problem, they only worked on the two-degree pose – yaw and pitch. On the other hand, [101] directly incorporated some modified training images with labeled poses into the basis matrix. When the test image was correlated to some elementary pose images in the basis matrix, the pose was determined as the same as those training images. The pose estimation capability was limited by the pose versatility in the training set.

Besides the facial pose, the human body pose can also be determined by the SR framework. [26] recovered the occlusion-free image or the feature image for the observed corrupted image with SR. Then the human pose was obtained by applying the recovered image into a mapping function between the feature image and the pose, which was built by Gaussian process regressor. A similar approach was introduced in [68]. These methods require a set of training images of the same target labeled with known poses. The capability and accuracy are determined by the resolution of training set.

More general object recognition was investigated in [126]. The SR framework was used in the learning stage to build a mapping between the target pose and its corresponding appearance. The basis matrix consisted of separable Gaussian bases. To achieve accurate pose estimation, a large number of densely sampled images from different viewing perspectives will be gathered. In their setting, only two translation plus one rotation degrees were considered.

The approaches above are mostly static applications. [103] applied SR on 2D target tracking in the video sequences. The basis matrix is composed of a trivial basis (unit matrix) for capturing the noise and the occlusion, and a set of target templates for expressing the target. A generic particle filter was used to determine the target position, where each particle was evaluated with SR solved by the interior-point method from [74]. To deal with the pose and illumination variations, the target templates were constantly updated during tracking. Inspired by the performance of this scheme, [83] used OMP for SR optimization and applied a random measurement matrix to reduce the computational burden. They showed robust 2D tracking with real-time efficiency.

SR was applied on multi-view/multi-camera surveillance in [133]. The target position on the ground of the scene was considered as sparse compared to the complete ground area. On each camera, the silhouette image was subtracted from the background and compressed with a random matrix before transmitted to a central computer. Moving target positions were then recovered with the second order cone programming within SR framework. The use of SR effectively reduced the demanded transmission load in such a distributed sensor network.

Due to the fact that cast shadows in a scene are illuminated by a small number of illumination sources, they can be considered as sparse with respect to the illumination from all possible directions. [104] utilized such a sparsity and applied SR on the illumination recovery. They generated a discrete set of all possible light directions by uniformly sampling from the surface of a unit hemisphere. Then some cast shadow images synthesized from these lighting conditions were put into the basis matrix. The cast shadows were also separated from the test image and used for lighting recovery with SR framework. The results were promising despite the requirement for scene geometry information.

2.1.3 Filtering in the High Dimensional State Space

Visual tracking and visual recognition problems in video sequences typically exhibit nonlinear (The target as well as the camera movement usually cannot be formulated as a linear transition model; The observation seldom can be expressed as a linear equation of the latent state variables.), non-Gaussian (The likelihood distribution across the image shows a multi-modal pattern, and the system noise cannot be considered as white Gaussian, because, for instance, some frames can be more blurred that the others). Therefore, conventional Kalman filter performs poorly in these tasks. In comparison, particle filter (PF) is commonly exploited for estimating and propagating more general probability distributions. PF was introduced in visual tracking by [70] under the name CONDENSATION (standing for CONditional DENSity PropagATION). Despite its success on the low dimensional state space problems, e.g. visual 2D tracking, as the dimension grows, the number of required particles increases dramatically. For example, a realistic articulated human body model can bear 25 degrees of freedom, and thus requires over 40,000 particles [38], which will result in an inefficient algorithm.

Many schemes can be adopted to reduce the number of particles required. For example, it is well known that when the observation and the transition model can be approximated with linear equations, Extended Kalman Filter (EKF) can be used for estimating the system state with high efficiency. By combining EKF for generating different proposal distributions for each particle, the most recent observation information can be incorporated before generating samples for the current time step, thus particles can be moved more effectively towards the high likelihood regions. Similarly, Unscented Kalman Filter (UKF) can be exploited for generating proposal distribution with more accurate mean and variance, which yields Unscented Particle Filter (UPF) [159]. However, the linearization of the observation and transition models in EKF requires simple and analytic representation of state space model. When, for instance, the observation process is sufficiently complex or has no analytic form, the EKF cannot be applied. Likewise, UKF also requires analytic observation model for propagating the mean and the variance [138], otherwise all *sigma points* should be evaluated for all particles, which will generally be a great computational burden. Furthermore, the performance of UKF combined PF and the UPF are case dependent [84], which should be determined by a careful study of the application in question.

PF has been applied on the 3D point cloud registration [141]. The same metric as the Iterative Closest Point (ICP) was used as the observer. Due to the high dimensionality in the 3D registration, the optimum cannot be reached with a small number of particles. Therefore, they exploited the Bayesian sequential procedure to combine with the iterative process in the ICP for performing the optimization. However, their approach only dealt with registration for static scenes instead of for sequential frames.

Some modifications have been proposed to solve the high dimensional data with PF. For instance, when the state space has special structures, e.g. the density distribution in some dimensions can be formulated as linear Gaussian, then Rao-Blackwellisation can be applied to partition the state space and increase the efficiency of the Bayesian framework [44]. Or parametric optimization can be combined with the Monte Carlo sampling scheme used in PF to increase the searching ability of the individual particle [19], where each particle is not only propagated with the system transition model or with random variation, but also is moved with an iterative optimization in its local region.

A more popular approach for dealing with the high dimensional problem is to split the processing in one time step into multiple layers, which forms the Annealed Particle Filter (APF). In APF, different layers are employed with different weighting functions with increasing distinctive powers, which yields a coarse-to-fine processing scheme.

This can dramatically reduce the number of particles required in the high dimensional problem. For example, [38] applied APF on tracking articulated human body. They demonstrated that with as little as 100 particles, a human model with 29DOF could be well tackled in high frame rate video sequences (e.g. 60fps). The performance can be further improved by making most of the state space structure, e.g. partitioning the state space according to the relation among dimensions. For the case of articulated human body, some degrees of freedom can move independently from other degrees, thus encourages independent evaluation [39]. Meanwhile, different body parts should use different observers, which also implies partitioned processing.

The 3D object tracking can also be categorized into the high dimensional problem due to the difficulty in searching an optimum point in a 6DOF state space. [3] applied APF and took the 2D silhouette edges to calculate the likelihood. They showed that when no background clutter was present, an accurate 6DOF pose of a rigid object can be achieved with 250 particle in three annealing layers.

Genetic approach is another direction that has been investigated for the high dimensional problem. [87] proposed to use Particle Swarm Optimization (PSO) for Mutual Information maximization instead of more commonly adopted Levenberg-Marquardt on the 3D model registration. On the other hand, genetic operations, e.g. the crossover and the mutation, can be combined with PF or APF to help overcoming the particle impoverishment limitations in the conventional PF framework [80, 124].

Besides developing methods that is capable of processing high dimensional data, [82] employed Gaussian Process Dynamic Model (GPDM) for the dimension reduction, where the original 29DOF human articulated body was cast into a low dimensional latent space. They showed that by combining with APF, their method can robustly estimate the human body pose under a relatively low frame rate (30fps or lower).

2.2 Theoretical Background

Some mathematical formulations for Sparse Representation (SR) are introduced in this section, which can help the interested readers to better understand what problems SR is developed to solve. This will be followed by some detailed descriptions with respect to orthogonal match pursuit and anneal particle filter that are adopted in the proposed coarse pose estimation algorithm. These derivations or formulations are provided, because they either are used in the development of the proposed algorithm or play an important role on the understanding of the problem to be discussed.

2.2.1 Formulation for Sparse Representation

A signal $\mathbf{I}$ (or an image sorted in 1D) can be linearly expressed under an orthonomal basis $\mathbf{A}$ as

$$\mathbf{I} = \mathbf{A}\mathbf{x},$$

where $\mathbf{x}$ is the transformation coefficient vector for $\mathbf{I}$, e.g. the wavelet coefficients if $\mathbf{A}$ contains a wavelet basis. Equivalently, we can interpret $\mathbf{x}$ as the underlying signal and $\mathbf{I}$ as the measurements for $\mathbf{x}$ under the measurement matrix (or feature extraction matrix) $\mathbf{A}$. The underlying signal $\mathbf{x}$ can be recovered with the observation $\mathbf{I}$ and the basis $\mathbf{A}$. If $\mathbf{A}$ is carefully chosen, $\mathbf{x}$ will be concentrated on a few elements with the rest equals or close to zero. However, without a priori knowledge about the number, the magnitudes and the locations of nonzero or non-negligible elements in $\mathbf{x}$, it seems exact recovery can only happen when the number of measurements in $\mathbf{I}$ equals to the size of $\mathbf{x}$, when the linear system can be uniquely solved.

Contradiction to the above common sense, recent researches on sparse representation [24, 42] argued that under mild conditions, a $n \times m$ measurement matrix $\mathbf{\Phi}$ with much less measurements than the problem dimension, i.e. $n \ll m$, will suffice for exact recovery. Now the linear relation is

$$\widetilde{\mathbf{I}} = \widetilde{\mathbf{A}}\mathbf{x}, \tag{2.1}$$

where $\widetilde{\mathbf{I}} = \mathbf{\Phi}\mathbf{I}$ and $\widetilde{\mathbf{A}} = \mathbf{\Phi}\mathbf{A}$. Such a linear system is underdetermined. To obtain a sparse $\mathbf{x}$, a ℓ_p-regularization term ($p \geq 0$) can be applied to the optimization:

$$\min_x \|\mathbf{x}\|_p \;\; s.t. \;\; \widetilde{\mathbf{I}} = \widetilde{\mathbf{A}}\mathbf{x}$$

As p gets closer to zero, more sparsity will be enforced, where ℓ_2 norm[1] (aka *ridge regression*) is the most frequently used regulation. However, ℓ_2 minimization requires no sparsity at all [42]. Although ℓ_0 norm provides a natural sparsity measure, it often requires a combinatorial optimization and is NP hard [167]. Therefore, it is often relaxed to ℓ_1 minimization, which can be solved by linear programming.

The measurements are often corrupted with noise as

$$\widetilde{\mathbf{I}} = \widetilde{\mathbf{A}}\mathbf{x} + \mathbf{z},$$

[1] The ℓ_2 norm of a vector $\mathbf{x}$ is the Euclidean length of $\mathbf{x}$, i.e. $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{N} x_i^2}$.

where $\mathbf{z}$ denotes the noise. Then above recovery can be formulated as:

$$\min_{x} \|\mathbf{x}\|_1 \;\; s.t. \;\; \|\widetilde{\mathbf{I}} - \widetilde{\mathbf{A}}\mathbf{x}\|_2^2 \leq \epsilon \tag{2.2}$$

and can be solved by *Second Order Cone Programming* (SOCP).

Equivalently, Eq. (2.2) can be reformulated as:

$$\min_{x} \|\widetilde{\mathbf{I}} - \widetilde{\mathbf{A}}\mathbf{x}\|_2^2 \;\; s.t. \;\; \|\mathbf{x}\|_1 \leq t, \tag{2.3}$$

which is known as the *LASSO* formulation and can be solved by *Quadratic Programming* (QP) [54].

Likewise, the optimization above can be expressed as:

$$\min_{x} \frac{1}{2} \|\widetilde{\mathbf{I}} - \widetilde{\mathbf{A}}\mathbf{x}\|_2^2 + \tau \|\mathbf{x}\|_1, \tag{2.4}$$

where the selection of τ depends on the sparsity as well as the noise level [28]. This is a BPDN (Basis Pursuit DeNoising) formulation and can be efficiently solved by interior-point methods [74] even for very large scale problems.

The equivalence between ℓ_1 and ℓ_0 recovery of a S-sparse signal (with at most S nonzero elements) happens when $\widetilde{\mathbf{A}} = \mathbf{\Phi}\mathbf{A}$ satisfies the *Restricted Isometry Property* (RIP) introduced by Candès and Tao [22], who stated if for any S-sparse signal $\mathbf{x}$, a matrix $\widetilde{\mathbf{A}}$ obeying RIP will satisfy

$$(1 - \delta_S)\|\mathbf{x}\|_2^2 \leq \|\widetilde{\mathbf{A}}\mathbf{x}\|_2^2 \leq (1 + \delta_S)\|\mathbf{x}\|_2^2,$$

where δ_S is the *isometry constant* of $\widetilde{\mathbf{A}}$ and should be less than 0.307 as proved in [20]. RIP implies that a S-sparse vector $\mathbf{x}$ transformed under $\widetilde{\mathbf{A}}$ roughly preserves its Euclidean length [23], which, otherwise, exact recovery with $\widetilde{\mathbf{A}}$ will make no sense. [24] demonstrated some methods for generating a matrix obeying RIP, including the random matrices, the Fourier ensembles and some general orthogonal ensembles. The measurement matrix generated with different methods will require different number of measurements (rows of the measurement matrix) for a robust recovery.

A closely related notion to RIP is *mutual incoherence* [65], which can be used to measure the similarity between the measurement matrix $\mathbf{\Phi}$ and the basis matrix $\mathbf{A}$. When mutual incoherence of $\mathbf{\Phi}$ and $\mathbf{A}$ has a really small value, $\widetilde{\mathbf{A}}$ is expected to obey RIP. [23] showed that a random measurement matrix $\mathbf{\Phi}$ is with high probability incoherent with any fixed basis $\mathbf{A}$. A Gaussian random matrix is considered as one such matrix for $\mathbf{\Phi}$, or with a more efficient random number generator – the Hash kernel proposed in [143].

2.2.2 Orthogonal Matching Pursuit

When a target signal $\mathbf{x}$ is extremely sparse (e.g. 50 nonzero elements in a 4096-vector [54]), Orthogonal Matching Pursuit (OMP) [153] is believed to be a very efficient yet robust recovery method for solving sparse representation problems.

The sparse problem solved by OMP can be formulated as:

$$\min_{x} \|\mathbf{x}\|_0 \ \ s.t. \ \ \|\widetilde{\mathbf{I}} - \widetilde{\mathbf{A}}\mathbf{x}\|_2^2 \leq \epsilon. \tag{2.5}$$

Different from Eq. (2.2), OMP tries to find solution for a ℓ_0-regularized problem. In each iteration, OMP selects the atom in the dictionary that most correlates with the current residual vector. Then the selected atom is integrated into the most recent orthogonal solution space. Put aside for now the measurement matrix $\mathbf{\Phi}$, the detailed procedure for OMP is shown in Algorithm 2.1.

Solving least squares in Step 2 (d) equals to projecting $\mathbf{I}$ onto a subspace spanned by all columns from $\mathbf{\Psi}_k$. Thus the residual $\mathbf{r}_k$ in Step 2 (e) is orthogonal to the current solution space $\mathbf{\Psi}_k$. This guarantees that in each iteration a new atom in $\mathbf{A}$ will be selected instead of some already chosen ones. To be more efficient with respect to solving least squares in Step 2 (d), the solution subspace $\mathbf{\Psi}_k$ can be maintained with the *QR* factorization instead of directly appending the selected column into $\mathbf{\Psi}_{k-1}$ as in Step 2 (c). As suggested by [54], this can be done with the *Modified Gram-Schmidt* (MGS) algorithm.

Step 2 (a) of Algorithm 2.1 is the most time consuming step especially for a large scaled problem. However, when implemented with GPU acceleration, the correlation can be calculated with marginal costs, which will be detailed in Subsection 2.4.2. Further speedup can be achieved by applying the measurement matrix $\mathbf{\Phi}$ as in Eq. (2.1). Theoretically, the number of measurements (i.e. rows n of matrix $\mathbf{\Phi}$) should satisfy $n \geq K\eta log(n/\delta))$ for η-sparse problem with failure probability δ [54]. However, for a recognition task that does not require exact signal recovery, [83] demonstrated that the number of measurements can be remarkably less than the theoretical bound.

Algorithm 2.1 OMP procedure

```
Inputs:
  1. Dictionary $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_n] \in \mathbb{R}^{m \times d}$ and data vector $\mathbf{I} \in \mathbb{R}^d$.
  2. Sparsity level $\eta$, predefined correlation bound $\tau$.
Procedure:
  1. Set the initial residual vector $\mathbf{r}_0 = \mathbf{I}$, the index set $\mathbf{\Lambda}_0 = \emptyset$ and set initial
     solution space $\mathbf{\Psi}_0$ to be an empty matrix.
  2. for $k = 1 \to \eta$ do
     (a) Calculate absolute correlation for all atoms as $e_{j=1,...,n} = |\langle \mathbf{r}_{k-1},\ \mathbf{a}_j \rangle|$.
     (b) Find the $p$-th atom in $\mathbf{A}$ that has largest absolute correlation value $e_p$. If
         $e_p < \tau$, terminate iteration; else continue.
     (c) Append $p$ to end of index set $\mathbf{\Lambda}_{k-1}$ to form $\mathbf{\Lambda}_k = \mathbf{\Lambda}_{k-1} \cup \{p\}$, expand
         solution space to incorporate most recent $p$-th atom as $\mathbf{\Psi}_k = [\mathbf{\Psi}_{k-1}, \mathbf{a}_p]$.
     (d) Get an estimate for the sparse solution with least-square
             $\mathbf{x}_k = arg\, min_{\mathbf{x}} \|\mathbf{I} - \mathbf{\Psi}_k \mathbf{x}\|_2$
     (e) Calculate the new residual as $\mathbf{r}_k = \mathbf{I} - \mathbf{\Psi}_k \mathbf{x}_k$.
  3. end for
Outputs:
  1. Nonzero index set $\mathbf{\Lambda}_k$ and the corresponding solution values in $\mathbf{x}_k$.
  2. Solution vector with entries specified by indices in $\mathbf{\Lambda}_k$ set to the values in $\mathbf{x}_k$,
     and all the rest to be zero.
```

2.2.3 Annealed Particle Filter

In Bayesian filtering, the posterior distribution $p(\boldsymbol{\theta}_{0:t}|\mathbf{z}_{0:t})$ offers all useful information necessary about the system state up to time t [44], where $\boldsymbol{\theta}_t$ and $\mathbf{z}_t$ are the system (hidden) state and the observation respectively at time t. In most cases, more interested is one of its marginal distributions, or the so called filtering density $p(\boldsymbol{\theta}_t|\mathbf{z}_{0:t})$, by which we can obtain the point estimate for desired properties. For instance, the expectation of some function $f(\boldsymbol{\theta}_t)$ defined on $\boldsymbol{\theta}_t$ can be calculated as

$$\mathrm{E}[f] = \int f(\boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t|\mathbf{z}_{0:t}) \mathrm{d}\boldsymbol{\theta}_t.$$

When a set of samples $\boldsymbol{\theta}_t^{(l)}$ $(l = 1, \ldots, L)$ can be independently drawn from

$p(\boldsymbol{\theta}_t|\mathbf{z}_{0:t})$, the integral can be approximated as [14]:

$$\mathrm{E}[f] \approx \frac{1}{L}\sum_{l=1}^{L} f(\boldsymbol{\theta}_t^{(l)}). \tag{2.6}$$

However, it is often impractical to sample directly from an arbitrary posterior distribution. In this case, *Bayesian Importance Sampling* is usually adopted, where a *proposal distribution* $q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{z}_{0:t})$ is used instead for generating the samples. The proposal distribution should be sufficiently simple for drawing samples and reasonably close to $p(\boldsymbol{\theta}_t|\mathbf{z}_{0:t})$. This leads to the following equation for approximating the expectation:

$$\mathrm{E}[f] \approx \sum_{l=1}^{L} f(\boldsymbol{\theta}_t^{(l)})w_t^{(l)}, \quad \text{where} \;\; w_t^{(l)} = \frac{w_t^{*(l)}}{\sum_{j=1}^{L} w_t^{*(j)}},$$

and the unnormalized importance weight is calculated recursively [44] as:

$$w_t^{*(l)} = w_{t-1}^{*(l)} \frac{p(\mathbf{z}_t|\boldsymbol{\theta}_t^{(l)})p(\boldsymbol{\theta}_t^{(l)}|\boldsymbol{\theta}_{t-1}^{(l)})}{q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{z}_{0:t})}. \tag{2.7}$$

Here $p(\mathbf{z}_t|\boldsymbol{\theta}_t^{(l)})$ and $p(\boldsymbol{\theta}_t^{(l)}|\boldsymbol{\theta}_{t-1}^{(l)})$ represent the observation model and the system transition model respectively. Eq. (2.7) forms the basis for *Sequential Importance Sampling (SIS)* and provides a mechanism for processing the sequential data, e.g. video sequences.

As demonstrated in [159], SIS bears a serious limitation that the variance of the importance weights increases over time. This will lead to a degeneracy of the algorithm as all but one samples will have importance weights quite close to zero. Although it can be solved by using an optimal proposal distribution $q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{z}_{0:t}) = p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{z}_{0:t})$, as the case of $p(\boldsymbol{\theta}_t|\mathbf{z}_{0:t})$, drawing samples from $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{z}_{0:t})$ is usually not straight forward. Another widely adopted strategy is, after the prediction and the observation, to implement a resampling step to enforce a reasonably effective sample size. The resampling process copies multiple times those samples with high importance, and drops samples with low weights. After resampling, unlike what Eq. (2.7) suggests, all survived particles will have same importance. This increases the number of effective samples at the cost of potentially losing sample diversity resulting in the *sample impoverishment* [124]. In such cases, genetic approaches, e.g. mutation and crossover, can help re-supplying the impoverished particles [80].

Above discussion indicates that the selection of the proposal distribution plays a crucial role on the success of applying the Bayesian Sequential Importance Sampling. Generic PF usually takes system transition model $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ as the proposal distribution [70], which leads to a simple implementation. In last decade, the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) have been investigated to provide the proposal distribution for each individual particle, which yield the Extended Kalman Particle Filter (EKPF) and the Unscented Particle Filter (UPF) respectively. These methods usually require the observation process to be simple or to have an analytic form.

When the potential system state spans a high dimensional space, a large number of samples will be required to represent the posterior distribution or to perform the stochastic search for the optimum in Bayesian SIS framework. APF is proposed to tackle such a situation [38]. Although it also adopts the transition model as the proposal distribution and employs a resampling step, APF does not output the estimated posterior mode or the expectation directly after resampling nor propagate the resampled particles to the next time step. Instead, it further processes the particles in multiple layers with weighting functions of increasing distinctive powers. The weighting function $w_m(\mathbf{z}, \boldsymbol{\theta})$ from layer m has the same form as the initial layer M and only differs slightly on the sharpness as

$$w_m(\mathbf{z}, \boldsymbol{\theta}) = w(\mathbf{z}, \boldsymbol{\theta})^{\beta_m} \tag{2.8}$$

with $\beta_0 > \beta_1 > \cdots > \beta_M$.

Under the above setting, layer M will yield a broad distribution indicating the general large-scale feature in the search space, while layer zero brings about sharp peaks showing the local features of the distribution function, by which an accurate estimate of the local mode can be achieved. Besides using the same observer for all layers, it is also possible to use different observers in different layers as the cascaded particle filter in [86].

The procedure of APF is as follows [39]:

Start from layer $m = M$, draw all L particles from $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$.

1. Evaluate all particles to obtain $w_m^{*(l)}(\mathbf{z}, \boldsymbol{\theta})$, and get the normalized weight $w_m^{(l)}(\mathbf{z}, \boldsymbol{\theta})$.

2. Apply resampling with replacement under normalized weight $w_m^{(l)}(\mathbf{z}, \boldsymbol{\theta})$. Impose zero mean Gaussian random variable $\mathbf{b}_m$ to all resampled particles: $\boldsymbol{\theta}_{t,m-1}^{(l)} = \boldsymbol{\theta}_{t,m}^{(l)} + \mathbf{b}_m$

3. If $m \neq 0$, $m \leftarrow m - 1$, go back to step 1; otherwise, terminate iteration and output desired expectation with Eq. (2.6).

The multi-layered processing enables most (or even all) particles to converge to the same point, which helps the mode determination. However, although the initial particles in APF are also drawn from $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$, different from the generic PF, the particle convergence from the last time step (or the last frame) reduces the diversity of the particle distribution. It works practically as initialing all particles from one single particle located at the predicted mode.

2.3 Coarse 6DOF Tracking

This section introduces the method for the coarse pose estimation. The 6DOF tracking algorithm is based on the Annealed Particle Filter (APF) framework, and each particle is evaluated by the sparse representation. Towards this purpose, first a new composition of the template matrix for sparse representation is proposed, which has better distinctive power than those presented in the literature. Then the multi-resolution strategy is adopted in APF to further harness the distinctive power. The target is allowed to move fast in a highly cluttered background. Moreover, several update steps are also discussed to accommodate to the environment changes during tracking.

2.3.1 Composition of Template Matrix for Sparse Representation

In the simple linear equation $\mathbf{Ax} = \mathbf{I}$, if $\mathbf{x}$ has only a small number of elements non-zero, $\mathbf{x}$ can be considered as the Sparse Representation (SR) for the signal $\mathbf{I}$ under the basis $\mathbf{A}$. In computer vision, $\mathbf{I}$ can be interpreted as an image patch. [167] presented an application with SR for the human facial identity recognition, where the basis matrix was composed of facial images from hundreds of different persons. They pointed out one major advantage of SR over previous approaches on the identity recognition, e.g. the nearest neighbour or the nearest subspace, was its distinctiveness. Namely, the sparse vector $\mathbf{x}$ will be quite different for different individuals, by which their identity can be effectively determined.

To employ SR, as well as to take advantage of its distinctive power, the core is to cast the problem under question into a sparse representation problem. Or more specifically, to find a basis that can sparsely express the interested signal. In the coarse estimation stage, $\mathbf{I}$ will be the image patch grabbed by each particle in the annealed particle filter. The aim is to find a basis or a template matrix $\mathbf{A}$ that can be used to sparsely express **I** and effectively distinguish a background image patch from a target patch. This subsection describes a flexible and extendable composition of the template matrix for visual tracking, which can effectively distinguish a target image patch from a background patch.

Wavelet Basis vs. Trivial Basis

SR was investigated for visual tracking in [103] and [83]. They composed the template matrix with a unit matrix **E** for capturing noise and background and a set of grabbed target templates **M** for modeling the target appearance. The identity of a grabbed image patch **I** can be determined by solving the following equation

$$[\mathbf{E}, \mathbf{M}] \begin{bmatrix} \mathbf{x}_E \\ \mathbf{x}_M \end{bmatrix} = \mathbf{I}, \tag{2.9}$$

where $\mathbf{x}_E$ and $\mathbf{x}_M$ are the subvectors of the coefficient vector **x** corresponding to the submatrices **E** and **M** in **A** respectively. After having solved Eq. (2.9), whether the patch is judged as from target or not is analyzed by calculating the reconstruction error

$$\varepsilon = \|\mathbf{M}\mathbf{x}_M - \mathbf{I}\|. \tag{2.10}$$

With such a scheme, when an image patch comes from the target region, it can be sparsely expressed under such a basis. However, when the image patch is from a background region, the sparsity cannot be guaranteed. What's worse, such a basis will lose the distinctive power that makes it superior over other methods, because the reconstruction error between the target and the background image patches will be tiny. This is illustrated in the second row of Fig. 2.1 (in (d), (e) and (f)), where the target image patch (the image in the 32×64 red box) in Fig. 2.1 (a) is used as the target template **M**. It can be observed from the reconstruction coefficients for a perfect target patch (d), an occluded target patch (e) and a background patch (f), that under the unit matrix, the variation between different input patches is quite small. The greatest non-zero element in **x** always corresponds to the same target template. This indicates that simply using a unit matrix cannot take advantage of the distinctive power of SR.

It is well known that the wavelet transform can be used for compressing a lot of natural signals. Therefore, under a wavelet basis, it is expected that a signal can be expressed sparsely. By this motivation, the unit matrix **E** is replaced by a wavelet basis **W**, the Symmlet-4 [28]. The resulting reconstruction coefficients are shown in the third row of Fig. 2.1. It is obvious that the most significant reconstruction coeffi-

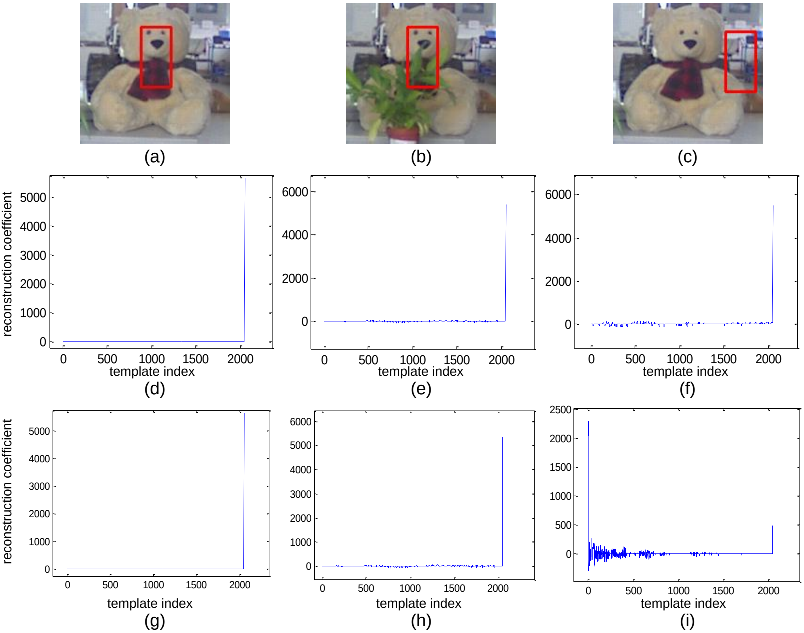


Figure 2.1: Decomposition coefficients under unit matrix (second row) or wavelet basis (third row). The three images in the top row show the target template in (a) used as **M**, the target image patch under occlusion in (b) and the background image patch in (c) respectively. The rest two rows illustrate the reconstruction coefficients if **I** is from its above grabbed image patch. The X axis in the reconstruction coefficient figures represents the column index in the template matrix with one pure target template put in the last column, and Y axis expresses the magnitude of the reconstruction coefficients.

cient for the background image patch is captured by the wavelet basis. This makes a huge disparity between the reconstruction coefficient for a background (Fig. 2.1 (i)) and for a perfect target patch (Fig. 2.1 (g)), or an occluded patch (Fig. 2.1 (h)).

It will be signal dependent regarding the choice of wave forms. For instance, the Fourier basis usually performs better on periodic signals. Whereas, for the rectangle pulse or step signals, Haar wavelet is preferred. After a number of experiments on DCT (Discrete Cosine Transform), Haar wavelet, Symmlet-4, Daubechies-4 and Meyer with various image patches grabbed from a variety of natural images, Symmlet-

4 shows the most reliable performance for a large number of cases. [41] proposed to use a mixed template matrix by combining different wave forms, e.g. edgelets or wavelets. However, this will increase the size of the basis and the required computations. During our experiments, it is observed that the high order half of the wavelet basis seldom has big impact on the coefficients, which can also be seen in Fig. 2.1 (i). Therefore, the higher order half of Symmlet-4 is replaced by the lower order half of the Haar basis, which can increase the versatility of the signal wave forms that the basis can express sparsely.

Multiresolution Consideration

In a particle filter based high dimensional mode estimate algorithm, it is less likely that the mode can be located with a limited number of particles. This implies, when the hypothesis state is close to the optimal state value, it will be desirable to assign the corresponding particle with high weight. Adopting wavelet basis for composing the template matrix can increase the distinctive power of SR, which also remarkably reduces the size of the high likelihood region. When the search space is small, such reduction of high likelihood region can help accurately locating the mode. However, for problems like tracking a fast moving object, the search space is large, even the best particle can be some distance away from the optimum. In this case, the distinctive power will be harmful to the robustness of the tracking algorithm. For instance, in top two rows of Fig. 2.2, for a slightly translated target image patch (b), it will be desirable to judge it as from target region. But the reconstruction coefficients in (e) are more similar to the background case in (f). Therefore, the distinctive power needs to be harnessed for use in high dimensional problems with a large search space.

This work employs annealed particle filter for filtering out the desired optimum. In higher layers, distribution with sharp peak will be more beneficial for accurately locating the optimum. While in lower layers, smooth distributions will be more helpful for robustly capturing the general trend of distribution with a small number of probes. The distinctiveness of SR will yield sharp probability distributions. A widely adopted strategy in computer vision for harnessing the distinctive power is multiresolution processing, which means low resolution (smoothed) images will be used in lower annealing layers and fine images are for higher layers.

The decomposition coefficients for smoothed image are shown in the bottom two rows of Fig. 2.2. The image patches are still grabbed from the same position as for the high resolution case in top two rows of Fig. 2.2. In contrast to the result for high resolution level, the most significant non-zero coefficient in (m) for slightly translated target image patch (h) is captured by target template. Therefore, it will

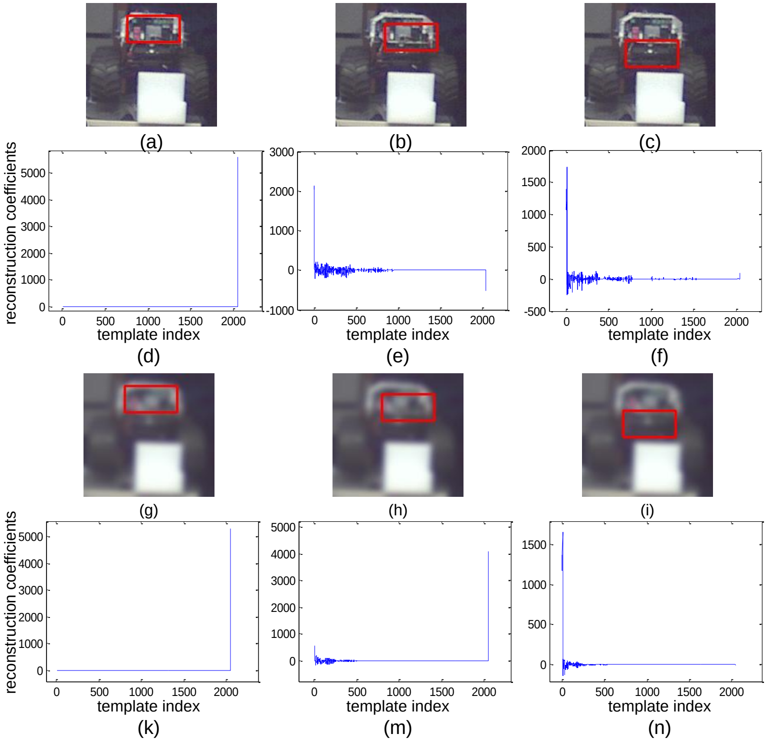


Figure 2.2: Reconstruction coefficients in high and low resolution levels. The template basis is composed of symmlet-4 and target template (from image in corresponding resolution level). Images in top two rows are for fine resolution and bottom two rows are for low resolution level. The X axis in images for reconstruction coefficients represents column index of template matrix, Y axis is for reconstruction values.

be determined as from target region by Eq. (2.10). Meanwhile, the background patch (i) is still modeled by wavelet basis. Combining the performance under fine resolution in Fig. 2.2, an effective particle filtering can be based upon.

Occlusion and Background Templates

Simply using one target template and two low order half wavelet bases cannot work robustly in complex scenarios, because the target can be occluded, the illumination can vary significantly and the background can be highly cluttered. The template matrix in SR provides a flexible framework for modeling these influences. In principle, there are infinitely many occlusion possibilities, e.g. occluded by a metal fence, by some plants, or by a box. If the occlusion type can be predicted, some corresponding templates can be established and incorporated into the template matrix. Although occlusion is not the major problem to be solved in this work, some block-like occlusion templates are built and put into the template matrix. Fig. 2.3 shows the employed occlusion templates for all resolution levels.

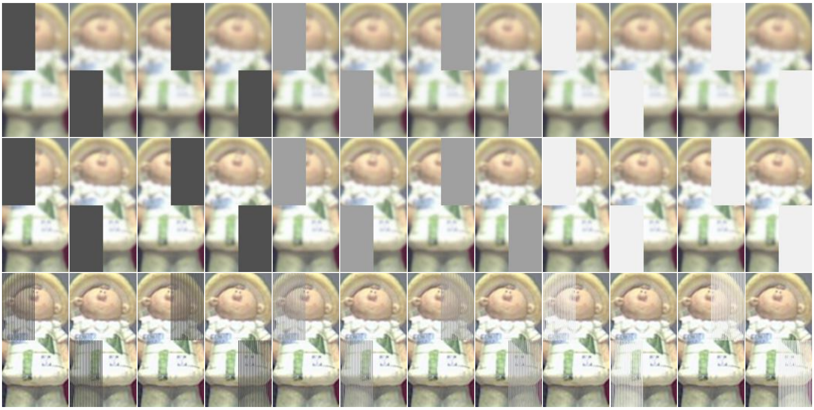


Figure 2.3: Target occlusion templates for the doll dwarf in all resolution levels. Images from the top row to the bottom row are for the low, the intermediate and the high resolution levels respectively.

The use of the occlusion templates will bring about the capability of handling these simple block-like occlusions. In addition, in some cases, the appearance changes caused by illumination variations can be interpreted as the occlusion, for example, when some shadows are cast on part of the target, or when the lighting comes from one side and the target appears half bright and half dark, or when the high light points occupy a large area of the target surface as can be seen in Sec. 5.3.

Since the intensity of the occlusion appearance usually cannot be known a priori, several occlusion intensity levels are employed, e.g. three intensity levels in Fig. 2.3. Moreover, different resolution levels will be used in different layers of the annealed particle filter. After several layers of processing, the particles should have already concentrated on the desired optimal state in the highest layer. The task in the highest layer is to determine which particles are more close to the optimum. When no occlusion is present, some background pixels may be captured by the occlusion blocks, thus an inaccurate alignment may be interpreted as an occluded target. Therefore, it is recommended to weaken the influence of the occlusion templates in this layer, as shown in the bottom row of Fig. 2.3. When the target is indeed occluded, a better alignment can still yield lower reconstruction errors under the weakened occlusion templates.

One major drawback of using the occlusion templates is, the modeling capability of the target templates (containing occlusion templates and pure target templates) will be too strong than the non-target templates (the wavelet bases). This will lead to the consequence that some of the background patches may be determined as from the target region. The success of the SR framework relies heavily on the balance between the target and non-target templates. Therefore, a number of randomly grabbed image patches from the background region are incorporated into the template matrix and used as non-target template. With the aforementioned two lower order half wavelet bases $\mathbf{W}_1$ and $\mathbf{W}_2$, the final composition for the template matrix is

$$\mathbf{A} = [\mathbf{W}_1, \mathbf{W}_2, \mathbf{B}, \mathbf{T}_{ocl}, \mathbf{T}_{pur}, \mathbf{T}_{ill}], \tag{2.11}$$

where $\mathbf{B}$ contains the background templates, $\mathbf{T}_{ocl}$ is for the occlusion templates in one resolution level and $\mathbf{T}_{pur}$ is for the pure target template. In Chapter 4, a group of illumination target templates $\mathbf{T}_{ill}$ for modeling the illumination variations are also employed. Now the target templates used in Eq. (2.10) have the composition of $\mathbf{M} = [\mathbf{T}_{ocl}, \mathbf{T}_{pur}, \mathbf{T}_{ill}]$, and the rest templates in $\mathbf{A}$ are used as the non-target templates. The composition of the template matrix $\mathbf{A}$ is also illustrated in Fig. 2.4.

Due to the use of wavelet bases, the template matrix $\mathbf{A}$ must have a dyadic number of rows. The experiments showed that 1024 or 2048 can be chosen as a good compromise between performance and computational efficiency. In current implementation, $\mathbf{A}$ contains 12 occlusion templates as shown in Fig. 2.3. Besides, 200 background patches are grabbed from the observation image in the initialization stage and will be updated during tracking, as will be discussed in Subsection 2.3.3. This work uses only one pure target template that is grabbed from the initialization frame. But it can be extended according to the actual application when more information about the target are available. The generation of the illumination templates will be introduced in Chapter 4. But some problems regarding the illumination modeling still remain open. Therefore, for most of the tests conducted in this thesis, the illumination templates are set to zero.

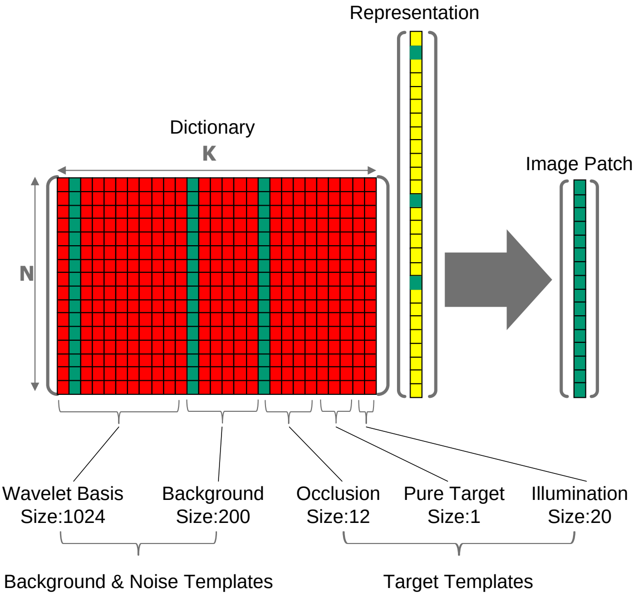


Figure 2.4: Illustration for the template matrix composition. This is a schematic illustration for $\mathbf{Ax} = \mathbf{I}$. where $\mathbf{x}$ is the sparse representation of $\mathbf{I}$ under the dictionary (or the template matrix) $\mathbf{A}$. The non-zero elements in the representation is marked in green, which corresponds to the green columns in the dictionary.

Some experimental results for recognition of the occluded Merlin robot under the above template matrix composition are illustrated in Fig. 2.5. For illustration convenience, the background templates are placed between the two wavelet bases, so that the location of the background templates in $\mathbf{A}$ is far away from the target templates and the coefficients on the background templates can easily be recognized. The upper two rows show the results in the low resolution level, while the lower two rows are for results in the high resolution level.

In the low resolution level, the background patches (Fig. 2.5 (a) and (b)) are captured by the randomly grabbed background templates (as shown in Fig. 2.5 (d) and (e)). As a contrary, the image patch from the target region (Fig. 2.5 (c)), although rotated and occluded, is still captured by the target templates (Fig. 2.5 (f)). In the annealed particle filter, this will provide a good basis for wiping out particles on the

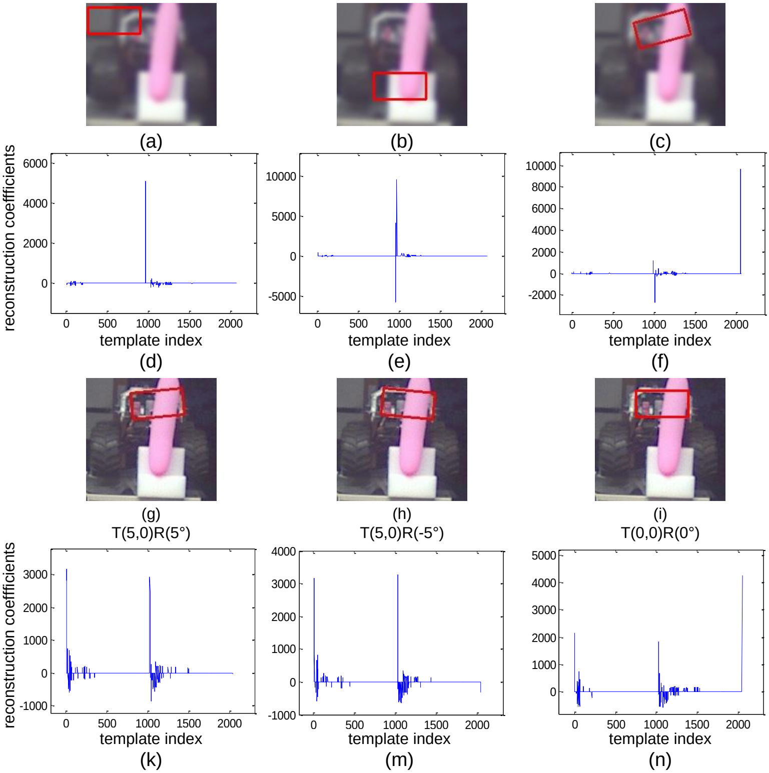


Figure 2.5: Reconstruction coefficients in low and high resolution levels. The composition for the template matrix in **A** is as specified in Eq. (2.11), but the background templates are placed between the two wavelet bases for illustration convenience.

background and propagating the particles on the target region to higher layers.

The particles are expected to have concentrated on the target region in high resolution levels, only their pose may be not accurate, as the case in Fig. 2.5 (g) and (h). Therefore, for the high resolution level, only particles on the target region but with different alignment are tested. The slightly translated and rotated particles (Fig. 2.5 (g) and (h)) are mostly captured by the wavelet bases and therefore can be distinguished from the accurately aligned particle (Fig. 2.5 (i)). Such a distinctiveness is essential for the accuracy of the pose estimation in the coarse stage.

2.3.2 Multiresolution Annealed Particle Filter

In high dimensional problems, e.g. the 6DOF pose estimation problem, it is difficult to accurately locate the optimum with a small number of probes, especially when the actual search space is large. As introduced in the related works in Subsection 2.1.3, Annealed Particle Filter (APF) can be exploited for searching the optimum point in a high dimensional space. The core of APF is layered processing, where the evaluated particles in one layer will be resampled and propagated to the next layer. From lower to higher layers, the variance of the particles, and correspondingly the actual search space, should be decreasing and all particles should be gradually concentrating on the desired optimum. Towards this end, the particle weights in one layer are just slightly different than the other layers as introduced in Eq. (2.8).

The APF presented in the literature uses the same observation data $\mathbf{z}$ for different layers, by which the exponential weight used in successive layers can exhibit a consistent behaviour. However, the discussion regarding the multiresolution consideration in the previous subsection indicates the observation images with different resolution levels can yield different distinctive power of SR. Therefore, besides smoothing the likelihood distribution, the observation data are also under different smoothing levels in different annealing layers. Instead of using Eq. (2.8), the weight of l-th particle in layer m will be given by

$$w_m^{(l)}(\mathbf{I}_m, \boldsymbol{\theta}_m^{(l)}) = w(\mathbf{I}_m, \boldsymbol{\theta}_m^{(l)})^{\beta_m}, \tag{2.12}$$

where $w_m^{(l)}$ is the unnormalized weight for the l-th particle in the layer m and $\mathbf{I}$ is the observation image. Consequently, the importance resampling in each layer is carried out with the normalized particle weight $\pi_m^{(l)} \propto w_m^{(l)}(\mathbf{I}_m, \boldsymbol{\theta}_m^{(l)})$, where $\sum_{l=1}^{L} \pi_m^{(l)} = 1$.

Determining the Exponent β_m

Although each layer still yields a sharper likelihood distribution than its previous layer, the exponent β_m in the multiresolution APF may not obey the relation $\beta_0 > \beta_1 > \cdots > \beta_M$, where layer 0 is the highest layer. Therefore, the setting of β_m will be essential for effectively choosing a desired number of particles and propagating to the next layer for further processing. Under a pre-specified particle survival rate α_m in one layer, β_m can be estimated with the evaluated particle weights.

The effective number of particles that will be chosen and propagated to the next layer can be measured by the *survival diagnostic* D from [38] as

$$D_m = \left(\sum_{l=1}^{L} (\pi_m^{(l)})^2 \right)^{-1}. \tag{2.13}$$

The particle survival rate α_m in layer m can then be calculated as

$$\alpha_m = \frac{D_m}{L}.$$

As introduced in Eq. (2.12), the particle weight $w_m^{(l)}(\mathbf{I}_m, \boldsymbol{\theta}_m^{(l)})$ in layer m of APF is the exponential output of the original weight evaluated with SR, thus D_m is a function of the exponent β_m as $D_m = D(\beta_m)$. When the particle survival rate α_m is pre-specified, the value of β_m can be obtained by minimizing the following cost function

$$e_\alpha(\beta_m) = D(\beta_m) - \alpha_m L.$$

Combining Eq. (2.13) and Eq. (2.12), the cost function can be reformulated as

$$e_\alpha(\beta_m) = \frac{\left(\sum_{l=1}^{L} (w^{(l)})^{\beta_m} \right)^2}{\sum_{l=1}^{L} (w^{(l)})^{2\beta_m}} - \alpha_m L,$$

where $w^{(l)}$ is the particle weight evaluated with SR. Then the optimization problem can be solved by a gradient based minimizer, e.g. the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.

The Procedure of the Multiresolution APF

The particles in APF are evaluated with the SR framework presented in Subsection 2.3.1. Since different layers use observation images under different smoothing levels, the template matrix in SR should also use the information obtained from the same resolution levels, e.g. the background templates $\mathbf{B}_r$ and the target templates $\mathbf{M}_r$ in resolution level r should be used for its corresponding annealing layer m.[1] The procedure of the multiresolution APF is described in Algorithm 2.2.

The input predicted pose $\boldsymbol{\theta}$ for Algorithm 2.2 can be calculated by using a system state transition model. However, due to the complex motion of the hand-held target for most of the tests, currently, it is simply obtained as the estimated pose from

[1] Although one resolution level can be used in multiple layers of APF, for simplicity, one resolution level only corresponds to one layer in this work. That is, the number of resolution levels equals to the number of the annealing layers, and in this case $r = m$.

Algorithm 2.2 Multiresolution APF with SR weighting

```
Inputs:
  1. Predicted pose $\boldsymbol{\theta}$.
  2. Observation image for current time step.
  3. Background usage count $\mathcal{C}$ for all background templates in all resolution levels.
Procedure:
  1. Generate all hypothesis poses with $\mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\sigma}_F^2)$.
  2. for $r = R - 1 \to 0$ do
     (a) Smooth input image with Gaussian filter; insert $\mathbf{B}_r$ and $\mathbf{M}_r$ into $\mathbf{A}$
     (b) Transform 3D points in $\mathcal{M}_d$ with hypothesis pose of each particle and
         generate $\mathbf{I}$ with grabbed image patch.
     (c) Evaluate weight for each particle with SR.
     (d) Increase background usage count $\mathcal{C}_r$ for $\mathbf{B}_r$; Store all grabbed image
         patches; Store all weights and hypothesis poses.
     (e) If $r \neq 0$, generate samples for next round, i.e. calculate $\beta_r$, perform
         importance resampling, impose $\delta\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_r^2)$ with $\boldsymbol{\sigma}_r = \frac{1}{R+1-r}\boldsymbol{\sigma}_L$ to each
         resampled particle and then perform crossover and mutation operations.
  3. end for
Outputs:
  1. Optimal pose as weighted sum of best samples.
  2. Stored information during tracking process.
```

last frame. Another input $\mathcal{C}$ (a non-negative integer set) contains the usage counts of the background templates for all resolution levels. More specifically, $\mathcal{C}_r$ represents the usage count for a resolution level r and has the same size as the number of background templates in the template matrix. It is utilized for analyzing the importance of each background template. If a background template is seldom used (e.g. $\mathcal{C}_r[n] = 0$ means the n-th background template in resolution level r is never used), it will be updated with a new grabbed background patch, which will be detailed in next subsection.

The initial particles are generated with the Gaussian distribution $\mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\sigma}_F^2)$ which is centered on the predicted pose $\boldsymbol{\theta}$. The variance $\boldsymbol{\sigma}_F$ of the initial particles should be set according to the magnitude of motion in a specific application, so that the particles can effectively cover the search space, where the target pose is expected to reside.

In each resolution level, the background templates $\mathbf{B}_r$ and the target templates $\mathbf{M}_r$ are used in the template matrix $\mathbf{A}$ of SR. The 3D model points in $\mathcal{M}_d$ are transformed under the hypothesis pose of each particle and projected onto the 2D image to sample the image intensity values for evaluating the likelihood of each particle. $\mathcal{M}_d$ contains the down-sampled model points that have passed the visibility test with the predicted pose. Details regarding the down-sampling will be introduced in the next section. The grabbed image patches, the evaluated weights of all particles, as well as the usage count of background templates are stored for further use in Algorithm 2.3 and 2.4 for updating the SR model.

The importance resampling is performed after the weights of all particles are evaluated. The resampled/ survived particles will be imposed with the Gaussian noise $\delta\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_r^2)$ for probing the state space surrounding the hypothesis pose, where the variance $\boldsymbol{\sigma}_r = \frac{1}{R+1-r}\boldsymbol{\sigma}_L$ decreases steadily as the APF proceeds from lower to higher layers. $\boldsymbol{\sigma}_L$ should be set according to the number of layers employed so that the surrounding state space to probe diminishes reasonably from layer to layer. In addition, some genetic approaches can be combined with the particle filter [80] to further prevent sample impoverishment (also as in [124]). In the APF implementation, the crossover and mutation operations, commonly adopted in particle swarm optimization, are performed after the addition of Gaussian noise.

The SR problem for each particle in Step 2 (c) of Algorithm 2.2 is solved with OMP introduced in Subsection 2.2.2. The more sparse, the less computation is required by OMP. The distinctive power of the SR framework can be further capitalized to achieve higher computational efficiency. The maximum sparsity could be set very aggressively to be one in the lower annealing layers. Although this degrades the SR to a template correlation algorithm, it shows reliable performance, because the task in the lower layers is to distinguish the target particles from the background particles. As shown in Fig. 2.5, the most significant reconstruction coefficient suffices for such a recognition task. In the highest layer, where most of the particles have already concentrated around the desired optimal pose, more accuracy will be required. Therefore, a greater sparsity setting should be adopted in the highest layer.

The particle output of all layers in APF can be seen in Fig. 2.6. In this test, as well as in most other tests in this work, a three-layer (also three resolution levels) setting is deployed. The top left figure shows the initial generated particles. With 320 particles, a large image area is covered for dealing with the potential fast target motion. Then as the processing proceeds, the particles gradually concentrate on the desired pose as shown in Fig. 2.6 (b), (c) and (d). The final pose output of APF is displayed in Fig. 2.6 (e). Although it is the output pose in the coarse estimation stage, it is already quite close to the desired pose.[1]

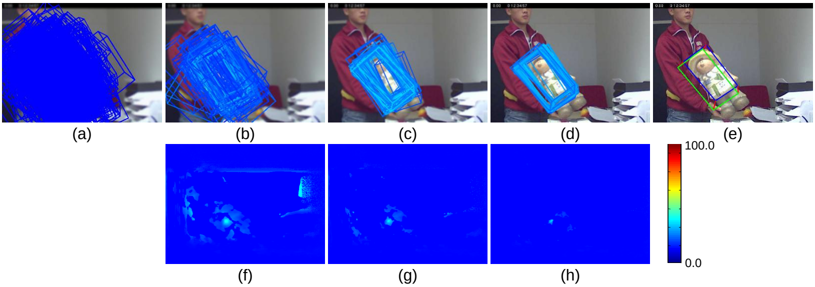


Figure 2.6: Output of each APF layer and the corresponding schematic likelihood image. The images in the first row from left to right illustrate the original scattering of all particles, the resampled particles in each layer and the final output of the coarse estimation stage. The second row shows the 2D schematic likelihood images for their above APF layers. The colors of each pixel in the likelihood images are encoded with the color bar at bottom right, where dark blue represents low likelihood. The color bar also defines the particle weights in figure (b), (c) and (d).

The figures in the second row of Fig. 2.6 illustrate the schematic likelihood maps for their above observation images under a certain resolution level. Each pixel in the image represents a hypothesis pose, under which the mass center of the target is projected onto this point. The likelihood values are encoded with color and calculated from Eq. (2.12) with $A = 100$ and $B = 10$, where 100 means a perfect match between the target model data and the observation data. Due to image and target pose variations between the initialization frame and the current frame, the importance values for a potential target region are typically between 15∼30. In comparison, the background regions usually have importance values below 0.005 (when no target templates are used in OMP procedure, $w(\mathbf{I}, \boldsymbol{\theta}) = 100 \cdot exp(-10) = 0.0045$). The distinction between the target and non-target is then quite obvious. Notice that in the lower resolution levels, more false positive could be produced. For example, the right edge area in the bottom left image of Fig. 2.6 can indicate potential target in this region. This is mostly caused by two reasons: first, the white wall and the black letter basket can exhibit similar overall pattern as the target appearance with a light body and dark leg when the image is smoothed significantly in the lowest resolution

[1] The feet of the dwarf is excluded from the target model, because the instep of the feet is almost parallel to the optical axis of the TOF camera and the color is dark, for which the PMD camera cannot produce reliable measurements.

level; second, this region is far away from the current target position. Therefore, the background information will not be updated nor modeled into the background templates (details regarding the update rules will be introduced in the next subsection). Nevertheless, as APF comes to the higher resolution levels, this region is determined as the background when more detailed texture information become available.

It can be seen that the high likelihood region shrinks from lower to higher layers. Especially in the highest layer (Fig. 2.6 (h)), almost only the pixel positions corresponding to the target center have high likelihood values. As pointed out in [167], when conventional template matching methods are considered, the distinction between the target and non-target regions in the likelihood image will be tiny, i.e. nowhere will be in dark blue. In consequence, the particle convergence through layers can be trapped into the local minimum on some background regions.

On the other hand, the likelihood image in the highest layer also shows the necessity for adopting the multiresolution strategy. With the proposed SR framework, only a small target region exhibits salient likelihoods in the highest resolution level. If only a small number of particles are available and the search space is quite large, the probability can be very low that some of the particles are correctly positioned on the small target region that has high likelihoods. In comparison, the likelihood image for the lowest resolution level (Fig. 2.6 (f)) shows the desired characteristics, where a large area surrounding the target region yields high likelihoods, while most of the background regions still gives low likelihoods.

It should be noted that it is impractical to evaluate the likelihood distribution for a full 6DOF state space, meanwhile it is impossible to visualize the 6DOF space either. Therefore, only the 2D schematic likelihood images are given, which are obtained by evaluating a set of image patches grabbed with the translated 3D model points.

2.3.3 Online Update Rules

During tracking, the background can vary dramatically and be significantly different from the background templates modeled in the initialization stage. Besides, the illumination condition for a frame may be quite different than the initial lighting condition, which can bring about remarkable target appearance variations. Furthermore, the target pose variations will cause the visibility changes of target surface points. The model points invisible to the camera should not be used for visual tracking. Therefore, the information used for tracking needs to be updated to accommodate to these changes for achieving robustness in a dynamic scenario. This subsection presents some update rules that can improve the performance of the proposed coarse estimation algorithm. This part provides the updates due to visibility changes and focuses on some rules for updating the background model. The update rules for the illumination model will be introduced in Subsection 4.4.1.

Updating Model Visible Points Set

The model points used for tracking in a frame should be visible to the camera under the target pose in this frame. However, before the pose has been estimated with the visible model information, it will not be known a priori. Therefore, the target pose in current frame t has to be predicted for obtaining a reasonable visible model point set. In this work, the predicted pose is approximated with the estimated pose $\boldsymbol{\theta}_{t-1}$ from last frame $t-1$. And the visible point set $\mathcal{M}_z$ is obtained by first transforming all model points in $\mathcal{M}$ with the pose parameter $\boldsymbol{\theta}_{t-1}$ then projecting all transformed points onto the image plane and performing z-buffering for the visibility test.

Furthermore, as previously discussed, one drawback of adopting the wavelet basis is the restriction on the dimension of the template matrix, i.e. the rows of which must be a dyadic number 2^p, which implies the number of model points used in SR should also be 2^p. This will require a down-sampling step from $\mathcal{M}_z$ to $\mathcal{M}_d$. $\mathcal{M}_d$ contains the down-sampled model points, and should also be updated, because some of its points will be invisible under a new target pose. The initial $\mathcal{M}_d$ is obtained through randomly (with uniform distribution) taking 2^p points from $\mathcal{M}_z$ (the initial $\mathcal{M}_z = \mathcal{M}$), and the selected points are organized in the same order (denoted as the pixel order) as they will be read from the image, e.g. from bottom left to top right in the image. In this way, the modeling power of the wavelet bases can be made most of. When in a frame, some points in $\mathcal{M}_d$ are not visible any more under the predicted pose, these points will be replaced by points from $\mathcal{M}_z$. Although it will be better that in any frame, the $\mathcal{M}_d$ can be organized (as well as reorganizing the background and target templates in the template matrix) in pixel order to guarantee the effectiveness of the wavelet bases, current tests are only performed on the object initialized with the frontal side, for which the influence of model points' ordering changes is less remarkable. Therefore, currently, the invisible points in $\mathcal{M}_d$ are replaced by points randomly selected from the points belonging to $\mathcal{M}_z$ but not to $\mathcal{M}_d$.

Background Update Phase I

The target update scheme in [103] and [83] can take into account the target appearance variations during tracking. They used the estimated target position to grab the target online observation information and incorporated the grabbed data into the model. However, the estimated pose will never be perfect, sometimes can even be very inaccurate. The grabbed data will inevitably contain some background data. Especially when the tracking fails for one frame and the target position is wrongly estimated to be on a background region, the background will be updated into the target templates. All these artefacts can gradually contaminate the target model and lead to tracking failure in a long time run. Thanks to the SR framework for being able to model the target and the background simultaneously, here another strategy is presented to update the background model instead of the target model. Background here is defined as the region which is a certain distance away from the estimated target position. In this way, inaccuracy in tracking will not influence the quality of the model. The update procedure is described in Algorithm 2.3.

Algorithm 2.3 Background Update phase I

```
Inputs:
  1. Estimated pose.
  2. Hypothesis poses, weights and grabbed image patches for all samples in all
     resolution levels.
  3. Usage count $\mathcal{C}$ for all background templates in all resolution levels.
Procedure:
  1. for $r = R - 1 \to 0$ do                                  ▷ run in parallel
     (a) Select all $n_b$ background templates $\mathbf{B}_{r,nb}$ that have usage count $= 0$
     (b) Select all $n_s$ image patches $\mathbf{I}_{r,ns}$ from samples that have weight $> w_\tau$ and
         image distance $> d_\tau$.
     (c) $n_u = mina(n_b, n_s)$, randomly select $n_u$ grabbed image patches from $\mathbf{I}_{r,ns}$
         to replace $n_u$ background templates from $\mathbf{B}_{r,nb}$.
     (d) Decrease all background usage count $\mathcal{C}_r$ by $h$, and set usage counts for all
         newly updated background templates to be $5h$.
  2. end for
Output:
  1. Updated background templates $\mathbf{B}_r$
  2. Usage count $\mathcal{C}_r$ for each resolution level.
```

The Inputs 2 and 3 in Algorithm 2.3 are the outputs from Algorithm 2.2. The Input 1 is the estimated pose from the accurate estimation stage. Since the updates for each resolution level are independent, they can be run in parallel to take advantage of a multi-core CPU.

After an accurate target pose has been estimated, the background region as well as the particles on the background region for each resolution level can be determined. Among these background particles, some are assigned with high weights during the evaluation in APF, which indicates that these particles, or equivalently the corresponding background regions, cannot be effectively handled by current SR template matrix. Therefore, the background image patches should be incorporated into the template matrix. However, only the background templates in a resolution level that are rarely used (e.g. $\mathcal{C}_r[n] = 0$) will be replaced by the new background patches.

After the background update, the usage count for a newly updated background template will set to be $5h$, and the counts for the remaining background templates will be decreased by h. This guarantees the new background templates will have a life time of at least 5 frames. The usage count will be increased by one each time the OMP selects the corresponding background template as the most correlated atom in the template matrix to the current residual vector. Therefore, the positive integer h determines how fast the importance of a background template drops with time.

The effect of the above update scheme is illustrated in Fig. 2.7. The resampled particles for two consecutive frames in the lowest resolution level are shown in Fig. 2.7(a) and (b). Similar to the likelihood images in Fig. 2.6, Fig. 2.7(c) and (d) are the 2D schematic likelihood images for their above observation images.

In Fig. 2.7(a), some background particles are resampled, which correspond to the high likelihood region in the red dashed box in Fig. 2.7(c). These particles are drawn in light cyan because they are assigned with high weights. Meanwhile, they are far away from the true target position, thus they will be integrated into the background templates according to the update rule in Algorithm 2.3. After the update, the high likelihood area in the same background region has remarkably reduced in the next frame as shown in Fig. 2.7(d). Correspondingly, no particles are resampled again at the same spot. On the other hand, some particles slightly above the target are also resampled. Whether to incorporate these particles into the background templates will depend on their distance to the estimated target position. Besides, the changes of the target pose can also bring about image appearance variations. In consequence, in Fig. 2.7, even if the background remains largely constant between the two frames, some new salient area appears on the right side of the target due to the change of target pose.

Background Update Phase II

Each time the target has been successfully tracked, the invisible points in $\mathcal{M}_d$ under the estimated pose will be replaced by some new target visible surface points from $\mathcal{M}_z$.

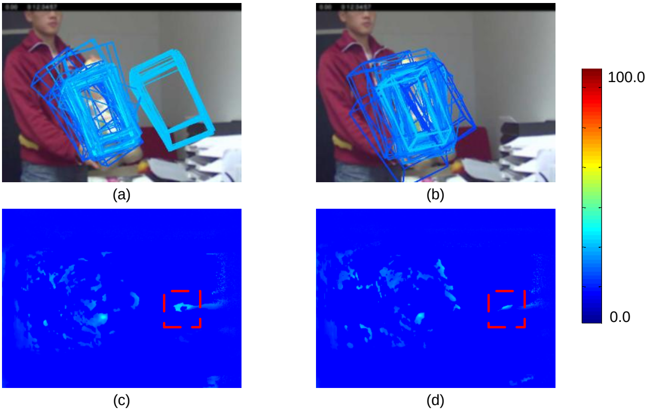


Figure 2.7: Background update result in the lowest resolution level. The images in the first row show the resampled particles of two consecutive frames in the lowest resolution level. The second row shows the schematic likelihood images for their above observation images. The region in the red dashed box depicts the reduction of the high likelihood area after background update. The likelihood values for all pixels in the importance image as well as for the resampled particles are encoded with the color bar on the left side.

This will result in a slight change of $\mathcal{M}_d$ from frame to frame. The background update phase presented in Algorithm 2.3 can only take into account the down-sampled point set $\mathcal{M}_d$ up to the most recent processed frame. The accumulated background templates cannot completely model the new integrated target model points, because they are image patches grabbed with points from previous $\mathcal{M}_d$. Therefore, another update is proposed, the Background Update Phase II, for incorporating the background patches with the most recent $\mathcal{M}_d$.

It will be ideal, if the background templates can be directly grabbed from the background region of the frame to process. Unfortunately, before the target pose has been estimated, the background region cannot be determined. However, the background usually dose not change much between two consecutive frames. This implies the background templates grabbed in the last frame can also effectively model the background in the current frame. Therefore, the Update Phase II is performed with the most recent $\mathcal{M}_d$ but on the observation image in the last frame.

As the discussion for Update Phase I, only the background regions that cannot be well handled should be updated into the background templates, which correspond to the particles on the background region and with high likelihood values. The demand on the likelihood values of particles with new $\mathcal{M}_d$ will require an evaluation stage. The background image patch can be obtained through generating a particle and evaluating the likelihood, until the required number of qualified background patches are collected. However, this scheme can be time consuming, because the particles have to be sequentially processed thus no GPU acceleration can be exploited. Furthermore, in the worst case it can require a huge number of particles to obtain the desired number of background particles with high likelihood values. Therefore, another scheme is proposed, where the update is performed after the particles are generated with APF for the new frame (i.e. the Step 1 of Algorithm 2.2). The generated particles will be evaluated on the just processed frame, where the background region can be determined. Then the qualified image patches will be considered for integration into the background templates. If no background particles yield high likelihoods, it indicates most probably the background can be well modeled by the current model, and no update is required. The procedure is detailed in Algorithm 2.4.

Algorithm 2.4 Background Update Phase II

```
Inputs:
  1. Estimated pose and its corresponding observation image.
  2. Hypothesis pose for all samples.
  3. Usage count for all background templates in the lowest resolution level $R-1$.
Procedure:                                   ▷ only in the lowest resolution level
  1. Transform 3D points in $\mathcal{M}_d$ with hypothesis pose of each particle, project onto
     2D image and grab image patches $\mathbf{I}$, store the projected 2D pixel sets of all
     particles.
  2. Evaluate weight for each particle with SR.
  3. Decrease usage count by $h$.
  4. Perform the update step 1 to 3 in Algorithm 2.2 but only for the lowest
     resolution level.
  5. Set usage count to be $5h$ for newly updated background template atoms.
Outputs:
  1. Updated background templates $\mathbf{B}_{R-1}$ and corresponding usage count.
  2. Projected 2D pixel sets of all particles.
```

The Output 2 of the above update can be reused in Step 2 (b) of Algorithm 2.2 for grabbing image patches, because the pixel positions for each particle are obtained through projecting the points from new $\mathcal{M}_d$ and transformed under the hypothesis pose of each particle in APF for the new frame. Different from the Update Phase I, the Phase II only updates the background templates in the lowest resolution level. One reason is for the consideration of the computational cost, because under the above update scheme, only the outputs in the lowest resolution level can be reused. Another reason is due to the different tasks different resolution levels aim to solve. The major task in the lowest resolution level is to identify particles on the target region from those on the background region, while other resolution levels aim to achieve a more accurate target pose. This indicates the lowest resolution level will raise a higher demand on the distinctive power of the template matrix. Therefore, the second update stage is only applied on the lowest level.

Besides efficiency, another advantage can be obtained with the update strategy in Phase II. When the background changes mildly between consecutive frames, because the hypothesis poses of all particles are the same between the lowest level in the new frame and the Update Phase II with the last frame, and the difficult particles have already been updated into background templates, in the lowest annealing layer for the new frame, only the particles on the target region will have high likelihoods. This implies the template matrix of SR will be more effective on distinguishing between the target and the background particles.

One possible drawback of such an update rules is that probably only the lowest resolution level is under active update, while the other levels will keep unchanged most of the time. This is result from the constraints in Step 1 (b) of Update Phase I, where a particle will be qualified for update only when it has high likelihood and meanwhile is a certain distance away from the estimated target position. However, as can be observed, in most cases, few particles on the background region could pass the first two layers and still have high likelihoods. Therefore, almost no candidate particles can be qualified for updating into the model in the higher resolution levels. One solution could be to use the candidate particles in low resolution levels also for updating the background model in higher levels, e.g. simply applying the qualified particles in Step 4 of Update Phase II to all resolution levels.

2.4 Some Implementation Issues

This section describes some details for implementing the algorithm presented in this chapter. First the choice on the state vector for representing the 6DOF pose will be discussed, because the pose representation, especially the rotation representation will influence the performance of the pose estimation algorithm. Then the parallelism of the algorithm will be investigated and the computational efficiency under the GPU acceleration will be given, showing the capability of the algorithm for real-time applications. In the end, some important parameters in the algorithm are introduced. The performance of the algorithm can be tuned through configuring these parameters.

2.4.1 Choice on State Space

The pose of a rigid object has 6 degrees of freedom, three for rotation and three for translation. The rotation part can be represented in many ways, e.g. with rotation matrix, unit quaternion, axis-angle, Euler angle, etc. When a pose estimation algorithm is developed with a specific representation, the question often arises, why this representation?

The rotation matrix can be directly applied for rotating a point in the 3D Cartesian coordinates or transforming a point in the 4D homogeneous coordinates when combined with a translation vector. However, it is seldom used as the rotation expression in the estimation or optimization of the pose due to its high dimensionality.

Quaternion provides a simple representation for a rigid pose and is robust to singularity. With quaternion, the pose is described as $\mathbf{q} = [\mathbf{q}_R^\top, \mathbf{q}_T^\top]^\top$ [13], with the 3×1 vector $\mathbf{q}_T$ describing the translation and the unit quaternion $\mathbf{q}_R = [q_0, q_1, q_2, q_3]^\top$ describing the rotation, where $[q_1, q_2, q_3]^\top$ represents the rotation axis and $\alpha = 2arccos(q_0)$ is the rotation angle. In APF, the particles are propagated from layer to layer with the predefined covariance. However, when this is performed on the pose represented by quaternion, the unit constraint $\|\mathbf{q}\|_2 = 1$ requires a normalization after the covariance are incorporated. This will cause intricate effects on the volume of the desired state space to probe, which means, the predefined covariance will not be able to effectively reflect the desired rotation on the state vector due to the normalization.

Another representation, the axis-angle representation (or axis-azimuth as in [144]), has a similar structure as quaternion, yet requires no normalization because the rotation axis and rotation angle are directly described in angle, and any imposed covariance will have straight forward effects on the state space. However, another problem appears and prevents the use of the axis-angle like representation in the particle filter based algorithms. The quaternion or the axis-angle are extremely effective for modeling the motion of a spinning spacecraft. In the gradient descent based optimization methods, these representations, due to free of singularity, can produce a reliable result because the pose is obtained through minimization of a cost function. In contrary, the particle filter based algorithms generate a number of hypothesis samples, and the optimal pose is retrieved by evaluating some similarity metric for all samples. When the object has a large rotation other than simple spinning, the rotation axis will have to vary between 0° and 90° in a 2D state space. Not only the span, what's worse, a small rotation in yaw or pitch will be reflected by a big change in the axis direction. In such a case, it is very hard to generate a small set of samples that can effectively model these motions under the axis-angle representation.

The problems mentioned above can be solved if Euler angle is adopted as the rotation representation. Contrary to the indirect values in quaternion, it uses angles, which have no normalization problem. And the Euler angle explicitly introduces rotation around three axes, which provides an efficient mechanism to model the rotation and thus requires only a small state space for a mild rotation in any manner. One well known drawback of the Euler angle is the singularity problem, aka the *gimbal lock*, which refers to when an angle reaches infinitesimally close to 0° or 90° (depends on the rotation convention), the other two degrees of freedom degenerate to one degree. The optimization problem will be ill-posed around the singularity angle.

Since the singularity only happens on specific angles, one solution is to use an incremental pose which is expected to avoid the neighborhood of the singular angle as in [118]. The output of the optimization process will thus be an incremental pose, which can be fused into the reference pose through the use of the transformation matrix to get a global pose. We also adopted this strategy in the pose optimization with the Textured-ICP algorithm proposed in Chapter 3. The same idea can as well be applied in the annealed particle filter used in this chapter. However, since currently no full target model with the 3D geometry and texture is available for our tests, the target will not rotate as much as 90° in any direction, because otherwise the known side of the target will largely disappear and the pose estimation will fail. This implies by taking an appropriate rotation convention for the Euler angle representation, the singularity will not happen for our current tests. Meanwhile, the in-plane rotation (with the rotation axis perpendicular to the image plane) brings about the most significant image variations than the rotations around other axes. Therefore, the hypothesis sample generation should take this into account and be able to effectively model these variations. To this end, the Euler angle with the ZYX convention is adopted, where the singularity will only happen when the rotation around the Y axis gets infinitesimally close to 90°.

The choice on the rotation center will not have impact on the rotation, rather, it will influence the translation parameters. One widely adopted rotation center is the origin of the coordinate system. This leads to a representation of the complete transformation as one simple matrix multiplication $\mathbf{v}' = [\mathbf{R}, \mathbf{t}]\dot{\mathbf{v}}$, where $\mathbf{R}$ is the 3×3 rotation matrix and $\mathbf{t}$ stands for the 3×1 translation vector. This requires translating all the original target 3D world points back to the origin of the coordinate system with the translation vector specified by the mass center of all the original target world points. In this case, any changes in the translation $\mathbf{t}$ reflect the real motion in the physical world. This will yield a simple implementation for the annealed particle filter, because a priori assumption of the target speed can be effectively applied, i.e. by setting the variance of the pose in each layer. In a word, the 3D model points in Eq. (1.1) are the initial target 3D coordinates translated to the system origin with the mass center. This concludes the reason for the choice on state space in this thesis.

2.4.2 GPU Acceleration

Many recent researches on General Purpose GPU (GPGPU) computing [2, 53] have pointed out that when the parallelism of a specific algorithm can be exploited, magnitudes of acceleration can be achieved when the parallel calculation is properly implemented on modern many-core GPUs. This subsection investigates the parallelism of the coarse pose estimation algorithm and provides the running time under a typical algorithm configuration with the GPU acceleration.

Parallelism of the Coarse Estimation Algorithm

The coarse pose estimation algorithm presented in this chapter is based on the Annealed Particle Filter (APF) framework. The particle filter [63], as well as the APF [81] are known as the ideal algorithms for a GPU implementation because of the independent evaluation of all particles. More specifically, in the proposed algorithm, all particles are evaluated with SR solved by OMP. In one iteration of the OMP optimization (see Algorithm 2.1), the most correlated atom in the compressed template matrix $\mathbf{\Phi A}$ to the current residual vector $\mathbf{\Phi e}_i$ for the i-th particle can be obtained through a matrix-vector multiplication $(\mathbf{\Phi A})^\top \mathbf{\Phi e}_i$. Furthermore, when the residual vectors of all particles are compiled into one matrix, the most correlated atoms of all particles can be calculated through a large scale matrix-matrix multiplication $(\mathbf{\Phi A})^\top \mathbf{\Phi}[\mathbf{e}_1, \cdots, \mathbf{e}_N]$, which can be solved with extremely high efficiency on a modern many-core GPGPU.

Another time consuming operation is the transformation and the projection of all down-sampled model points for obtaining the image values under the hypothesis poses of all particles. The transformation of all 3D points can be expressed as a matrix-

matrix multiplication. The down-sampled point set is the same for all particles, only the transformation matrices are different. Similar to the above processing for OMP, when the transformation matrices for all particles are compiled into one big matrix, the transformation of all down-sampled target points for all particles can be done with one matrix-matrix multiplication

$$\begin{bmatrix} \mathbf{U}_1' \\ \vdots \\ \mathbf{U}_N' \end{bmatrix} = \begin{bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_N \end{bmatrix} \dot{\mathbf{U}}_d^{init},$$

where $\dot{\mathbf{U}}_d^{init}$ is a 4×2^p matrix containing the homogeneous coordinates of all down-sampled points from $\mathcal{M}_d$ in each of its columns. $\mathbf{U}_i'$ is the 3×2^p matrix containing all 2^p transformed down-sampled points for the i-th particle. $\mathbf{T}_i$ is the 3×4 transformation matrix formed from the hypothesis pose of the i-th particle. On the other hand, after the transformation, the projection is a pin-hole camera based perspective non-linear operation, it has to be implemented with the self-written GPU kernels.

The calculation of the cost function values as well as its derivatives for all particles required for determining the β_m in APF are also implemented with GPU kernels. However, since only a moderate number of particles are used in the APF (thanks to the multi-layered processing), the acceleration of these calculations achieved with GPU is not remarkable.

Overlap between CPU and GPU

If the parallel calculations are simply implemented on the GPU, when the GPU is busy, the CPU will be idle and waiting for the results from the GPU. Likewise, when the CPU is doing calculation, the GPU will be idle. This can result in a great waste of the computing resources the hardware can provide. For example, although the correlation between the atoms in $\mathbf{\Phi A}$ and the residuals is performed on GPU, the operation for maintaining the QR decomposition (see Subsection 2.2.2) will be more appropriate if run on CPU. Under such the above configuration, in each iteration of Algorithm 2.1, the CPU needs to wait for the result of the most correlated atom selection on the GPU and the GPU has to wait for the new residual vectors for calculating $(\mathbf{\Phi A})^\top \mathbf{\Phi}[\mathbf{e}_1, \cdots, \mathbf{e}_N]$.

By a careful implementation, the above issue can be solved through increasing the overlap between CPU and GPU operation periods. Still taking the OMP procedure as an example, if the particles are divided into two groups, when the correlation calculation is being performed on GPU for one group, the residual calculation and the maintenance of the QR decomposition can be conducted on CPU for another group. In this way, both computing units can be kept occupied by calculations. Thus the usage of the computing resources are increased. This strategy will be most helpful when the number of particles or the amount of calculations exceeds the computing capability of the GPU and dividing the calculations into groups does not decrease the GPU efficiency.

Running Time with GPU Acceleration

The algorithm is implemented with C++ for the CPU part and CUDA for the GPU part. The running time is tested on a gaming laptop with a 4-core CPU and a 336-CUDA-core GPU. The result as well as the configurations for the test are listed in Table 2.1.

Coarse estimate stage	
Number of APF layers	3
Smoothing kernel size	(3,9,15)
Max. sparsity	(15,1,1)
Number of particles	300
Size of $\mathcal{M}_d$	$2^p = 1024$
Rows of $\mathbf{\Phi}$	$d_0 = 100$
Running time	$45 \sim 55$ ms

Table 2.1: Running time of the coarse estimate stage.

The APF has three annealing layers, the maximum sparsity in the low and the intermediate resolution levels are set aggressively to one, while in the highest resolution level it is set to 15. With 300 particles, the algorithm can be performed with $45 \sim 55$ ms. As previously discussed, some calculations are performed on CPU, some are on GPU. It is inevitable that the data need to be transferred back and forth between both computing units. It has been observed that in quite some case, the data transfer takes more time than the calculation itself. Since the testing GPU has a PCI-E 2.0 interface, when a GPU with PCI-E 3.0 can be used, which doubles the bandwidth of PCI-E 2.0, a non-negligible amount of data transfer time can be saved. Furthermore, if more recent GPUs with thousand of CUDA cores can be exploited, the running time can still be reduced significantly. Therefore, it can be concluded the coarse pose estimation algorithm is competent for real-time applications.

2.4.3 Important Parameters

The proposed coarse estimation algorithm aims at dealing with the fast or mild motion of various targets. The background can be highly cluttered, the illumination condition can vary significantly (as will be discussed in Chapter 4). The capability or versatility of the algorithm is gained at the cost of the requirement for tuning some parameters for different application scenarios. Although automatic parameter tuning is more desired, it is not researched in this work. This subsection lists some of the most important parameters for the algorithm. When the tracking exhibits an unstable performance, these parameters, when appropriately configured, may make a difference. In the code, the parameters are specified in the header file "./Src4PoseEst/config.h".

The threshold for initializing and updating the background templates. In the proposed sparse representation framework for tracking, the most important principle is the balance between the non-target and the target templates, so that the image patches on the target region can be effectively separated from those on the background region. The threshold likelihood w_τ in Step 1 (b) of Algorithm 2.3 controls which and how many background patches will be integrated into the background templates. Especially at the initialization stage, if this threshold is set too low, a lot of image patches that can already be handled by the current template matrix will be considered as the candidates for integrating into the background templates. This can reduce the chances for integrating more important image patches. On the other hand, if it is set too high, not enough background templates can be accumulated and thus the ability of the non-target templates will be weakened. This threshold is scene dependent. It should always be checked whether s sufficient but not excessive amount of the background templates are considered as the valid candidates for update.

The strength of the occlusion blocks. As depicted in Fig. 2.3, in the highest resolution level, the occlusion blocks are not as strong as in the other resolution levels, because most of the particles should have already concentrated on the target region and the diluted occlusion blocks can be more helpful for determining the accurate pose. Likewise, if it can be expected that the occlusion does not happen much and most of the particles can converge to the target region after the processing in the lowest layer, the occlusion blocks in the intermediate resolution level can also be set diluted. This will bring about an increased distinctive power on the intermediate level, i.e. the probability will be higher that the background particles survived from the lowest resolution level can be wiped out in the intermediate level with the diluted occlusion blocks in the occlusion templates.

The number of particles. The number of particles will influence the capability for handling the large inter-frame motion and the computational efficiency. If the inter-frame motion is expected to be significant, more particles will be required. But using more particles will also raise higher computation cost. The setting should consider the computing resources available. For example, if the GPU has 336 CUDA cores, 200 particles or 300 particles will not cause much difference from the perspective of the GPU's computing power. For most tests reported in this thesis, 300 particles were used.

The input variance for each annealing layer. When dealing with a large inter-frame motion, besides increasing the number of particles, the volume of the state space that the particles can effectively cover should also be increased. This is done by tuning the input variance for each layer in the APF, i.e. $\boldsymbol{\sigma}_F$ and $\boldsymbol{\sigma}_L$ in Algorithm 2.2.

The weight threshold for the final output. The weight threshold for the final pose output determines the fidelity of the coarsely estimated pose. With the terminology from ROC (Receiver operating characteristic) analysis, a higher threshold value will bring about a better true positive estimation. However, it can also result in a higher false negative result. This means, once an estimation result has passed the threshold, with high probability it is a correct estimation for the target pose. Meanwhile, it will more easily report target tracking failure. Lower thresholds will have an opposite effect. The target will less likely be lost, but it can happen that an incorrect target pose is determined as a valid coarse pose. For robotic applications, relatively higher values are recommended, because it is better to lose the target than to track a wrong one. As described in the algorithm overall workflow in Fig. 1.1, when the coarse pose estimation fails, it will not come to the accurate pose estimation stage, and some failure handling can be performed.

2.5 Summary

The coarse pose estimation algorithm is presented in this chapter, which takes the target colored point cloud and the observation color image as the input and outputs a coarse pose estimation that can be further refined in a gradient-based accurate pose estimation stage. The proposed algorithm is based on Annealed Particle Filter (APF) for dealing with the high dimensional state space of the 6DOF problem, and each particle is evaluated with Sparse Representation (SR). The major innovation towards state-of-the-art approaches for coarse estimation or tracking can be summarized as

- A new and flexible composition of the template matrix in SR is proposed, which can better distinguish the image patches grabbed on the target region from those on the backgrounds.

- Multiresolution strategy is adopted in APF, which can further harnessing the distinctive power of the proposed SR for the 6DOF tracking problem.
- Several online update rules are discussed for accommodating the changes during tracking. Compared to the rules in the literature for updating target information, the methods proposed in this chapter update the non-target information instead. Thus the inaccuracies in tracking will not be accumulated and propagated from frame to frame.
- The major computations in this chapter are implemented with GPU acceleration and real-time performance is achieved.

The large inter-frame motion of the target is handled in the annealed particle filter based coarse estimation stage. However, since this stage only employs the 2D appearance information, it can be very difficult to accurately estimate the complete six degrees of freedom, especially with a small number of particles. Therefore, another gradient decent based pose refinement stage is introduced using the fused data from the range and the color cameras. This is described in the next chapter.

Chapter 3

Accurate Estimation with Textured-ICP

This chapter introduces the accurate pose estimation algorithm extended from the conventional Iterative Closest Point (ICP) algorithm for point clouds alignment. The conventional range data based ICP cannot deal with geometrically symmetric objects and can perform poorly when the range data is corrupted. The proposed method is based upon the ICP framework but also takes the object appearance into account. The combination of the range and the texture information makes the pose estimation robust to the range artefacts common for current TOF cameras. Meanwhile, it can also tackle the geometrically symmetric but sufficiently textured objects, thus improves the capability of ICP. After the problem to be addressed is clarified, the related works regarding the alignment between the measurement and the model, the methods for the accurate pose estimation and some major variants of ICP will be discussed. Then details will be given for the conventional ICP with the projective data association and point-to-plane error metric. The extension for obtaining the proposed Textured-ICP is then described. Some experiments are conducted to evaluate the convergence of the proposed algorithm. Some implementation issues regarding the surface normal estimation and the GPU acceleration will be discussed. In the end, the major points in this chapter will be summarized.

3.1 Problem Statement and Related Works

This section discusses the accurate pose estimation problem to be solved in this chapter, where the coarse pose output from the previous chapter will be refined by the proposed Textured-ICP. Some state-of-the-art researches that are related to the range measurements registration and the pose estimation problems will also be introduced.

3.1.1 Problem Statement and Contributions

Previous chapter presents a 2D data based algorithm that can track a fast moving rigid object and provide a coarse 6DOF pose estimate. This chapter will propose a range data based estimation stage that can further refine the coarse pose and obtain an accurate pose estimate. In this field, ICP is the most prevalent method for point clouds alignment. However, range data based methods have the limitation that they cannot be applied on the symmetric geometry due to lack of constraints. Fig. 3.1 shows a 2D schematic illustration for the conventional ICP with projective data association on the geometrically symmetric measurements.

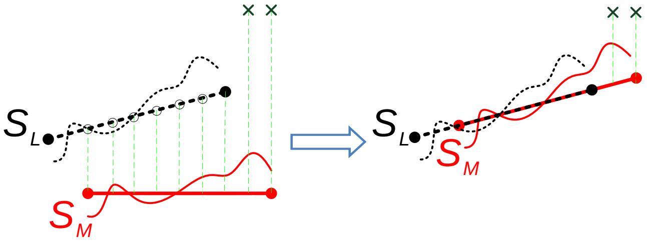


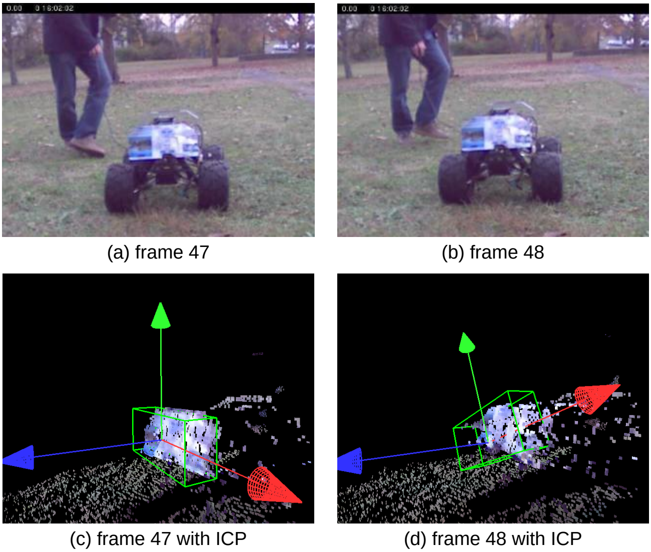
Figure 3.1: Conventional ICP on symmetric geometry. The solid section represents the geometrically symmetric range data, while the dashed curve stands for the intensity distribution of all points on the section. The model data is in red and the live observation data is in black. The matched points on the live data found by the projective data association are marked with small circles, while the none-correspondence projective rays (in green dash) are ended with crosses. The registration result with ICP is shown on the right.

The red line section in Fig. 3.1 represents the model range data, while the black dashed section is the live range measurement. The red and the black curves are the spatial intensity distributions for all points on the model and the live measurements respectively. The green dashed lines can be interpreted as the projecting rays coming from the principle point of the camera lens and passing through each pixel on the image plane. The projective data association projects each model point onto the image plane and finds its correspondence point on the live measurement along the projective ray. When the initial alignment between the model and the live data is not close enough or when some of the range measurements are marked invalid, some of the model points will not be able to find correspondence points, as marked by crosses in Fig. 3.1. All matched correspondence pairs will be used in the cost function calculation. Under such a configuration, the output of the final alignment will look similar to the right image in Fig. 3.1. Obviously, the result fails on some degrees of freedom. However, from the perspective of the cost function, the alignment is already optimal, because all matched point pairs are already perfectly aligned.

The misalignment in Fig. 3.1 is caused by two major reasons. First, not all model points can find correspondence points under projective data association. With other association techniques, e.g. the nearest point searching scheme for unorganized data [13], better matching may be achievable with the price of much higher computation load. However, when another structure appears nearby, the nearest point searching scheme can run into problem. Another reason for the misalignment is because conventional ICP simply ignores the intensity distribution. As shown in the right image of Fig. 3.1, when the range data for all matched point pairs are perfectly aligned, their intensity distributions do not agree with each other.

Besides the restriction on geometrically symmetrical surfaces, conventional ICP also has problem when the range data is corrupted, which is common for current TOF cameras. Fig. 3.2 shows an example for the conventional ICP on noisy range measurements. The images were grabbed with the fused cameras mounted on a mobile robot driving on grassland, where the uneven ground caused remarkable motion artefacts on the range measurements in some frames. The left tow images in Fig. 3.2 show the observation color image and the estimated target pose when the range data are in good quality. Whereas the left two images are from the successive frame, where the range data is corrupted and the pose estimation with ICP fails on some degrees of freedom. However, although the range data on the robot exhibit quite different quality between the two consecutive frames, their appearances remain largely unchanged. This can lead to the implication that when the target appearance is incorporated into the range data based ICP, the influence of the noisy range data may be mitigated or compensated.

This chapter aims at solving the accurate 6DOF pose estimation problem when a coarse pose estimate is available. The input range data are obtained from the PMD camera and can be corrupted by motion artefacts since the algorithm is desired to be applied on a fast moving object. Furthermore, the target object may be geometrically symmetric, e.g. a planar object. Based on the above discussions, the target appearance (texture) information is combined with the range data to achieve a robust



(a) frame 47

(b) frame 48

(c) frame 47 with ICP

(d) frame 48 with ICP

Figure 3.2: Conventional ICP on noisy range data. The images (a) and (b) are the color images in two consecutive frames captured by the AXIS camera. The images (c) and (d) depict the corresponding fused RGBD images overlaid with the bounding boxes for the pose estimated by ICP. The fusion images are rotated for better visual illustration.

pose estimation algorithm. The major contribution in this chapter is the proposed Textured-ICP algorithm, which extends the conventional ICP by incorporating the texture into the ICP framework. The proposed method, as demonstrated in this thesis, is capable of dealing with geometrically symmetric objects as well as noisy range measurements. Meanwhile, the accurate pose estimation algorithm is implemented with GPU acceleration, by which the real-time performance is achieved.

3.1.2 Methods for 3D Registration and Pose Estimation

In this section, some methods appeared in the literature for point sets registration or 3D pose estimation are discussed. These methods are categorized into two groups: the methods using 2D visual observation data and the methods using 3D range data. Since computational efficiency is of significance for real-time applications as in the work of this thesis, only online approaches are considered and introduced.

The 2D/3D Registration

Despite the ambiguities in 3D problems using 2D visual information, under mild assumptions, some approaches have been presented with great success. [122] made use of the *epipolar constraints* for matching FAST corner features between frames. The *essential matrix* between the current frame and a previous keyframe was calculated for determining the relative pose up to a scalar. Based on the estimated pose, they also provided an online 3D model reconstruction with an interactive efficiency. Or when multiple cameras can be used, the feature extraction can be used for online 3D map building [116]. When the target 3D model or a set of registered keyframes are available, the 2D-3D feature correspondences can be established and used for determining the pose [158]. Besides corner features, [59] adopted the SIFT features extracted from a pair of stereo cameras. But since they used the range information constructed from the stereo vision, their method falls more into the 3D/3D registration. [33] also used SIFT feature for estimating poses of multiple targets with an iterative feature clustering. These features, however, will be problematic for a textureless object. Furthermore, as stated in [88], the feature correspondence performances can be influenced by rotation, scaling and perspective projection variations. [30] extended the range data based point pair features with the color information to improve the performance for handling the self-symmetric objects. The colored point pair features were validated with various daily objects. However, the computational efficiency can be a drawback for the real-time applications.

The pose estimation methods above are based on matched feature correspondences between the current frame and the keyframes or a known model. Another prevalent strategy is to perform pose estimation with a stochastic searching. [128] exploited annealed particle filter with the target edge junction as the observation. [106] also relied on matching Canny edge features for calculating the 6DOF pose. Online processing at video frame rate was made possible with GPU acceleration. The use of an edge based observer restricted these methods on objects with strong edges. In comparison, [3] used silhouette profile to determine the pose. Therefore, it can be extended to objects with more general shapes. However, the use of silhouette cannot make most of the 2D appearance information, thus their approach suffers from the same limitation as the range data based approaches and cannot handle geometrically symmetric objects.

When the observation image is blurred by motion, the feature extraction process, the corner features, the edge features, the SIFT features, etc., can be problematic and the matching can be unreliable. In contrast, the template matching is more robust in such cases. For example, Mutual Information (MI) is a frequently adopted method for evaluating the similarity between two distributions and can provide a robust metric for the optimization on image registration, e.g. the MRI image reconstruction [147]. Levenberg-Marquardt (LM) method is usually used for the MI maximization, as presented in [123] for the rigid object pose estimation. It has been demonstrated that the similarity metric provided by MI is robust under illumination variation and partial occlusion. To tackle the local optimum problem often encountered in an iterative optimization, [87] used Particle Swarm Optimization (PSO) instead for maximizing the MI in the 3D model registration.

The 3D Point Sets Registration

The 3D point sets registration problem has been vastly studied for over two decades due to the development of range measuring devices. When the point correspondences are already accurately established, the desired relative transformation parameters can be optimized through solving an eigensystem for the unit quaternion expression [66] or through SVD of the data covariance matrix for the rotation matrix expression. When the point correspondences cannot be known a prior, the most prevalent solution is Iterative Closest Point (ICP) algorithm. In each iteration, the point correspondences are determined as the closest pairs from the two sets. After the cost function is minimized either from a point-to-point [13] or a point-to-plane [173] error metric, the relative transformation parameters are obtained for the current iteration and the point sets are transformed with the parameters. Then the new point correspondences will be established. The iteration continues until some termination criterion is met.

One limitation of the conventional ICP is that only the local minimum can be reached. This requires the initial point configuration should be sufficiently close to the true alignment. One way to solve this problem is to adopt a stochastic searching. For example, [141] used particle filtering for optimizing the relative pose between the two point sets. The iterative process in ICP was adopted to find the local optimum for each particle in one sample propagation step. The ICP error metric was used to evaluate the particle weights. Besides Bayesian sequential filtering, Genetic Algorithm (GA), due to their capability on a free-form objective function with many local optimum, can also be used to find the transformation parameters between two point sets as in [32].

The local optimum limitation can also be solved through combining ICP with an algorithm that can provide a coarse pre-alignment. [79] employed a Kalman filter for an online 3D model acquisition for an object hold by a manipulator. In the Kalman filter, the inaccurate joint angles from the encoders on the robot were used as the prediction and the inaccurate ICP registration information as the measurement. By segmenting the point cloud of the manipulator from a known 3D model, the object point cloud was subtracted and registered to construct a 3D target model online. [97] adopted GA to perform a pre-registration, where the correspondence pairs were established for each sample and the fitness was evaluated with a point-to-point error metric.

Another limitation of ICP is the computational cost, mostly for the correspondence search procedure. However, when the point set is organized, e.g. the 3D data perceived by a range camera, ray tracing [125] can be employed to provide a projective point pair association. [140] exploited such a scheme and proposed a real-time ICP algorithm for building the 3D model of an object hold in hand with a interactive performance. A more recent approach presented by [118] took advantage of modern many-core GPUs, accelerated the ray tracing, and was applied on a real-time large volume scene modeling. Even higher efficiency can be achieved by using partial image information, e.g. by using only the edges instead of all image points as in [45].

Besides working on two point sets, it is also possible to register multiple range data sets simultaneously. Compared to registering point sets pair by pair, aligning multiple data sets simultaneously can avoid error accumulation that is essential on building a complete 3D model for an object. [16] reported some multiview (six range views) range data registration results with the projective data association. However, no details about the optimization for multiple transformation parameter sets were given. [164] cast the problem into an optimization of the transformation parameters between all point sets and a fixed point set. The optimization was done in a two-step iteration. However, due to the introduction of a fixed point set that has the same number of points as the individual input point set, their method is limited to problems with sufficient overlap among all input point sets.

Most point sets registration methods aim at dealing with a rigid transform (including articulated point sets). It is also possible to handle points under non-rigid and non-linear transformations, e.g. the affine transform or elastic deformation. [115] proposed an algorithm for registering a set of points to another non-rigid transformed set. Different from ICP, their method used all points in the set for calculating the energy function instead of only with the matched point pairs. Meanwhile, a coherent motion constraint was imposed with the spatial smoothness determined by a Gaussian kernel. The registered points were optimized through an EM (Expectation Maximization) procedure, and therefore can be time consuming.

3.1.3 Brief Overview on ICP Variants

After ICP was developed, it has been widely modified and extended. [139] classified these modifications as six stages and analyzed some major variants for each stage. For a convenient discussion of the Textured-ICP algorithm proposed in this thesis, a brief overview of some of the ICP variants closely connected to the work in this thesis are summarized in this subsection.

The point correspondence pairs used in the cost function calculation can be built either through the closest-point search [13] or by the projective data association [16]. The closest-point scheme finds for each point in one set its correspondence point in another set with the closest Euclidean distance. Such a scheme can be applied when one point set is a subset of another. It works most robustly for complex geometries. However, in each iteration, building a correspondence map between the two sets can be quite time consuming even with a k-d tree acceleration. In contrast, the projective data association searches the matching points along the projection rays of the range sensor. If the point sets are well organized, e.g. obtained from a range camera, it is remarkably faster than the closest-point scheme. For real-time applications on not very "difficult" geometries [139], the projective data association is often preferred.

The cost function of ICP can be calculated from either point-to-point [13] or point-to-plane [173] error metrics. The point-to-point metric builds least squares equation with the matched point pairs and has a closed-form solution, e.g. through SVD, quaternion, etc. A good evaluation for four relevant optimization methods can be found in [51]. The point-to-plane metric projects the distance of the correspondence point pairs on the local surface normal direction and is reported to be more effective than the point-to-point metric [140]. The point-to-plane cost function cannot be solved in a closed-form, and non-linear optimization methods like Levenberg-Marquarddt algorithm [55] or the stochastic search schemes [141] are usually adopted.

Some ICP variants aim to deal with the noisy range measurements produced by current range sensors. Having observed that outliers usually provide large residuals in the cost function, it is proposed to sort the correspondence pairs by their residuals, and only use some of them for calculating the cost function, e.g. with least median squares or the trimmed-ICP [29]. The robustness can as well be improved by incorporating the surface texture information in ICP. [162] presented to use the texture-closest points in the neighborhood of distance-closest points also into the correspondence pair set. [72] integrated the texture distance with Euclidean distance for obtaining the matching pairs. Incorporation of the texture can also help handling the surface geometrical symmetry. However, existing methods use the texture in a point-wise level, which will be sensitive to the surface appearance variations.

Besides a rigid surface, ICP can be extended to be applied on more complex surfaces. For a articulated body, [79] built the transformation parameters of all links into the ICP cost function, which was minimized with the Leverberg-Marquardt method. They combined the Kalman filter with the articulated ICP to track an object hold by a robotic manipulator and construct a 3D model of the object online. For deformable surfaces, [115] proposed to use all the points instead of only the matched point pairs for calculating the cost function and output the registered point set. However, their method differs a lot from the ICP procedure, and may not be appropriate to be classified as a variant of ICP.

3.2 Theoretical Background

This section details a variant of the conventional range data based ICP, i.e. the projective point-to-plane ICP, which will be extended in this chapter to incorporate the object texture (appearance) information. This variant has the advantage of being very efficient for dealing with organized range data obtained from a range camera, because the correspondence pairs can be associated by projecting each model point onto the image plane and reading the required observation range data [118].

The cost function in one iteration step is formulated as:

$$E(\mathit{\Delta}\boldsymbol{\theta}) = \sum_{n=1} g^2(\mathbf{u}_{m,n}^{k-1}, \mathit{\Delta}\boldsymbol{\theta}), \tag{3.1}$$

where $\mathbf{u}_{m,n}^{k-1} = [u_x, u_y, u_z]^\top$ is the n-th 3D model point on the target surface from iteration $k-1$. $\mathit{\Delta}\boldsymbol{\theta} = [\mathit{\Delta}\theta_\alpha, \mathit{\Delta}\theta_\beta, \mathit{\Delta}\theta_\gamma, \mathit{\Delta}\theta_x, \mathit{\Delta}\theta_y, \mathit{\Delta}\theta_z]^\top$ is the state vector for the incremental pose, which is used for transforming a model point $\mathbf{u}_{m,n}^{k-1}$ in iteration $k-1$ to $\mathbf{u}_{m,n}^{k}$ in iteration k. As discussed in Subsection 2.4.1, Euler angle $[\theta_\alpha, \theta_\beta, \theta_\gamma]^\top$ in ZYX convention is adopted to model the rotation and $[\theta_x, \theta_y, \theta_z]^\top$ in Cartesian coordinate for the translation.

The point-to-plane distance metric for the n-th point is given by:

$$g(\mathbf{u}_{m,n}^{k-1}, \mathit{\Delta}\boldsymbol{\theta}) = (\mathit{\Delta}\mathbf{T}_k \dot{\mathbf{u}}_{m,n}^{k-1} - \mathbf{u}_{l,n}^{k-1})^\top \mathbf{N}_{l,n}^{k-1}, \tag{3.2}$$

where $\mathbf{u}_{l,n}^{k-1}$ is the live observation correspondence point for the n-th 3D model point $\mathbf{u}_{m,n}^{k-1}$ found by the projective data-association. $\mathbf{N}_{l,n}^{k-1}$ is the live surface normal at point $\mathbf{u}_{l,n}^{k-1}$, which can be estimated with a set of live points in its neighbourhood. The operation $\dot{\mathbf{u}} = [\mathbf{u}^\top, 1]^\top$ is used to get a homogeneous point. The target model point in iteration k is calculated as:

$$\mathbf{u}_{m,n}^{k} = \mathbf{T}_k \dot{\mathbf{u}}_{m,n}^{ref} = \mathbf{T}_k \dot{\mathbf{T}}_{ref} \dot{\mathbf{u}}_{m,n}^{init}, \tag{3.3}$$

where the reference 3D model point $\dot{\mathbf{u}}_{m,n}^{ref}$ for the current frame (or time step) is obtained by rotating and translating the initial model point $\dot{\mathbf{u}}_{m,n}^{init}$ with the target pose from the last frame. $\mathbf{T}_k$ is the 3×4 transformation matrix in the current time step up to iteration k, which can be used to update a reference 3D point to the current iteration. $\dot{\mathbf{T}}_{ref}$ is used to transform the initial model point to the reference point in the current frame. In a video sequence, $\dot{\mathbf{T}}_{ref}$ is the composite transformation matrix up to the last frame. The 4×4 homogeneous transformation matrix is composed of a 3×3 rotation matrix $\mathbf{R}$ and a 3×1 translation vector $\mathbf{t}$ as:

$$\dot{\mathbf{T}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}.$$

The $\mathit{\Delta}\mathbf{T}_k$ in Eq. (3.2) then provides an update between iterations:

$$\mathbf{T}_k = \mathit{\Delta}\mathbf{T}_k \dot{\mathbf{T}}_{k-1}. \tag{3.4}$$

The benefit of optimizing a small incremental update in the cost function instead of estimating a global pose is twofold. First, the singularity of Euler angle can be avoided because for the ZYX convention the singularity will only happen when the rotation around Y axis is close to 90° but the incremental pose is supposed to be small; Second, by assuming a small incremental angle between iterations, the rotation matrix can be linearised and the incremental transformation matrix $\mathit{\Delta}\mathbf{T}_k$ can be approximated as (in the case of Euler angle in ZYX convention):

$$\mathit{\Delta}\mathbf{T}'_k = [\mathbf{R}'_k, \mathbf{t}_k] = \begin{bmatrix} 1 & -\mathit{\Delta}\theta_\alpha & \mathit{\Delta}\theta_\beta & \mathit{\Delta}\theta_x \\ \mathit{\Delta}\theta_\alpha & 1 & -\mathit{\Delta}\theta_\gamma & \mathit{\Delta}\theta_y \\ -\mathit{\Delta}\theta_\beta & \mathit{\Delta}\theta_\gamma & 1 & \mathit{\Delta}\theta_z \end{bmatrix}.$$

Accordingly, the model points $\mathbf{u}_{m,n}^{k}$ used in iteration k are obtained by:

$$\mathbf{u}_{m,n}^{k} = \mathit{\Delta}\mathbf{T}'_k \dot{\mathbf{u}}_{m,n}^{k-1}.$$

With the approximated incremental transformation matrix, Eq. (3.2) can alternatively be rearranged as:

$$g(\mathbf{u}_{m,n}^{k-1}, \varDelta\boldsymbol{\theta}) = (G(\mathbf{u}_{m,n}^{k-1})\varDelta\boldsymbol{\theta} + \mathbf{u}_{m,n}^{k-1} - \mathbf{u}_{l,n}^{k-1})^{\top}\mathbf{N}_{l,n}^{k-1}, \tag{3.5}$$

where

$$\mathbf{G}(\mathbf{u}) = \begin{bmatrix} -u_y & u_z & 0 & 1 & 0 & 0 \\ u_x & 0 & -u_z & 0 & 1 & 0 \\ 0 & -u_x & u_y & 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

Setting the derivative of $g^2(\mathbf{u}_{m,n}^{k-1}, \varDelta\boldsymbol{\theta})$ with respect to $\varDelta\boldsymbol{\theta}$ to zero and summing up all associated points leads to a linear equation for the optimal increment pose:

$$\sum_n (\mathbf{A}_n^{\top}\mathbf{A}_n)\varDelta\boldsymbol{\theta} = -\sum_n g(\mathbf{u}_{m,n}^{k-1}, \mathbf{0})\mathbf{A}_n^{\top}, \tag{3.7}$$

where $\mathbf{A}_n^{\top} = \mathbf{G}(\mathbf{u}_{m,n}^{k-1})^{\top}\mathbf{N}_{l,n}^{k-1}$.

Notice that the linearised 3 × 3 matrix $\mathbf{R}'_k$ in $\varDelta\mathbf{T}'_k$ may not be a valid rotation matrix. To avoid the error being propagated and accumulated between iterations, after $\varDelta\boldsymbol{\theta}$ has been estimated, the transformation matrix $\mathbf{T}_k$ in Eq. (3.4) will still be updated with $\varDelta\mathbf{T}_k$ rather than $\varDelta\mathbf{T}'_k$.

3.3 Incorporating Texture into ICP

This section describes the approach to incorporate texture (target appearance) into the range data based ICP. Combining the texture with the range data can yield a registration algorithm that can deal with symmetric geometry that cannot be handled by conventional ICP. In comparison to the schematic illustration in Fig. 3.1, where the range data based ICP fails on aligning a line section due to short of constraints, Fig. 3.3 shows schematically, if the texture distribution is used in addition to the range data, after some iterations, the desired registration can be achieved.

Previous approaches for combining texture with range data mostly used texture for determining the correspondence pairs. For instance, [72] added the color consistency into the range Euclidean distance and matched points were the pairs with the minimal combined distances. In [162] the matched point set contained both the closest range pairs and the color consistent pairs. They all used texture in a pairwise level. This work, in contrast, builds texture information into the optimization stage of ICP, i.e. into the cost function. The minimization process calculates the spatial derivative

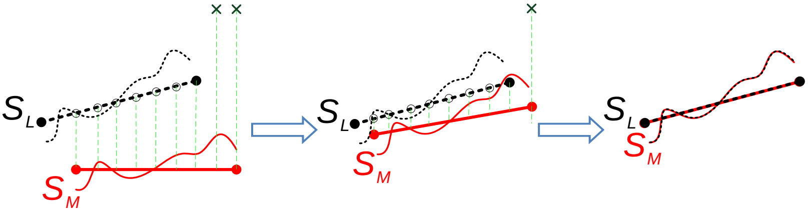


Figure 3.3: Schematic illustration for the Texutured-ICP on a symmetric object. The solid section represents the geometrically symmetric model range data, while the dashed curve stands for the intensity distribution of all points on the section. The model data is in red and the live observation data is in black. The matched points on the live data found by the projective data association are marked with small circles, while the none-correspondence projective rays (the green dashed lines) are ended with crosses.

for the texture of the matched live points, thus implicitly takes the neighborhood information into account. In this way, the proposed method performs more robust than by using the pairwise color consistency.

The pose estimation method proposed in this section is term as Textured-ICP, because it integrates the texture into the ICP framework. However, after it has been derived, the close connection to Lukas-Kanade method for solving the optical flow [6] is revealed. Therefore, the proposed Textured-ICP is also denoted as LKICP. In following discussions, both terms are used interchangeably.

It should be noted that during the writing of this thesis, a similar approach to the proposed LKICP/Textured-ICP is published in [73], where the optical flow from intensity data is combined with the range flow. They applied their method on the sensor ego-motion estimation in a static scene and also demonstrated the real-time performance. In comparison, our method combines the optical flow with ICP and focuses on the pose estimation of a rigid object in the cluttered backgrounds. Besides, as will be introduced in the following subsections, compared with the optical flow derivation in [73], the proposed LKICP uses B-Spline interpolation for a subpixel accuracy on the texture image and adopts a normalization step for formulating the texture consistency. Therefore, the LKICP is expected to be more accurate and more robust to illumination variations. The cost coming with these operations is the raised computation.

The cost function of the proposed Textured-ICP is expressed as

$$E(\mathit{\Delta}\boldsymbol{\theta}) = \sum_{\mathit{\Omega}_k(n)} \left[g^2(\mathbf{u}_{m,n}^{k-1}, \mathit{\Delta}\boldsymbol{\theta}) + \rho f^2(\mathbf{u}_{m,n}^{k-1}, \mathit{\Delta}\boldsymbol{\theta}) \right], \tag{3.8}$$

where $g^2(\mathbf{u}_{m,n}^{k-1}, \mathit{\Delta}\boldsymbol{\theta})$ and $f^2(\mathbf{u}_{m,n}^{k-1}, \mathit{\Delta}\boldsymbol{\theta})$ are the distance metrics for the n-th point under a specified pose for ICP and for the texture respectively. ρ is a constant that controls the influence of the texture in the optimal pose search. In following sections, The metrics above are written as $g_n^2(\mathit{\Delta}\boldsymbol{\theta})$ and $f_n^2(\mathit{\Delta}\boldsymbol{\theta})$ instead for notation simplification. Similar to the discussion in Sec. 3.2, the Textured-ICP also works with the incremental pose $\mathit{\Delta}\boldsymbol{\theta}$, which is similar to the cost function for the point-to-plane ICP in Eq. (3.1), because the optimization process relies heavily on the linearization of the transformation matrix under the small angle assumption. For the sake of derivation convenience, where no confusion would occur, $\boldsymbol{\theta}$ is used for expressing the incremental pose instead of $\mathit{\Delta}\boldsymbol{\theta}$ in the following subsections.

As in [118], $\mathit{\Omega}_k(n)$ represents the 3D consistency test. It takes place in each iteration k on all z-buffered visible points and checks whether the observation 3D measurements and the surface normals are consistent with its projective-associated transformed model data. Only those matched point pairs that have passed $\mathit{\Omega}_k(n)$ will be summed into the cost function. Details with respect to the proposed texture distance and the optimization process will be given in the following subsections.

3.3.1 2D Texture Model

Eq. 3.8 will be minimized with a gradient based optimization scheme, which will require smooth data with a certain order of differentiability. This section parameterizes the 2D texture image captured by the color camera. The parametric form of the 2D image is obtained by interpolating the image with the B-Spline interpolation [123]:

$$v(y_1, y_2) = \sum_j \sum_i c[i,j] B(y_1 - i) B(y_2 - j), \tag{3.9}$$

where $v(y_1, y_2)$ is the image value at pixel position (y_1, y_2). $B(x)$ stands for the quadratic B-Spline basis function and $c[i,j]$ represents the B-Spline coefficients, which has the same size as the observation image. One advantage of the B-Spline interpolation compared to e.g. the polynomial interpolation is that the B-Spline coefficients can be efficiently calculated by an inverse filtering technique as introduced in [146].

Since the quadratic B-Spline is adopted, which has a compact support of 3 units. Above summation will be non-zero for only three elements in both i and j directions. If denoting $h, w = -1, 0, 1$ (representing the indices for the three non-zero support of B-Spline) and $r_j = y_j - round(y_j)$, the interpolation in Eq. (3.9) can be rewritten as

$$v(\mathbf{y}) = \sum_h \sum_w c_{w,h}(a_w r_1^2 + b_w r_1 + d_w)(a_h r_2^2 + b_h r_2 + d_h),$$

where $c_{w,h} = c[round(y_1) + w, round(y_2) + h]$. In addition, $a_{\{-1,0,1\}} = \{0.5, -1, 0.5\}$, $b_{\{-1,0,1\}} = \{-0.5, 0, 0.5\}$ and $d_{\{-1,0,1\}} = \{0.125, 0.75, 0.125\}$ are the parameters used for calculating the quadratic B-Spline function values.

The 2×1 gradient vector

$$\nabla_{\mathbf{y}} v = \left[\frac{\partial v}{\partial y_1}, \frac{\partial v}{\partial y_2}\right]^\top$$

of intensity with respect to $\mathbf{y}$ is calculated as

$$\begin{cases} \dfrac{\partial v}{\partial y_1} = \sum_h \sum_w c_{w,h}(2a_w r_1 + b_w)(a_h r_2^2 + b_h r_2 + d_h) \\ \dfrac{\partial v}{\partial y_2} = \sum_h \sum_w c_{w,h}(a_w r_1^2 + b_w r_1 + d_w)(2a_h r_2 + b_h) \end{cases}.$$

When required, the 2×2 Hessian matrix $\mathbf{H}_v(\mathbf{y})$ of intensity with respect to $\mathbf{y}$ can also be obtained accordingly. But as will be discussed in Subsection 3.3.3, instead of Hessian, the Gauss-Newton approximation will be used to calculate the pose update during optimization, which only requires the first order derivatives. Therefore, the second order derivatives are not given here.

3.3.2 Perspective Projection Model for 2D Camera

Besides the differentiable observation data derived in the previous subsection, the way to sample data should also be analytically established with a certain order of smoothness. Since the aim is to compare the appearance value of each model point with the texture value of its correspondent observation point, and the correspondence is determined through the projective data association as introduced in Sec. 3.2, the mapping $\mathbf{y} = p(\mathbf{u}^k, \boldsymbol{\theta}) = [y_1, y_2]^\top$ from the 3D model point $\mathbf{u}^k = \varDelta\mathbf{T}_k \dot{\mathbf{u}}^{k-1} = [u_x^k, u_y^k, u_z^k]^\top$ to the 2D image pixel should also be obtained through the projective model. By assuming a pin-hole camera model, the perspective projection model is expressed as

$$\mathbf{y} = \left[F_x \frac{u_x^k}{u_z^k} + l_{y1}, F_y \frac{u_y^k}{u_z^k} + l_{y2} \right]^\top,$$

where (F_{y1}, F_{y2}) are the focal lengths scaled by the physical pixel sizes for the pin-hole camera in both x and y directions. (l_{y1}, l_{y2}) is the pixel position for the intersection of the image plane and the optical axis of the lens. They are the intrinsic parameters for the pin-hole camera as in Eq. (1.4), which can be obtained through a calibration process with the method propose in [180]. $\boldsymbol{\theta}$ represents the incremental pose.

Similar to the processing from Eq. (3.2) to Eq. (3.5), the incremental transformation matrix $\mathit{\Delta}\mathbf{T}_k$ can be linearised under the small angle assumption, and the above projection model can be approximated as:

$$\mathbf{y} = \left[F_{y1} \frac{\mathbf{G}_0\boldsymbol{\theta} + u_x^{k-1}}{\mathbf{G}_2\boldsymbol{\theta} + u_z^{k-1}} + l_{y1}, F_{y2} \frac{\mathbf{G}_1\boldsymbol{\theta} + u_y^{k-1}}{\mathbf{G}_2\boldsymbol{\theta} + u_z^{k-1}} + l_{y2} \right]^\top$$

where the 1×6 vector $\mathbf{G}_j$ is the j-th row of $\mathbf{G}(\mathbf{u}^{k-1})$ in Eq. (3.6). Given a 3D model point $\mathbf{u}^{k-1}$, the interested thing in a gradient based optimization method is the gradient of intensity with respect to the incremental pose $\boldsymbol{\theta}$, which can be obtained from the chain rule:

$$\nabla_{\boldsymbol{\theta}} v = \frac{\partial v}{\partial y_1} \frac{\partial y_1}{\partial \boldsymbol{\theta}} + \frac{\partial v}{\partial y_2} \frac{\partial y_2}{\partial \boldsymbol{\theta}} = \mathbf{J}_{\mathbf{y}}(\boldsymbol{\theta}) \nabla_{\mathbf{y}} v. \tag{3.10}$$

Here $\mathbf{J}_{\mathbf{y}}(\boldsymbol{\theta})$ is the 6×2 Jacobian matrix

$$\mathbf{J}_{\mathbf{y}}(\boldsymbol{\theta}) = \mathbf{J}_p(\mathbf{u}^{k-1}, \boldsymbol{\theta}) = \left[\frac{\partial y_1}{\partial \boldsymbol{\theta}}, \frac{\partial y_2}{\partial \boldsymbol{\theta}} \right], \tag{3.11}$$

which can be calculated by:

$$\begin{cases} \dfrac{\partial y_1}{\partial \boldsymbol{\theta}} = \dfrac{F_{y1}}{(\mathbf{G}_2\boldsymbol{\theta} + u_z^{k-1})^2} (\mathbf{D}_0\boldsymbol{\theta} + \mathbf{B}_0) \\ \dfrac{\partial y_2}{\partial \boldsymbol{\theta}} = \dfrac{F_{y2}}{(\mathbf{G}_2\boldsymbol{\theta} + u_z^{k-1})^2} (\mathbf{D}_1\boldsymbol{\theta} + \mathbf{B}_1) \end{cases}$$

with

$$\begin{aligned}\mathbf{D}_j &= \mathbf{G}_j^\top \mathbf{G}_2 - \mathbf{G}_2^\top \mathbf{G}_j \\ \mathbf{B}_0 &= u_z^{k-1}\mathbf{G}_0^\top - u_x^{k-1}\mathbf{G}_2^\top . \\ \mathbf{B}_1 &= u_z^{k-1}\mathbf{G}_1^\top - u_y^{k-1}\mathbf{G}_2^\top\end{aligned}$$

Correspondingly, the second order derivatives can also be obtained as

$$\begin{cases}\dfrac{\partial^2 y_1}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top} = \dfrac{F_{y1}}{(\mathbf{G}_2\boldsymbol{\theta} + u_z^{k-1})^2}\mathbf{D}_0 - \dfrac{2}{\mathbf{G}_2\boldsymbol{\theta} + u_z^{k-1}}\dfrac{\partial y_1}{\partial\boldsymbol{\theta}}\mathbf{G}_2 \\ \dfrac{\partial^2 y_2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top} = \dfrac{F_{y2}}{(\mathbf{G}_2\boldsymbol{\theta} + u_z^{k-1})^2}\mathbf{D}_1 - \dfrac{2}{\mathbf{G}_2\boldsymbol{\theta} + u_z^{k-1}}\dfrac{\partial y_2}{\partial\boldsymbol{\theta}}\mathbf{G}_2\end{cases}.$$

3.3.3 Texture Consistency

Previous two subsections introduced the method for sampling the observed intensity value for a specified model point under an input (incremental) pose. This subsection derives the distance metric used for comparing the model intensity and the observation intensity values, as well as necessary derivatives.

The Normalized Sum of Squared Difference (NSSD) is exploited for formulating the error metric, for which the normalized intensity difference is:

$$f(\mathbf{u}_{m,n}^{k-1}, \varDelta\boldsymbol{\theta}) = \frac{v(p(\mathbf{u}_{m,n}^{k-1}, \varDelta\boldsymbol{\theta})) - \mu_l^k}{\sigma_l^k} - \frac{v_{m,n}^{k-1} - \mu_m^{k-1}}{\sigma_m^{k-1}}, \tag{3.12}$$

where (μ_l^k, σ_l^k) and $(\mu_m^{k-1}, \sigma_m^{k-1})$ are mean and standard deviation of the intensities of the projective-associated live points and the target model points respectively. Similar to the Mean Squared Error (MSE), although NSSD may not really reflect the true perception variation [161], it yields an error metric that can be solved analytically and is by far the most adopted distance metric in many fields.

The mean and standard deviation of intensity are determined by all matched point pairs, which are influenced by $\varDelta\boldsymbol{\theta}$ and the consistency test $\varOmega_k(n)$. Thus the model statistics $(\mu_m^{k-1}, \sigma_m^{k-1})$ and the live statistics (μ_l^k, σ_l^k) will vary with respect to $\varDelta\boldsymbol{\theta}$, i.e. they are functions of $\varDelta\boldsymbol{\theta}$. However, similar to the linearisation of the transformation matrix $\varDelta\mathbf{T}_k$, $\varDelta\boldsymbol{\theta}$ is expected to change gently between successive iterations. Thus variation of the statistics is also expected to be small. Therefore, both (μ_l^k, σ_l^k) and $(\mu_m^{k-1}, \sigma_m^{k-1})$ are assumed constant between two iterations. Likewise, (μ_l^k, σ_l^k) calculated with the current matched live points can be approximated by $(\mu_l^{k-1}, \sigma_l^{k-1})$ from the matched live points in the previous iteration. Under such simplifications, the 6×1 gradient vector $\nabla f_n^2(\boldsymbol{\theta})$ of $f_n^2(\mathbf{u}^{k-1}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ (instead of writing

$f^2(\mathbf{u}_{m,n}^{k-1}, \varDelta\boldsymbol{\theta})$ with respect to $\varDelta\boldsymbol{\theta}$ for the sake of notation clarity) can be derived as

$$\nabla f_n^2(\boldsymbol{\theta}) = \frac{2}{\sigma_l} f_n^\top(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} v_n,$$

where as a scalar, $f_n^\top(\boldsymbol{\theta}) = f_n(\boldsymbol{\theta})$.

And the 6 × 6 Hessian matrix $\mathbf{H}_{f_n^2}(\boldsymbol{\theta})$ is calculated as:

$$\mathbf{H}_{f_n^2}(\boldsymbol{\theta}) = \frac{2}{\sigma_l^2} \nabla_{\boldsymbol{\theta}} v_n (\nabla_{\boldsymbol{\theta}} v_n)^\top + \frac{2}{\sigma_l} f_n(\boldsymbol{\theta}) \mathbf{H}_v(\boldsymbol{\theta}). \tag{3.13}$$

The gradient $\nabla_{\boldsymbol{\theta}} v$ of intensity with respect to $\boldsymbol{\theta}$ is obtained with Eq. (3.10), and the Hessian matrix of intensity with respect to $\boldsymbol{\theta}$ is calculated by

$$\mathbf{H}_v(\boldsymbol{\theta}) = \left[\frac{\partial^2 y_1}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}, \frac{\partial^2 y_2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \right] \nabla_{\mathbf{y}} v + \mathbf{J}_{\mathbf{y}}(\boldsymbol{\theta}) \mathbf{H}_v(\mathbf{y}) \mathbf{J}_{\mathbf{y}}^\top(\boldsymbol{\theta}).$$

The first term on the right side of the above equation is quite small compared to the second term, it could simply be neglected when calculating $\mathbf{H}_v(\boldsymbol{\theta})$ for a full Hessian based gradient descent optimization algorithm. However, it has been demonstrated in [27] and [6] that for an image alignment application, a Gauss-Newton approximation to the Hessian $\mathbf{H}_{f_n^2}(\boldsymbol{\theta})$ in Eq. (3.13) will yield a better convergence when the initial pose is far from the optimum, i.e. the initial input has large residuals. The Gauss-Newton approximation to the Hessian is given by:

$$\mathbf{H}_{GN,f_n^2}(\boldsymbol{\theta}) = \frac{2}{\sigma_l^2} \nabla_{\boldsymbol{\theta}} v_n (\nabla_{\boldsymbol{\theta}} v_n)^\top.$$

Moreover, employing a Gauss-Newton update forms an algorithm that is consistent with the structure of the projective point-to-plane ICP, which will be further presented with more details in the next subsection. For a better discussion, now $\varDelta\boldsymbol{\theta}$ is used to express the incremental pose. The Gauss-Newton update $\delta\boldsymbol{\theta}$ to the incremental pose $\varDelta\boldsymbol{\theta}$ for optimizing $f_n^2(\mathbf{u}^{k-1}, \boldsymbol{\theta})$ is obtained by solving the linear equation

$$\mathbf{H}_{GN,f_n^2}(\varDelta\boldsymbol{\theta}^{k-1}) \delta\boldsymbol{\theta} = -\nabla f_n^2(\varDelta\boldsymbol{\theta}^{k-1}),$$

where $\varDelta\boldsymbol{\theta}^{k-1}$ is the incremental pose upto the last iteration. However, according to Eq. (3.3) and (3.4), the input 3D model points are calculated with $\mathbf{u}^{k-1} = \varDelta\mathbf{T}_{k-1}\dot{\mathbf{T}}_{k-2}\dot{\mathbf{u}}^{ref}$. The use of $\varDelta\mathbf{T}_{k-1}$ implies that the input model point has already been transformed by applying the incremental pose $\varDelta\boldsymbol{\theta}^{k-1}$ upto the last iteration, and the incremental pose $\varDelta\boldsymbol{\theta}^{k-1}$ should be set to $\mathbf{0}$ since it is already used in transforming the model points. In this case, the update $\delta\boldsymbol{\theta}$ is in effect the incremental pose $\varDelta\boldsymbol{\theta}$ for transforming the model points between iterations. Therefore, above update can be calculated by

$$\mathbf{H}_{GN,f_n^2}(\mathbf{0})\varDelta\boldsymbol{\theta} = -\nabla f_n^2(\mathbf{0}). \tag{3.14}$$

And evaluation of $\mathbf{J}_{\mathbf{y}}(\boldsymbol{\theta})$ at $\boldsymbol{\theta} = \mathbf{0}$ will be simplified as:

$$\mathbf{J}_{\mathbf{y}}(\mathbf{0}) = \left[\frac{F_{y1}}{(u_z^{k-1})^2}\mathbf{B}_0, \frac{F_{y2}}{(u_z^{k-1})^2}\mathbf{B}_1,\right].$$

Furthermore, the 3×6 matrix $\mathbf{G}(\mathbf{u})$ is quite sparse, thus only a moderate amount of calculations are required.

3.3.4 Iterative Optimization for Textured-ICP

The incremental pose solved by the texture consistency in Eq. (3.14) shares the same form as the one solved with the conventional ICP in Eq. (3.7) (except Eq. (3.7) is formulated for all matched points). This indicates that the texture and range distances can be combined for solving the incremental pose by simply summing up both equations as

$$\sum_{\varOmega_k(n)} \left(\mathbf{A}_n^\top\mathbf{A}_n + \rho\mathbf{H}_{GN,f_n^2}(\mathbf{0})\right)\varDelta\boldsymbol{\theta} = -\sum_{\varOmega_k(n)} \left(g_n(\mathbf{0})\mathbf{A}_n^\top + \rho\nabla f_n^2(\mathbf{0})\right), \tag{3.15}$$

where as for Eq. (3.7), the summation is performed upon all matched point pairs that have pass the matching consistency test $\varOmega_k(n)$.

The variable ρ in Eq. (3.15) controls the influence of texture in the pose optimization. The texture and the range error values usually have different magnitudes. ρ should also take this into account. In this work, it is set as

$$\rho = \rho_c\frac{\|\mathbf{A}_n^\top\mathbf{A}_n\|_2}{\|\mathbf{H}_{GN,f_n^2}(\mathbf{0})\|_2}, \tag{3.16}$$

where $\|\cdot\|_2$ calculates the matrix norm. Thus the fraction part balances the numeric magnitudes between range and texture consistencies. ρ_c is a pre-specified constant that determines the texture influences. It should be set according to the noise level of the range and the color images. For example, when the motion in the application is smooth and the range data is reliable or when the ambient light is weak and the color camera only produces low quality images, $\rho_c = 0.5$ could be used. On the other hand, if the range data is contaminated, $\rho_c = 2.0$ is preferred. In most tests in this thesis, $\rho_c = 1.0$ is used.

The iterative optimization procedure for the proposed Textured-ICP (LKICP) is described in Algorithm 3.1. It uses the pre-aligned pose from the coarse pose estimation stage as the initial pose guess. The fused RGBD image is taken as the input observation data. The model colored point cloud will be tested with z-buffering under the coarsely estimated pose, as carried out in the first step of Algorithm 3.1. Later on, the z-buffered point set will be used as the model data for the accurate pose estimation. Although the visible point set determined by the coarse pose is in general different with the true visible set, small variations of the set either can be handled by the matching consistency test $\Omega_k(n)$ or can be simply neglected without causing much error.

Algorithm 3.1 Iterative Optimization for LKICP

```
Inputs:
  1. Coarse pose estimate.
  2. Live (or observation) fused RGBD image.
Procedure:
  1. Get z-buffered transformed target point set $\mathcal{M}_{z,0}$ and extended target image
     region with input pose.
  2. Calculate surface normal and B-Spline coefficients in extended target region.
  3. for $k = 1 \rightarrow K_{max}$ do
      (a) Get matched point pairs with projective data association for all points in
          $\mathcal{M}_{z,k-1}$.
      (b) Calculate $\mathbf{A}_n^\top$ and $g(\mathbf{u}_{m,n}^{k-1}, \mathbf{0})$ in Eq. (3.7) for all matched point pairs, sort
          the match pairs with cost function values $g_n^2(\mathbf{0})$.
      (c) Calculate $(\mu_l^{k-1}, \sigma_l^{k-1})$, $(\mu_m^{k-1}, \sigma_m^{k-1})$, $\mathbf{H}_{GN,f_n^2}(\mathbf{0})$ and $\nabla f_n^2(\mathbf{0})$ for 50% point
          pairs with smallest values of $g_n^2(\mathbf{0})$.
      (d) Calculate $\Delta\boldsymbol{\theta}$ with Eq. (3.15) using trimmed points pairs, if $\|\Delta\boldsymbol{\theta}\| < \varepsilon_{\boldsymbol{\theta}}$,
          exit iteration.
      (e) Transform point set $\mathcal{M}_{z,k-1}$ to $\mathcal{M}_{z,k}$ with transformation matrix $\Delta\mathbf{T}_k$.
  4. end for
  5. Get final transformation matrix $\mathbf{T}_{est}$ and estimated pose $\boldsymbol{\theta}_{est}$.
Output: Final transformation matrix $\mathbf{T}_{est}$ and estimated pose $\boldsymbol{\theta}_{est}$
```

The Step 1 also outputs an extended target 2D upright bounding rectangle for all the projected target points on the image. The bounding rectangle is enlarged a bit to account for the small difference between the coarse pose and the final accurate pose. The computations required for calculating the B-Spline coefficients and the surface normals in Step 2 are only performed within the extended target region, which avoids unnecessary calculations on non-target image regions.

Step 3(b) in Algorithm 3.1 calculates the range terms for all matched point pairs. Step 3(c) calculates the texture terms for 50% of the matched point pairs that have the smallest ICP cost function values. The proposed LKICP is based on least squares, which is known to be sensitive to outliers. Therefore, the Trimmed-ICP suggested by [29] is adopted, where the cost function values from ICP are sorted and only a portion of the matched pairs with the smallest values will be used for the optimization. Since the range data from the PMD camera can be quite noisy under motion, the trimmed version usually yields a more robust pose estimation than using all matched pairs.

The iteration can be terminated either when the incremental pose is too small or when the pre-specified maximum number of iterations has been reached. The final transformation matrix can be retrieved by

$$\mathbf{T}_{est} = \mathit{\Delta}\mathbf{T}_k \dot{\mathbf{T}}_{k-1} \dot{\mathbf{T}}_{ref}.$$

Then the final accurate pose $\boldsymbol{\theta}_{est}$ can be obtained accordingly.

3.3.5 Tests on Symmetric Geometry and Noisy Range Data

Incorporating texture into the range data based ICP can improve the estimation performance when dealing with a geometrically symmetric object or noisy range measurements. This subsection shows such improvements by comparing the pose estimation result from LKICP with the result from ICP.

The test on the symmetric geometry was carried out with a cylindrical bucket. Some frames from a video sequence are illustrated in Fig. 3.4, where the cylinder was rotating around its axis for some degrees. For this test, the coarse pose estimation stage was completely deactivated, and the estimated pose from the previous frame was taken as the initial pose for the current frame.

The pose estimated with ICP are shown in images in the first row of Fig. 3.4. The range data based ICP is ill-posed on symmetric geometries. When the cylinder rotates around its axis, although the rotation is visually perceptible, it cannot be reflected from range measurements variations. Therefore, poses from ICP remain unchanged throughout the frames. In contrast, the proposed LKICP can effectively capture the pose changes as illustrated in the second row of Fig. 3.4.

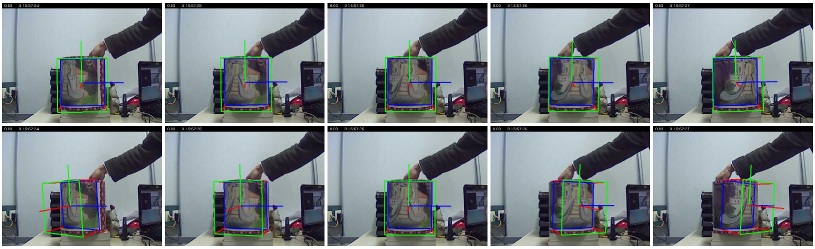
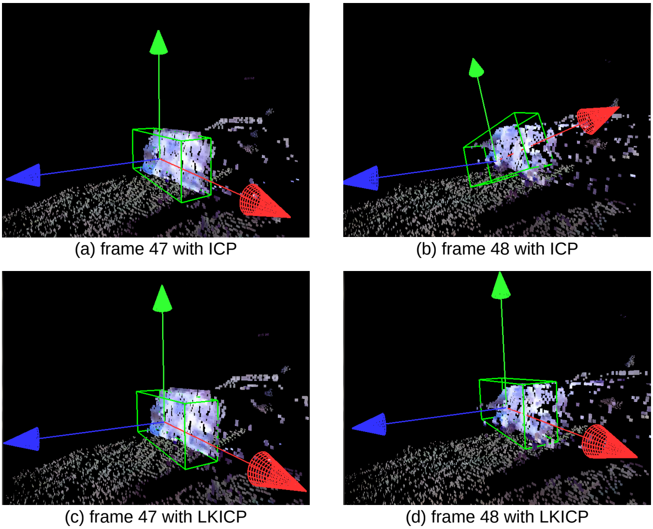


Figure 3.4: Test on symmetric geometry. A cylindrical bucket is used as the test object. The poses estimated with ICP and LKICP are shown in the first and the second rows respectively.

Although the results in Fig. 3.4 were obtained with the accurate stage only, the gradient based optimization requires a good initial pose input, due to the use of the projective data association and the quadratic B-Spline for interpolating 2D image. When the target has a large motion between frames, the pose from the coarse estimation stage will play a crucial role on the final performance. Since the coarse stage also uses the texture for determining the pose, it can tackle the symmetric geometry as well. Thus, with an initial pose from the coarse stage, the performance of ICP can also be significantly improved. On the other hand, as will be discussed in Sec. 5.2, the proposed LKICP also outperforms ICP in terms of accuracy.

Moreover, the conventional ICP will run into problems when the range data is contaminated by motion artefacts, as shown in Fig. 3.2, where the fused sensors were mounted on a mobile robot driving on grassland. The trimmed version of ICP yielded a better result but can still fail on estimating the pose on some degrees of freedom. In comparison, Fig. 3.5 shows the robustness of the LKICP on the noisy range measurements. This is because the 2D color camera produced reliable texture images for the target.

Normally, the motion will also decrease the quality of the 2D image. But the experiment shows that the influences are quite different. Fast motion usually blurs the 2D image. For a gradient feature (e.g. corner feature) based algorithm, such an artefact can be detrimental. However, the coarse stage, as well as the LKICP in the accurate stage, uses a large number of target surface points and works more like a template matching method. Therefore, the proposed method is robust under the 2D image blurring. In contrast, the PMD camera takes four shots for calculating one range image. The influence of motion on the range image will be much severer than


(a) frame 47 with ICP

(b) frame 48 with ICP

(c) frame 47 with LKICP

(d) frame 48 with LKICP

Figure 3.5: Test on noisy range measurement. (a) and (b) are two successive frames with the pose estimated with ICP. The range data are contaminated by the motion artefacts in the second frame. (c) and (d) show the estimation results with the proposed LKICP.

on the color image. The four shots are expected to come from the same reflected signal with a constant amplitude. Under motion, the real reflected signals can vary dramatically not only due to different range between the target object and the surrounding background surfaces, but also because of the change of the signal amplitude result from the reflectivity differences between the target and the background. All in all, it can be much more complicated than the simple aliasing for the case of the 2D image. In some cases, with the corrupted range data, the shape of the target can barely be recognized. Thus using range data alone will yield poor results when the PMD range measurements are contaminated by fast motion.

In above test, the constant ρ_c in Eq. (3.16) for controlling the influence of the texture in the pose optimization is set to $\rho_c = 1.0$. No special preference is placed on the texture nor the range measurements. Full video of this test can be found in the supplementary CD and more evaluations will be given in later chapters.

3.4 Implementation and Convergence Evaluation

This section discusses some implementation issues with respect to the surface normal estimation and GPU acceleration. The estimated surface normal will be used in Eq. (3.7) for the pose optimization and for determining the validity of the matched point pairs. Also, the surface normal estimation plays an important role on the illumination modeling which will be discussed in the next chapter. The parallelism of the proposed Textured-ICP is investigated and the running time in this stage with GPU acceleration is given. Furthermore, the convergence of the Textured-ICP is also evaluated and compared with the conventional ICP and the texture-only optimization. It is demonstrated that combining the texture and the range data not only improves the stability for dealing with the initial pose with large errors, but also reduces the number of iterations required to reach a pre-specified accuracy.

3.4.1 Surface Normal Estimation

The Surface Normal (SN) vector $\mathbf{n} = [n_x, n_y, n_z]^\top$ at a target point can be estimated by grabbing a number of points in its neighbourhood, and fitting the sample points to a local plane. If the plane function is given by

$$n_x x + n_y y + n_z z - d = 0,$$

the fitting can be achieved by minimizing an objective function [107]

$$\min_{\mathbf{n}} \sum_{i=1}^{k} (\mathbf{p}_i^\top \mathbf{n} - d)^2 \;\; s.t. \;\; \|\mathbf{n}\|_2 = 1, \tag{3.17}$$

where $\mathbf{p}_i$ represents a sample point in the neighbourhood.

The Traditional Least Squares (TLS) problem in Eq. (3.17) can be solved analytically by finding the eigenvector corresponding to the smallest eigenvalue of the 3×3 covariance matrix

$$\mathbf{M} = \frac{1}{k} \sum_{i=1}^{k} (\mathbf{p}_i - \overline{\mathbf{p}})(\mathbf{p}_i - \overline{\mathbf{p}})^\top,$$

where $\overline{\mathbf{p}}$ is the mean of all the 3D sample points. This is denoted as the PlanePCA method in [75]. Besides PlanePCA, [75] also evaluated other methods for solving the least squares problem and concluded that for medium-sized neighbourhoods, the PlanePCA method yields better performance in terms of accuracy and computational efficiency.

Although the TLS solution obtains an optimal estimation, it is based on *i.i.d.* (identical independent distribution) noise assumption for the three dimensions of all sample points, which means the noises in X, Y and Z directions are independent. However, if expressing a sample point in the spherical coordinates as

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = r\mathbf{v}, \text{ where } \mathbf{v} = \begin{bmatrix} sin\theta cos\phi \\ sin\phi \\ cos\theta sin\phi \end{bmatrix},$$

where r represents the radial distance directly measured by the PMD camera, since most of the measurement uncertainties from the PMD camera are reflected in r, it is clear that the noise propagates linearly along different coordinate dimensions. Such a violation to the *i.i.d.* assumption can result in the poor performance when the range data are corrupted. Fig. 3.6 (a) shows the SNs estimated by TLS on a plane contaminated with Gaussian random noise, where most of the estimated SNs have large errors.

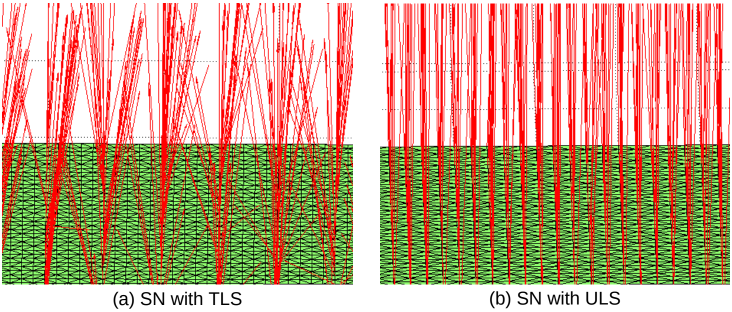


Figure 3.6: Surface Normal (SN) estimation on a noisy plane. The resulting SN estimated with the Traditional Least Squares (TLS) and the Unconstrained Least Squares (ULS) are shown in (a) and (b) respectively, where the plane is depicted in green and the SNs are drawn in red.

[4] presented a method that can handle the violation of TLS to the i.i.d. assumption. If the spherical coordinate of a sample point $\mathbf{p}_i = r_i[v_{xi}, v_{yi}, v_{zi}]^\top$ is put into the plane function, the following equation can be obtained:

$$r_i v_{xi} n_x + r_i v_{yi} n_y + r_i v_{zi} n_z - d = 0,$$

which can be reformulated as $r_i k_i = d$ with $k_i = v_{xi} n_x + v_{yi} n_y + v_{zi} n_z$. In a small neighbourhood, k_i varies little between points. Therefore, dividing d^2 in the objective function in Eq. (3.17) can be interpreted as an approximation for removing r_i from $\mathbf{p}_i$. The new objective function is expressed as

$$\min_{\widetilde{\mathbf{n}}} \sum_{i=1}^{k} (\mathbf{p}_i^\top \widetilde{\mathbf{n}} - 1)^2.$$

Notice that after removing d^2, the new $\widetilde{\mathbf{n}}$ to optimize is not necessarily a unit vector. Thus a following normalization step is required. Due to lack of the unit vector constraint, this method is termed as the Unconstrained Least Squares (ULS) method in [4]. The SN estimation result for a contaminated plane is illustrated in Fig. 3.6 (b). Compared to the result with TLS in Fig. 3.6 (a), ULS yields much better SNs, meanwhile it has a lower computational cost. Therefore, it is adopted in this thesis for the SN estimation. More evaluations for different SN estimation approaches can be found in [4, 75, 107].

3.4.2 GPU Acceleration

The computational efficiency is essential in the robotic applications. Taking advantage of the modern many-core GPUs, this section investigates the parallelism in the proposed accurate pose estimation algorithm, and gives the running time of this estimation stage with GPU acceleration under typical configurations. Unfortunately, no CPU version of implementation is available, thus the detailed speedup cannot be presented.

The surface normal estimation can be naturally calculated in parallel for all the image points inside the target bounding rectangle. For the implementation on GPU, the matrix inversion operation involved in the surface normal estimation will require double precision, otherwise the result can be quite poor. Therefore, double precision GPUs should be considered. Further acceleration can be achieved by using the box-filtering as was suggested in [4], although it is currently not implemented yet in the code.

The data association process involves the matrix computations for transforming then projecting the model points onto the image plane, as well as testing the matching consistency between the transformed model points and the observation points. These operations can be efficiently calculated on GPU. The terms in the main iteration of the optimization, e.g. $\mathbf{A}_n^\top \mathbf{A}_n$ and $g(\mathbf{u}_{m,n}^{k-1}, \mathbf{0})$ in Eq. (3.7), $\mathbf{H}_{GN,f_n^2}(\mathbf{0})$ and $\nabla f_n^2(\mathbf{0})$ in

Eq. (3.14), are computed on GPU in parallel. Since the trimmed version of ICP is adopted, the matched point pairs will be sorted according to the cost function values of ICP. Moreover, the summation in Eq. (3.15) will operate on thousands of matrices, for which the computation time cannot be neglected. Besides, the calculation of the model and the observation statistics $(\mu_l^{k-1}, \sigma_l^{k-1})$ and $(\mu_m^{k-1}, \sigma_m^{k-1})$ in Step 3 (c) of Algorithm 3.1 also raises a noticeable cost if it will be computed sequentially on CPU. Above sorting and summation, however, can be solved by the parallel prefix sum algorithms with high efficiency. This work exploited the *Thrust* parallel algorithm library for a quick implementation of the parallel prefix sum algorithms.

The running time of the proposed algorithm is evaluated on a laptop introduced in Subsection 2.4.2. When the extended target bounding box calculated in Step 2 of Algorithm 3.1 has around 105 pixels in width and 170 pixels in height, and with around 2,500 matched pairs, this stage takes 15∼20 ms.

accurate estimate stage	
Size of $\mathcal{M}$	≈10,000
Size of $\mathcal{M}_z$	≈9,000
Matched pairs	≈2,500
Bounding box size	≈(105,170)
Max. number of iterations	15
Running time	15 ∼ 20 ms

Table 3.1: Running time for the accurate estimate stage

Similar to the discussions in Subsection 2.4.2, some of the computations are running on CPU. Partly because some calculations will be much more efficient when running on CPU, e.g. solving the linear equation for the incremental pose in Eq. (3.15); partly because some source codes are available for the CPU calculation, which can remarkably ease the implementation and debugging, e.g. the inverse filtering in [146] for calculating the B-Spline coefficients. Such a hybrid CPU/GPU computation inevitably raises costs for the data transfer between the two computing units. When a GPU with PCI-E 3.0 can be used, the cost for the data transfer can be significantly reduced. Or when the calculation for the B-Spline coefficient is implemented on the GPU, it is expected that around 4∼5 ms can be saved in the above test. Or when more recent GPUs with over one thousand CUDA cores can be employed, the parallel computation can be remarkably accelerated. Therefore, it can be concluded that the proposed algorithm is competent for real-time applications.

3.4.3 Convergence Evaluation

The proposed LKICP is a gradient based iterative optimization algorithm. For an iterative approach, often two questions arise: how fast it can converge to the optimum and how much error it can tolerate for the initial input. In this subsection, both are evaluated through experiments on real objects and the results for the Merlin robot are given in Fig. 3.7 and Fig. 3.8.

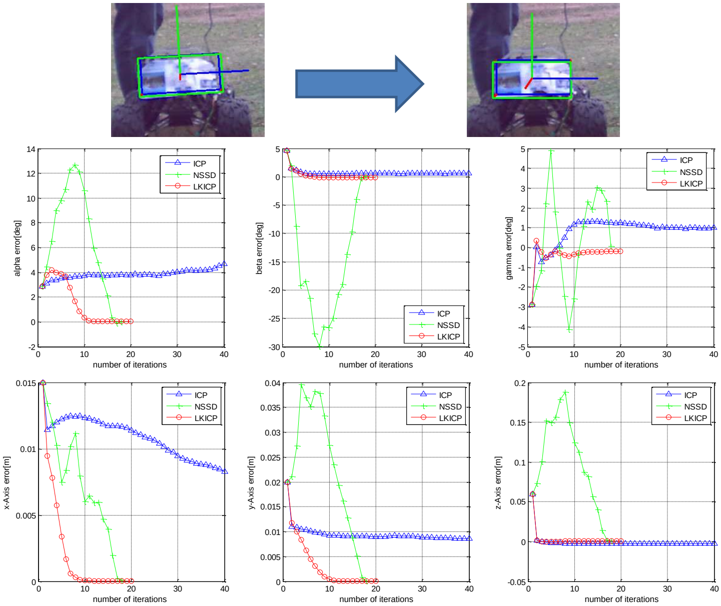


Figure 3.7: Convergence rate evaluation. The top two images illustrate the input erroneous pose and the desired pose. The middle three are for Euler angle rotation around the Z, Y and X axes, in radian. The bottom three are for the translation along X,Y and Z axes, in meter. The number of iterations and the corresponding residual pose are shown on the X and Y axes respectively.

The convergence rate (or convergence speed) is evaluated by inputting an erroneous initial pose and checking the residual errors after each iteration as well as the number of iterations required to reach a pre-specified acceptable error bound. The initial pose and desired pose are depicted in the two images on the top of Fig. 3.7.

For demonstrating the advantage of combining the texture with the range data, three error metrics are compared, i.e. the texture based NSSD, the range based ICP and the combined LKICP. Results for all six degrees are illustrated in Fig. 3.7, where the three images in the middle row are for the rotational degrees and the bottom three are for the translation.

For the input erroneous pose, within 40 iterations, only ICP fails to converge to the specified accuracy. This is largely due to the planar shape of the target. Although the Merlin robot has a complex shape, this work only considers a target model initialized with one frame due to the lack of a complete CAD model. In this test, only the rear part of the robot is taken as the target. The wheels are excluded because the dark texture make them hard to sense with the PMD camera. Therefore, even if the initial pose is quite close to the desired pose, the ICP iterations cannot correctly converge.

On the other hand, the NSSD converges to the desired accuracy with around 18 iterations. However, it takes a quite zigzag path, which can clearly be seen in the images for rotation (the middle row in Fig. 3.7). This may result from the use of the 2D quadratic B-Spline for interpolating the 2D image. In theory, the 3 units compact support of the quadratic B-Spline can only effectively model pixel intensities within a 3×3 window. When the initial alignment exceeds this range, it can work in a complicated manner. In contrast to ICP, when it gets closer to the optimum especially when within the 3-unit region, NSSD shows a rather fast convergence.

In comparison, the proposed LKICP demonstrates the most stable convergence rate. It combines the advantages of ICP and NSSD methods and yields fast convergence at both beginning stage of the iteration and near the optimum. Although the desired accuracy is achieved with 20 iterations, the pose is already quite close the desired one within 10 iterations. A lot more tests indicate that the number of iterations required to tackle input error poses is usually less than 20. Therefore, a maximum iteration number of 15∼20 is normally adopted, as the case in Table 3.1. Such a setting can be further confirmed in the following convergence stability evaluation.

The convergence stability or convergence region refers to how much initial state error the algorithm can tolerate and achieve the optimal state. The 6DOF pose estimation is a high dimensional state space problem, for which it is impractical to evaluate the convergence stability for all six degrees simultaneously. Therefore, each degree of freedom is evaluated separately by setting the input error for one degree and the rest to the desired optimal values. The results for all six degrees of freedom are shown in Fig. 3.8, where the two poses under the positive and negative maximum input test errors are illustrated above the evaluation images for each degree. As before, the top row shows the Euler rotations under ZYX convention, and the bottom row depicts the translations. As for the convergence rate test, the three error metrics, i.e. ICP, NSSD and LKICP, are compared.

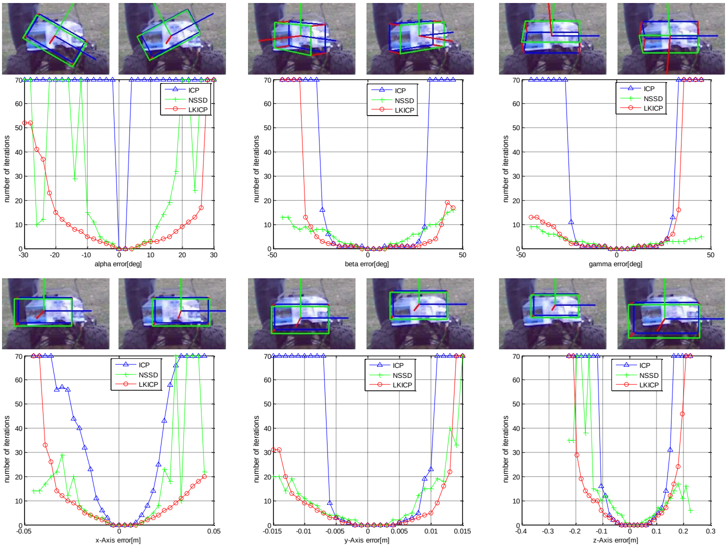


Figure 3.8: Convergence stability evaluation. The six figures illustrate the number of iterations required for handling the input errors for each degree separately. Three methods - ICP, NSSD and LKICP - are compared. The X and Y axes represent the initial input error for one freedom and the corresponding required number of iterations respectively. Above each figure, two target images are given for illustrating the tested positive and negative extreme input errors.

Among the three compared methods, the ICP yields the worst performance, especially for the rotation around Z axis. For some degrees of freedom, the NSSP shows comparable results as the LKICP. But in general, the proposed LKICP demonstrates the best convergence rate and convergence stability. In most cases, the input errors can be effectively tackled within 15 iterations. Therefore, the maximum number of iterations is set to 15 for most of the tests in this work. The results in this subsection also imply the necessity of a good initial pose guess, otherwise the algorithm still cannot converge to the desired pose. The pose output from the coarse estimation stage can provide the required initial accuracy. Details will be further discussed in Chapter 5.

3.5 Summary

This chapter presents the accurate pose estimation stage, which takes the pose from the coarse estimation stage as the initial pose guess and outputs a refined pose through a gradient based iterative optimization procedure. The major innovation in this chapter is the incorporation of the target appearance (texture) information into the conventional point-to-plane ICP framework with projective data association. The derivation of the texture consistency in the proposed Textured-ICP is closely connected to the Lukas-Kanade method for solving the optical flow. Therefore, it is also denoted as the LKICP in short for Lukas-Kanade ICP. The proposed LKICP exhibits several advantages over the conventional ICP: the ability to deal with geometrically symmetric object; better performance under range measurements contaminated by motion artefacts; faster convergence and better tolerance to initial errors. Despite of the additional computations compared to ICP, the real-time performance can still be achieved by making use of GPU acceleration.

Chapter 4

Handling Illumination Variation

The coarse and the accurate pose estimation algorithms proposed in previous chapters all make use of the target appearance for aligning the model and the observation data. It is well known that the appearance of an object can vary significantly under different lighting conditions, which may cause great difficulty for a vision based object recognition task. Therefore, to achieve robustness under the illumination variations, such appearance changes must be taken into account. This chapter aims at modeling the illumination variation into the pose estimation algorithm by using the theoretical works for illumination modeling with the spherical harmonics. First, the problem will be discussed in detail and the related works on this topic will be introduced. This is followed by some theoretical background knowledge used in this chapter, which can help the description of the algorithm to be proposed. Then some surface reflectance estimation ideas are evaluated and the estimated reflectance is used in the following illumination invariant pose estimation algorithm. In the end, the work is summarized.

4.1 Problem Statement and Related Works

This section clarifies the problem to be solved and describes the contributions made on the illumination invariant pose estimation. Some state-of-the-art works in the literature are introduced, which are related to the illumination invariant tracking, the illumination modeling methods and the inverse lighting algorithms. The introduction to the related works not only provides an overview for the illumination related vision topics, but also motivates the research done in this chapter, for example, the spherical harmonic illumination modeling and the vision based reflectance estimation methods.

4.1.1 Problem Statement

The target appearance plays an important role in the pose estimation algorithms proposed in previous chapters. As described in algorithm workflow in Fig. 1.1, the target model data is initialized with the information from one captured fused RGBD image. During tracking, the target appearance model is compared with the live observed appearance in each frame. However, the object appearance is greatly influenced by the illumination conditions. As an example, Fig. 4.1 shows four images of the same person from the same viewing angle but varying lighting conditions. Under this condition, it is even hard for human eyes to identify whether it is the same person or not. Although this is an extreme case that usually will not be encountered in most scenarios, it gives an impression regarding how much appearance changes different lighting can bring about.

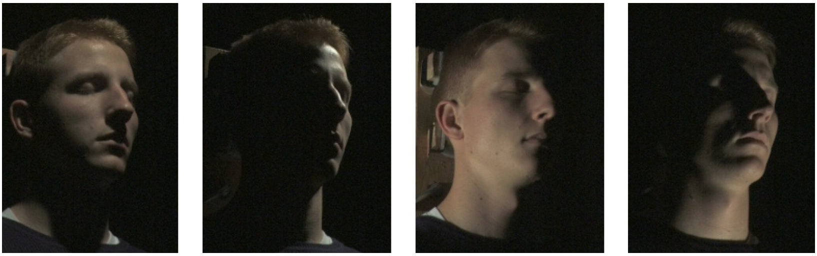


Figure 4.1: Images of the same human face obtained from the same viewing angle but various illuminations. Images courtesy of [130].

In some state-of-the-art researches, the target model data are updated with the information grabbed from recent frames during tracking [103, 137]. In this way, the appearance changes can be gradually accumulated. The problem for such a scheme is that when the target position cannot be accurately estimated, which is the case for most practical applications, the tracking inaccuracy will also be accumulated and propagated from frame to frame. Different from the above online target model updates, the updates described in the workflow in Fig. 1.1 are performed over the background model in sparse representation. The inaccuracy of the target pose will not contaminate the target model nor the background model as detailed in Subsection 2.3.3. The drawback is that it cannot account for the target appearance changes caused by the illumination.

However, for the coarse pose estimation stage presented in Chapter 2, as specified in the template matrix composition for the sparse representation in Fig. 2.4, the target appearance variation caused by illumination can be built into the target model if it can be expressed with a fixed low-dimensional subspace. Such a subspace can be interpreted as the target intrinsic properties, which is irrelevant to the ambient illuminations. Likewise, in the accurate pose estimation stage using the Textured-ICP in Chapter 3, some synthetic images constructed with the subspace can be used for calculating the cost function in the Textured-ICP instead of using the original appearance values initialized before the tracking.

Based on the above analysis, the problem is formulated as:

- **Model determination**. Find an appropriate algorithm that can model the illumination variations with a low-dimensional subspace. The subspace should represent the intrinsic characteristics of a specific object and can model most of the illumination conditions.

- **Model parameter estimation**. Estimate the parameters required for the illumination model in the initialization stage with as less frames as possible (in the ideal case, one initialization frame is desired).

- **Integration and tests**. Integrate the illumination model into the framework of the pose estimation algorithms presented in previous chapters and test the improvement of the pose estimation under various lighting conditions.

The contributions of the work in this chapter can be summarized into several aspects. The state-of-the-art works regarding illumination invariant tracking, illumination modeling and inverse lighting are reviewed in depth, which can help the interested readers to build the horizon in this field. The idea for estimating the reflectance in the visible spectrum with the reflectance in the infra-red spectrum is briefly discussed and a theoretical LED array model is evaluated for the active infrared lighting on the PMD camera. Although results show that due to practical reasons (e.g. the manufacturing and assembling inaccuracies) the theoretical model cannot accurately approximate the spatial intensity distribution for the LED arrays on the PMD camera, this work can give the other researchers some hints with respect to what is realistic and what is not for the current theory and real hardware. The performance of the Spherical Harmonic (SH) illumination modeling with the PMD measurements is investigated, for which the required surface reflectance is estimated with a calibration object. Then the SH model is incorporated into the pose estimation framework proposed in previous chapters and tested on real objects in the 3D video tracking scenarios with significant illumination variations.

4.1.2 Related Works

Focusing on the illumination and object appearance related topics, this section introduces some state-of-the-art works in this realm that are closely connected to the algorithm to be presented in this chapter. It includes a brief overview of illumination invariant tracking, theories and methods for modeling the illumination and reflectance, as well as the approaches for estimating the parameters required in the theoretical models.

Illumination Invariant Tracking

Tracking has been vastly investigated in computer vision community. Here only a brief overview of the approaches will be discussed, which are reported or expected to be robust under illumination variations. They can be categorized into 2D feature based, 2D template based, and 3D range data based methods.

One of the most prevalent features used for tracking and registration is the SIFT feature (Scale-Invariant Feature Transform) from [99]. It uses the Difference of Gaussian (DOG) for determining the feature location and scale. Due to the use of a histogram of gradient directions in the neighborhood as the feature descriptor, it can largely deal with intensity variations caused by illuminations, therefore achieves robustness under varying lighting. Similarly, the SURF (Speeded-Up Robust Feature) feature, developed from SIFT [10], by taking advantage of the integral image and using Haar wavelet response in the neighborhood as the feature descriptor, also provides a robust feature under illumination changes. Another widely adopted feature is the HOG (Histogram of Oriented Gradients) feature from [37], which also builds histograms for gradients and is expected to be illumination invariant.

Although being efficient in term of computation, feature based methods often bear the drawback of being sensitive to the image blur, which is frequently encountered in real video sequences. In contrast, the template based approaches usually yield a more reliable performance. However, the direct template matching will be influenced by the intensity variations. To tackle this, Probabilistic Principal Component Analysis (PPCA) [150] can be used to extract the statistical principal subspace for the target appearance, thus can be applied on face recognition [108]. PPCA is further extended to incrementally learn the subspace with the online data for the illumination invariant target tracking [137]. The online learning strategy was also adopted in [103], where the grabbed target image patches were accumulated into the template matrix for the sparse representation, which made the method capable of accommodating to the gradual changes caused by illumination.

Derived from *Kullback-Leibler divergence*, Mutual Information (MI) provides a metric for evaluating the similarity between two probability distributions by building the joint distribution [14]. When applied in vision applications, MI can be used to conduct appearance based 3D registration. For example, [123] used MI to measurement the distance of two intensity histograms from the model and the observation images. Due to the statistic measurement (in the form of intensity histograms) adopted in their method, the illumination variations will not change the sparseness in the joint-histograms matrix. Therefore, the MI based 3D tracking is more robust under varying lighting conditions compared to the Sum of Squared Difference (SSD) or the Normalized Cross-Correlation (NCC) metric based approaches.

For a Lambertian object, the spherical harmonics can be used to accurately approximate the reflected light with as low as nine basis images. [170] integrated the basis images and the motion effects for 3D tracking. They derived that the appearance of a moving object lies closely in a bilinear subspace defined by motion variables and spherical harmonic light coefficients. Their method took some monocular 2D image sequences and the target 3D model (with reflectance information) as inputs. Through an iterative or bootstrap procedure, it can output simultaneously the estimated light coefficients and the 6DOF pose.

Contrary to the appearance based approaches, range data is independent upon the lighting condition. Especially for the measurement from TOF cameras that use active illumination and employ Suppression of Background Illumination (SBI), the range measurements are not influenced by the ambient lighting, therefore can be used for illumination invariant applications. [155] took advantage of the range measurements for determining the scale parameter and extended the SIFT feature for pose estimation. For a planar object, edge features can be combined with the 3D plane fitting to yield a robust and accurate pose estimate [134].

Since its introduction, ICP (Iterative Closest Point) became the dominant method for registering a set of observation range data to the 3D model points. [118] used ICP on point clouds in a large volume and applied on the camera pose estimation as well as on online modeling. ICP can also be extended to track articulated objects as in [162]. Although invariant to ambient illumination, the range data based ICP cannot tackle geometrically symmetric objects.

Illumination Modeling Methods

Illumination modeling refers to the methods characterizing the appearance variation caused by illumination changes. It is a fundamental problem in computer vision and has been vastly studied for decades. Due to the complexity of lighting conditions, object shapes, and material configurations, assumptions are usually made to simplify the problem. For instance, the incident light is often modeled as from one of or a combination of several simple source types, e.g. the directional light source, the point light source, or the area source with/without sharp edges [181]. Meanwhile surface reflectance properties are modeled as a product of the Bidirectional Reflectance Distribution Function (BRDF) describing the light scattering property and the texture addressing the light absorbance ratio of the surface. Some BRDFs commonly used in the photo-realistic rendering include mirror BRDF, Lambertian BRDF (homogeneous for perfect diffuse situation), Phong BRDF (integrating ambient, diffuse and specular reflection into one model) and Torrance-Sparrow BRDF (using microfacets for specular component), etc.

Empirical works demonstrated that by decomposing a group of images of a Lambertian object captured under the same pose but varying distant illuminations, the appearances of an approximate convex Lambertian object can be modeled by a low-dimensional subspace with high accuracy [178]. Thus a low-dimensional basis suffices to model the illumination variations. This observation was proved by [12], showing when shadow is not considered, the set of images of n pixels, although forms a convex cone (which they dubbed as the *illumination subspace*) in $\mathbb{R}^n$, in general lies in a 3-Dimensional subspace. By taking the *attached shadow* into account, they further introduced the concept of *illumination cone*, which is a convex polyhedral cone and can be obtained by setting the negative pixel values to zero in the images from the illumination subspace. The illumination sphere, as well as the illumination cone of a convex Lambertian object, can be constructed from as few as three images without shadow. [58] applied the illumination cone on face recognition under varying lighting.

By assuming a distant lighting, [131] adopted the *Spherical Harmonics* (SH) to model the reflectance function for objects with arbitrary but homogeneous BRDF. They demonstrated that the reflected light can be interpreted as an incident light convolved with a transfer function (a product of the clamped cosine and the textured BRDF). Focusing on convex Lambertian surfaces, [9] and [130] gave more detailed derivation and showed the clamped cosine function works as a low pass filter. By using as few as nine SH basis images, 98% energy of the incident lighting can be captured, which means the SH explicitly provides a low-dimensional basis sufficient to model the illumination variation of a Lambertian object with high accuracy. Analytic study in [129] linked the empirical low-dimension subspace observation with the SH illumination modeling theory by applying PCA on a dense (continuous) set of images constructed from the SH basis functions instead of on a set of sampled real images.

They showed that under appropriate assumptions, the eigenvectors and eigenvalues are equivalent to the SH basis images and SH coefficients respectively.

Above SH modeling works in high accuracy on objects with convex Lambertian surfaces. However, when non-convexity comes to stage, the *cast shadow* cannot be neglected and using low order SH basis images will result in large residuals. In the presence of cast shadows, [119] researched on the wavelet bases that can provide a better approximation to the real lighting with a small number of wavelet basis functions. However, the fixed harmonic basis images are calculated with reflectance and geometry information, whereas the wavelet basis is determined through analyzing the captured images. When the lighting has changed, the basis will also be altered. Therefore, it will involve a lot of computation and prevent from real-time applications.

Inverse Rendering - Reflectance and Illumination Estimation

Illumination modeling provides physical schemes for rendering photo realistic images of an object. Before the model can be deployed, the model parameters need to be determined, e.g. the lighting condition, the texture and the BRDF. This can be done by a direct measurement with dedicated instruments. But more adopted in the computer vision field are methods that can estimate the parameters through a set of captured photographs of the target or the scene under consideration, which is called inverse rendering.

Due to the complication of the inverse-rendering problem, e.g. the generalized bas-relief ambiguity in [11] when the object geometry is unknown, or the ambiguity between light and BRDF, assumptions or experimental simplifications are usually made to make the problem tractable, for example, assuming a single directional light source and homogeneous reflectance as in [69, 168], using a large number of photographs captured under known poses as in [105, 177], assuming the presence of specular points [61], incorporating the prior statistic knowledge about the target in [15], using a calibration object [181], or some other heuristic approaches [184], etc.

As introduced in the previous contents, SH provides a powerful tool for modeling the reflectance function under the distant lighting assumption, it leads to a natural way to inverse the procedure and recover the lighting or BRDF when the reflected light filed is known. [131] used the SH model to factorize the lighting and BRDF in the frequency space. Their method can estimate the lighting and arbitrary BRDF simultaneously when the parameters for both are unknown. However, the calculation requires SH coefficients for the reflected light, which in their paper was acquired by using 60 densely sampled photographs with known camera poses. This limited its use to the laboratory experiments.

Due to the fact that the specular highlights usually result from reflecting the high frequency part of the incident lighting, they are often used for estimating a point light source with the assumption that at least one specular point is visible in the query image. For instance, by assuming a single distant light source and homogeneous reflectance, [69] estimated the light direction and reflectance parameters by segmenting the Lambertian and specular components.

The distant light source assumption in SH modeling was removed in [61] by using both homogeneous diffuse and Torrance-Sparrow specular models. The Lambertian and specular components were separated by a *polarization filter*. The light source distance and the reflection parameters were recovered in an iterative manner. With a homogeneous specular reflectance assumption, the algorithm can be further extended for recovering the distances of multiple point light sources by using only the specular components. A similar approach in [62] also worked on the specular point sets to estimate the specular reflectance parameters without the single light source assumption. They provided the *spherical specular reflection model* based on directional statistics to approximate the original Torrance-Sparrow model. An initial estimate of the specular reflectance was obtained by using the spherical specular reflection model. These initial parameters were then refined to get parameters for the original Torrance-Sparrow model. Although multiple light source directions can be recovered, they assumed that all light sources had the same properties and only differed from each other by their directions.

Apart from relighting, specularities can also be applied on recognition tasks. In [121], the specular or glossy points usually considered as noise were exploited for the object recognition. They worked on a single light source, for which the direction should either be roughly known a priori for a purely specular surface or the reflectance is known for a Lambertian object. When the Lambertian component is also available, the rough light source direction can be estimated with the help of SH modeling. In their work, both the light source and the viewing point were assumed distant. They also provided an approach to recover the Lambertian surface reflectances with two input images. However, the approach was highly heuristic, and yielded only a coarse approximation to the underlying reflectance.

Besides searching for the specular points in the scene, it was also reported in several researches to use the known information in the scene or to use a calibration object for the inverse lighting. [160] used objects or structures in the scene with a known geometry and homogeneous Lambertian reflectance and integrated the information from shades and shadows for estimating multiple directional light sources.

[181] used a calibration sphere with both specular and Lambertian components to perform the inverse lighting. They also proposed a light model, which can incorporate point light sources, directional light sources, and area light sources. The light source direction was recovered by the specular component of the calibration sphere and the light intensity was estimated with the diffuse component.

In some cases, e.g. in the human face recognition, the reflectance estimation for an individual subject can be performed when the statistic model of the objects in question are available. [15] and [157] researched on the optimal reflectance estimation with Wiener filter, by which the prior information and the observation were fused. [179] proposed to combine the statistical model of human faces with the SH modeling, where the reflectance for the observed subject was acquired with bootstrap steps.

For the absence of neither prior knowledge nor the calibration objects, some heuristic approaches were proposed to human face reflectance recovery. [67] proposed to apply PCA on a large number of training images obtained by illuminating faces with floodlight from different directions and determine the most intrinsic face images and their corresponding lighting conditions. The lighting conditions were then clustered to get the basis point light sources for synthesizing a virtual image under arbitrary illuminations. In [184], the spatial consistency of the reflectance on human faces was leveraged. Based on spherical harmonics, their approach estimated the reflectance and the lighting coefficients in an iterative manner with an initial guess of the reflectance map. But there was no guarantee for the heuristic methods in general that the iteration would converge to the underlying true reflectance.

A lot of inverse rendering researches based their methods on the use of SH modeling. However, [119] pointed out that for estimating the high frequency part of the lighting, a large number of basis functions were required (similar to the Fourier expansion). They argued that the clamped cosine function for a Lambertian surface will not be a low pass filter in the cast shadow region and therefore cannot recover the high frequency part of the lighting. Furthermore, they demonstrated that the wavelet could be an alternative basis for the light recovery, which yielded a better accuracy than using SH with respect to estimate the high frequency components. This was further confirmed by the works in [156]. Upon the observation that the cast shadow will only appear when the directional light sources are sparse, [104] utilized sparse representation for lighting recovery from cast shadow regions.

[169] proposed a method to recover the BRDF and multiple light sources simultaneously with stereo vision. They recovered the Phong-Blinn BRDF and required the stereo pair having the same intrinsic parameters. Their method removed the dif-

fuse component and estimated the lighting and specular reflectance in the first phase. Then with the estimated lighting location and intensity, the diffuse component was obtained in a second phase. Although they claimed "one shot" estimation (only one viewpoint stereo images were required) and no intervention on the scene, they do need the number of light sources were known a priori, and restricted the light recovery on point sources.

Previously introduced inverse rendering researches were all based on homogeneous BRDF assumption, i.e. one BRDF model was applied on the complete object surface. However, there are some objects that are composed of different surface materials, thus cannot be assumed homogeneous. [105] relieved this restriction by segmenting and classifying the 3D surface voxels into different BRDF types, for which the recovery was done separately. But the approach required a large number of images captured with known viewpoints.

4.2 Theoretical Background

In this subsection, the theories related to Light Emitting Diode (LED) array modeling and illumination modeling with Spherical Harmonics (SH) are introduced. The LED array model will be used in Subsection 4.3.1 for analyzing the intensity distribution of the LED array mounted on the PMD camera. The SH illumination model will be used in Subsection 4.3.2 for the reflectance estimation and in Sec. 4.4 for the pose estimation.

4.2.1 LED Array Modeling

The spatial intensity distribution of a LED can be very different from a point light source. If the intensity along the optical axis of a LED at distance r is denoted as $E_0(r)$, the spatial distribution can be approximated as [114]:

$$E(r, \beta) = E_0(r) \cos^m(\beta),$$

where β is the angle between the optical axis and a space point. Point light source is usually considered as a Lambertian light with exponent round $m = 1$, and a typical LED would have value $m > 30$. This implies the LED light can be very directional like Fresnel spotlight used for the opera lighting [43].

The "inverse square law" is widely adopted to model $E_0(r)$ [120], and the exponent m is determined by the half intensity angle $\beta_{1/2}$ (the viewing angle where the LED light intensity is half the magnitude at angle 0°) as [113]:

$$m = \frac{-ln2}{ln(\cos\beta_{1/2})}. \tag{4.1}$$

When the interested scene is largely planar and perpendicular to the optical axis of the LED, Eq. (4.1) can be approximated for computational convenience [174, 175]. In [112], the author provided more detailed formulation for modeling the roughness of the chip faces, the encapsulating lens, and the reflecting cup also into the LED model.

In this thesis, it is of greater interest of the overall intensity distribution of a LED array because the complete spatial intensity distribution from the illumination source on the PMD camera can be obtained as a superposition of the lights from the two separate LED arrays on each side of PMD camera lens. [113] studied the influence of different array patterns, e.g. the linear array, the ring array, the square array, etc., on the intensity distributions. However, [111] pointed out when a far-field condition is met, the spatial intensity distribution can be modeled as a single directional point source. Here the far-field refers to the distance, over which the measured *radiant intensity* (radiant flux per solid angle) is practically independent upon the distance from the source. The far-field condition can be determined by the array pattern together with the half intensity angle of the individual LEDs in the array.

In the far field region, where the inverse square law applies [111], the spatial intensity distribution of the LED arrays on PMD camera can be modeled as a linear combination of the lights from both illumination units:

$$L = k\frac{\rho I_0}{\pi}\sum_{i=1,2}\frac{1}{r_i^2}\cos(\theta_i)\cos^m(\beta_i), \tag{4.2}$$

where r_i is the distance between the center of the i-th illumination unit and the interested surface point, θ_i is the angle between the surface normal on the interested point and the light direction, β_i is the angle between optical axis of the i-th illumination unit and the light direction.

4.2.2 Illumination Modeling with Spherical Harmonics

The Lambert's cosine law states that the reflected light intensity on a surface point P_i can be model as:

$$\mathbf{I}_i = \rho_i l(\mathbf{u}_l) max(\cos\theta'_l, 0),$$

where ρ_i is the reflectance on the point, $l(\mathbf{u}_l)$ is the intensity of the incident light in direction $\mathbf{u}_l$, and θ_l' is the angle between the incident light $\mathbf{u}_l$ and the surface normal $\mathbf{n}_i$ on point P_i. By integrating over all possible light directions, the overall reflected light intensity can be written as [9]:

$$\mathbf{I}_i = \rho_i \int_{S^2} l(\mathbf{u}_l) max(\cos\theta_l', 0) d\mathbf{u}_l,$$

where S^2 represents the surface of a unit sphere. If define a transfer function $k(\theta_l') = max(\cos\theta_l', 0)$, the above formulation can be viewed as a convolution defined on a sphere

$$\mathbf{I}_i = \rho_i l * k = \rho_i \int_{S^2} l(\mathbf{u}_l) max(\cos\theta_l', 0) d\mathbf{u}_l. \tag{4.3}$$

Analogous to the Fourier transform applied on circle, spherical harmonics is a powerful signal processing tool applied on sphere, by which a function $f(\mathbf{u})$ defined on the unit sphere can be decomposed as:

$$f(\mathbf{u}) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} f_{n,m} Y_{n,m}(\mathbf{u}),$$

where $f_{n,m}$ are the spherical harmonic transform coefficients for $f(\mathbf{u})$, and $Y_{n,m}(\mathbf{u})$ are the basis functions. The coefficients $f_{n,m}$ are obtained by

$$f_{n,m} = \int_{S^2} f(\mathbf{u}) Y_{n,m}^*(\mathbf{u}) d\mathbf{u}.$$

By separating the azimuth angle ϕ and the zenith angle θ in $\mathbf{u}$, the basis function can be factorized into two parts $Y_{n,m}(\theta, \phi) = g_{n,m}(\theta) e^{im\phi}$.

Applying the spherical harmonic transform, the light $l(\mathbf{u}_l)$ is decomposed by:

$$l(\mathbf{u_l}) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} l_{n,m} Y_{n,m}(\mathbf{u}_l). \tag{4.4}$$

The transfer function $k(\theta_l')$ has no dependence on the azimuth angle ϕ, the coefficients will vanish as $k_{n,m} = 0$ for $m \neq 0$ [130], for which the spherical harmonic transform is

$$k(\theta_l') = \sum_{n=0}^{\infty} k_n Y_{n,0}(\theta_l'), \tag{4.5}$$

where k_n are constant real numbers.

Combining Eq. (4.3), (4.4), and (4.5), [130] and [9] used different techniques and derived the same formulation for decomposing the intensity of the reflected light

$$\mathbf{I}_i = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \alpha_n l_{n,m} Y_{n,m}(\mathbf{n}_i),$$

where $\alpha_n = \sqrt{\frac{4\pi}{2n+1}} k_n$. Combining the basis function $Y_{n,m} = g_{n,m}(\theta_i) e^{im\phi}$ and $e^{im\phi} = \cos(m\phi) + i \sin(m\phi)$, it yields:

$$\begin{aligned}
\mathbf{I}_i &= \rho_i \sum_{n=0}^{\infty} \alpha_n \sum_{m=-n}^{n} L_{n,m} g_{n,m}(\theta_i)(\cos(m\phi) + i\sin(m\phi)) \\
&= \rho_i \sum_{n=0}^{\infty} \alpha_n \{ l_{n,0} g_{n,0}(\theta_i) + \sum_{m=1}^{n} l_{n,m} g_{n,m}(\theta_i)[\cos(m\phi) + i\sin(m\phi)] \\
&\qquad\qquad\qquad + \sum_{m=-1}^{-n} l_{n,m} g_{n,m}(\theta_i)[\cos(m\phi) + i\sin(m\phi)] \} \\
&= \rho_i \sum_{n=0}^{\infty} \alpha_n \{ l_{n,0} g_{n,0}(\theta_i) + \sum_{m=1}^{n} (l_{n,m} + l_{n,-m}) g_{n,m}(\theta_i) \cos(m\phi) \\
&\qquad\qquad\qquad + \sum_{m=1}^{n} i(l_{n,m} - l_{n,-m}) g_{n,m}(\theta_i) \sin(m\phi) \}. \qquad (4.6)
\end{aligned}$$

Here $g_{n,m}(\theta_i)$ is even with respect to m [9]. Analogous to the Fourier transform, by separating the complex light coefficient $l_{n,m}$ to be the even $l^e_{n,m}$ and the odd $l^o_{n,m}$ parts as $l_{n,m} = l^e_{n,m} + il^o_{n,m}$, and combining $l_{n,m} + l_{n,-m} = 2l^e_{n,m}$ and $i(l_{n,m} - l_{n,-m}) = -2l^o_{n,m}$ into the above equation, the complex term vanishes and yields:

$$\mathbf{I}_i = \rho_i \sum_{n=0}^{\infty} \sum_{m=-n}^{n} B_{n,m}(\mathbf{n}_i) L_{n,m},$$

where $B_{n,m}(\mathbf{n}_i)$ and $L_{n,m}$ are real numbers. [130] pointed out that upto an order of $n = 2$, the approximated accuracy can be at least 98%. $n = 0 \sim 2$ forms 9 basis images for reconstructing the reflected light

$$\mathbf{I}^{p\times 1} \approx \mathbf{D}_{\rho}^{p\times p} \mathbf{B}^{p\times 9} \mathbf{L}^{9\times 1}, \qquad (4.7)$$

where p is the number of pixels, $\mathbf{D}_{\rho}^{p\times p}$ is a diagonal matrix with the reflectance of all points on its diagonal. $\mathbf{B}^{p\times 9}$ contains nine albedo free basis images in each of its columns, and the vector $\mathbf{L}$ contains the coefficients for the corresponding basis images. The nine lower order basis images are given by:

$$\begin{aligned}
B_{0,0}(\mathbf{n}) &= \pi Y_{0,0}(\mathbf{n}) = \sqrt{\frac{\pi}{4}} \\
B_{1,0}(\mathbf{n}) &= \frac{2}{3}\pi Y_{1,0}(\mathbf{n}) = \sqrt{\frac{\pi}{3}}z \\
B_{1,-1}(\mathbf{n}) &= \frac{2}{3}\pi Y^o_{1,1}(\mathbf{n}) = \sqrt{\frac{\pi}{3}}y \\
B_{1,1}(\mathbf{n}) &= \frac{2}{3}\pi Y^e_{1,1}(\mathbf{n}) = \sqrt{\frac{\pi}{3}}x \\
B_{2,0}(\mathbf{n}) &= \frac{1}{4}\pi Y_{2,0}(\mathbf{n}) = \sqrt{\frac{5\pi}{256}}(3z^2-1) \qquad (4.8) \\
B_{2,-1}(\mathbf{n}) &= \frac{1}{4}\pi Y^o_{2,1}(\mathbf{n}) = \sqrt{\frac{15\pi}{64}}yz \\
B_{2,1}(\mathbf{n}) &= \frac{1}{4}\pi Y^e_{2,1}(\mathbf{n}) = \sqrt{\frac{15\pi}{64}}xz \\
B_{2,-2}(\mathbf{n}) &= \frac{1}{4}\pi Y^o_{2,2}(\mathbf{n}) = \sqrt{\frac{15\pi}{64}}xy \\
B_{2,2}(\mathbf{n}) &= \frac{1}{4}\pi Y^e_{2,2}(\mathbf{n}) = \sqrt{\frac{15\pi}{256}}(x^2-y^2),
\end{aligned}$$

where the normal is expressed in Cartesian coordinates $\mathbf{n} = [x, y, z]^\top$. The image sensed by the camera will be a scaled version of the reflected light intensity.

By using Eq. (4.7), the image of an object can be reconstructed by a combination of positive and negative lightings, which are physically unrealistic. [9] proposed an approach to enforce a nonnegative light constraint by approximating the lighting as a group of uniformly sampled positive lighting on an unit sphere.

4.3 Reflectance Estimation

Among the three most influential factors on the appearance of an object, i.e. surface geometry, surface reflectance, and illumination condition, the illumination is the one that we do not have accurate control in most real lie applications. Surface geometry can be determined with the help of a range sensor as well as a pose estimation algorithm. When the surface reflectance can be obtained, the accuracy of the synthetic image generated under some realistic illuminations can be evaluated by comparing with the real captured image. Such an evaluation can produce useful implications on object recognition under arbitrary and unknown illuminations. Therefore, the reflectance estimation plays an important role in the appearance based illumination invariant object recognition and tracking.

It is ideal to estimate the surface reflectance without any intervention or prior knowledge on the scene, i.e. no modification nor control to the illumination, no special requirements (available shadow or specular points, no particular calibration objects) on the scene. It is also desirable that the estimation can be done with one shot, i.e. no needs for multiple training images captured either from different viewing angle but under the same lighting, or from some different camera view points but with varying illuminations. With these analysis, the following two tests for reflectance estimation have been performed.

4.3.1 Trial with LED Array Modeling

This section briefly introduces an attempt work to estimate the surface reflectance with LED array modeling. The motivation for performing such a research and developing the method for evaluating the theoretical LED array model on the real LED array on PMD camera are discussed. Although the verification results showed that the LED array model is not accurate enough for further investigation on the reflectance estimation, the work done in this section still provides useful information for interested readers who might have similar ideas. Or in the future, when the LED array on the PMD camera can be assembled more precisely or when the theoretical LED array model has improved, the ideas presented in this section can still be a potential research direction.

Motivation

PMD cameras take two Near Infra-Red (NIR) LED arrays as its active illumination unit. The amplitude image is thoroughly determined by the NIR LED arrays and is not influenced by the ambient lighting, which can be shown in Fig. 4.2. When both LED arrays are covered, the amplitude image will be completely dark. Although the ambient light (especially the sun light) comprises the light components in the NIR spectrum as well, the light emitted from the LED array on the PMD camera is modulated and the measurement is reported to be invariant to the ambient lighting due to Suppression of Background Illumination (SBI) function. This means when the spatial intensity pattern of the LED arrays can be accurately modeled, the lighting condition for the amplitude image will be known. Furthermore, when the amplitude image, the lighting condition, and the surface geometry information are all available, the surface reflectance in the NIR spectrum can be estimated with the spherical harmonic illumination model.

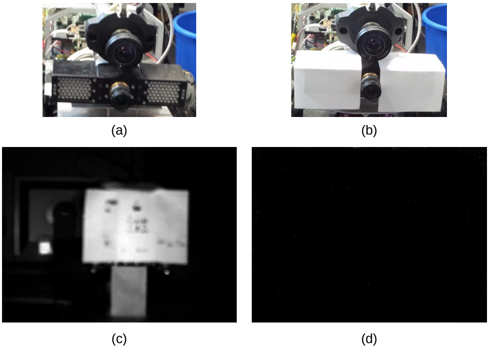


Figure 4.2: PMD amplitude images with and without LED arrays illumination. The amplitude images illuminated with and without the mounted infra-red LED arrays are shown on the bottom left and the bottom right images respectively.

The algorithm proposed in previous chapters uses the image captured with the color camera, because current color camera is more mature and can provide a high quality image with a high resolution. More importantly, the object appearance in the color image will not vary with the distance from the camera to the object, which is a crucial factor for an appearance-based object recognition in the dynamic scenarios. The illumination invariant pose estimation algorithm to be presented in this chapter relies heavily on the reflectance information in the visible spectrum. However, the aforementioned reflectance is the surface characteristic in the NIR spectrum. A natural question arises: will the reflectance information from the NIR spectrum help the reflectance estimation in the visible spectrum?

In Fig. 4.3, two reading lamps were place closely on both sides of the AXIS camera, so that the illumination direction is similar to the LED arrays on PMD camera, and the amplitude image from PMD camera and intensity image in red channel from AXIS camera for doll dwarf are displayed. Under this configuration, despite the huge difference between LED lighting and reading lamp and the slight directional difference between LED lighting and reading lamp as well as difference between viewing direction, the similarity between two images implies knowing reflectance under one spectrum can be helpful for determining reflectance under the other spectrum.

Unfortunately, for an arbitrary object, there is no rule of thumb available for the relation between reflectances under different spectrums. For example, the spectral

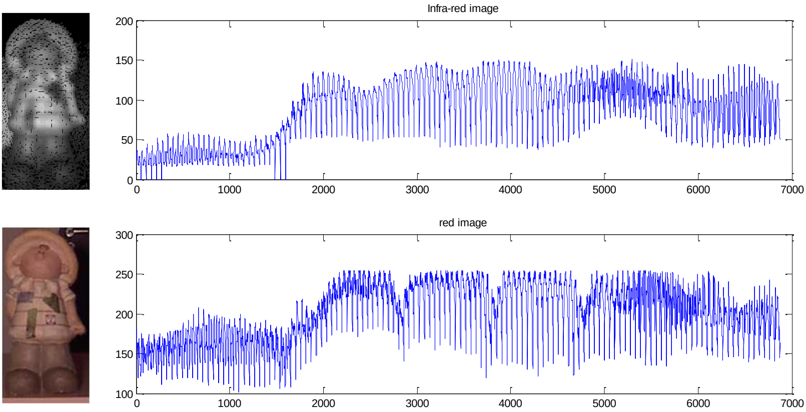


Figure 4.3: Images from PMD and AXIS Under Similar Lighting Directions. Top row displays amplitude data from PMD camera; bottom row is for data from AXIS camera. 2D images are shown on the left, and 1D vector form of 2D image is displayed on right. The 1D illustration of intensity image captured by AXIS is composed of red component only.

reflectances for two rocks are displayed in Fig. 4.4. The reflectances in the visible spectrum are quite similar for both rocks, but they diverges greatly in the NIR spectrum. For Rock 1, the reflectance between the NIR and red spectrum are quite similar. Whereas Rock 2 shows rather different NIR and red reflectances. Although this is an extreme case, it gives us the insight regarding how complex the problem can be and reflectance under NIR spectrum cannot be directly used as the reflectance under other spectra. More sophisticated strategies are required and appropriate assumptions need to be made for estimating e.g. red reflectance by taking advantage of the NIR reflectance. For example, assuming target surface is made of a small number of materials, the NIR reflectance can help segmenting different material regions as in [177], or estimating desired reflectance by minimizing the entropy of the reflectance pattern similar to [1].

Evaluation of LED Array Modeling

Before investigating more into the use of NIR reflectance, it is important to ensure that the LED arrays can be accurately modeled and the NIR reflectance can be estimated. Towards this end, the LED array modeling approach introduced in the theoretical background Subsection 4.2.1 is evaluated. Particularly, the image syn-

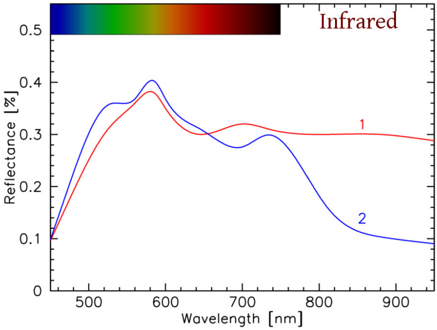


Figure 4.4: Reflectance spectra of two rocks. X axis is the wavelength of light, Y axis shows the reflectance. The red and blue curves are for Rock 1 and Rock 2 respectively. Image courtesy of [151].

thetic method in Eq. (4.2) for modeling the illumination from LED array is verified.

The image reconstruction with the LED array model in Eq. (4.2) requires the information about the half intensity angle for a single LED, the optical axis of the LED array, the distance between the light source and the target surface, the surface reflectance, and the surface normal. The LEDs used on PMD 19K and CamCube2.0 are TSFF5410/TSHF5400 from *VISHAY* company (`http://www.vishay.com/`) with half intensity angle 22°. According to Eq. (4.1), the exponent value m used in Eq. (4.2) is about 9. Moreover, due to assembling inaccuracies, the optical axis of the LED array cannot be assumed parallel to the optical axis of the PMD lens. Therefore, it is manually tuned for the following tests.

The model in Eq. (4.2) applies in the far-field region as discussed in Subsection 4.2.1. The far-field condition is much relaxed for a single LED array than considering both arrays on the PMD camera as one point source. Therefore, the following test is performed on one side of the LED array only, and the other side is covered by a hard paper.

In the test, a white planar board is used which can be assumed to be Lambertian with constant homogeneous reflectance. Distances between the camera and surface points are provided by PMD camera measurements and are used to calculate the distance r_i between the center of single LED array and the surface point. The surface normal for the plane is estimated by using all available points on the board. Combining with the above discussed optical axis, the half intensity angle, etc., all the information are available for applying the image reconstruction model in Eq. (4.2). Results obtained by using PMD 19k with the board placed at different distances, i.e. 100 cm and 150 cm, are shown in Fig. 4.5 and Fig. 4.6.

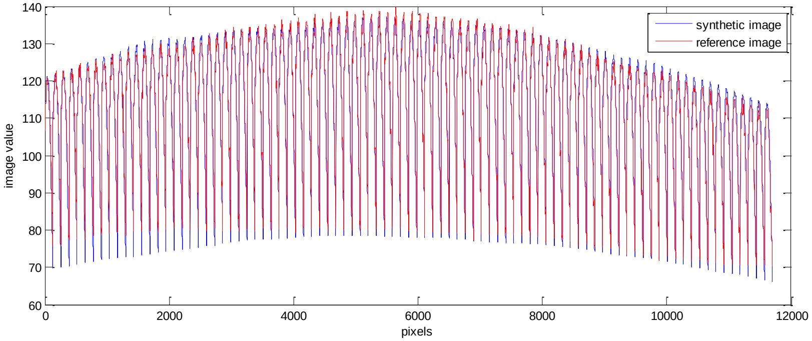


Figure 4.5: Synthetic image values for a white board placed at 100 cm. The synthetic image is obtained with $m = 45$ and a manually tuned direction of the optical axis of LED array. The reference image is captured by PMD 19k.

The white board is tested at multiple distances. However, with $m = 9$ as indicated by the half intensity angle of a single LED, the synthetic image never complies with the real amplitude image captured by PMD 19k. The real image value decreases more rapidly from the image center to the boarder than the synthetic image. Only with increased m values, the synthetic images can be close to the real images. For example, the synthetic image for the white board at 100 cm in Fig. 4.5 is obtained with $m = 45$ for low reconstruction error. However, as the white board is moved further away, the intensity decays again more rapidly from image center to the boarder than the synthetic image with $m = 45$. Meanwhile, at further distances, the region with the highest intensity values in the captured image apparently shifts from the synthetic intensity center. Therefore, both m value and the direction of the optical axis of the LED array should be adjusted. The result for the white board at 150 cm is demonstrated in Fig. 4.6 with $m = 75$ with the optical axis manually probed.

Results for Camcube2.0 are much better than PMD 19k, as in the near ranges (< 1 m), the synthetic image with $m = 9$ and the optical axis of the LED array set to be parallel with the optical axis of lens can be quite close to the real amplitude image. However, it suffers the same problem as PMD 19k that when the white board is moved far away, both m value and the direction of the optical axis need to be

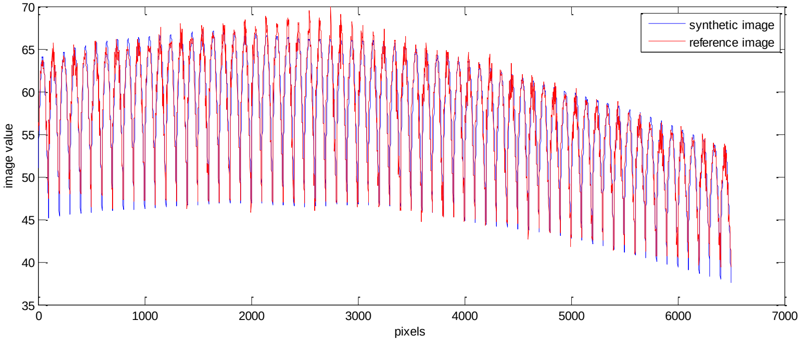


Figure 4.6: Synthetic image values for a white board placed at 150 cm. The synthetic image is obtained with $m = 75$ and a manually tuned direction for the optical axis of the LED array. The reference image is captured by PMD 19k.

reconfigured.

On one hand, with manually adjusted parameters, the image synthesized through modeling the LED array can be quite close to the real image. On the other hand, both the optical axis direction and the m value are required to be reconfigured. Meanwhile, results for CamCube2.0, although still problematic, have been significantly improved compared to PMD 19k. This yields two implications: the LED array model provides useful information for the real LED arrays on the PMD cameras; Either current model misses some important components or the assembling of the LED array is not precise enough as it is practically very difficult to install all LEDs pointing to the same direction as assumed in Eq. (4.2). More results regarding this topic can be found in [49]. Since the LED array cannot be accurately modeled, it cannot help much with the reflectance estimation and other methods should be investigated.

4.3.2 Reflectance Estimation with a Calibration Object

This section describes the reflectance estimation method adopted in this thesis, which is based on the Spherical Harmonic (SH) illumination modeling introduced in the theoretical background in Subsection 4.2.2. The related works in Subsection 4.1.2 describe some reflectance or illumination estimation methods. Some required a large number of images captured under the same lighting condition and varying but known viewing angles [131], or some assumed the knowledge for the number of light sources

[169]. These methods restricted their use in the laboratory experiments. Some took the statistic models of the interested object [15, 157, 179] or required the specular points [61, 62, 69, 121], which can only be applied on some specific objects. In contrast, some methods for probing the lighting condition with a calibration object [160, 181], although also interfering the scene, did not lay much restrictions on the environment nor on the target. Therefore, the scheme is adopted for the illumination estimation.

Method for Reflectance Estimation

Object appearance is dependent upon surface geometry, surface reflectance[1] and illumination condition. The object appearance can be captured from a camera image. The surface geometry can readily be retrieved with the range measurements from a PMD camera. However, the reflectance and the illumination are coupled. Object appearance can be an arbitrary combination of the reflectance and the lighting. Once one is determined, the other can be straight forwardly calculated when combined its captured appearance information.

An object with known geometry and reflectance can thereof be used to estimate the illumination. Then under the same illumination, the reflectance of another convex Lambertion object can be estimated with the SH modeling [9]. To be used in the SH framework, a desirable calibration object should have sufficiently rich surface normals, a convex shape, as well as known homogeneous reflectance for computational convenience. A sphere covered with the white paper satisfies all the above requirements and is therefore adopted as the calibration object. The calibration sphere and the test scene with multiple light sources are shown in Fig. 4.7. Besides the four labeled light sources, the background light from the ceiling is also used in the tests.

The reflectance of a white calibration sphere can be assumed to be one, thus $\mathbf{D}_\rho^{p\times p} = \mathbf{E}^{p\times p}$ in Eq. (4.7). The basis image matrix $\mathbf{B}^{p\times 9}$ can be calculated with Eq. (4.8) from PMD range measurements. The nine SH basis images are illustrated in Fig. 4.8, where the first row shows the $0th$ and the $1st$ order spherical harmonic basis images and the second row displays the $2nd$ order basis images.

After solving Eq. (4.7) with least squares for the calibration sphere, the lighting condition can be obtained in the form of SH coefficients $\mathbf{L}^{9\times 1}$. Although it is a filtered version of the real lighting and can only represent the low frequency components, for a Lambertian object, the filtered lower order nine components already capture

[1] Strictly speaking, reflectance refers to the textured BRDF [131]. Since the SH model also builds BRDF into the model, here the reflectance is only the ratio of the amount of reflected light to the incident light.

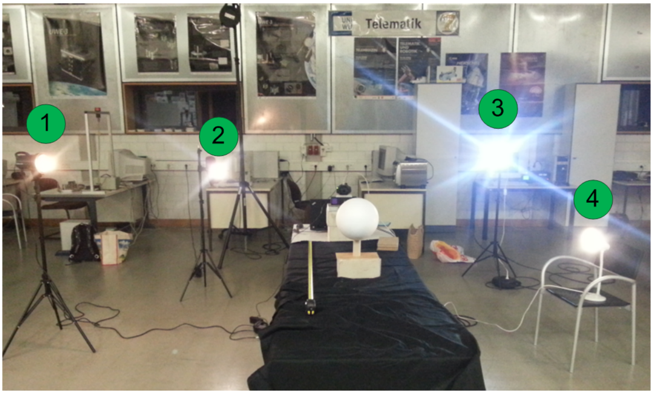


Figure 4.7: Test scene with a calibration sphere and multiple light sources. The four labeled light sources together with the lighting from the ceiling are switched on and off for achieving different lighting conditions.

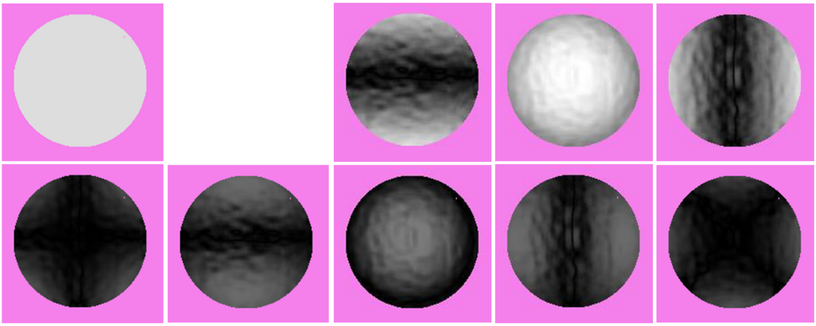


Figure 4.8: Low order nine SH basis images for the calibration sphere.

99.2% of the reflected light energy [130], and therefore are accurate enough for most applications.

Then remove the calibration sphere, and place the interested object where the calibration sphere was located,[1] and the desired reflectance is calculated as

$$\rho_i \approx \frac{\mathbf{I}_i}{(\mathbf{BL})_i},$$

where $\mathbf{I}_i$ is the image intensity for the i-th point on the target, and $(\mathbf{BL})_i$ is the i-th target point intensity in the synthetic albedo free image.

Evaluation

The accuracy of the above reflectance estimation method is evaluated by taking another homogeneous object as the interested target. In following test, a white homogeneous cylinder is used, which is expected to have reflectance one as the white calibration sphere. Although the cylinder does not have sufficiently rich surface normal directions to be applied for inverse rendering problems, e.g. for estimating the lighting condition, the forward rendering derivation of the SH modeling still applies. The estimated reflectance is shown in Fig. 4.9, where most of the surface points on the cylinder have reflectance within the range [0.9, 1.1].[2]

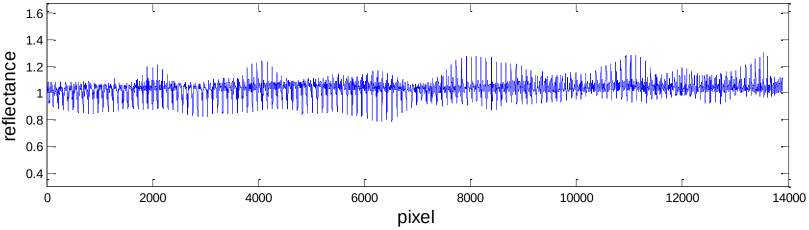


Figure 4.9: Estimated reflectance for a homogeneous cylinder.

Besides the accuracy evaluation for the reflectance estimation, it is more interested whether using the estimated reflectance can improve the performance of object recognition under significant illumination changes. For this purpose, the following test is performed. The target reflectance is estimated under a frontal lighting, so that most of the visible target points can be lit. The normalized target image during initialization is recorded as a 1D vector $\widetilde{\mathbf{I}}_{init}$, and the target illumination model is

[1] Although SH modeling assumes distant lighting and the object appearance only depends on the viewing angle, there are also some lights that cannot be considered as distant, e.g. the multi-reflected light from ground. To be more consistent between the $\mathbf{L}^{9\times1}$ estimated with the calibration sphere and the light used on the target, it is better to place them on the same location.

[2] The estimated reflectance can exceed one because of the inaccuracy from the surface normal estimation.

expressed with the $p \times 9$ matrix $\mathbf{B}_{\rho}^{p\times 9} = \mathbf{D}_{\rho}^{p\times p}\mathbf{B}^{p\times 9}$ in Eq. (4.7). Then remarkably change the illumination condition so that there are obvious target appearance variations. The normalized test image under the new lighting is denoted as $\mathbf{I}_t$. Here the normalization of a vector $\mathbf{x}$ refers to

$$\widetilde{x}_i = \frac{x_i - \mu}{\sigma},$$

where x_i is the i-th component of $\mathbf{x}$. μ and σ are the mean and the standard deviation of all elements in $\mathbf{x}$.

The intensity value $\widetilde{\mathbf{I}}_{init}$ is used in the target model in Chapter 2 and Chapter 3 for evaluating the similarity between the model and a captured image in a video sequence. This has the setback that when the target appearance has changed significantly due to illumination variation or target pose changes with respect to the illumination direction, even the image patch grabbed perfectly from the target can yield a low likelihood. The aim of this recognition test is to verify whether $\mathbf{I}_t$ can be well modeled by the low-dimensional subspace $\mathbf{B}_{\rho}^{p\times 9}$ and whether the recognition performance under the subspace $\mathbf{B}_{\rho}^{p\times 9}$ can notably outperform the results from using $\widetilde{\mathbf{I}}_{init}$.

More concretely, the performance of the subspace $\mathbf{B}_{\rho}^{p\times 9}$ is evaluated by first calculating the synthetic image $\mathbf{I}_{syn}$ from solving $\mathbf{I}_t \approx \mathbf{B}_{\rho}^{p\times 9}\mathbf{L}^{9\times 1}$ to get the estimated light coefficients $\mathbf{L}_{est}^{9\times 1}$ for a new illumination condition, then generate and normalize $\mathbf{I}_{syn} = \mathbf{B}_{\rho}^{p\times 9}\mathbf{L}_{est}^{9\times 1}$ to get $\widetilde{\mathbf{I}}_{syn}$. Both the Normalized Cross-Correlation (NCC) and the Normalized Sum of Squared Difference (NSSD) are considered as the error metric, because the procedure of the Orthogonal Matching Pursuit (OMP) in Algorithm 2.1 calculates the cross-correlation in each step, and the texture consistency in Chapter 3 is derived with NSSD. The NSSD is formulated as:

$$\epsilon_x = \|\widetilde{\mathbf{I}}_x - \widetilde{\mathbf{I}}_t\|_2,$$

where $\widetilde{\mathbf{I}}_x = \widetilde{\mathbf{I}}_{init}$ is used to get ϵ_{init} and $\widetilde{\mathbf{I}}_{syn}$ for obtaining ϵ_{syn}. NCC is expressed as the dot product of two normalized vectors

$$\xi_x = \langle \widetilde{\mathbf{I}}_x, \widetilde{\mathbf{I}}_t \rangle.$$

Several illumination conditions are applied to obtain $\mathbf{I}_t$ for the tests. The recognition results are shown in Table 4.1, where the column for the illumination condition specifies which light sources are used as in Fig. 4.7 (B refers to the background lighting from the ceiling). It is clear that under both NSSD and NCC error metrics, the recognition performance can be remarkably improved when using the synthetic image from the illumination model. Only under the lighting condition in last row of Table 4.1, the recognition performance are comparable, because the observation image is quite similar to the initialization image. This may be because the background illumination is dominating compared to the light source 4, thus produces a homogeneous image as in the initialization image.

illumination	ϵ_{init}	ϵ_{syn}	ξ_{init}	ξ_{syn}
B	0.3606	0.2853	0.8197	0.8573
1,B	0.9438	0.4882	0.5281	0.7559
1,4,B	0.6140	0.3873	0.6930	0.8064
4,B	0.2452	0.2535	0.8732	0.8774

Table 4.1: Recognition results for initialization and synthetic images. The tests are carried out under four illumination conditions, where the specified light sources as labeled in Fig. 4.7 are used. B refers to the background illumination from the ceiling.

All above tests were carried out in a spacious room (the robotic hall) when the sun set. In this way, the influence of the controlled illumination on the target appearance can be guaranteed to be dominating. The target was placed on a table covered with black curtains. The great size of the room together with the black curtain can be helpful for satisfying the distant lighting assumption and reducing the influence of multi-reflected light from walls and the table surfaces as much as possible.

4.4 Illumination Invariant 6DOF Pose Estimation

This section incorporates the SH illumination model discussed in previous sections in this Chapter into the framework proposed in Chapter 2 and 3, by which the illumination invariant 6DOF tracking is achieved. In short, the synthetic images constructed from SH basis images are put into the template matrix in sparse representation in the coarse pose estimation stage. Meanwhile, the Textured-ICP algorithm for the accurate pose also uses the synthetic image values instead of the target initial intensity for pose refinement. These processings provide a simple yet effective mechanism to model the appearance changes caused by illumination during 3D motion tracking.

4.4.1 Incorporating Illumination Model into SR framework

This subsection incorporates the SH illumination model into the SR framework used for the coarse pose estimation and analyzes the recognition performance under varying illumination conditions. The 20 illumination templates in Fig. 2.4 are the synthetic images constructed with the estimated lighting in the most recent frames, where the synthetic images are obtained with the nine basis images in $\mathbf{B}_\rho^{p\times 9}$ (e.g. the basis images for the doll dwarf are displayed in Fig. 4.10.

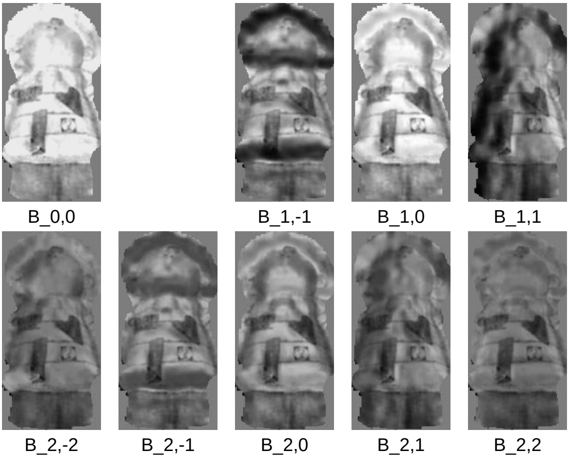


Figure 4.10: Low order nine basis images for the doll dwarf. The positive values are in the range [125,250], the negative values are between [0,124]. The background is set to the neutral gray.

Although Subsection 4.3.2 evaluated the recognition performance under the SH subspace $\mathbf{B}_\rho^{p\times 9}$, if the SH basis images are directly applied on the SR framework proposed in Chapter 2, it may not be able to effectively model the illumination variation due to the use of Orthogonal Matching Pursuit (OMP) introduced in Subsection 2.2.2 as well as due to the influence of other templates in the template matrix.

From Eq. (4.8), it is clear that most probably some of the basis image values will be positive and some will be negative due to the rich distribution of the surface normal orientations. On the other hand, the image patches grabbed from the observation image will be non-negative. This indicates that most of the basis images will be less correlated with the grabbed image patch. The SR used for the coarse estimation is solved through OMP. The evaluation tests discussed in Subsection 4.3.2 are carried out with the illumination subspace $\mathbf{B}_\rho^{p\times 9}$ solved by least squares only. The fundamental difference between the OMP and the least squares used to generate $\widetilde{\mathbf{I}}_{syn}$ in Subsection 4.3.2 is that the OMP selects one atom from the template matrix at one time that most correlates with the current residual vector, whereas the least squares simultaneously considers all atoms. With OMP, the atoms selected to reconstruct $\widetilde{\mathbf{I}}_{syn}$ will most probably be a mixture of some of the nine basis images in Fig. 4.10 and some of other target templates or even wavelet atoms. This means the existence of the other templates will break the theoretical completeness of SH modeling if the SH basis images are directly incorporate into the template matrix depicted in Fig. 2.4.

Furthermore, for an object image under severe illumination changes, it is less likely that the first selected atom from the OMP procedure can be from any of the basis images. Although the basis images calculated from $B_{0,0}$, $B_{1,0}$ and $B_{2,0}$ in Eq. (4.8) can be guaranteed non-negative, they alone cannot reflect the severe illumination changes. In consideration of efficiency, the maximum sparsity in Chapter 2 is set to one for the coarse and the intermediate resolution levels in the annealed particle filter. In this case, the SH basis images cannot be directly used for capturing the illumination changes.

The fact that lighting between consecutive frames usually does not change much has been used in a number of approaches to model the illumination variations. For example, [103] integrated the target image from the last frame into the model and the most recent grabbed target image patch can be used to accommodate to any gradual changes between frames, e.g. lighting, occlusion or viewing perspectives. However, the tracking is imperfect or even can fail for some frames. Therefore, under such a simple update scheme, the model will be contaminated by the background pixels and the inaccuracy can be accumulated. Fig. 4.11 (b) illustrates the grabbed image patch under the error pose depicted in Fig. 4.11 (a). From one side, it contains a lot of background pixels; from another, the left part of the target in the patch is grabbed from the target right part in the observation image due to the erroneous pose. When such a patch is put into the target templates, it will significantly influence the tracking performance in the following frames.

This thesis, as a contrary, uses another strategy for updating the target templates. The template used for update is obtained from the synthetic image from SH basis images with the estimated lighting. Since the synthetic image is a linear combination of the basis images, it can be inaccurate but will always purely reflect the target appearances and will not be corrupted by image values from the background regions. Fig. 4.11 (c) shows the synthetic target image patch. Although it cannot correctly reflect the illumination condition due to pose failure, it still provides a meaningful target appearance, therefore will not have negative impact in future processing.

The balance between the target and non-target templates after the illumination

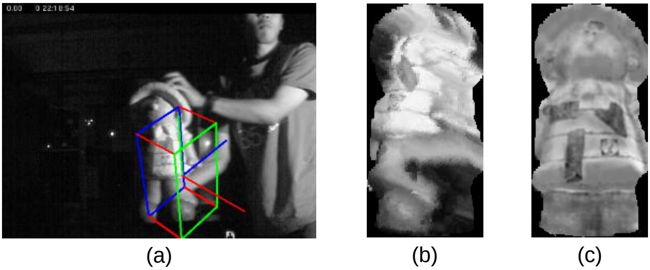


Figure 4.11: Target patch for updating the template matrix under an erroneous pose estimate. (a) shows a frame with an erroneous estimated pose; (b) depicts the grabbed image patch with the error pose; (c) is the synthetic image using the erroneous light coefficients.

templates are incorporated seems to be not critical when using the synthetic images grabbed from previous 20 frames. Fig. 4.12 and Fig. 4.13 show the reconstruction coefficients of SR for two frames from a video sequence, where a target is moving in an illumination varying scene.

The target is illuminated from left in Fig. 4.12 and from right in Fig. 4.13. The target image patch is grabbed with pose parameters indicated with the 3D bounding box in the top left image. The sparse solution is obtained with the OMP procedure under the maximum sparsity 15.

The illumination templates shown in the bottom two rows in Fig. 4.12 are obtained from previous 20 frames. The template pointed by the green dashed arrow corresponds to the largest coefficient for SR in the top right figure. Although being rough compared to the observation image due to the noisy surface normal estimates on the target, the synthetic image template still has the best correlation than the other target (The target is initialized with the frontal light) and non-target templates, thus are effective on handling the illumination variations. Fig. 4.13 shows another example for reconstructing the coefficients under the right lighting.

In comparison, the feature extraction methods will encounter great difficulties under the severe illumination changes. For instance, the well known SURF and SIFT features are reported to be robust under illumination variations [10, 99]. However, under severe illumination changes as the case in Fig. 4.13, the correctly matched feature correspondences can be quite insufficient for a reliable pose estimation. Fig. 4.14 shows the result for the SURF feature extraction and matching, where the correspon-

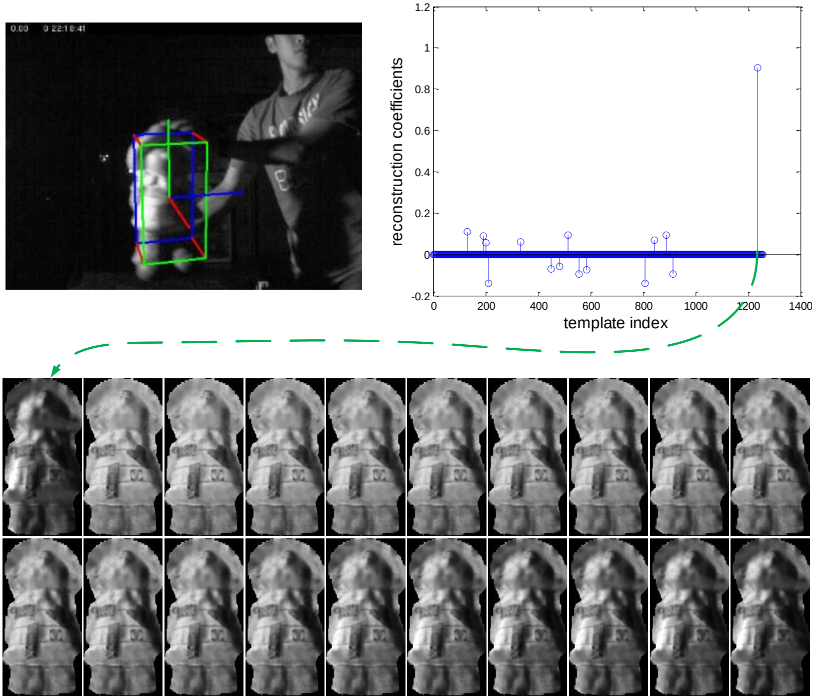


Figure 4.12: SR reconstruction coefficients under left illumination. The estimated pose is shown in the top left image, the reconstruction coefficients for the estimated pose is displayed in the top right figure. All 20 synthetic image templates are illustrated in the bottom row, where the image template corresponds to the largest nonzero coefficients is pointed by the green dashed arrow.

dence pairs between the initialization frame and the observation frame are linked with line segments. It can be seen that a lot of features are incorrectly matched due to the target appearance changes caused by illumination. Under such conditions, even the robust M-estimator or RANSAC can run into problems.

The use of the synthetic images from recent frames also works well with the OMP procedure even under the aggressive settings (sparsity one). The synthetic image is not influenced by the other templates because they do not require a group of templates to model a specific illumination. The cost is the restriction on the inter-frame lighting changes. For most scenarios, this mild inter-frame variation assumption

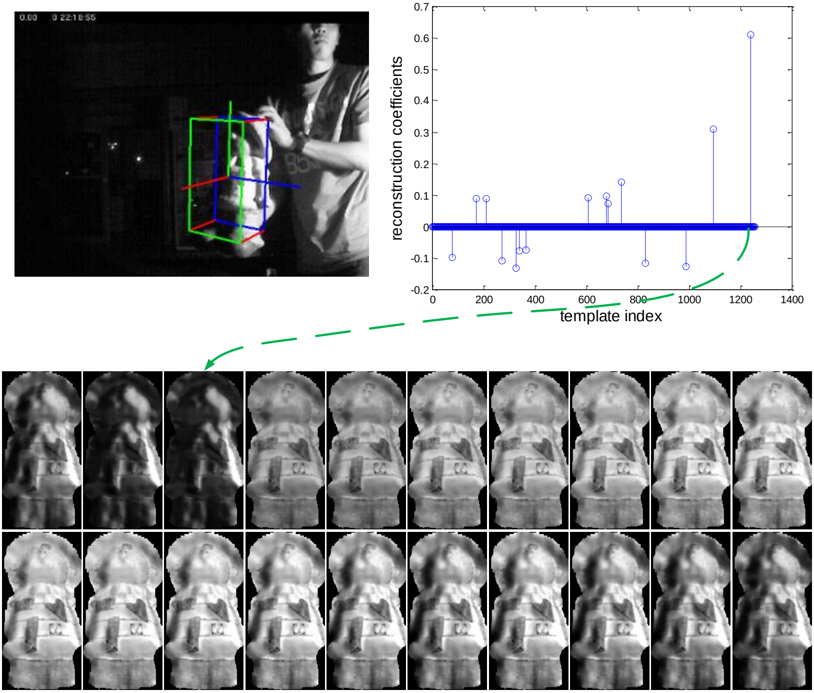


Figure 4.13: SR reconstruction coefficients under right illumination. The estimated pose is shown in the top left image, the reconstruction coefficients for the estimated pose is displayed in the top right figure. All 20 synthetic image templates are illustrated in the bottom row, where the image template corresponds to the largest nonzero coefficients is pointed by the green dashed arrow.

can be satisfied.

It should be noted that although the use of the synthetic image reconstructed with SH basis images improves the tracking performance under varying lighting conditions, experiments also indicate that it will still run into problems when the light changes fast between frames, especially when the target is under fast out-of-plane rotation at the same time. In such cases, it is desirable to have some light prediction mechanism to incorporate the synthetic images under the predicted lighting condition into the illumination templates. Meanwhile, as can be implied from the roughness of basis images in Fig. 4.8, a more accurate surface normal estimation will also improve the

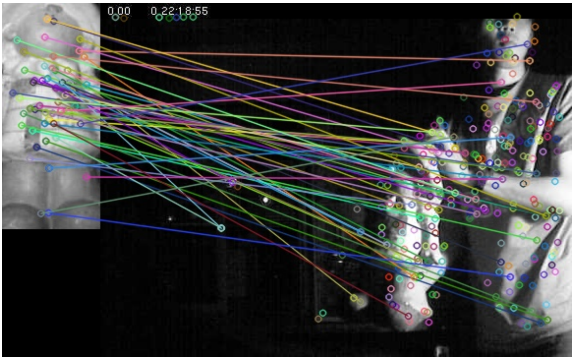


Figure 4.14: SURF feature under significant illumination variation for comparison. The target appearance in the initialization frame is shown in the top left figure. The right observation image is the same frame as Fig. 4.13. The extracted SURF features are marked in small circles and the correspondence features between the initialization frame and the observation frame are linked with line segments.

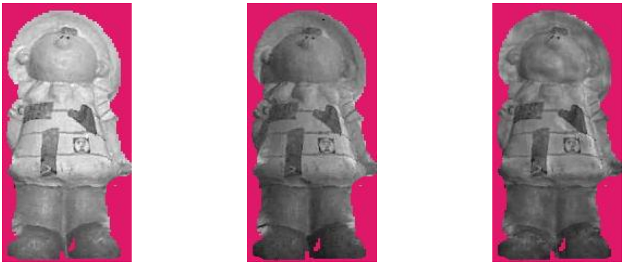
tracking performance, which will require better range measurements.

4.4.2 Incorporating Illumination Model into Textured-ICP

This subsection incorporates the SH illumination model into the Textured-ICP proposed in Chapter 3 and tests the tracking performance under significant illumination changes. In Chapter 3, the target texture information $\mathbf{I}_{init}$ is obtained in the initialization stage. When the lighting has changed dramatically than the initial lighting condition, the texture consistency between the model and the live observation data cannot provide much helpful information for determining the pose.

Similar to the approach presented in the previous subsection, the SH basis images can be used to cope with the illumination variation, where the target texture information is provided by a synthetic image $\mathbf{I}_{syn}$ reconstructed from the lighting condition estimated in the last frame. For instance, Fig. 4.15 gives a comparison between the initial target image $\mathbf{I}_{init}$ in Fig. 4.15 (a), the observation target image $\mathbf{I}_t$ in Fig. 4.15 (b) and the synthetic target image $\mathbf{I}_{syn}$ in Fig. 4.15 (c). Despite some "dirty" speckle on the doll face in the synthetic image caused by the noisy surface normal, $\mathbf{I}_{syn}$ still well mimics $\mathbf{I}_t$.

The incorporation of the synthetic image into Textured-ICP is tested under severe lighting changes. Fig. 4.16 shows some frames grabbed from two result videos on the



(a)
initialization image

(b)
reference image

(c)
synthetic image

Figure 4.15: Real and synthetic images for the doll dwarf with SH modeling.

doll dwarf and the Merlin robot. Full videos can be found in the supplementary materials. To guarantee sufficient lighting variations, the experiments are carried out in a dark and spacious hall, where the target is lit by one moving illumination source. The color camera AXIS PTZ 212 cannot produce a color image under a weak ambient lighting condition, therefore only gray scale images are provided.

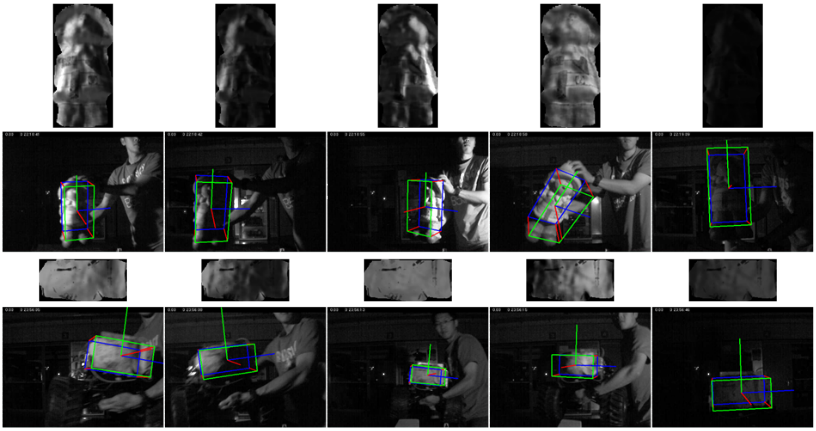


Figure 4.16: Pose estimation under significant illumination variations. The 2nd and the 4th rows show some frames for the estimated pose of the doll dwarf and the Merlin robot under different lighting conditions. Above them are the synthetic target images used as the appearance information in the Textured-ICP.

As shown in Fig. 4.16, the targets are under both significant 3D motion and severe illumination variations. The target model is initialized with multiple light sources from different orientations so that all visible surface points can be lit. For testing the performance under extreme lighting conditions, the light source is moved approximately along a half circle, e.g. some frames are lit from the left, some frames from the right. Also in a few frames, the light slightly points away from target region and the target appears dark, as can be seen in the two images on the last column. The synthetic target image used in the texture consistency calculation for the Textured-ICP is shown above the corresponding result image. Despite of the roughly reconstructed appearances caused by the inaccurate surface normals, they can effectively capture the most important changes due to illumination. The results are obtained with the setting $\rho_c = 1.0$ in Eq. (3.16), where no special preference is made on the texture nor the range data in the Textured-ICP. As can clearly be seen in the synthetic images for the Merlin robot, under weak lighting, the image sensed by the color camera is not as sharp as when the light intensity is sufficient. With such smoothed images, the texture cannot contribute much for the accurate pose estimation. If the target motion is mild and the range data are reliable, it is recommended to set $\rho_c = 0.5$ for a fast varying lighting condition.

4.5 Summary

This chapter deals with the influence of the appearance variations caused by illumination in the object pose estimation problems. First the state-of-the-art methods for the illumination invariant visual tracking, the illumination modeling, and the inverse rendering are reviewed. Then the theoretical background knowledge regarding the LED array modeling and the illumination modeling with Spherical Harmonics (SH) is introduced. Previous researches on the SH modeling mostly focused on human face recognition problems. To be applied on more general Lambertian objects, the target surface reflectance information is required.

With the idea that the surface reflectance in the near infra-red spectrum may help determining the reflectance in the other spectra, e.g. the red spectrum, the modeling method for the intensity distribution of the LED array mounted on the PMD camera is investigated. If the intensity distribution can be accurately approximated, estimating the near infra-red surface reflectance will be straight forward. However, experiments show that there is a non-negligible gap between the theoretical LED array model and the real LED arrays on the PMD camera. The model parameters, which are supposed to be constant, need to be determined for measurements at different distances. Since the aim of modeling the LED array is to help estimating the target surface reflectance, to avoid too much complication of the problem, instead of investigating more into the LED modeling, another simple reflectance estimation method is adopted, where a calibration object is exploited.

Under the SH illumination modeling framework, a Lambertian calibration object with homogeneous reflectance can be used to determine the lighting condition. Once the lighting is known, together with the target geometrical information obtained from the PMD measurement, the target surface reflectance can be estimated. Then the estimated reflectance is applied on the object recognition under significantly different lighting conditions, which yields a better recognition performance especially for the extreme lighting conditions, e.g. sideways illumination.

After the reflectance information is obtained, SH model is incorporated into the pose estimation algorithm proposed in previous chapters. The major contribution of this chapter is to put the theoretical research on illumination modeling into pose estimation problem of real objects. The effectiveness of the illumination modeling scheme is demonstrated through the target pose estimation results in video sequences with varying illumination conditions. However, it should be pointed out when the inter-frame illumination variation is dramatic, the pose estimation algorithm can still run into problem. Therefore, fast changing lighting can still be an open problem for future researches.

Chapter 5

Experiments and Application

This chapter gives a series of experimental results for evaluating the pose estimation algorithm proposed in previous chapters. The reference evaluation is performed by comparing the estimated pose with the highly accurate reference measurement from the iSpace system. A calibration algorithm is proposed for transforming the data in iSpace coordinates to the camera system. Besides evaluating the accuracy of the estimated pose, the robustness of the proposed algorithm are demonstrated with tests on various targets in both indoor and outdoor environments. In addition, a leader-follower mobile robot formation application is conducted with the estimated pose.

5.1 Unifying Coordinates with the Reference Pose

iSpace measurement system can provide pose information for iSpace sensor frames with a very high accuracy. In following experiments, one iSpace sensor frame is mechanically attached to the fused cameras and another is attached to the target object. For evaluation, the target pose estimated by the proposed algorithm will be compared with the pose measured by iSpace, where both poses should be in the same coordinate system. Here AXIS world coordinate system is chosen as the common coordinate system where evaluations are performed, because the pose estimation algorithms in previous chapters are derived in this system. Therefore, transformation from iSpace sensor poses to target and camera poses are required to yield valid reference data for evaluation. In the following, the formulation for transforming pose measured by iSpace into camera coordinate system is derived.

5.1.1 Target Pose in iSpace and in Camera Systems

As shown in Fig. 5.1, there are two iSpace frames: the target iSpace frame which is attached to the target, with origin $S_{t,w}$ and pose $\mathbf{T}_{w,St}$ in world coordinate system; the camera iSpace frame which is attached to the camera with origin $\mathbf{S}_{C,w}$ and pose $\mathbf{T}_{w,SC}$ in world coordinate system. Here the 3×4 matrix $\mathbf{T}_{w,St}$ represents the pose of the target iSpace frame in world coordinate system, or equivalently it can be used to transform a point in the target iSpace coordinates to the world coordinates.

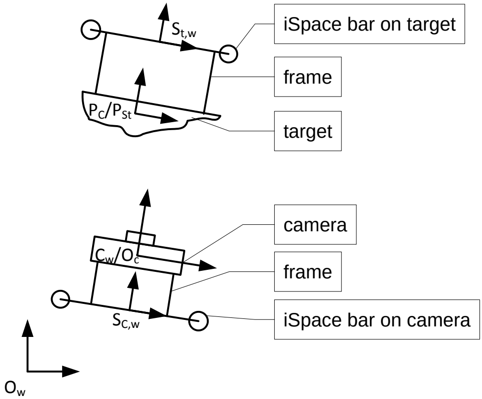


Figure 5.1: 2D schematic illustration for camera, target and iSpace configurations.

For conducting the evaluation with iSpace measurements, the target pose $\mathbf{T}_{C,t}$ in the camera coordinates is desired when the iSpace measurements $\mathbf{T}_{w,St}$ and $\mathbf{T}_{w,SC}$ are available. To this end, the relative transformation matrix $\mathbf{T}_{C,SC}$ and $\mathbf{T}_{t,St}$ are required, by which the target pose in the camera coordinates can be formulated as:

$$\dot{\mathbf{T}}_{C,t} = \dot{\mathbf{T}}_{C,SC}\dot{\mathbf{T}}_{SC,w}\dot{\mathbf{T}}_{w,St}\dot{\mathbf{T}}_{St,t}, \tag{5.1}$$

where $\dot{\mathbf{T}}$ is the 4×4 homogeneous version of $\mathbf{T}$. Under such a notation, the inversion of the homogeneous transformation matrix can be denoted in a simple way, e.g. $\dot{\mathbf{T}}_{SC,C} = \dot{\mathbf{T}}_{C,SC}^{-1}$ represents the camera pose in the camera iSpace coordinates.

In Eq. (5.1), $\mathbf{T}_{C,SC}$ and $\mathbf{T}_{St,t}$ are two unknown relative transformations that are hard to measure and should be estimated. Assuming the initial orientation of the camera coordinate system is the same as the target coordinate system,[1] the rotation part of $\mathbf{T}_{St,t}$ can be retrieved from $\mathbf{T}_{St,C}$ and the translation part from $\mathbf{P}_{St}$, where $\mathbf{P}_{St}$ is the target center in the target iSpace frame coordinates. Observing $\mathbf{T}_{St,C} = \mathbf{T}_{St,SC}\dot{\mathbf{T}}_{SC,C}$, where $\mathbf{T}_{St,SC} = \mathbf{T}_{St,w}\dot{\mathbf{T}}_{w,SC}$ is readily known, in the following contents, the estimation methods for $\mathbf{T}_{SC,C}$ and $\mathbf{P}_{St}$ are derived.

5.1.2 Transformation between iSpace and Camera Coordinates

In this section, a chess board is used as a calibration object for estimating $\mathbf{T}_{SC,C}$. The experimental setup is illustrated in Fig. 5.2, where the fused sensors mounted on a mobile robot are placed in front of the chess board pattern.

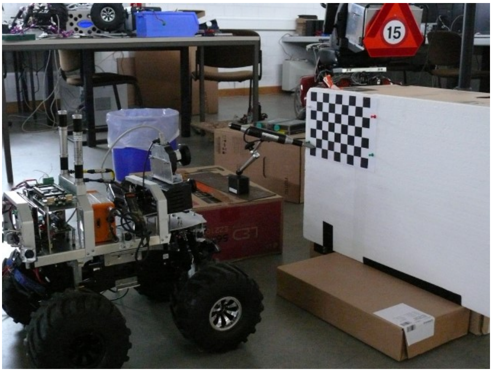


Figure 5.2: Experimental setup for determining the relative transformation between iSpace and camera systems.

When the chess board pose $\mathbf{T}_{C,ch}$ in the camera system and the pose $\mathbf{T}_{w,ch}$ in world coordinates can be acquired, $\mathbf{T}_{C,SC}$ can be derived as

$$\dot{\mathbf{T}}_{C,SC} = \dot{\mathbf{T}}_{C,ch}\dot{\mathbf{T}}_{w,ch}^{-1}\dot{\mathbf{T}}_{w,SC}.$$

$\mathbf{T}_{w,ch}$ can be obtained by measuring three or four corner points on the chess board with the iSpace vector (in Fig. 5.2, the bottom left point on the chess board is being measured as the origin of the chess board coordinates). iSpace produces

[1] Actually the initial target pose is defined in the implementation to have the same orientation as the camera coordinate system.

measurements with a very high accuracy (absolute error<1 mm) in real-time (40 fps). Therefore, $\mathbf{T}_{w,ch}$ is supposed to be quite reliable.

$\mathbf{T}_{C,ch}$ is estimated with the routine *cvFindExtrinsicCameraParams2* from the OpenCV library. The input intrinsic and distortion parameters for the color camera are estimated beforehand. During calibration, it is recommended to place the chess board as parallel as possible to the image plane of the camera. Because under the hood of the OpenCV routine, the constraints for solving the rotation matrix are not complete and there will follow a singular value decomposition step enforcing the computed rotation matrix to be a valid rotation matrix. When the underlying rotation matrix is close to a unit matrix, the solution matrix will be quite close to the final output rotation matrix, and thus can yield better result.

Since it is impractical to directly measure the true rotation and translation between the color camera's image plane and the camera iSpace frame, we only empirically evaluated the accuracy of $\mathbf{T}_{C,SC}$ by placing the camera and the camera iSpacec frame as parallel as possible and examining the estimated relative transformation. Results indicated that the absolute rotation errors are within $2^\circ \sim 3^\circ$ and the absolute translation errors are within $(5, 5, 15)$ millimeters in three directions. The error will be further discussed in the next subsection.

5.1.3 Estimating Transformation between Target and iSpace

For an arbitrary target with non-planar shape, the mass center will reside inside the target body and can be infeasible to measure directly. Moreover, when the interested target is only a part of an object which is determined during the initialization, the mass center should also be calculated with the points selected for tracking in the pose estimation program. This means that the relative transformation between the target coordinate system and the coordinates for the target iSpace frame can only be determined with the help of the fused TOF and color cameras.

On the other hand, the Iterative Closest Point (ICP) algorithm has been investigated substantially since its introduction and has been demonstrated to be able to provide an accurate pose estimation when the range measurement is reliable. Since under a well-controlled condition, CamCube2.0 is reported to produce range data with precision better than 3 mm. Therefore, the range measurement provided by CamCube2.0 along with the pose estimated with ICP could be exploited for estimating $\mathbf{P}_{St}$.

In the Fig. 5.3, $\mathbf{P}_C$ is the target mass center in the camera coordinates during initialization and $\mathbf{P}'_C$ is the estimated mass center obtained by applying ICP on the range data from CamCube2.0. The uncertainty of the range measurement, as well as of the the ICP algorithm, is mostly reflected in a scalar h representing the Cartesian depth along Z axis. The mapping from a pixel position $[x, y]^\top$ to a point in the 3D coordinate of the camera system is expressed by the pin-hole camera model

$$\mathbf{P}_C = h\mathbf{M}^{-1}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$

where $\mathbf{M}$ is the 3×3 intrinsic matrix of the color camera.

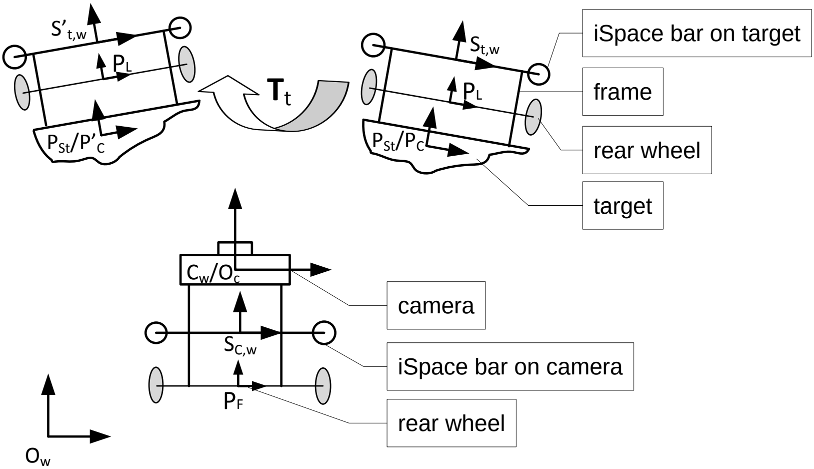


Figure 5.3: Relative transformation between iSpace and target systems.

The target mass center $\mathbf{P}_w$ in world coordinates can be obtained by transforming $\mathbf{P}_C$ as

$$\mathbf{P}_w = \mathbf{T}_{w,SC}\dot{\mathbf{T}}_{SC,C}\dot{\mathbf{P}}_C. \tag{5.2}$$

Also, $\mathbf{P}_w$ can be obtained by transforming $\mathbf{P}_{St}$ as

$$\mathbf{P}_w = \mathbf{T}_{w,St}\dot{\mathbf{P}}_{St}.$$

Combining above three equations yields:

$$\mathbf{T}_{w,St}\dot{\mathbf{P}}_{St} = \mathbf{T}_{w,SC}\dot{\mathbf{T}}_{SC,C}\dot{\mathbf{M}}^{-1}\begin{bmatrix} hx \\ hy \\ h \\ 1 \end{bmatrix}. \tag{5.3}$$

Every time the target is moved, a new unknown h is introduced, alone with three functions corresponding to the three rows of Eq. (5.3). For N target positions, there will be $3N$ equations and $3+N$ unknowns (3 for P_{St}, N for all h). With $3N \geq 3+N$, i.e. $N \geq 2$ target positions, all unknowns could be solved.

As discussed in the previous subsection, most of the inaccuracies in $\mathbf{T}_{C,SC}$ will be reflected in the range translation along Z axis. This inaccuracy will be propagated to the right side of Eq. (5.3) in the form of $\mathbf{P}_w$. Since the above linear equation system can provide more equations than unknowns, the inaccurate range element along Z axis in $\mathbf{T}_{C,SC}$ can also be incorporated as an unknown and be solved with the above system. In our experiment setup, the range element in $\mathbf{T}_{C,SC}$ corresponds to the X element in $\mathbf{T}_{SC,C}$. When the 3×4 transformation matrix $\mathbf{T}$ is expressed as a rotation matrix and a translation vector as $\mathbf{T} = [\mathbf{R}, \mathbf{t}]$, and when the translation vector for $\mathbf{T}_{SC,C}$ is denoted as $\mathbf{t}_{SC,C} = [d_x, d_y, d_z]^\top$, and the rotation term of $\mathbf{T}_{w,C} = \mathbf{T}_{w,SC}\dot{\mathbf{T}}_{SC,C}$ as $\mathbf{R}_{w,C} = [\mathbf{r}_{wC,1}, \mathbf{r}_{wC,2}, \mathbf{r}_{wC,3}]$, Eq. (5.3) can be reformulated as:

$$\mathbf{R}_{w,St}\mathbf{P}_{St} - d_x\mathbf{r}_{wC,1} - h\mathbf{R}_{w,SC}\mathbf{P}_C = \mathbf{t}_{w,SC} - \mathbf{t}_{w,St} + d_y\mathbf{r}_{wC,2} + d_z\mathbf{r}_{wC,3}, \tag{5.4}$$

which is a linear system with $3N$ equations and $4+N$ unknowns (4 for $\mathbf{P}_{St}$ and d_x). The solution d_x will be used as the corrected range element in $\mathbf{T}_{SC,C}$. Experimental results showed that after such a correction, the absolute error of the range element for $\mathbf{T}_{C,SC}$ and $\mathbf{T}_{St,t}$ can be within 5$\sim$10 mm.[1]

5.2 Evaluation with Reference Pose Available

The estimated pose is evaluated by comparing with the reference pose obtained from iSpace, which is transformed into the camera coordinate system with the unification method presented in the previous section. The experimental setup is illustrated in Fig. 5.4, where the rear part of the Merlin robot is used as the target object. The

[1] For instance, the estimated range terms in $\mathbf{T}_{C,SC}$ and $\mathbf{T}_{St,t}$ were 315 mm and 335 mm, where the corresponding values roughly measured by a ruler were 312 mm and 332 mm.

fused cameras are mounted on another robot as shown in Fig. 5.2.

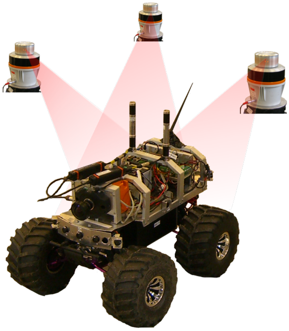


Figure 5.4: Experimental setup with iSpace system.

The evaluation result is shown in Fig. 5.5, where the results for the rotational degrees of freedom are shown in the three figures in the first row. They are measured in degrees with Euler angle under the ZYX rotation convention (X and Y are horizontal and vertical axes on the image plane). The second row shows the results for the translation along X, Y, Z directions in millimeter in Cartesian coordinates. The comparison is performed for both the coarse pose (SR) in Chapter 2 and the refined accurate pose (SR+LKICP) in Chapter 3.

The coarse pose complies well with the pose from iSpace for the rotation around Z axis and the translation along X and Y axes (i.e. the in-plane motion). The pose changes in these degrees can bring about significant appearance variations in the 2D image, therefore can be handled by the 2D texture based coarse estimation. The object appearance is much less sensitive to the rotation around X and Y axes or the translation along Z axis (i.e. the out-of-plane motion), which can result in large errors for 2D data based coarse pose. In comparison, by taking the coarse estimate as the initial pose guess and refining by the LKICP, the accurate pose estimation yields reliable results for all 6DOF. However, the accurate pose module alone cannot produce a robust and efficient pose estimation algorithm, partly because the LKICP is a gradient-based procedure and can be trapped into the local minimum, partly because it is relatively less effective on the in-plane motions, which can be seen in the convergence evaluation in Fig. 3.8, where more iterations are required to handle a similar amount of in-plane motion than the out-of-plane motion. In a word, the coarse estimation can handle the in-plane motion, while the accurate estimation tackles well

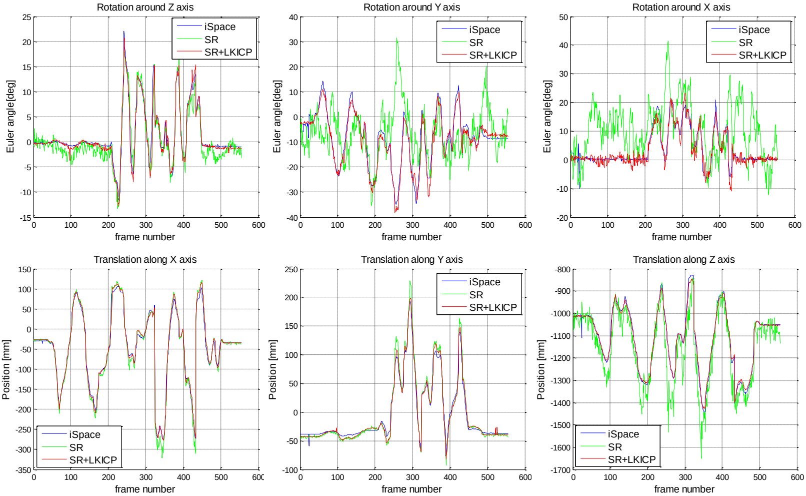


Figure 5.5: The estimated pose and the reference pose from iSpace. The pose from iSpace, from the coarse stage, and from the accurate stage are displayed in blue, green, and red respectively.

the out-of-plane motion and the combination of both yields the reliable performance on all 6DOF.

It should be noted that the coarse pose in Fig. 5.5 is obtained by completely deactivating the accurate pose estimation module. If the coarse pose can take the accurate pose output from the last frame as the initial pose input for the current frame, which is the case for the real implementation, the coarse estimation result for the rotation around Y and X axis can be significantly improved.

With the reference data from iSpace, the errors for the pose calculated from three algorithms - the conventional ICP, the texture based NSSD and the proposed LKICP - are compared and illustrated in Fig. 5.6. They are the alternative error metrics for calculating the cost function Eq. (3.8), i.e. ICP for $g_n^2(\varDelta\boldsymbol{\theta})$, NSSD for $f_n^2(\varDelta\boldsymbol{\theta})$, and LKICP for $g_n^2(\varDelta\boldsymbol{\theta}) + \rho f_n^2(\varDelta\boldsymbol{\theta})$. All three pose refinement algorithms are combined with the proposed coarse pose estimation.

The rotational errors are shown in the three figures in the first row of Fig. 5.6. The texture data based NSSD produces large errors on the rotation around Y axis and the range data based ICP yields non-negligible errors on the rotation around Z axis.

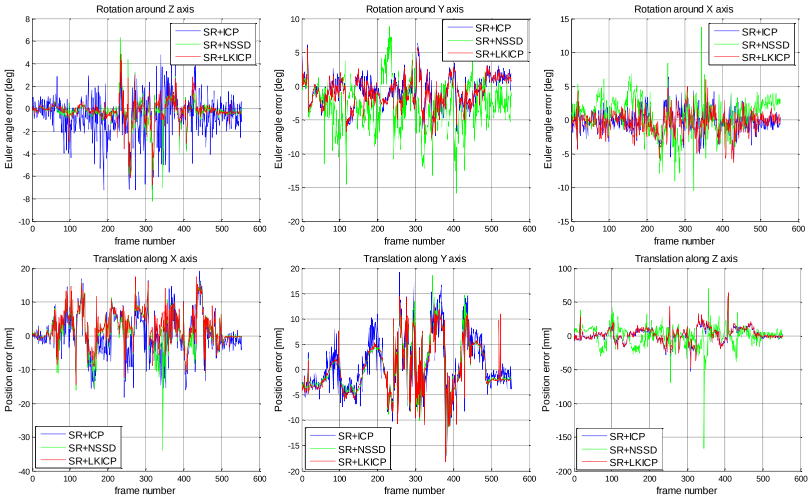


Figure 5.6: Error comparison for ICP, NSSD and LKICP. The pose from iSpace is used as the reference. The pose estimation from ICP, NSSD and LKICP are displayed in blue, green and red respectively.

In comparison, the LKICP performs the best estimation. For most of the frames, the errors are within 3°, which is sufficient for most mobile robot applications. All three algorithms have similar performance on the three translational degrees of freedom as shown in the figures in the second row of Fig. 5.6. The errors in most of the frames are within 1.5 cm.

However, there are also some frames, the translational error can be as large as 2.0 cm. By checking the result video, there are no frames having such large errors (2.0 cm errors on the translation along X or Y axes are supposed to be visibly salient). Although the variation of the measurement produced by iSpace is reported to be around 0.25 mm for the typical environment [64], during our test, we seldom reached such a level. The variation was always around 3∼6 mm. The camera measurements are also an error source, including the target appearance variation from the color camera, or the range measurement uncertainties from the PMD camera. Both will result in errors for the relative transformation between camera/target and the iSpace coordinate systems. Such an error can better be observed from the translational error along Y axis in the middle figure of the second row of Fig. 5.5, where for a lot frames the pose measured by the iSpace is clearly shifted from the estimated pose. Therefore, the error is a composite of the errors from the pose estimation, from the iSpace measurement, from the coordinates unification between iSpace and the camera, from the imperfect synchronization for data acquisition between iSpace and the camera, and also from the inaccurate sensor measurements (range data inaccuracies from the CamCube2.0 and the intensity variations from the color camera).

Practically it is very difficult to synchronize the camera and the iSpace measurements. Even if the capturing of both sensors can be triggered by some complicated hardware, the integration time of the two cameras cannot be completely controlled. For the experiments discussed above, the target was first placed still. When it started to move, the time stamps for the data from both iSpace and camera were manually determined. Then the measurement from one source (camera or iSpace) was translated by the time stamp difference, so that the time in both systems can be aligned. On the other hand, the iSpace runs at 20 Hz or 40 Hz, but the frame rate for the fused camera system is usually lower than 20 Hz (including the capturing time, the data recording time, etc.). Therefore, to perform a valid comparison, after the overall time translation for the measurement alignment, the final iSpace data used for the comparison are the interpolated data by using the time stamps recorded for both iSpace and camera systems. Since these synchronization and interpolation operations cannot achieve perfect time alignment, it should also be considered as an error source in terms of comparison. The comparison error caused by synchronization can be clearly observed from the spikes in the second row of Fig. 5.6. The translation error indicated by these spikes can be as large as 15∼20 mm. But the result video shows there is seldom a frame with error larger than 10 mm. The synchronization error can also be seen in Fig. 5.7, which shows the detailed data between frame 380 and 388 in the middle image of the second row in Fig. 5.5 (only results for iSpace and the proposed algorithm are displayed). At around frame 384, the estimated pose goes one frame later than the measured pose from iSpace, which brings about an absolute difference of around 18 mm at frame 384. The work in [135] provides a method for determining the time delay between the measurement system and the estimation algorithm. For our system, the iSpace data are transmitted via Ethernet. More importantly, the iSpace API is running in Windows and the pose estimation algorithm is in Linux. Therefore, two computers are used, which makes the synchronization problem more complicated.

Based on the above discussions, the error from iSpace measurement, from coordinates unification process, especially the large differences caused by synchronization

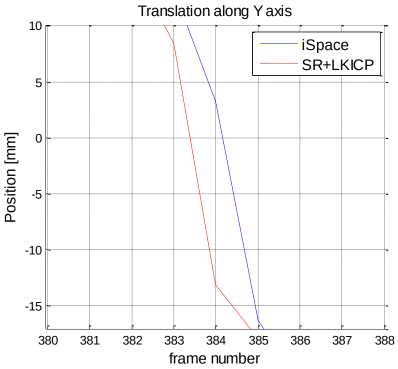


Figure 5.7: Error from synchronization between the iSpace and the camera systems. This figures shows the detailed pose comparison from the iSpace and the proposed algorithm between frame 380 and 388 for the translation along Y axis. It is a scaled figure for the middle image in the second row of Fig. 5.5.

between iSpace and camera systems, should not be deemed as real errors for the estimated pose. Therefore, the proposed algorithm can be expected to yield pose with absolute translation error within 1.0 cm and absolute rotation error within 3°.

5.3 Tests on Various Targets

The proposed algorithm is also tested on various targets in both indoor and outdoor environments. Although no reference data are available for these target objects, the quality of the estimated pose can be shown in the result videos. Fig. 5.8 illustrates some of the result frames performed in the indoor environment. Full result videos can be found in the attached CD. The test objects include a pottery cock, a small testing satellite, a cylinder bucket, the outdoor Merlin robot, and a doll dwarf. A desired target should have appropriate size, because a big object do not fit the small field of view of CamCube2.0, whereas a small object cannot be covered by sufficient image pixels. The object should have a Lambertian surfaces with good reflectivity, because current TOF cameras cannot produce reliable measurements on glossy or dark surfaces due to the insufficiency of the modulated near infra-red light reflected back into the camera.

For testing the performance under fast motion, both the target and the fused

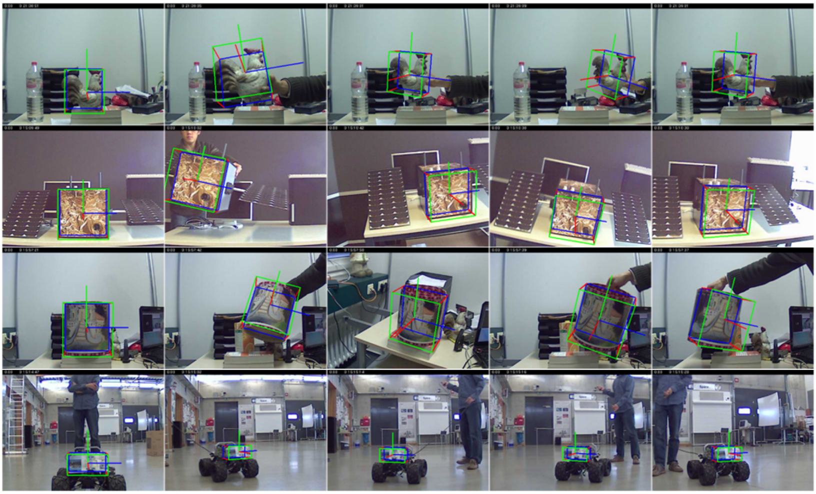


Figure 5.8: Tests on various targets in indoor environment.

cameras were moving in all 6DOF (except the bottom row in Fig.5.8, where the robot drove on a flat floor). The inter-frame motion can be as large as approximately 10∼15 cm in translation and 15∼20° in rotation. In this tests, 400 particles were used in the coarse pose estimation stage for robustly handling the large inter-frame motions. Even faster movement will require more particles for effectively covering a larger volume in the state space. However, this will also raise higher demand on both CPU and GPU. On the other hand, the increased particles will reduce the processing frame rate, which will make the inter-frame motion even severer. Nevertheless, when the target moves too fast, the color image will be blurred and the range image can be quite noisy. Therefore, the upper boundary for the restriction on the target speed will be drawn by the capability of the cameras and processors.

The targets shown in Fig. 5.8 have various shapes and textures. The proposed algorithm works robustly on all these shapes. Among them, the small satellite and the cylinder bucket are of particular interest for their geometrically symmetric shapes. The range data based conventional ICP is ill-posed for such objects due to short of constraints. The proposed LKICP, on the other hand, can well handle the symmetric targets because of the incorporation of the texture information.

Although diffuse surfaces are assumed, practically no objects have perfect Lambertian surfaces. In some cases, the diffuse component can have strong influence on the target appearance. For instance, during the test on the small satellite, some parts of the target surface were highlighted in some frames, one of which is shown in the left image of Fig. 5.9 with the upper right corner of the satellite highlighted. Despite of these unexpected appearances, the pose of the small satellite can always be correctly estimated. To be clearly on how such a situation is handled, the reconstruction coefficients for sparse representation is investigated in greater depth. A particle is manually placed on the final optimized pose, with which an observation image patch in the finest resolution level is grabbed. The reconstruction coefficients for the grabbed image patch are solved by orthogonal matching pursuit and are illustrated in the middle image of Fig. 5.9. It can be seen that the most significant reconstruction coefficient corresponds to an occlusion template depicted in the right bottom image.

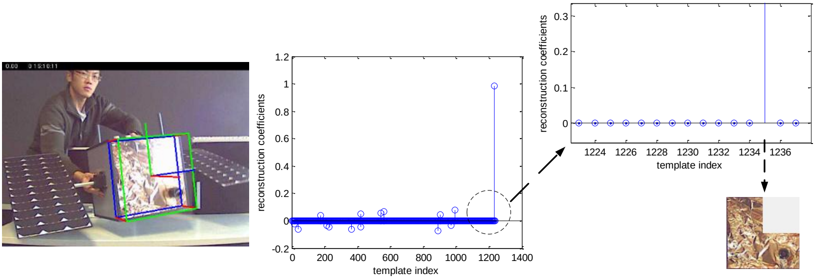


Figure 5.9: Reconstruction coefficients for the partially highlighted satellite. The left image shows the partially highlighted satellite overlaid with the estimated pose box. The middle image depicts the reconstruction coefficients for sparse representation for the estimated pose in the highest resolution level, where the most significant coefficient and its corresponding template are illustrated in the right images.

It should be pointed out that there are a lot of occlusion types, e.g. occluded by hand, by grass or by metal fences. If a certain occlusion type in an application can be foreseen, the corresponding occlusion templates can be built and incorporated into the template matrix in sparse representation. In this thesis, only the block type occlusion templates are implemented as shown in Fig. 2.3. The occlusion templates are meant for dealing with occlusions. However, even if the coarse pose module could handle occlusion, the proposed LKICP will perform poorly if the occlusion is close to the target.[1] Some results for the pose estimation under occlusion are given in

[1] The consistency test $\Omega_k(n)$ in Eq. (3.8) cannot exclude occlusion points which are close to the

Fig. 5.10, where the occlusion objects are about 30 cm closer to the cameras than the target. The occlusion templates shown in Fig. 2.3 can be used to model the block-wise occlusion. Therefore, most of the occlusion situations in Fig. 5.10 can be handled. However, when the middle part of the target is occluded, no corresponding occlusion templates are available, the pose estimation can fail, as shown in the middle image of Fig. 5.10. This also indicates that if the occlusion is often encountered in an application, more occlusion templates should be incorporated.

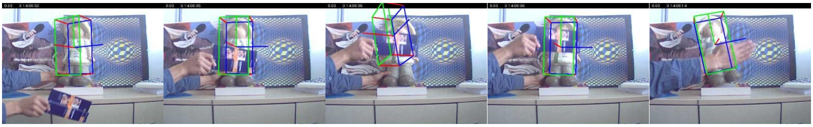


Figure 5.10: The estimated pose under occlusion. The pose estimation is performed with the occlusion objects about 30 cm closer to the cameras than the target.

The pose estimation algorithm is also tested in the outdoor environment, where the strong near infra-red component from the sunshine can have remarkable negative influence on the range measurements. Despite of the corrupted range data, the proposed algorithm can robustly track the 3D motion as in the indoor case. Fig. 5.11 shows some of the result frames for various targets in the outdoor test.

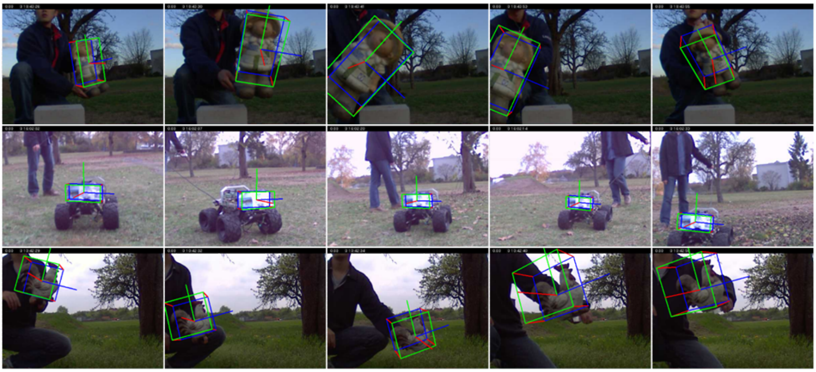


Figure 5.11: Tests on various targets in the outdoor environment.

The estimated pose of the doll dwarf is as accurate as the indoor test. Because the

target from being calculated into the cost function for LKICP.

dwarf is made of paper with rich texture, and has a convex and asymmetric shape, which means the proposed LKICP can have sufficient constraints from both range and texture information. The test on the outdoor Merlin robot was carried out by mounting the fused cameras on another robot and both robots were driving on the uneven grassland. The rough terrain causes additional motion artefacts besides the noise cause by sunshine. The conventional ICP fails on some rotational degrees of freedom as is also shown in Fig. 3.5 due to the contaminated range measurements. Whereas the LKICP yields robust estimates that can be used in real outdoor leader-follower formation applications. The most difficult object, as revealed by the experiment, is the cock. Its pottery material raises more problems on the range measurements under sunshine. Its small size (less than 20 cm in both height and width) decreases the number of points that can be used for estimating the pose. Its largely homogeneous texture cannot provide enough constraints for the texture information in LKICP. Although the target could still be located thanks to the coarse pose estimation stage, the final estimated pose exhibits some jumping behaviour for a lot of frames.

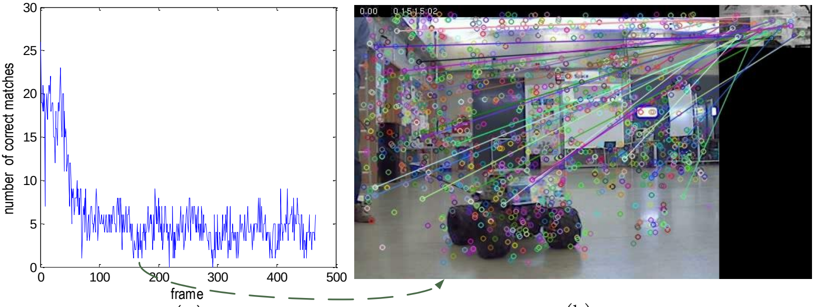
Since keypoints (SIFT, SURF, etc.) are often adopted for pose estimation, as a comparison, the SURF feature matching is applied for all above objects with some of the video sequences. The SURF feature extraction and correspondence are implemented with OpenCV routines. As mentioned in [33], usually 8 - 10 correct correspondences are necessary for a reliable pose estimation. Meanwhile, the ratio of the number of outlier correspondences to the inliers should be less than 1.0 even with the robust estimators. Therefore, the results for these two criteria are given in Table 5.1.[1] The proposed algorithm yields a robust pose estimation for these video sequences. In comparison, the SURF feature matching has problems with large numbers of frames for all test sequences.

	dwarf	satellite	cock	robot	bucket
# model SURF features	39	102	28	27	68
# f. inliers<10/# f.	460/1098	6/761	100/298	415/466	24/762
# f. inliers<50%/# f.	827/1098	327/761	152/298	425/466	309/762

Table 5.1: SURF feature correspondences for various targets. The second row lists the number of SURF features extracted on target region from the initial frame. The third row gives the number of frames with less than 10 correct correspondences and the number of frames in the sequence. The fourth row shows the number of frames with less than 50% inlier correspondences.

[1] When the distance between a matched keypoint and the projected pixel position of the keypoint transformed with true pose is more than 5 pixels, it will be determined as an outlier correspondence.

The SURF feature matching is carried out with the exhaustive search throughout the complete image. The left image in Fig. 5.12 shows the number of correct correspondences for Merlin robot in the indoor video sequence, and the right image illustrates the correspondences for one frame. It can be seen that when the target has a noticeable out-of-plane rotation, only a small number of correspondences are correct, which is insufficient for the pose estimation. Although one big advantage of the keypoint based methods is the ability to perform pose estimation without a predicted pose, it seems a good prediction is still required to narrow down the search area for improving the correct matching rates. Another method is to apply some rectifications to decrease the image distortions caused by perspective projection or rotation for achieving better correct correspondence rates [88].



(a) (b)

Figure 5.12: SURF feature matching for Merlin robot. Correspondences with error more than 5 pixels are considered as mismatches.

The presented algorithm is a frame-to-frame pose estimation approach, which means it can only deal with a restricted amount of inter-frame motions. Therefore, the maximum allowed inter-frame motions for various objects are also evaluated. As before, it is impractical to evaluate the full 6DOF simultaneously. Thus the evaluation is performed for each degree of freedom separately. More specifically, the target is moved/rotated approximately in one degree with all model updates in the algorithm disabled. The maximum allowed inter-frame motion will be recorded at the point when the pose estimation becomes very unstable or even fails. The tests are conducted under typical parameter settings, e.g. 300 particles with an initial rotational variance 10° (0.17 in radian) and translational variance 8 cm, i.e. $\boldsymbol{\sigma}_F = [0.17, 0.17, 0.17, 8.0, 8.0, 8.0]$ in Algorithm 2.2. The results are shown in Table 5.2.

It can be seen that in general the results indicate very fast inter-frame motions. However, in practice, the real motion will be a combination of movements in several degrees of freedom. Furthermore, the motion artefacts should also be considered. Therefore, the results in Table 5.2 should serve as the upper bound for the maximum allowed inter-frame motion for these objects.

target	θ_α	θ_β	θ_γ	θ_x	θ_y	θ_z
dwarf	26°	23°	31°	18 cm	16 cm	34 cm
satellite	17°	21°	25°	20 cm	19 cm	41 cm
cock	22°	29°	24°	19 cm	18 cm	25 cm
robot	21°	21°	28°	13 cm	17 cm	23 cm
bucket	38°	23°	32°	20 cm	21 cm	28 cm

Table 5.2: Maximum allowed inter-frame motion for different targets. The results were obtained under a typical setting for the proposed algorithm, e.g. 300 particles, with initial rotation variance 10° (0.17 in radian) and translation variance 8 cm, i.e. $\boldsymbol{\sigma}_F = [0.17, 0.17, 0.17, 8.0, 8.0, 8.0]$ in Algorithm 2.2.

5.4 Application on Non-Cooperative Leader Follower Formation

The pose estimation algorithm is applied on a non-cooperative leader-follower formation scenario. Two car-like mobile robots - the outdoor Merlin robot - are used in this application, for which the kinematic model can be expressed as

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) \\ \sin\theta(t) \\ (\tan\beta(t))/l \end{bmatrix} v(t),$$

where v is the translational speed of the robot, β is the steering angle of the front wheels, $[x(t), y(t)]^\top$ is the position of the mid-point on the rear-wheel axle in the global world coordinate system at time t, θ represents the orientation, and l is the length between frontal and rear wheel axles.

The geometric configuration of the two robots in formation is illustrated in Fig. 5.13. The leader robot is tele-operated while the follower robot is controlled with the formation scheme in [172], where the objective is to maintain a pre-specified constant distance ρ and a relative angle α between the two robots.

The estimated pose of the leader robot in the camera system will be transformed

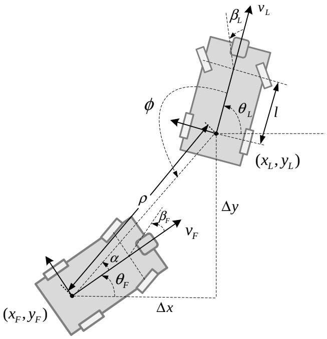


Figure 5.13: Model of the leader-follower formation for two car-like vehicles. Parameters for the Leader and the Follower robots are marked with suffix L and F respectively. Image courtesy of [171].

into the follower's coordinate system and used as the input information for the leader. The required information for the follower can be obtained from the mounted sensors, e.g. the incremental sensor, gyroscope, etc. The control input $\mathbf{u}_F = [v_F, \beta_F]^\top$ contains the translational speed v_F and the steering angle β_F of the follower.

Some snapshots for the formation test are shown in Fig. 5.14. In this test, the relative distance and angle to be maintained are set to be $\rho = 1.2$ m and $\alpha = 0°$. The leader is tele-operated to drive in an approximate circular path. Therefore, for most of the frames, the leader robot appears more on the right side of the follower's observation image, which can be seen from the images in the second row of Fig. 5.14.

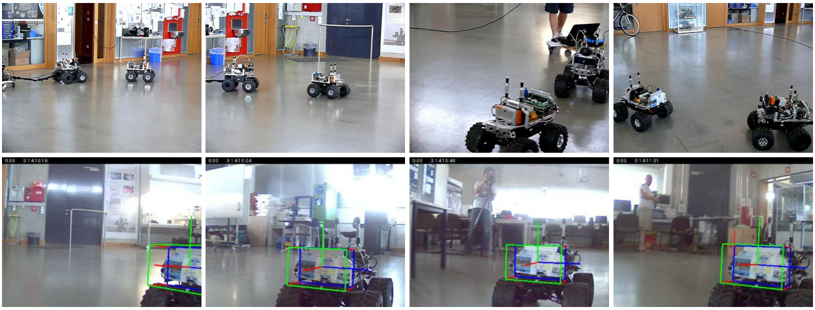


Figure 5.14: Snapshots from a complete formation sequence. The first row shows some frames grabbed from the formation test sequence. The second row gives the estimated leader pose for its above frame.

The left frame in Fig. 5.14 is of special interest, where the leader made a sharp turn towards right and almost moved out of the field of view of the fused cameras on the follower. Although in the subsequent frames the follower effectively steered and successfully maintained the formation, it can happen that the leader moves out of the observation of the follower robot and the formation will fail in such a condition. The 2D camera AXIS PTZ 212 used in this work is a fish eye color camera, which can have a quite large field of view. Currently, the target will be judged as successfully located when the pose estimation succeeds in both coarse and accurate stages. Under this configuration, when the output pose of the coarse stage lies out of the PMD camera's field of view, the accurate stage will report estimation failure. Consequently, the formation will be paused. Therefore, the proposed algorithm can be implemented to report pose results from both stages and some control schemes can be developed to maintain the formation even if the accurate stage is determined as failed.

In the above test, the poses of both leader and follower robots are measured with iSpace system and the results are shown in Fig. 5.15. The top view in Fig. 5.15 (a) illustrates the overall paths the formation robots drove along.

The robots started from a left middle point on the path and went upwards. The leader traveled an outer circle than the follower, because the objective of the formation scheme was to maintain the leader in the middle of the follower's lateral observation range and keep a pre-specified distance rather than following the path of the leader. The X and Y positions for both robots at all time steps are shown in Fig. 5.15 (b) and (c) respectively. When the fused cameras are mounted on a mobile robot, the abrupt motion of the robot, especially at start and stop, can cause significant motion artefacts on the camera measurements. The above formation test, together with the outdoor tests presented in Sec. 5.3, demonstrated the robustness (i.e. the reliable performance under significant background changes and under noisy range measurements caused by driving on grassland) of the proposed algorithm.

5.5 Summary

A series of experiments are performed in this chapter, including the reference evaluation with iSpace measurements, the tests on various targets in both indoor and outdoor environments. For the reference comparison, a calibration method is proposed for unifying the estimated pose and the iSpace measurement into one coordinate system. Despite of the error sources from the camera measurements, from the iSpace measurements, from the unification process, as well the time synchronization

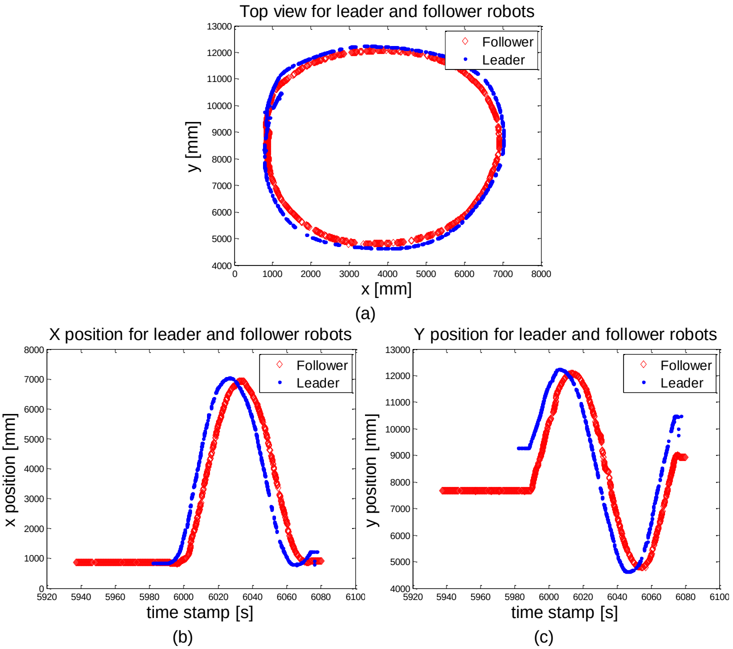


Figure 5.15: iSpace measurements for the formation scenario. (a) shows the top view of the paths for both robots. (b) and (c) depict the translational positions along X and Y axes with respect to time for both robots. The leader and follower data are drawn in red and blue respectively.

between the two systems, the translational and rotational errors are within 15 mm and 3° respectively.

The robustness of the pose estimation algorithm is demonstrated through experiments on a variety of targets. The proposed algorithm can handle symmetric objects, which will be ill-posed for the range data based conventional ICP. In both indoor and outdoor environments, the target pose can be robustly estimated under a arbitrary cluttered background. Furthermore, the proposed algorithm is applied on a non-cooperative leader-follower mobile robot formation scenario. The result shows that the pose estimation algorithm can provide a reliable input for such applications.

Chapter 6

Conclusion and Future Works

In this monograph, the 6DOF object pose estimation problem is discussed, where the target with a complex or symmetric shape can move fast in a cluttered background. Many practical robotic applications can benefit from a reliable pose estimation algorithm that can deal with such conditions with real-time efficiency, e.g. the leader-follower formation driving, the in-plant transportation, the self-localization for mobile robots, the rendezvous and docking for spacecrafts, etc. Recent advances on the TOF sensor for producing a high quality range image in a scene provides new possibilities for achieving this goal. This work is developed upon the use of a TOF sensor - PMD camera - for the pose estimation. To overcome some deficiencies of the current TOF cameras, i.e. lack of color information for each pixel and sensitive to motion, a commodity color camera is fused with the PMD camera to produce the RGBD data for all pixels.

The pose estimation in this work is divided into two stages, a coarse estimation stage and an accurate estimation stage. The major task in the first stage is to separate the target from the background and provide a coarse pose estimate that can be effectively refined in the gradient based accurate estimation stage. Due to the high dimensionality of the 6DOF problem, the Annealed Particle Filter (APF) is employed to estimate the probability distribution of the pose. Each particle in APF is evaluated with Sparse Representation (SR) solved by orthogonal matching pursuit. A new flexible composition of the template matrix in SR is introduced, which can better distinguish an image patch grabbed on the target region from a patch on the background. The distinctive power is further harnessed with a multiresolution strategy. Thus in the lower layers of APF, a small number of particles suffice to approximately locate the target; and in the higher layers, minor differences between the closely positioned particles can be detected, which increases the accuracy of the estimated pose. Some rules for updating the template matrix is proposed to adapt to the changes during tracking, e.g. the background variations, or the target appearance changes.

The accurate estimation stage refines the output from the coarse stage. Since the coarse pose estimate usually is quite close to the desired pose, the accurate stage adopts a gradient based optimization method - the point-to-plane ICP with a projective data association. For dealing with the symmetric geometry as well as the noisy range measurements from the PMD camera, the conventional ICP is extended to incorporate the target texture information, yielding the Textured-ICP or LKICP in this thesis. Compared to the range data based conventional ICP, the LKICP converges faster to the optimum, meanwhile it can better tolerate the initial pose errors.

Since both coarse and accurate estimation stages use the target appearance information, which can be significantly influenced by the illumination conditions, Spherical Harmonics (SH) illumination modeling is exploited to cope with the illumination variations. The required surface normal is calculated with the PMD range measurements, the target reflectance is estimated with the help of a calibration object. After incorporating the online synthetic images from the SH model into both estimation stages, even the extreme lighting conditions (e.g. illuminated from one side) can be tackled.

The proposed algorithm is evaluated with a set of experiments. The reference evaluation is conducted by comparing the estimated pose with the iSpace measurements. A calibration algorithm for converting the iSpace data into the camera coordinate system is introduced. Despite the error sources from the iSpace data, from the camera measurements, and from the coordinate unification process, the errors for the estimated pose are within 15 mm in translation and 3° in rotation. The robustness of the algorithm can be demonstrated through the tests on various targets in both indoor and outdoor environments, where the fast inter-frame movements, the motion artefacts, and the symmetric geometries can be well handled. The parallelism inside the algorithm is investigated for improving the computational efficiency and the real-time performance can be achieved with GPU acceleration.

The major innovations and contributions in this work can be briefly summarized as:

- In the coarse estimation stage, a flexible composition of the template matrix in SR is presented. Compared to the state-of-the-art methods, it can better distinguish a target image patch from a background one. Several online update rules are proposed to effectively incorporate the changes into the model, which do not accumulate and propagate the estimation errors from frame to frame.

- In the accurate estimation stage, a gradient based pose optimization algorithm is proposed, which integrates texture into the conventional range data based ICP. Compared to conventional ICP, the resulting Textured-ICP can handle not only geometrically symmetric objects, but also noisy range measurements.

- By investigating the parallelism of both coarse and accurate stages, it is demonstrated that real-time performance can be achieved with GPU acceleration.

- SH illumination model is incorporated into the proposed pose estimation framework. Compared to some prevalent approaches (e.g. the SURF keypoints based methods), the target pose estimation can be reliably performed under a varying lighting condition by using online SH synthetic images.

- A calibration method is presented which can unify the pose estimated in the camera coordinates with the pose measured with iSpace system for a reference evaluation.

The work in this thesis can be extended in several aspects. Currently, the target surface reflectance required in the SH modeling is estimated by using a calibration object. In most practical robotic applications, such an intervention to the scene is not desired. Therefore, an automatic estimation method with no intervention and less camera shots to the target can be a future research direction. Also the multi-target pose estimation as demonstrated in [33] can be a good research direction.

The proposed algorithm is tested on various targets but only for the side that is visible during initialization. When a complete 3D model of the target with all the required information is available, the pose estimation algorithm can be adopted and evaluated for a complete rotation. The rules for updating the visible points set $\mathcal{M}_d$ in Subsection 2.3.3 should be further modified, because after several updates, the visible points in $\mathcal{M}_d$ will be in a random sequence in the current configuration. Whereas the wavelet basis will be most useful for modeling natural signals, the randomness of the points in $\mathcal{M}_d$ will make the wavelet basis much less useful. One possible improvement can be made that each time $\mathcal{M}_d$ is updated, it should be reorganized according to the pixel reading sequence of all points in the observation image. Meanwhile, the background templates should also be rearranged to be consistent with the arrangement of the visible points in $\mathcal{M}_d$.

For an arbitrary object, a 3D model with all required information will not be available. This can be solved by a 3D online modeling approach with the estimated pose. This will involve a series of research topics. [71] proposed an online method for building the 3D geometric model of the scene. However, the question regarding how to incorporate the required color information is still left open. Besides, their method only constructs a 3D model for the complete scene. For modeling a single object, a efficient separation of the object and the backgrounds also needs to be investigated. Other problems, like the error accumulation [165] and the loop closure [163], the choice on the model expression, e.g. voxel model [36], polynomial model [5] or surfel model [163], also deserve an in-depth research.

The SR framework in the coarse stage is capable of dealing with the occlusion types that can be approximately predicted a priori. However, the Textured-ICP in the accurate stage does not consider the occlusion conditions. Since the Textured-ICP uses the range measurements, if combining the information that can be conveyed from the usage of the occlusion templates in the SR modeling, the Textured-ICP can be extended for handling the occlusion. Furthermore, the proposed 6DOF pose estimation algorithm is for the 3D tracking of a single target. How multi-target tracking can be realized is still an open question.

Bibliography

[1] Neil G Alldrin, Satya P Mallick, and David J Kriegman. Resolving the Generalized Bas-Relief Ambiguity by Entropy Minimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7. IEEE, 2007. 115

[2] M. Andrecut. Fast GPU implementation of sparse signal recovery from random projections. *Engineering Letters*, 17(3):151–158, 2009. 63

[3] Pedram Azad, D Munch, Tamim Asfour, and Rüdiger Dillmann. 6-DoF Model-Based Tracking of Arbitrarily Shaped 3D Objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5204–5209. IEEE, 2011. 33, 73

[4] Hernan Badino, Daniel Huber, Y Park, and Takeo Kanade. Fast and Accurate Computation of Surface Normals from Range Images. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3084–3091. IEEE, 2011. 92, 93

[5] Chandrajit L Bajaj, Fausto Bernardini, and Guoliang Xu. Automatic Reconstruction of Surfaces and Scalar Fields from 3D Scans. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 109–118. ACM, 1995. 156

[6] Simon Baker and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision (IJCV)*, 56(3):221–255, 2004. 80, 85

[7] Richard G Baraniuk. Compressive Sensing [lecture notes]. *IEEE Signal Processing Magazine*, 24(4):118–121, 2007. 26

[8] Bogumil Bartczak and Reinhard Koch. Dense Depth Maps from Low Resolution Time-of-Flight Depth and High Resolution Color Views. In *Advances in Visual Computing*, pages 228–239. Springer, 2009. 10

[9] R. Basri and D.W. Jacobs. Lambertian Reflectance and Linear Subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25 (2):218–233, 2003. 104, 110, 111, 112, 119

[10] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3): 346–359, 2008. 11, 102, 126

[11] Peter N Belhumeur, David J Kriegman, and Alan L Yuille. The Bas-Relief Ambiguity. *International Journal of Computer Vision (IJCV)*, 35(1):33–44, 1999. 105

[12] P.N. Belhumeur and D.J. Kriegman. What Is the Set of Images of an Object under All Possible Illumination Conditions? *International Journal of Computer Vision (IJCV)*, 28(3):245–260, 1998. 104

[13] Paul J Besl and Neil D McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14 (2):239–256, 1992. 61, 71, 74, 76

[14] Christopher M Bishop et al. *Pattern Recognition and Machine Learning*, volume 1. Springer, New York, 2006. 38, 103

[15] S. Biswas, G. Aggarwal, and R. Chellappa. Robust Estimation of Albedo for Illumination-Invariant Matching and Shape Recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(5):884–899, 2009. 105, 107, 119

[16] Gérard Blais and Martin D. Levine. Registering Multiview Range Data to Create 3D Computer Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8):820–824, 1995. 75, 76

[17] Martin Böhme, Martin Haker, Thomas Martinetz, and Erhardt Barth. Shading Constraint Improves Accuracy of Time-of-Flight Measurements. *Computer Vision and Image Understanding*, 114(12):1329–1335, 2010. 9

[18] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008. 14

[19] Matthieu Bray, Esther Koller-Meier, and Luc Van Gool. Smart Particle Filtering for High-Dimensional Tracking. *Computer Vision and Image Understanding*, 106(1):116–129, 2007. 32

[20] T Tony Cai, Lie Wang, and Guangwu Xu. New bounds for restricted isometry constants. *IEEE Transactions on Information Theory*, 56(9):4388–4394, 2010. 35

[21] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009. 29

[22] Emmanuel J Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005. 35

[23] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008. 26, 35

[24] Emmanuel J Candès, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006. 34, 35

[25] Derek Chan, Hylke Buisman, Christian Theobalt, Sebastian Thrun, et al. A noise-aware filter for real-time depth upsampling. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2*, 2008. 10

[26] Cheng Chen, Yi Yang, Feiping Nie, and Jean-Marc Odobez. 3D human pose recovery from image by efficient visual feature selection. *Computer Vision and Image Understanding*, 115(3):290–299, 2011. 30

[27] Pei Chen. Hessian Matrix vs. Gauss-Newton Hessian Matrix. *SIAM Journal on Numerical Analysis*, 49(4):1417–1435, 2011. 85

[28] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1): 33–61, 1998. 27, 35, 41

[29] Dmitry Chetverikov, Dmitry Stepanov, and Pavel Krsek. Robust Euclidean Alignment of 3D Point Sets: The Trimmed Iterative Closest Point Algorithm. *Image and Vision Computing*, 23(3):299–309, 2005. 76, 88

[30] Changhyun Choi and Henrik I Christensen. 3D Pose Estimation of Daily Objects Using an RGB-D Camera. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3342–3349. IEEE, 2012. 73

[31] Edwin KP Chong and Stanislaw H Zak. *An introduction to optimization*. Wiley, 2013. 27

[32] Chi Kin Chow, Hung Tat Tsui, and Tong Lee. Surface Registration Using A Dynamic Genetic Algorithm. *Pattern recognition*, 37(1):105–117, 2004. 74

[33] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The MOPED framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30(10):1284–1306, 2011. 73, 147, 155

[34] Ryan Crabb, Colin Tracey, Akshaya Puranik, and James Davis. Real-time foreground segmentation via range and color imaging. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–5. IEEE, 2008. 11

[35] Yan Cui, Sebastian Schuon, Derek Chan, Sebastian Thrun, and Christian Theobalt. 3D Shape Scanning with a Time-of-Flight Camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1173–1180. IEEE, 2010. 11

[36] W Bruce Culbertson, Thomas Malzbender, and Greg Slabaugh. Generalized Voxel Coloring. *Lecture notes in computer science*, pages 100–115, 1999. 156

[37] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005. 102

[38] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 126–133. IEEE, 2000. 31, 33, 39, 50

[39] Jonathan Deutscher, Andrew Davison, and Ian Reid. Automatic Partitioning of High Dimensional Search Spaces Associated with Articulated Body Motion Capture. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–669. IEEE, 2001. 33, 39

[40] James Diebel and Sebastian Thrun. An Application of Markov Random Fields to Range Sensing. *Advances in Neural Information Processing Systems*, 18: 291–298, 2005. 10

[41] David L Donoho and Xiaoming Huo. Combined image representation using edgelets and wavelets. In *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, pages 468–476. International Society for Optics and Photonics, 1999. 43

[42] David Leigh Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 25, 26, 34

[43] J.O.B. Dorsey, F.X. Sillion, and D.P. Greenberg. Design and Simulation of Opera Lighting and Projection Effects. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 41–50. ACM, 1991. 108

[44] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. *Statistics and computing*, 10 (3):197–208, 2000. 32, 37, 38

[45] Ivan Dryanovski, William Morris, Ravi Kaushik, and Jizhong Xiao. Real-Time Pose Estimation with RGB-D Camera. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 13–20. IEEE, 2012. 75

[46] Marco F Duarte and Richard G Baraniuk. Recovery of Frequency-Sparse Signals from Compressive Measurements. In *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 599–606. IEEE, 2010. 26

[47] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008. 26, 29

[48] Marco F Duarte, Michael B Wakin, and Richard G Baraniuk. Wavelet-Domain Compressive Signal Reconstruction Using a Hidden Markov Tree Model. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5137–5140. IEEE, 2008. 28

[49] Theresa Eckert. Object Visual Recognition and Tracking under Varying Illumination with Fused ToF and RGB cameras. Master's thesis, University of Würzburg, July 2013. 118

[50] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004. 28

[51] David W Eggert, Adele Lorusso, and Robert B Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5-6):272–290, 1997. 76

[52] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 28

[53] Yong Fang, Liang Chen, Jiaji Wu, and Bormin Huang. GPU Implementation of Orthogonal Matching Pursuit for Compressive Sensing. In *IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1044–1047. IEEE, 2011. 63

[54] Mário A.T. Figueiredo, Robert D Nowak, and Stephen J Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007. 28, 35, 36

[55] Andrew W Fitzgibbon. Robust Registration of 2D and 3D Point Sets. *Image and Vision Computing*, 21(13):1145–1153, 2003. 76

[56] Simon Flöry and Michael Hofer. Surface fitting and registration of point clouds using approximations of the unsigned distance function. *Computer Aided Geometric Design*, 27(1):60–77, 2010. 29

[57] Stefan Fuchs and Stefan May. Calibration and Registration for Precise Surface Reconstruction with Time-of-Flight Cameras. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):274–284, 2008. 9

[58] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(6):643–660, 2001. 104

[59] Thilo Grundmann, Robert Eidenberger, Martin Schneider, Michael Fiegert, and Georg v Wichert. Robust High Precision 6D Pose Determination in Complex Environments for Robotic Manipulation. In *Proc. Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation at the Int. Conf. Robotics and Automation*, pages 1–6, 2010. 73

[60] W Hannemann, A Linarth, B Liu, and G Kokai. Increasing depth lateral resolution based on sensor fusion. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):393–401, 2008. 10

[61] K. Hara, K. Nishino, and K. Ikeuchi. Determining Reflectance and Light Position from a Single Image without Distant Illumination Assumption. In *International Conference on Computer Vision (ICCV)*, pages 560–567. IEEE, 2003. 105, 106, 119

[62] K. Hara, K. Nishino, and K. Ikeuchi. Mixture of Spherical Distributions for Single-View Relighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(1):25–35, 2008. 106, 119

[63] Gustaf Hendeby, Jeroen D Hol, Rickard Karlsson, and Fredrik Gustafsson. A Graphics Processing Unit Implementation of the Particle Filter. In *Proceedings of European Signal Processing Conference, Poznan, Poland*, 2007. 63

[64] Robin Heß and Klaus Schilling. GPS/Galileo testbed using a high precision optical positioning system. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 87–96. Springer, 2010. 141

[65] Olga Holtz. Compressive sensing: a paradigm shift in signal processing. *arXiv preprint arXiv:0812.3137*, 2008. 26, 35

[66] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of Optical Sociery of America A*, 4(4):629–642, 1987. 74

[67] Yuan-Kui HU and Zeng-Fu WANG. A Low-dimensional Illumination Space Representation of Human Faces for Arbitrary Lighting Conditions. *Acta Automatica Sinica*, 33(1):9–14, 2007. 107

[68] Jia-Bin Huang and Ming-Hsuan Yang. Estimating Human Pose from Occluded Images. *Computer Vision–ACCV*, pages 48–60, 2010. 30

[69] K. Ikeuchi and K. Sato. Determining Reflectance Properties of an Object Using Range and Brightness Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(11):1139–1153, 1991. 105, 106, 119

[70] Michael Isard and Andrew Blake. CONDENSATION—Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision (IJCV)*, 29(1):5–28, 1998. 31, 39

[71] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011. 155

[72] Andrew Edie Johnson and Sing Bing Kang. Registration and integration of textured 3D data. *Image and vision computing*, 17(2):135–147, 1999. 76, 79

[73] Graeme A Jones. Combining Optical Flow and Range Flow to Recover RGBD Sensor Ego-Motion. In *RGB-D workshop at Robotics Science and Systems (RSS) Conference*, Berlin, Germany, June 2013. 80

[74] Seung-Jean Kim, Kwangmoo Koh, Michael Lustig, Stephen Boyd, and Dimitry Gorinevsky. An Interior-Point Method for Large-Scale ℓ_1-Regularized Least Squares. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):606–617, 2007. 27, 30, 35

[75] Klaas Klasing, Daniel Althoff, Dirk Wollherr, and Martin Buss. Comparison of Surface Normal Estimation Methods for Range Sensing Applications. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3206–3211. IEEE, 2009. 91, 93

[76] Ralph Klose, Jaime Penlington, and Arno Ruckelshausen. Usability study of 3D Time-of-Flight cameras for automatic plant phenotyping. *Bornimer Agrartechnische Berichte*, 69:93–105, 2009. 12

[77] S. Knoop, S. Vacek, and R. Dillmann. Fusion of 2D and 3D sensor data for articulated body tracking. *Robotics and Autonomous Systems*, 57(3):321–329, 2009. 11

[78] Andreas Kolb, Erhardt Barth, Reinhard Koch, and Rasmus Larsen. Time-of-Flight Sensors in Computer Graphics. In *Proc. Eurographics (State-of-the-Art Report)*, 2009. 6

[79] Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and Object Tracking for In-hand 3D Object Modeling. *The International Journal of Robotics Research (IJRR)*, 30(11):1311–1327, 2011. 74, 77

[80] NM Kwok, Gu Fang, and Weizhen Zhou. Evolutionary Particle Filter: Re-sampling from the Genetic Algorithm Perspective. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2935–2940. IEEE, 2005. 33, 38, 52

[81] Nicolas H Lehment, Dejan Arsic, Moritz Kaiser, and Gerhard Rigoll. Automated Pose Estimation in 3D Point Clouds Applying Annealing Particle Filters and Inverse Kinematics on a GPU. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 87–92. IEEE, 2010. 63

[82] Raskin Leonid, Rivlin Ehud, and Rudzsky Michael. Using Gaussian Process Annealing Particle Filter for 3D Human Tracking. *EURASIP Journal on Advances in Signal Processing*, 2008:1–13, 2008. 33

[83] Hanxi Li, Chunhua Shen, and Qinfeng Shi. Real-time visual tracking using compressive sensing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1305–1312. IEEE, 2011. 24, 30, 36, 41, 55

[84] Peihua Li, Tianwen Zhang, and Arthur EC Pece. Visual Contour Tracking Based on Particle Filters. *Image and Vision Computing*, 21(1):111–123, 2003. 32

[85] Xiaoxing Li, Tao Jia, and Hao Zhang. Expression-insensitive 3d face recognition using sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2575–2582. IEEE, 2009. 29

[86] Yuan Li, Haizhou Ai, Takayoshi Yamashita, Shihong Lao, and Masato Kawade. Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Life Spans. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(10):1728–1740, 2008. 39

[87] Joerg Liebelt and Klaus Schertler. Precise Registration of 3D Models to Images by Swarming Particles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 33, 74

[88] João Paulo Lima, Veronica Teichrieb, Hideaki Uchiyama, Eric Marchand, et al. Object Detection and Pose Estimation from Natural Features Using Consumer RGB-D Sensors: Applications in Augmented Reality. In *IEEE Int. Symp. on Mixed and Augmented Reality (doctoral symposium), ISMAR'12*, Atlanta, USA, November 2012. 73, 148

[89] AG Linarth, J. Penne, B. Liu, O. Jesorsky, and R. Kompe. Fast fusion of range and video sensor data. In *Advanced Microsystems for Automotive Applications*, volume 16, pages 119–134. Springer, 2007. 10, 15, 16

[90] M. Lindner, A. Kolb, and K. Hartmann. Data-fusion of PMD-based distance-information and high-resolution RGB-images. In *International Symposium on Signals, Circuits and Systems (ISSCS)*, volume 1, pages 1–4. IEEE, 2007. 13, 17

[91] Marvin Lindner and Andreas Kolb. Lateral and Depth Calibration of PMD-Distance Sensors. In *Advances in Visual Computing*, pages 524–533. Springer, 2006. 9

[92] Marvin Lindner and Andreas Kolb. Calibration of the Intensity-Related Distance Error of the PMD ToF-Camera. In *Optics East*, pages 67640W–67640W. International Society for Optics and Photonics, 2007. 9

[93] Marvin Lindner and Andreas Kolb. Compensation of Motion Artifacts for Time-of-Flight Cameras. In *Dynamic 3D Imaging*, pages 16–27. Springer, 2009. 7, 10

[94] Marvin Lindner, Andreas Kolb, and Thorsten Ringbeck. New Insights into the Calibration of ToF-Sensors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–5. IEEE, 2008. 9

[95] Marvin Lindner, Martin Lambers, and Andreas Kolb. Sub-Pixel Data Fusion and Edge-Enhanced Distance Refinement for 2D/3D Images. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):344–354, 2008. 17

[96] Marvin Lindner, Ingo Schiller, Andreas Kolb, and Reinhard Koch. Time-of-flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding*, 114(12):1318–1328, 2010. 8

[97] Evgeny Lomonosov, Dmitry Chetverikov, and Anikó Ekárt. Pre-Registration of Arbitrarily Oriented 3D Surfaces Using A Genetic Algorithm. *Pattern Recognition Letters*, 27(11):1201–1208, 2006. 75

[98] O Lottner, A Sluiter, K Hartmann, and W Weihs. Movement Artefacts in Range Images of Time-of-Flight Cameras. In *International Symposium on Signals, Circuits and Systems (ISSCS)*, volume 1, pages 1–4. IEEE, 2007. 10, 14

[99] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 11, 102, 126

[100] Jun Luo, Liu Xiang, and Catherine Rosenberg. Does compressed sensing improve the throughput of wireless sensor networks? In *IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2010. 29

[101] Bingpeng Ma and Tianjiang Wang. Head pose estimation using sparse representation. In *Second International Conference on Computer Engineering and Applications (ICCEA)*, volume 2, pages 389–392. IEEE, 2010. 30

[102] Stephane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 28

[103] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(11):2259–2272, 2011. 24, 30, 41, 55, 100, 102, 125

[104] X. Mei, H. Ling, and D.W. Jacobs. Illumination Recovery from Image with Cast Shadows via Sparse Representation. *IEEE Transactions on Image Processing*, 20(8):2366–2377, 2011. 25, 31, 107

[105] B. Mercier, D. Meneveaux, and A. Fournier. A Framework for Automatically Recovering Object Shape, Reflectance and Light Sources from Calibrated Images. *International Journal of Computer Vision (IJCV)*, 73(1):77–93, 2007. 105, 108

[106] Philipp Michel, J Chestnut, Satoshi Kagami, Koichi Nishiwaki, James Kuffner, and Takeo Kanade. GPU-Accelerated Real-Time 3D Tracking for Humanoid Locomotion and Stair Climbing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 463–469. IEEE, 2007. 73

[107] Niloy J Mitra and An Nguyen. Estimating Surface Normals in Noisy Point Cloud Data. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 322–328. ACM, 2003. 91, 93

[108] Baback Moghaddam and Alex Pentland. Probabilistic Visual Learning for Object Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):696–710, 1997. 102

[109] Tobias Möller, Holger Kraft, Jochen Frey, Martin Albrecht, and Robert Lange. Robust 3D Measurement with PMD Sensors. *Range Imaging Day, Zürich*, 2005. 7, 8

[110] Hankyu Moon and Matt L Miller. Estimating facial pose from a sparse representation [face recognition applications]. In *International Conference on Image Processing*, volume 1, pages 75–78. IEEE, 2004. 30

[111] I. Moreno and C.C. Sun. LED array: where does far-field begin? In *8th International Conference on Solid State Lighting, Proc. of SPIE*, volume 7058, pages 70580R1–70580R9, 2008. 109

[112] I. Moreno and C.C. Sun. Modeling the radiation pattern of LEDs. *Optics Express*, 16(3):1808–1819, 2008. 109

[113] I. Moreno and R.I. Tzonchev. Effects on illumination uniformity due to dilution on arrays of LEDs. In *Proc. of SPIE*, volume 5529, pages 268–275, 2004. 109

[114] I. Moreno, M. Avendaño-Alejo, and R.I. Tzonchev. Designing light-emitting diode arrays for uniform near-field irradiance. *Applied optics*, 45(10):2265–2272, 2006. 108

[115] Andriy Myronenko, Xubo Song, and Miguel A Carreira-Perpinán. Non-Rigid Point Set Registration: Coherent Point Drift. *Advances in Neural Information Processing Systems*, 19:1009, 2007. 75, 77

[116] C Netramai and H Roth. Real-Time 3D Motion Estimation and Map Building Using Enhanced Multi-Camera System. In *11th International Conference on Modern Information and Electronic Technologies*, May 2010. 73

[117] C Netramai, O Melnychuk, C Joochim, and H Roth. Combining PMD and Stereo Camera for Motion Estimation of a Mobile Robot. In *Proceedings of the 17th World Congress, The International Federation of Automatic Control (IFAC)*, pages 5417–5422, 2008. 11

[118] Richard A Newcombe, Andrew J Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136. IEEE, 2011. 62, 75, 77, 81, 103

[119] Takahiro Okabe, Imari Sato, and Yoichi Sato. Spherical Harmonics vs. Haar Wavelets: Basis for Recovering Illumination from Cast Shadows. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–50. IEEE, 2004. 105, 107

[120] International Commission on Illumination. Measurement of LEDs, 2007. 108

[121] M. Osadchy, D. Jacobs, and R. Ramamoorthi. Using specularities for recognition. In *International Conference on Computer Vision (ICCV)*, pages 1512–1519. IEEE, 2003. 106, 119

[122] Qi Pan, Gerhard Reitmayr, and Tom Drummond. ProFORMA: Probabilistic Feature-Based On-line Rapid Model Acquisition. In *Proc. 20th British Machine Vision Conference (BMVC)*, 2009. 73

[123] Giorgio Panin and Alois Knoll. Mutual Information-Based 3D Object Tracking. *International Journal of Computer Vision (IJCV)*, 78(1):107–118, 2008. 74, 81, 103

[124] Seongkeun Park, Jae Pil Hwang, Euntai Kim, and Hyung-Jin Kang. A New Evolutionary Particle Filter for the Prevention of Sample Impoverishment. *IEEE Transactions on Evolutionary Computation*, 13(4):801–809, 2009. 33, 38, 52

[125] Steven Parker, Peter Shirley, Yarden Livnat, Charles Hansen, and P-P Sloan. Interactive Ray Tracing for Isosurface Rendering. In *Proceedings on Visualization*, pages 233–238. IEEE, 1998. 75

[126] Thang V Pham and Arnold WM Smeulders. Sparse representation for coarse and fine object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(4):555–567, 2006. 30

[127] A. Prusak, O. Melnychuk, H. Roth, and I. Schiller. Pose estimation and map building with a Time-Of-Flight-camera for robot navigation. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):355–364, 2008. 11

[128] Mark Pupilli and Andrew Calway. Real-Time Camera Tracking Using Known 3D Models and A Particle Filter. In *18th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 199–203. IEEE, 2006. 73

[129] R. Ramamoorthi. Analytic PCA Construction for Theoretical Analysis of Lighting Variability in Images of a Lambertian Object. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(10):1322–1333, 2002. 104

[130] R. Ramamoorthi. Modeling Illumination Variation with Spherical Harmonics. *Face Processing: Advanced Modeling and Methods*, 2005. 100, 104, 110, 111, 120

[131] R. Ramamoorthi and P. Hanrahan. A Signal-Processing Framework for Inverse Rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 117–128. ACM, 2001. 104, 105, 118, 119

[132] K Ravandoor, S Busch, L Regoli, and K Schilling. Evaluation and Performance Optimization of PMD Camera for RvD Application. In *19th IFAC Symposium on Automatic Control in Aerospace (ACA)*, volume 1, pages 149–154, 2013. 9

[133] Dikpal Reddy, Aswin C Sankaranarayanan, Volkan Cevher, and Rama Chellappa. Compressed sensing for multi-view tracking and 3-D voxel reconstruction. In *15th IEEE International Conference on Image Processing (ICIP)*, pages 221–224. IEEE, 2008. 31

[134] L. Regoli, K. Ravandoor, M. Schmidt, and K. Schilling. Advanced Techniques for Spacecraft Motion Estimation Using PMD Sensors. In *1st IFAC Confer-*

ence on Embeded System, Computational Intelligence and Telematics in Control (CESCIT), pages 320–325, 2012. 11, 103

[135] L Regoli, K Ravandoor, C Herrmann, and K Schilling. New Testing Facility for Proximity Operations. In *19th IFAC Symposium on Automatic Control in Aerospace (ACA)*, volume 1, pages 348–353, 2013. 142

[136] Kevin Rosenblum, Lihi Zelnik-Manor, and Yonina C Eldar. Dictionary optimization for block-sparse representations. In *AAAI Fall 2010 Symposium on Manifold Learning*, pages 50–58, 2010. 28

[137] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision (IJCV)*, 77(1):125–141, 2008. 100, 102

[138] Yong Rui and Yunqiang Chen. Better Proposal Distributions: Object Tracking Using Unscented Particle Filter. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–786. IEEE, 2001. 32

[139] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algorithm. In *Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. IEEE, 2001. 76

[140] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-Time 3D Model Acquisition. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 438–446. ACM, 2002. 75, 76

[141] Romeil Sandhu, Samuel Dambreville, and Allen Tannenbaum. Particle Filtering for Registration of 2D and 3D Point Sets with Stochastic Dynamics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 32, 74, 76

[142] Yoav Sharon, John Wright, and Yi Ma. Computation and relaxation of conditions for equivalence between $\ell 1$ and $\ell 0$ minimization. Technical report, University of Illinois, 2007. 26

[143] Qinfeng Shi, James Petterson, Gideon Dror, John Langford, Alex Smola, and SVN Vishwanathan. Hash kernels for structured data. *The Journal of Machine Learning Research*, 10:2615–2637, 2009. 36

[144] Malcolm D Shuster. A Survey of Attitude Representations. *The Journal of the Astronautical Sciences*, 41(4):439–517, 1993. 61

[145] Michael Sturmer, Jochen Penne, and Joachim Hornegger. Standardization of Intensity-Values Acquired by Time-of-Flight-Cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–6. IEEE, 2008. 9

[146] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited [medical images application]. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000. 81, 94

[147] Philippe Thévenaz and Michael Unser. Optimization of Mutual Information for Multiresolution Image Registration. *IEEE Transactions on Image Processing*, 9(12):2083–2099, 2000. 74

[148] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 27

[149] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2004. 25, 28

[150] Michael E Tipping and Christopher M Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. 102

[151] Toby Smith. Reflectance Spectra Tutorial. `http://www.astro.washington.edu/users/smith/Astro150/Tutorials/Spectra/`, 2009. 116

[152] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (ICCV)*, pages 839–846. IEEE, 1998. 10

[153] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007. 25, 28, 36

[154] T Tzschichholz and K Schilling. Range Extension of the PMD Sensor with Regard to Applications in Space. In *19th IFAC Symposium on Automatic Control in Aerospace (ACA)*, volume 1, pages 319–324, 2013. 8

[155] T. Tzschichholz, L. Ma, and K. Schilling. Model-based spacecraft pose estimation and motion prediction using a photonic mixer device camera. *Acta Astronautica*, 68(7):1156–1167, 2011. 11, 103

[156] C. Upright, D. Cobzas, and M. Jagersand. Wavelet-based Light Reconstruction from a Single Image. In *Fourth Canadian Conference on Computer and Robot Vision (CRV)*, pages 305–312. IEEE, 2007. 107

[157] Philipp Urban, Mitchell R Rosen, and Roy S Berns. A Spatially Adaptive Wiener Filter for Reflectance Estimation. In *Proceedings of the IS&T/SID 16th Color Imaging Conference (Society for Imaging Science and Technology/Society for Information Display)*, pages 279–284, 2008. 107, 119

[158] Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Stable Real-Time 3D Tracking Using Online and Offline Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(10):1385–1391, 2004. 73

[159] Rudolph Van Der Merwe, Arnaud Doucet, Nando De Freitas, and Eric Wan. The Unscented Particle Filter. *Advances in neural information processing systems*, pages 584–590, 2001. 32, 38

[160] Yang Wang and Dimitris Samaras. Estimation of multiple directional light sources for synthesis of augmented reality images. *Graphical Models*, 65(4): 185–205, 2003. 106, 119

[161] Zhou Wang and Alan C Bovik. Mean squared error: love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1): 98–117, 2009. 84

[162] Thibaut Weise, Bastian Leibe, and Luc Van Gool. Accurate and robust registration for in-hand modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 76, 79, 103

[163] Thibaut Weise, Thomas Wismer, Bastian Leibe, and Luc Van Gool. In-hand Scanning with Online Loop Closure. In *12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1630–1637. IEEE, 2009. 156

[164] Gaojin Wen, Zhaoqi Wang, Shihong Xia, and Dengming Zhu. Least-Squares Fitting of Multiple M-Dimensional Point Sets. *The Visual Computer*, 22(6): 387–398, 2006. 75

[165] Mark D Wheeler, Yoichi Sato, and Katsushi Ikeuchi. Consensus Surfaces for Modeling 3D Objects from Multiple Range Images. In *6th International Conference on Computer Vision (ICCV)*, pages 917–924. IEEE, 1998. 156

[166] Matthias Wiedemann, Markus Sauer, Frauke Driewer, and Klaus Schilling. Analysis and characterization of the PMD camera for application in mobile robotics. In *Proceedings of the 17th IFAC World Congress*, pages 6–11, 2008. 9

[167] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(2):210–227, 2009. 25, 26, 29, 34, 40, 54

[168] F. Xie, L. Tao, and G. Xu. Estimating Illumination Parameters Using Spherical Harmonics Coefficients in Frequency Space. *Tsinghua Science and Technology*, 12(1):44–50, 2007. 105

[169] S. Xu and AM Wallace. Recovering Surface Reflectance and Multiple Light Locations and Intensities from Image Data. *Pattern Recognition Letters*, 29 (11):1639–1647, 2008. 107, 119

[170] Yilei Xu and Amit K Roy-Chowdhury. Integrating Motion, Illumination, and Structure in Video Sequences with Applications in Illumination-Invariant Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(5):793–806, 2007. 103

[171] Zhihao Xu. *Cooperative Formation Control Design for Time-delay and Optimality Problems*. PhD thesis, University of Würzburg, 2013. 150

[172] Zhihao Xu, Martin Schroter, Dan Necsulescu, Lei Ma, and Klaus Schilling. Formation control of car-like autonomous vehicles under communication delay. In *Control Conference (CCC), 2012 31st Chinese*, pages 6376–6383. IEEE, 2012. 149

[173] Chen Yang and Gérard Medioni. Object Modelling by Registration of Multiple Range Images. *Image and Vision Computing*, 10(3):145–155, 1992. 74, 76

[174] H. Yang, J.W.M. Bergmans, T.C.W. Schenk, J.P.M.G. Linnartz, and R. Rietman. An analytical model for the illuminance distribution of a power LED. *Optics express*, 16(26):21641–21646, 2008. 109

[175] H. Yang, J.W.M. Bergmans, T.C.W. Schenk, J.P.M.G. Linnartz, and R. Rietman. Uniform Illumination Rendering Using an Array of LEDs: A Signal Processing Perspective. *IEEE Transactions on Signal Processing*, 57(3):1044–1057, 2009. 109

[176] Qingxiong Yang, Ruigang Yang, James Davis, and David Nistér. Spatial-depth super resolution for range images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 10

[177] Tianli Yu and Narendra Ahuja. Simultaneous Estimation of Texture Map and Pseudo Illumination from Multiple Views. In *In Texure 2005: Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, volume 51, pages 19–24, 2005. 105, 115

[178] Alan L Yuille, Daniel Snow, Russell Epstein, and Peter N Belhumeur. Determining Generative Models of Objects under Varying Illumination: Shape and Albedo from Multiple Images Using SVD and Integrability. *International Journal of Computer Vision (IJCV)*, 35(3):203–222, 1999. 104

[179] L. Zhang and D. Samaras. Face Recognition from a Single Training Image under Arbitrary Unknown Lighting Using Spherical Harmonics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(3):351–363, 2006. 107, 119

[180] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(11):1330–1334, 2000. 14, 83

[181] Wei Zhou and Chandra Kambhamettu. A Unified Framework for Scene Illuminant Estimation. *Image and Vision Computing*, 26(3):415–429, 2008. 104, 105, 107, 119

[182] Jiejie Zhu, Liang Wang, Ruigang Yang, and James Davis. Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 20

[183] Jiejie Zhu, Miao Liao, Ruigang Yang, and Zhigeng Pan. Joint Depth and Alpha Matte Optimization via Fusion of Stereo and Time-of-Flight Sensor. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 453–460. IEEE, 2009. 11

[184] X. Zou, J. Kittler, M. Hamouz, and J.R. Tena. Robust Albedo Estimation from Face Image under Unknown Illumination. In *Proc. of SPIE*, volume 6944, pages 69440A–69440A–11, 2008. 105, 107