Master's Thesis

Towards Precise Point Cloud based 3D Reconstruction: from Multi-view Inpainting to Online Pose Estimation

Feiran Li

August 1, 2019

Graduate School of Information Science
Nara Institute of Science and Technology

A Master's Thesis
submitted to the Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
MASTER of ENGINEERING

Feiran Li

Thesis Committee:

Professor Tsukasa Ogasawara	(Supervisor)
Professor Kenji Sugimoto	(Co-supervisor)
Associate Professor Jun Takamatsu	(Co-supervisor)
Assistant Professor Ming Ding	(Co-supervisor)
Assistant Professor Gustavo A. Garcia Ricardez	(Co-supervisor)

Towards Precise Point Cloud based 3D Reconstruction: from Multi-view Inpainting to Online Pose Estimation*

Feiran Li

Abstract

Point clouds based 3D reconstruction plays a critical role in many robotics and computer vision applications such as robot navigation, autonomous driving and medical image processing. As precision is always the first concern of point clouds, in this thesis, we specify it into two concrete issues: how to remove clutters from a point cloud and how to construct it accurately. Consequently, a multi-view based RGB-D inpainting framework and a Bingham distribution based adaptive filter-type pose estimation algorithm are proposed. Compared with past work, our inpainting framework is able to deal RGB-D sequences rather than simple RGB ones and our pose estimation filter frees the users from manual tuning as required in the prototype. And experiments show that each of them can outperform the state-of-the-art approaches.

Keywords:

RGB-D Images, Multi-view Inpainting, Pose Estimation, Bingham Distribution

*Master's Thesis, Graduate School of Information Science,
Nara Institute of Science and Technology, August 1, 2019.

Acknowledgements

This thesis would not have been possible without the all the active supports I have ever got :

First, I would like to wish my deepest thank to Prof. Tsukasa Ogasawara of the robotics lab. It was his advice that inspires me during the two-year study.

I am very grateful to my supervisor Associate Prof. Jun Takamatsu. The door to his office was always open whenever I ran into a trouble spot or have a question about my research or writing. He consistently let me know what research is and steered me in the right direction whenever he thought I needed it.

I would also like to thank Assistant Prof. Ming Ding and Assistant Prof. Gustavo A. Garcia Ricardez. They enlightened me in every detail in my research and helped me to make the paper professional.

I would also like to thank my friends and labmates, from whom I have got valuable comments and supports to make the ideas come true.

Also, I must express my very profound gratitude to my parents and Miss Yanjiao Ao for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. Thank you.

Finally, I wish this thesis won't be the end of my willingness to explore the unknown, and wish the wish to go far beyond.

Contents

List of Figures

List of Tables

1. Introduction

Nowadays people are eager to project objects and scenes in our real lives onto the screen. And 3D reconstruction is such a technique that enables us to first shoot some pictures and then shape them into a 3D model.

Precise 3D reconstruction algorithms are critical as they serve as the basis for many other applications. For example, they are necessary for constructing the high accuracy map used in autonomous driving [1], they can be used in medical imaging for health analysis [2], and they may also play a role in heritage protection [3]. Nevertheless, some work is yet to be done with respect to enhancing the real-world robustness and time efficiency.

1.1. Motivation and Research Themes

Many applications would desire precise point clouds that only contain useful scenarios and however it might be hard to configure the real environment. And therefore, reconstruction algorithms that can only focus on the useful scenes and ignore the clutters would be preferred. Also, since the final point cloud is usually constructed by unifying the poses of numerous scattered small ones, its accuracy is significantly affected by the estimated relative poses among segments.

Hereby I propose two algorithms in order to deal with the two problems as mentioned above. The first one is a multi-view inpainting framework which enables the users to remove the undesired objects from a given RGB-D image and inherently fill in the holes by leveraging information from the other frames in the same sequence. The second proposal is a filter-type, online pose estimation algorithm which is suitable for the cases where data amount is huge or the data is retrieved as times goes by.

1.2. Thesis Overview

The remainder of this thesis is organized as follows: In chapter 2 the multi-view RGB-D inpainting algorithm is introduced. In chapter 3 the filter for pose estimation problem is presented. And in chapter 4 we conclude the thesis and desicribe the potential future work. It is also worth to mention that both chapter 2 and chapter 3 are organized in the: *1) brief introduction; 2) related work; 3) proposed method; 4) experiments; 5) summary* structure.

2. Multi-view Inpainting for RGB-D Sequence

2.1. Introduction of Multi-view Inpainting

Point cloud based 3D reconstruction algorithms usually take a group of images as input and return the 3D point cloud model. Many of them have provided RGB-D versions for the easily achievable distance information. However, with such classical methods, undesired objects (*i.e.* moving objects) would be inevitably introduced to the reconstructed point clouds. Some methods [4, 5] present solutions to deal with the rigidly moving objects by clustering features into either static or dynamic classes. It was not until the recent blossom of image semantic segmentation that demonstrates new insight on how to remove the undesired objects in 2D image level with more flexibility.

However, simply removing the unwanted objects would leave blanks on the images and hence we focus on filling in these holes with inpainting techniques. Within the off-the-shelf approaches, the single image inpainting methods [6, 7] are powerless to handle large holes. As for the video completion algorithms [8, 9], they are specifically designed to deal with videos and hence not suitable for the RGB-D sequences. Therefore, we propose to solve the RGB-D inpainting problem from the multi-view inpainting perspective.

In this chapter we introduce a unified framework to inpaint a certain RGB-D frame by taking a corresponding sequence as input. With a slight abuse of notation we inherently define a pair of color and depth images as a "frame", which is the minimum unit in our algorithm. For a certain frame with blanks to fill in, our algorithm searches correspondences from the other counterparts. We consider our system as semi-automatic since the users only need to mask out the objects to remove, which can be easily done

with modern semantic segmentation neural networks [10].

2.2. Related Work

Image inpainting contains broad research fields and hereby we briefly review those closely related to our work. Our approach is a combination of multi-view image inpainting and depth Inpainting.

2.2.1. Single Image Inpainting

Exemplar based methods are popular in single image inpainting for their ability to deal with textured images. The beginning of them can be traced to the work of Criminisi *et al.* [11], in which the mask is completed via searching for similar patches from the rest region and inherently copying them. The PatchMatch algorithm proposed by Barnes *et al.* [6] uses random search for quickly finding approximate nearest neighbor matches between patches, which is widely employed as the basis in the follow-up work for its several orders higher time efficiency. Kawai *et al.* [7] extend the energy function by taking into account of brightness changes and spatial locality of texture to deal with unnatural matches. Lee *et al.* [12] propose to take Laplacian pyramid as an error term in patch synthesis in order to protect edges. In summary, the single image inpainting approaches leverage information from the image itself. In contrast, we use the other frames as additional sources.

2.2.2. Video Completion

Video completion aims at dealing with color image sequences. Some work requires interaction: Klose *et al.* [9] propose to inpaint a given video by using SfM and manually drawing 3D masks. Other methods either completely copy information from other frames or generate new textures by searching from them: Granados *et al.* [13] enable free movement of camera via using multiple homographies to estimate the geometric registration between frames; whose applications are limited for being required to satisfy the assumption that the missing pixels on the target frame can be completely achieved from the others. On the other hand, Newson *et al.* [14] propose an exemplar based method to search for similar patches on a group of aligned source frames, which is pretty time consuming for minimizing a global energy function. Similarly, Ebdelli *et al.* [8] shrink the searching range by only considering a small number of aligned

neighboring frames of the target one. Differently with these methods that handle color videos, our approach is designed for RGB-D sequences; also we take advantages from both direct copying and multi-view searching and hence more suitable for highly textured scenes.

2.2.3. Multi-View based Inpainting

Multi-view inpainting techniques leverage information from multiple source frames. Hays and Efros [15] gather photos from Internet as a huge database to help with image completion. Similarly, Whyte *et al.* [16] cover an undesired region on the query image with Internet photographs of the same scene, in which multi-homography and photometric registration are used to achieve geometric registration between the query image and the source ones. Also a Markov random field optimization [17] is employed for selecting the optimal sources. This kind of methods are pretty unstable since the masked objects are easy to be reintroduced as a result of lacking necessary means to filer the source information.

Recent research begins to show interests on using the geometric connections among different views. Baek *et al.* [18] present a multi-view based method to complete the user-defined region by jointly inpaint the color and depth image, which takes advantages from SfM to achieve geometric registration among different views. Similarly, Thonat *et al.* [19] enable free-viewpoint image based rendering with reprojected information from neighboring views. Also a refined method is proposed in the following work [20] that performs inpainting on intermediate, local planes in order to preserve perspective as well as to ensure multi-view coherence. In contrast, we use local homography for achieving pixel-wise correspondences, with which the information loss caused by SfM could be effectively avoided. Also they assume that the input images are of high quality, while in contrast, our approach aims at dealing with more common scenarios, such as those taken with moving cameras and hence glutted with blurriness.

2.2.4. Depth Inpainting

Depth inpainting is similar to the propagation methods designed for the color images to a certain degree. Result quality may however be limited if the algorithms designed for color images are simply transplanted to the depth counterparts. Therefore pop-

ular solutions use color images as guidance to complete the holes on the depth ones. Miao *et al.* [21] introduce a texture assisted inpainting technique via dividing the target area into smooth and edge classes and distribute different partial differential equations (PDE) to each class. Atapour-Abarghouei *et al.* [22] perform semantic segmentation on the color images to get the object edges and the depth value is coherently propagated within every object. Such work targets on assigning value to each unknown pixel. In this work, however, we take the unknown as one of the existing values and only inpaint the mask left by the removed undesired objects.

2.3. Proposed Method

2.3.1. Framework Overview

The proposed pipeline can be found in Fig. 2.1. A masked RGB-D sequence F and one target frame F_t are taken as input. The RGB-D sequence serves as source frames from where inpainting information can be gotten and the masks stand for the objects that we would like remove, which are semantically generated by PSPNet [10] in our work. Our goal is to fill in the masks on F_t with realistic content by using multi-view information. In order to achieve it, we carefully select a set of source frames from F for F_t. Also we take benefits from local homography based image warping method [23] to warp each source F_i into the same image coordinate with F_t. Considering that the warpings are not equally accurate, we use an MRF approach similar to those proposed in [16, 17] to reduce bias. After this, we separately use exemplar based multi-view inpainting algorithm to cope with the color image I_t and coherently inpaint the correlated depth one under guidance.

2.3.2. Select Useful Source Frames for Inpainting

In a multi-view inpainting system, we first need to sort out a set of source frames from the input RGB-D sequence that can fill in the blanks in F_t. This is a quite challenging mission considering the giant quantity and variable qualities of the input frames. In this work, we comprehensively appraise the suitability of each frame to be used as source.

Selection Strategy

Given a potential source frame F_i, the inpainting accuracy depends on both the image quality itself and the correlations between it and the target frame F_t. Therefore, we respectively grade the similarity, the inter-frame distance and the image quality of each F_i to evaluate whether it is suitable or not to inpaint F_t. It is reasonable to select these three factors because similarity ensures texture consistency; Larger distance would lead to lower warping accuracy by reducing the amount of matched feature points; as well as blurred image would not only weaken the quality of the inpainted image,

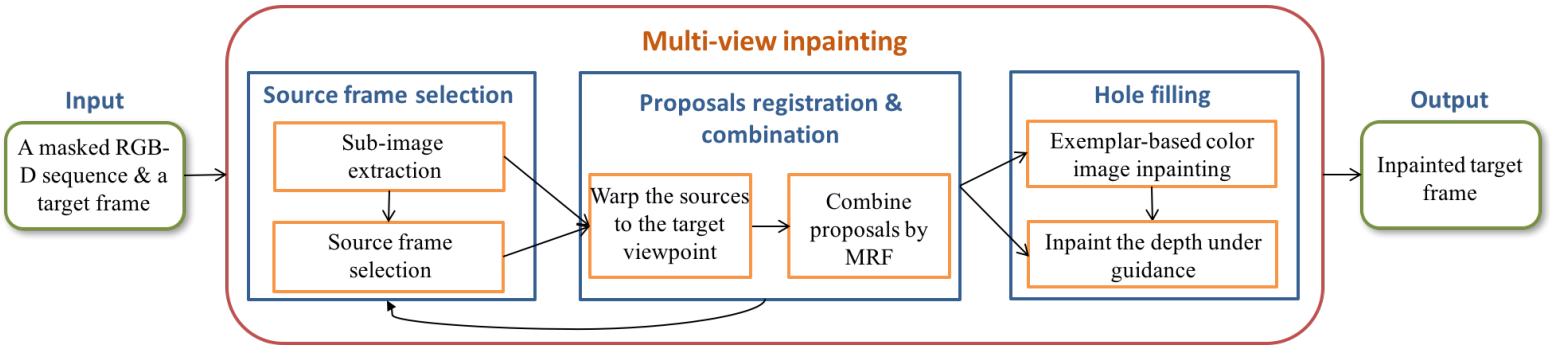


Figure 2.1.: Overview of the inpainting framework. For every input target frame, the first step is to select satisfactory source frames for inpainting. Then for every sub-image, information from the corresponding sources is comprehensively combined. Finally the left holes on the color and depth images are inpainted separately.

(a) (b) (c)

Figure 2.2.: (a) The color image of a target frame. Red color indicates completion region and each green box is a minimum circumscribed rectangle of one mask. (b) and (c) Every box is one sub-image and in each one of them, the red region indicates M_t and the left part is M_t^r

.

but may also trigger mismatches among feature points, which would further lead to high-bias warping results.

A certain target frame F_t can contain several masks. Instead of inpainting them simultaneously, we emphasize local similarity and treat each mask M_t (a tightly bounded box around the object) on F_t separately. Specifically, we split F_t into a set of sub-images by extracting the minimum circumscribed rectangle for every M_t, as shown in Fig. 2.2. Each M_t independently gets its own source frames and has not influence on the selections of the others.

We evaluate the suitability $S(F_i, M_t)$ for inpainting M_t with potential source frame F_i by Eq. 2.1

$$S(F_i, M_t) = \frac{(w_1 s(F_i, M_t) - w_2 d(F_i, M_t))q(F_i)}{q(F_t)}, \tag{2.1}$$

where $s(F_i, M_t)$ stands for the similarity between F_i and M_t; $d(F_i, M_t)$ represents the distance between F_i and M_t; as well as $q(F_i)$ and $q(F_t)$ shows the image quality of F_i

and F_t respectively. w_1 and w_2 are positive numbers for weight parameters. We use linear normalization to normalize all the factors to $[0, 1]$ range.

In practicing, it is pretty burdensome to calculate the distance term for all the F_i in F and hence we employ a two-step selection strategy. Namely, we first select the most similar n source frames by only using the similarity term $s(F_i, M_t)$ and then the $S(F_i, M_t)$ is calculated within these n frames.

Tunable Factors for similarity measurement

We consider the similarity evaluation as a typical image retrieval problem and in this work we use the bag of visual word (BoVW) model to solve it. In practice we use the SIFT feature [24] and vocabulary tree [25] for description and searching. The feature points are extracted from unmasked area of the sub-image M_t^r as shown in Fig. 2.2 (b) and Fig. 2.2 (c).

For the distance term, we take advantages from the extrinsic parameters of the camera. Since the distance relation between F_i and M_t is the same as that between F_i and F_t, we define the distance $d(F_s, M_t)$ as follows

$$d(F_i, M_t) = \left\| r(F_i, F_t) \right\| + \left\| t(F_i, F_t) \right\|, \tag{2.2}$$

where $r(F_i, F_t)$ and $t(F_i, F_t)$ represent the rotation and translation distances respectively between the potential source frame F_i and the target one F_t estimated by the PnP algorithm [26]. $\|\cdot\|$ stands for the vectorized L2-norm. And we set $d(F_i, M_t) = +\infty$ for unsolvable conditions, which are discarded when doing normalization.

We use gradient based methods [27–29] for image quality assessment, which base their evaluation criteria on the proportion of relatively large gradients in all of them. A higher proportion indicates a larger amount of explicit edges and hence stands for higher image quality. This kind of methods is suitable for our case since we consider edges consistency as one of the factors in MRF (specifically described in section 2.3.3) and coherently use gradients in the energy function of it.

2.3.3. Register and Combine Selected Sources

So far the source frames have been retrieved, the first thing to do is warping them into the same viewpoint with the target frame. For RGB-D sequence, the classical

method to achieve pixel-wise correspondence is to use depth information for calculating the rigid transformation. However, a pixel with unknown depth value cannot be transformed and the information loss triggered by such invalid transformation would significantly depress the inpainting accuracy. Hence instead of it we propose to use local homography based warping method in this work.

Given the registered source frames, we need to decide from which one of them the mask pixel should get its value. A naive method is to project all the source frames into the target one and then blending them. However it is easy to trigger blurriness. In this thesis we propose to use MRF to make optimal choices among sources by considering it as a multi-label problem, similar to [16, 17, 19].

We also carry out post refinements for more natural and preciser results after solving the MRF. The entire framework of the proposed warping and combination approach is summarized in algorithm 1.

Image Warping for Registration

A single global homography is the simplest solution to describe pixel-wise correspondences between images. However it is restricted to subject to planar scene or pure rotation motion assumptions. Multi-homography methods [16, 30] have been employed in various kinds of research to deal with scenes that contain multiple planes, in which the images are divided into several planes and for each of them, an independent homography matrix is calculated. In this work we use the grid based local homography method for warping images taken with freely moving cameras. Whether a local homography can be calculated or not depends on the abundance of matched feature points, hence we use the affine SIFT [31] for points extraction.

Combination of Multiple Sources

Each source image (for convenience, without ambiguity we hereafter use source image to refer to the color image in the source frame) is considered as a label l . Our goal is to assign a label l_p for each pixel p in the mask area. We use the data cost term to represent the cost of assigning l_p to p and the smooth cost term to encourage avoiding explicit boundaries among distinct sources. The energy function we would like to minimize is

formulated in the form of

$$\begin{aligned} E(l) = &\sum_{p \in \varsigma} (\lambda_1 T_1(p, l_p) + \lambda_2 T_2(p, l_p)) \\ &+ \sum_{(p,q) \in \zeta} \lambda_3 W(p, q, l_p, l_q), \end{aligned} \tag{2.3}$$

where the sum of $T_1(p, l_p)$ and $T_2(p, l_p)$ indicate the data cost and ς represents all the pixels in the sub-image. $W(p, q, l_p, l_q)$ stands for the smooth cost; (p, q) is a pair of neighboring pixels of which we use the 4-neighbor system and ζ is the set of all such pairs in the sub-image region. λ_1, λ_2 and λ_3 are weight parameters. It is important to notice that the unmasked pixels are also taken into account as contributions of the total energy since it can serve as constraint for the masked ones via the smooth cost term.

$T_1(p, l_p)$ is set to infinity if $l_p(p)$ belongs to mask, representing waiting to be inpainted in the hole filling section later on. For other cases, we choose $T_1(p, l_p)$ to have the form

$$T_1(p, l_p) = \begin{cases} \left\| I_{l_p}(p) - I_t(p) \right\| & p \in M_t^r \\ \\ \left\| I_{l_p}(p) - I_m(p) \right\| & p \in M_t \end{cases}, \tag{2.4}$$

where $I_{l_p}(p)$ indicates the value of pixel p on the source image l_p; $I_t(p)$ is that on the target image and $I_m(p)$ represents the one on the median image, which is considered as a weighted combination of all source images with the rule

$$I_m(p) = \sum_{i \in N} w_i I_{l_i}(p), \tag{2.5}$$

where N is the set of source images; $I_{l_i}(p)$ is the value of pixel p on the source image l_i and w_i is weight. The weight is imported basing on the fact that the warping accuracy of all the source are not equal. Therefore we use the w_i for describing the relative accuracy and define it as

$$w_i = 1 - \frac{SSD(I_{l_i}(p), I_t(p))/n_i}{\sum_{j \in N} SSD(I_{l_j}(p), I_t(p))/n_j}, \tag{2.6}$$

where SSD is the sum of square difference between two pixel values; and n_i and n_j are the amounts of overlapped unmasked pixels between the source and target, which is used for normalization. Also we linearly normalize the RGB value to avoid influence caused by illumination changes.

As described earlier, the warping accuracy drops notably when distance is increased between viewpoints. Therefore we choose $T_2(p, l_p)$ to have the form

$$T_2(p, l_p) = \exp(\|t(I_{l_i}, I_t)\|) - 1, \tag{2.7}$$

where $t(I_{l_i}, I_t)$ is the translation distance between I_{l_i} and I_t; different from Eq. 2.2, we ignore rotation distance since, as mentioned above, in principle, pure rotation would not introduce errors in homography computation. Also this function can provide approximately linear and distinguishable error term.

For the smooth cost term, we use the same gradient cost function proposed in [16], where $W(p, q, l_p, l_q) = 0$ if $l_p = l_q$ and otherwise

$$\begin{aligned} W(p, q, l_p, l_q) = &\left\|\nabla I_{l_p}(p) - \nabla I_{l_q}(p)\right\| \\ &+ \left\|\nabla I_{l_p}(q) - \nabla I_{l_q}(q)\right\|, \end{aligned} \tag{2.8}$$

where $\nabla I_{l_p}(p)$ indicates the gradient of I_{l_p} at pixel p and so on. We use the Sobel operator to get it.

Finally, to minimize the energy function described in Eq. 2.3, we use the graph-cut algorithm proposed in [32–34].

Color Adjustment

The brightness of the same scene may vary among distinct source images because of the different illumination conditions, which will cause obvious contrasts among the boundaries of sources. Therefore, in practice the mask is first expanded to a few more pixels with a dilation filter before solving the MRF and then the Poisson image editing [35] technique is used for blending the varied lighting.

Depth Transformation

Within previous procedures we have established pixel-wise correspondence between the target frame and the source ones. Now the task is to transform the depth from sources to the target. In principle we achieve it via following the classical "inverse projection, SE(3) transformation and projection" steps. For those source pixels whose depth values are unknown, we set their correspondence on the target as blank to represent waiting to be inpainted in the hole filling section later on. Namely, for each source

Algorithm 1 Warp and combine multiple sources

for Each Mask M_t in the target frame **do**
  for Each selected source frame F_i **do**
    Warp F_i into the same viewpoint with the target frame F_t;
  end for
  for All the color images I of F_i **do**
    Calculate the average SSD between each I_i and I_t;
    Create the median image by weighted overlaying each I_i;
  end for
  Minimize the energy function of MRF by graph-cut;
  for Each source frame F_i **do**
    Compute the transformation matrix T_{ir} between F_i and F_t;
    for Each source frame $F_j(j \neq i)$ **do**
      Compute the transformation matrix T_{ij} between F_i and F_j;
    end for
  end for
  Pose graph initialization;
  for Each vertexes V_i in the graph **do**
    Assign T_{ir} to V_i;
    for Each vertexes $V_j(j \neq i)$ in the graph **do**
      if T_{ij} **then**
        Assign T_{ij} to the edge E_{ij} that connects V_i and V_j;
      else
        Disconnect the two vertexes;
      end if
    end for
  end for
  Do graph optimization;
end for
Do Poisson image editing;

frame a transformation matrix $T \in \mathbb{R}^{4\times4}$ should be calculated. However, the estimation of pairwise transformation is not such accurate. Also considering that information

for inpainting is collected from multiple sources, it is easy to trigger inconsistencies on the inpainted depth image.

As a solution, we implement a global optimization process to get the preciser transformation matrices by modeling it into a graph optimization problem. Specifically, we take the pose of the target frame as the origin and its camera coordinate as the world one. Then the vertexes in the graph can be set to the transformation matrices T_{sr} from each source frame to the target. Coherently, the edges connecting pairwise vertexes are conditionally either set to the transformation matrices T_{ij} between the two source frames if the T_{ij} can be computed or left as blank to represent disconnections. We employ the g2o framework [36] for implementation.

2.3.4. Fill in the Still Existing Holes

The above step usually returns a partly inpainted frame since some pixels may not find their correspondences from the source frames and therefore left as blank. And hereby we fill in them with classical inpainting algorithms. Specifically, for the color image, a multi-view exemplar based inpainting method is proposed. And then, the inpainted color image will serve as guidance for completing the corresponding depth one.

Color Image Inpainting

Exemplar based inpainting methods basically synthesize values for the mask from the source by minimizing an energy function describing the similarity between them. In this work, we base our multi-view exemplar inpainting approach on the method proposed in [7, 37] and define the energy function in the form of

$$E = \sum_{p_i \in \phi, p_j \in \Phi} SSD(p_i, p_j), \tag{2.9}$$

where, the same as the general definitions in image inpainting work, ϕ is the boundary area of the mask; Φ is the source area from where the information is gotten; p_i and p_j respectively indicate pixels in ϕ and Φ; and SSD represents the patch similarity in the form of

$$\begin{aligned} SSD(p_i, p_j) = &\sum_{s \in \omega} \left\| I(p_{i+s}) - \alpha_{p_i p_j} I(p_{j+s}) \right\| \\ &+ \left\| \nabla I(p_{i+s}) - \nabla I(p_{j+s}) \right\|, \end{aligned} \tag{2.10}$$

where ω is the patch size; s is a shift vector used for traverse all the pixels within ω; $I(p_{i+s})$ and $I(p_{j+s})$ indicate values of pixel p_{i+s} and p_{j+s}; $\nabla I(p_{i+s})$ and $\nabla I(p_{j+s})$ are gradients used as constraints for preserving texture consistency. $\alpha_{p_i p_j}$ is a parameter used for dealing with brightness changes [7], which is defined as

$$\alpha_{p_i p_j} = \sqrt{\frac{\sum_{s\in\omega} I^2(p_{i+s})}{\sum_{s\in\omega} I^2(p_{j+s})}}, \tag{2.11}$$

For multi-view inpainting, we expand the definition of Φ as the unmasked area in the single image to that in both the target image and the source ones used for combining sources. It is worth to mention that the source images used in our approaches are the warped ones as described in section 2.3.3 to ensure texture consistency between the inpainted area and the original unmasked one. It is reasonable to do so since the warped images are in the same viewpoint with the target one, which means that we are seeking information from a spatially consistent region rather than several independent counterparts.

Guided Depth Image Inpainting

Depth images are usually low textured and propagation based methods are hence applicable. Similar to [21], we extract edges from the color image as guidance and coherently divide the masked pixels of the depth image into either smooth or edge classes. Specifically, a pixels would be classified into edge class if its color correspondence is edge and smooth otherwise. Propagations are processed separately on each class.

A given pixel can reckon its value from propagation only if its neighbors can provide enough information. In this work we use the 8-neighbor system and define respective rules for smooth and edge classes to evaluate the conditions of their neighbors. For the smooth class, a creditable neighbor should has no less than 4 available pixels that also belong to the smooth; and for the edge one, a neighbor region is reliable if the distribution of available edge pixels within it satisfies one of the conditions shown in Fig. 2.3.

The mask area is iteratively inpainted until convergence. In each iteration, a pixel is either set to the propagated value provided satisfying the rules above or skipped otherwise. An edge pixel would be consider as mis-masked and move to the smooth class if it still cannot be assigned value after several loops. For propagation we use the

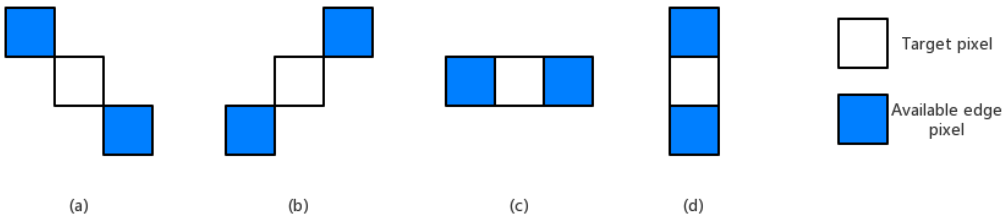


Figure 2.3.: Creditable distribution samples for edge class

Laplace equation in its discrete form

$$I(p, t+1) = \frac{\sum_{p' \in \mu_8(p)} \kappa(p') I(p', t)}{\sum_{p' \in \mu_8(p)} \kappa(p')}, \tag{2.12}$$

where $I(p, t+1)$ is the value of pixel p at time $t+1$; $\mu_8(p')$ indicates the 8-neighbor pixels of p; and $\kappa(p')$ is an indicator function which is set to 1 if the pixel is in the same class with p and 0 otherwise.

2.4. Experiments

We test our approach on different datasets from the TUM RGB-D benchmark [38] (freiburg3-walking-rpy, freiburg3-walking-xyz and freiburg3-walking-halfsphere), in which two persons randomly walk across the scenes and the cameras are also on moving. The "human" class is defined as undesired. The three datasets are totally different because of the distinct camera motions although they present similar scenes. These datasets are pretty challenging for being highly textured. For each dataset we use the first 800 frames and one target frame within it as input. The results are shown in Fig. 2.4.

2.4.1. Compare with Other Approaches

Since similar work designed for RGB-D inpainting barely exists, we could only compare our algorithm with other color image inpainting techniques. Specifically, we first compare our method with the classical single image inpainting ones [6, 12] and the video completion one [14]; and the results are presented in Fig. 2.5. Note that in order to run the video completion algorithm, we first down-sample the images and use the neighboring 100 frames of the target as input for its extremely high time cost.

Besides the comparisons above, we also compare our method with another two state-of-the-art multi-view inpainting approaches, as proposed in [19, 20]. The results are shown in Fig. 2.6. As can be seen, our method can effective maintain the texture consistency compared with [20] and will not suffer from information lost as [19] does, which is triggered by camera undistortion.

2.4.2. An Application Example

Hereby we present an application example of our work, as shown in Fig. 2.7. For a set of selected frames, still the fine-tuned PSPNet [10] is used to detect and draw masks on the "human" class. Also we expand the masks to a bit more pixels by dilation in order to cope with unmasked edges [39]. We use MeshLab [40] to present the final results as shown in Fig. 2.7. As shown, the humans are effectively removed from the constructed point clouds.

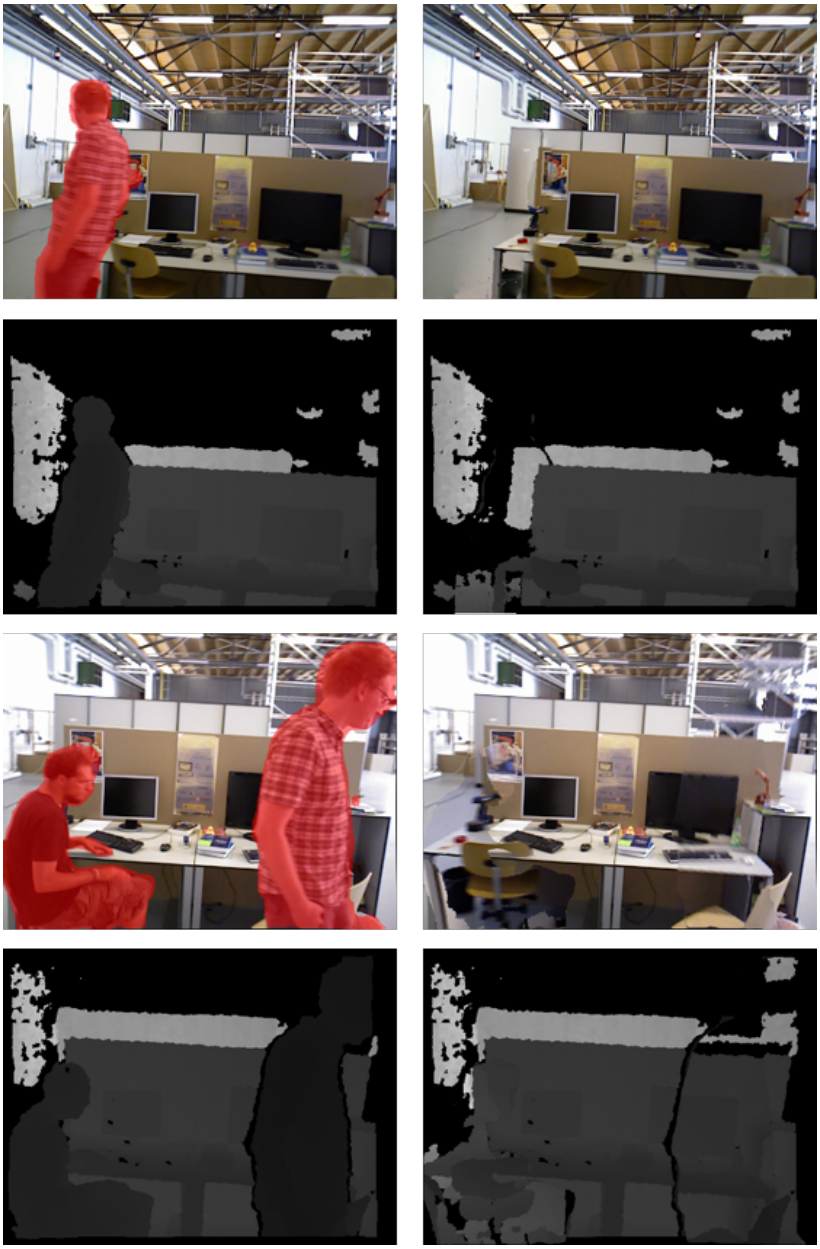


Figure 2.4.: Test results of our proposal on different datasets. The first row indicates the original frames and the second one is the inpainted results with our method. The first two columns come from the freiburg3-walking-xyz dataset; middle two columns: freiburg3-walking-rpy; last two columns: freiburg3-walking-halfsphere.

Figure 2.5.: Compare with other methods. Each column shows one image inpainted by different methods. The first and second rows show the origin images and inpainting results with our method. The third and fourth rows present the results of single image inpainting methods respectively proposed by Barnes *et al.* [6] and Lee *et al.* [12]. The last row is the results of the video completion algorithm proposed by Newson *et al.* [14].

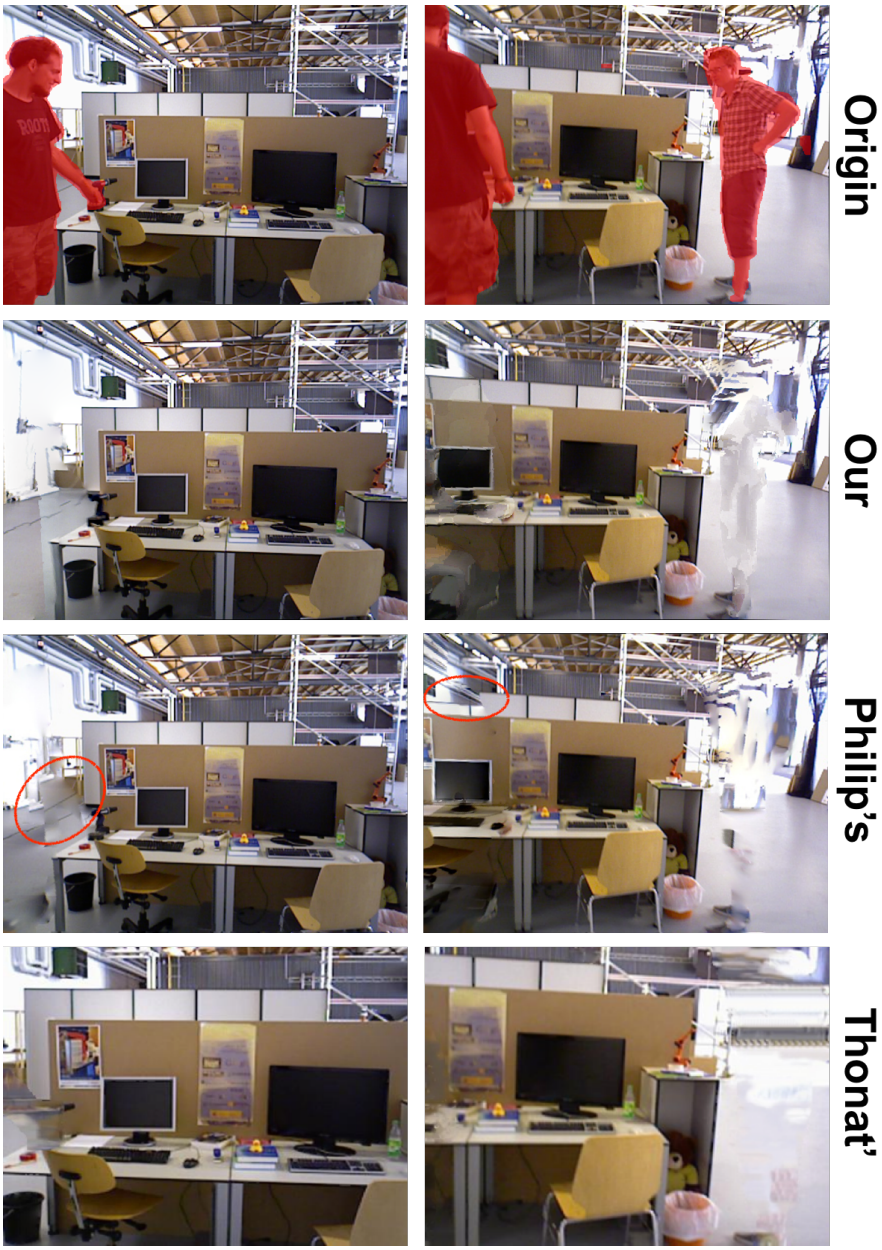
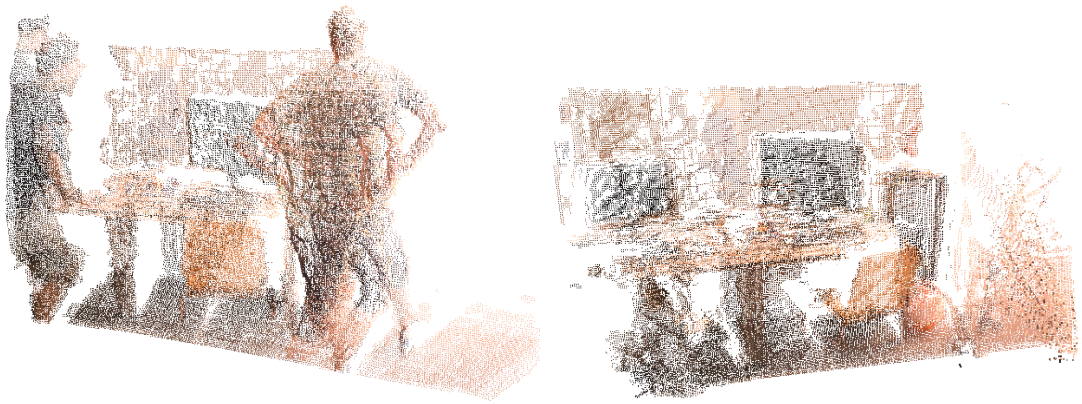


Figure 2.6.: Compare with other multi-view inpainting approaches [19, 20]. Our method can effectively preserve texture consistencies (*e.g.* the circled places) and won't suffer from scene missing.



(a) (b)

Figure 2.7.: Application example of our proposed algorithm. Above: point clouds projected from original images. Below: the counterpart after inpainting with our method.

2.5. Summary

In this chapter we have introduced a multi-view inapinting framework. Different from the past ones, our proposal is designed for inpainting RGB-D frames. There are 3 steps in the proposed framework: first, we select the sources for the inpainting task; second, similar to other approaches, we formulate the multi-view inpainting problem into an MRF manner to gather information; and at last, exemplar-based algorithm for RGB images and propagation-based one for the depth in post-processing are respectively used. And experiments shows that our proposal can out-perform the other candidates in terms of maintaining the texture consistency and naturality.

3. Pose Estimation

3.1. Introduction of Pose Estimation

Pose estimation (*aka. SE(3)* estimation) problem is directly related to the precision of 3D reconstruction tasks and similar application scenarios such as multi-camera system calibration [41], gaze tracking [42] and medical image registration [43]. And in this chapter, an efficient online pose estimation algorithm is introduced for enhancing accuracy.

Within most of the classical methods the uncertainty of the pose parameters is modeled by Gaussian distribution. However, since it is majorly used in vector spaces and states that are unimodally distributed, natural unfitness would be triggered when it is applied to rotation elements considering their unique properties that:

- The periodicity of the underlying manifold. Namely, a rotation θ and its correspondence $\theta + 2\pi$ are actually the same.
- The antipodal symmetry of the quaternions that $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. Namely, the quaternions should be modeled by a bimodal distribution instead of a unimodal one.

Although much work has tried to asymptotically fix a Gaussian distribution on the quaternions, it is still an approximation. Therefore, in order to exactly exploit the structure of the rotation parameter space, Bingham distribution [44] is proposed for usage, which is a specially designed bimodal distribution for modeling the uncertainty on the *n*-sphere manifold and hence more suitable to capture the properties of the quaternions as mentioned above.

Srivatsan *et al.* first propose a dual-quaternion based linear model to separately estimate *SE*(3) elements with Kalman filter [45]. Then the model is combined with Bingham distribution in [46,47] to achieve higher accuracy. However, the new method

still suffers from a common shortcoming that, similar to the traditional filter-based approaches, the noise covariance requires manual tuning, which is especially important to the Bingham filter due to its high sensitivity.

In this chapter we propose to solve this problem by taking advantage of the adaptive filtering theory and develop an adaptive Bingham distribution based filter, which has the following merits:

- *Self-ruling:* the proposed method can adjust the noise covariance autonomously and consequently free users from manual tuning.
- *Adaptive:* the adaptive Bingham filter can converge with varied noise conditions.
- *Accurate:* the stated approach can outperform the other candidates in precision.

3.2. Related Work

In this section we will go over the related pose estimation algorithms, which can be classified into batch-based approaches and filter-based counterparts. Also the adaptive filtering theories are reviewed.

3.2.1. Batch-based Pose Estimation

Batch-based methods estimate the $SE(3)$ elements via formulating it as a least square problem, whose objective function can be solved in closed form via computing either the singular value decomposition (SVD) or the eigen-system when the correspondences between the two point sets are already known [48]. Arun *et al.* proposed the first method by first calculate the rotation matrix and then the translation vector [49]. However, this method uses a standard matrix for rotation representation and its orthogonal property is ignored. On the other hand, Horn *et al.* propose a superior version by constraining the rotation matrix to be orthogonal in [50]. However, this method does not hold if the two point sets are planar. Therefore, in their following work [51], the orthogonal matrix representation is replaced by the unit quaternion, which is applicable to all cases. While all the three methods above compute rotation and translation separately, Walker *et al.* propose to represent them jointly with a dual-quaternion system [52], which is also the basis of the filter model used in this work. Years later, Myronenko and Song propose that degenerated problem encountered when the rotation is represented in matrix form [50, 53] can be formulated into a trace-maximizing manner [54].

If the point-wise correspondences are unknown, the iterative closest point (ICP) algorithm and its variants are widely used [55, 56]. Such methods take advantages from the expectation–maximization (EM) algorithm where the construction of point-wise correspondences and pose estimation are processed recursively. Another class of solutions is based on the Normal distribution transformation [57–59], where each point set is modeled as a Gaussian mixture model and a divergence is used to formulate the objective function. A detailed summary can be found in [60].

Although the batch-based methods can give closed-form solutions, the iterative way is more preferred in some applications. Especially in the cases where the data is retrieved incrementally [41] or the parameters for estimation vary as time goes by [42].

3.2.2. Filter-based Pose Estimation

The filter-based approaches are suitable alternates to solve the optimization problems in an iterative, model-based way. The classical methods employ the Kalman filter and its derivatives (EKF, UKF, etc.) for registration [61, 62]. Such methods however have two main flaws. First, their models are not truly linear but approximated by the Taylor expansion or unscented transformation, which may lead to diverge provided high initialization error. To deal with it, Srivatsan *et al.* come up with a novel true linear model by first estimating the rotation and then the translation with multiple measurements [45].

Second, as described previously, the Gaussian distribution cannot perfectly model the uncertainty of the $SO(3)$ elements. Particle filter [63] is an alternative solution but would be extremely time-consuming. Thus adopting the Bingham distribution is a better choice considering its merits as described above. However, it has not attracted enough attention as it deserves and most of the existing work can only deal with pure rotation cases. For example, Glover and Kaelbling present the prototype to estimate 3D rotation with the Bingham distribution [64]. Gilitschenski *et al.* combine unscented transformation with Bingham filter to cope with non-linear models. It was not until recently when Srivatsan *et al.* merge the Bingham distribution into their previous work [45] that a theoretically superior filter-based pose estimation algorithm makes its debut. This approach is however still at its childhood and one noticeable flaw is its sensitiveness to the manually set noise covariance, which is solve in this work via using the adaptive filtering technique.

3.2.3. Adaptive Filtering

Many adaptive filtering approaches have been developed to autonomously adjust the noise covariances in the process and measurement models. A strategical instruction of the classical adaptive filtering theories can be found in [65], where the algorithms are classified into four categories: Bayesian, correlation, covariance matching and maximum likelihood. When applied to pose estimation problem, Janabi and Marey combine the maximum likelihood method with the iterative Kalman filter for visual servoing [66]. Aghili and Su integrate maximum likelihood based adaptive Kalman filter with the ICP algorithm to merge Laser and IMU data for navigation [67]. Zhou *et*

al. develop a covariance matching based adaptive unscented Kalman filter for target tracking [68]. Although most of the existing adaptive filtering algorithms are originally developed for Kalman filter and its variants, the basic theories behind them can still be applied to the Bingham filter.

3.3. Proposed Method

3.3.1. Preliminaries of the Bingham Distribution

The Bingham distribution on the circle or (hyper-)sphere naturally arises when the d-dimensional Gaussian distribution is conditioned to the d-dimensional unity sphere S^{d-1} with zero mean value [69]. The relationship between a Bingham distribution and the corresponding Gaussian distribution can be found in Fig. 3.1.

Probability Density Function

The probability density function (PDF) of Bingham distribution is given by

$$f(\mathbf{x}) = \frac{1}{N} \cdot \exp(\mathbf{x^T M Z M^T x}), \tag{3.1}$$

where $\mathbf{M} \in \mathbb{R}^{4\times4}$ is an orthogonal matrix; $\mathbf{Z} = diag(0, z_1, .., z_{d-1}) \in \mathbb{R}^{4\times4}$ with $0 \geq z_1 \geq ... \geq z_{d-1}$ is named as the concentration matrix and N is the normalization constant.

$$N = \int_{S^{d-1}} \exp(\mathbf{x^T Z x}) d\mathbf{x}. \tag{3.2}$$

Computation of N is shown in Eq. 3.2, which is pretty challenging and hence is often achieved by some forms of approximation or precomputed lookup tables (details can be found in [69, 70]). In this work the second approach is employed.

Also it is worth to point out that adding a multiple of the identity matrix $\mathbf{I}_d$ to $\mathbf{Z}$ will not change the distribution [44], which clarifies that why $\mathbf{Z}$ can be always set to such a non-positive form. This property also provides foundations to derive the mode and covariance of the distribution.

Mode of the Distribution

The mode of a Bingham distribution is the normalized eigenvector corresponding to the largest eigenvalue. Since the columns in $\mathbf{M}$ are swapped to ensure the concentration matrix $\mathbf{Z}$ is non-positive definite and descending, the mode is the first column in $\mathbf{M}$ with 0 as the corresponding eigenvalue.

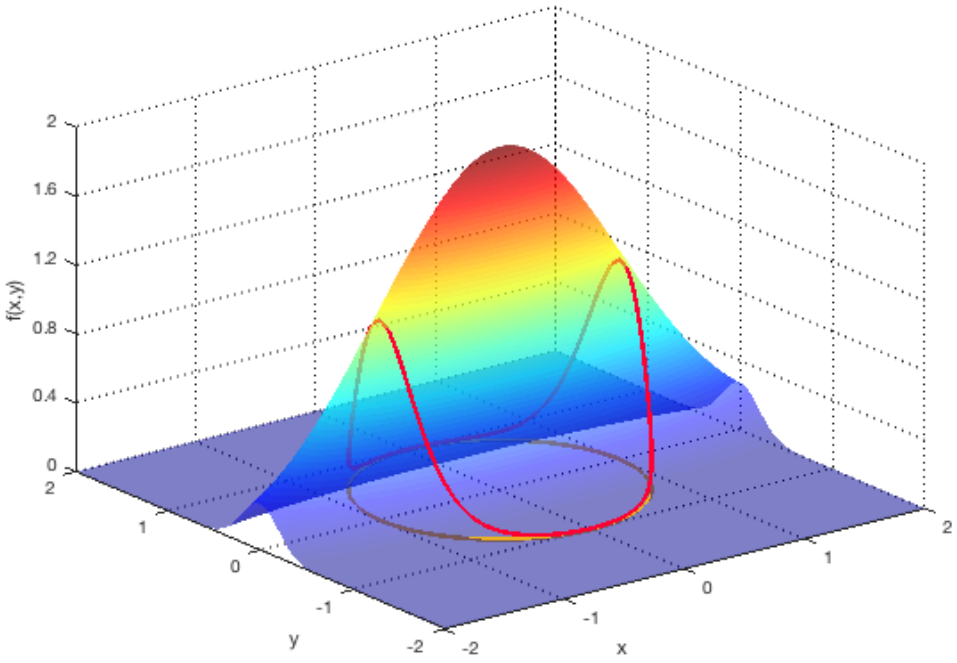


Figure 3.1.: A 2D Bingham distribution with $\mathbf{M} = \mathbf{I}_2$ and $\mathbf{Z} = diag(0, -2)$, which can be achieved by conditioning the Gaussian distribution $N(\mathbf{0}, diag(1, \frac{1}{5}))$ to the unit circle. Better to see in color.

Multiplication of Two PDFs

The product of two given Bingham distribution is a prerequisite of the Bingham density-related Bayesian inference. Similar to the Gaussian counterpart, it can be written into a rescaled form. Consider two Bingham PDF:

$$f_i(\mathbf{x}) = \frac{1}{N_i} \cdot \exp(\mathbf{x^T M_i Z_i M_i^T x}), \quad i = 1, 2. \tag{3.3}$$

The product can be written as

$$\begin{aligned} f_1(\mathbf{x}) \cdot f_2(\mathbf{x}) &= \frac{1}{N_1 N_2} \cdot \exp(\mathbf{x^T}(\underbrace{\mathbf{M_1 Z_1 M_1^T} + \mathbf{M_2 Z_2 M_2^T}}_{\mathbf{S}})\mathbf{x}) \\ &\propto \frac{1}{N} \cdot \exp(\mathbf{x^T M Z M^T x}), \end{aligned} \tag{3.4}$$

where N is a new normalization constant. $\mathbf{M}$ is orthogonal and refers to the eigenvectors of $\mathbf{S}$; and in order to keep the non-positive definite, $\mathbf{Z}$ is represented as $\mathbf{Z} = \mathbf{D} - \mathbf{D}_{11}\mathbf{I}_d$, where $\mathbf{D}$ is the descending sorted eigenvalues of $\mathbf{S}$ and hence $\mathbf{D}_{11}$ refers to the maximum entry.

Covariance

The covariance of a Bingham distribution can still be calculated in $\mathbb{R}^d$ regardless of the unit sphere-defined property in the form of

$$\begin{aligned} cov(\mathbf{x}, \mathbf{y}) &= E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{y} - E[\mathbf{y}])] \\ &= \mathbf{M}(diag(\frac{\frac{\partial N}{\partial z_1}}{N}, ..., \frac{\frac{\partial N}{\partial z_d}}{N}))\mathbf{M}^{\mathbf{T}}. \end{aligned} \tag{3.5}$$

Since the Bingham distribution can be equivalently characterized as a conditioned zero-mean Gaussian distribution where the mean is invariant, it can be seen as possessing the same covariance as the Gaussian one $N(0, -(2\mathbf{M}(\mathbf{Z} + c\mathbf{I})\mathbf{M}^{\mathbf{T}})^{-1})$, where c is an arbitrarily selected value in order to make sure that $(\mathbf{Z} + c\mathbf{I})$ is negative definite [44]. Same as [46], we chose $c = min(z_i)$ in this work without loss of generality.

3.3.2. Model of the Linear Filter

The model of the linear filter is originally proposed in [45, 46], which can estimate the rotation between two point clouds when a pair of corresponding points is taken as input. Hereby we briefly review the model to support further derivation. Also as quaternion is used to represent rotation in the model, we give a brief introduction of it in the appendix A.1.

System Model

The static pose estimation problem is typical in 3D registration and similar robotic vision problem. In this case multiple corresponding points can be achieved simultaneously and hence the process model can be treated as time-invariant. The transformation between each pair of points can be written as

$$\mathbf{b_i} = \mathbf{q} \odot \mathbf{a_i} \odot \mathbf{q}^{-\mathbf{1}} + \mathbf{t} \quad (i = 1, ..., n), \tag{3.6}$$

where $\mathbf{a_i}, \mathbf{b_i} \in \mathbb{R}^3$ are within the two corresponding point sets; $\mathbf{q}$ is the quaternion representing the rotation and $\mathbf{t}$ is the translation between them.

Within every update step two pairs of points are retrieved and mutually added and subtracted in the form of

$$\mathbf{b_2} - \mathbf{b_1} = \mathbf{q} \odot (\mathbf{a_2} - \mathbf{a_1}) \odot \mathbf{q^{-1}}, \tag{3.7}$$

and

$$\begin{aligned} \mathbf{b_2} + \mathbf{b_1} &= \mathbf{q} \odot (\mathbf{a_2} + \mathbf{a_1}) \odot \mathbf{q^{-1}} + 2\mathbf{t} \\ \Rightarrow \mathbf{t} &= \frac{(\mathbf{b_2} + \mathbf{b_1}) - \mathbf{q} \odot (\mathbf{a_2} + \mathbf{a_1}) \odot \mathbf{q^{-1}}}{2}. \end{aligned} \tag{3.8}$$

A brief introduction of quaternion is given in the appendix. Applying quaternion multiplication as shown in Eq. A.2, Eq. 3.7 can be rewritten as

$$\mathbf{z} = \mathbf{H} \cdot \mathbf{q} = \mathbf{0}, \tag{3.9}$$

where $\mathbf{z}$ is the pseudo-measurement and equals to $\mathbf{0}$ ideally. And $\mathbf{H}$ is written as

$$\mathbf{H} = \begin{bmatrix} 0 & (\mathbf{b} - \mathbf{a})^T \\ \mathbf{a} - \mathbf{b} & (\mathbf{a} + \mathbf{b})^\times \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{3.10}$$

in which $\mathbf{a} = \mathbf{a_2} - \mathbf{a_1}$ and $\mathbf{b} = \mathbf{b_2} - \mathbf{b_1}$.

It is also worth to mention that in real applications the definite should be slightly changed into $\mathbf{a} = \mathbf{a_2} - \mathbf{a_c}$ and $\mathbf{b} = \mathbf{b_2} - \mathbf{b_c}$ where $\mathbf{a_c}$ and $\mathbf{b_c}$ respectively denote the centroids of $\mathbf{a_i}$ and $\mathbf{b_i}$ [50]. Such a modification is used to ensure that the origins of their original bases coincide.

Update via Bayes' Theorem

The Bingham filter is iterated to obtain the estimated roation $\mathbf{q}$. And its a priori can be represented by Bingham distribution as

$$P(\mathbf{q}_k) = \frac{1}{N} \exp(\mathbf{q}_{k-1}^T \underbrace{\mathbf{M}_{k-1}\mathbf{Z}_{k-1}\mathbf{M}_{k-1}^T}_{\mathbf{D}_1} \mathbf{q}_{k-1}), \tag{3.11}$$

And once two pairs of measurement points are obtained, by assuming the measurement uncertainty is with Gaussian distribution, the measurement probability can be written

as

$$
\begin{aligned}
P(\mathbf{z}_k|\mathbf{q}_k) &= \frac{1}{N}\exp\left\{-\tfrac{1}{2}\left[\mathbf{z}_k - h(\mathbf{q}_k)\right]^T \mathbf{R}_k^{-1}\left[\mathbf{z}_k - h(\mathbf{q}_k)\right]\right\} \\
&= \frac{1}{N}\exp\left[-\tfrac{1}{2}(\mathbf{H}\mathbf{q}_k)^T {\mathbf{R_k}}^{-1}(\mathbf{H}\mathbf{q}_k)\right] \\
&= \frac{1}{N}\exp(\mathbf{q}_k^T(\underbrace{-\frac{1}{2}\mathbf{H}^T\mathbf{R}_k^{-1}\mathbf{H}}_{\mathbf{D}_2})\mathbf{q}_k),
\end{aligned}
\tag{3.12}
$$

where $\mathbf{z}_k$ and $h(\mathbf{q})$ are the measurement value and measurement model respectively as demonstrated in Eq. 3.9; and $\mathbf{R_k}$ is the measurement uncertainty.

Then the estimation can be updated via Bayesian inference and the a posteriori is

$$
\begin{aligned}
P(\mathbf{q}_k|\mathbf{z}_k) &= P(\mathbf{q}_k)P(\mathbf{z}_k|\mathbf{q}_k) \\
&\propto \frac{1}{N}\exp(\mathbf{q}_k^T(\mathbf{D}_1 + \mathbf{D}_2)\mathbf{q}_k) \\
&= \frac{1}{N}\exp(\mathbf{q}_k^T\mathbf{M}_k\mathbf{Z}_k\mathbf{M}_k^T\mathbf{q}_k).
\end{aligned}
\tag{3.13}
$$

Now the filter can be updated by passing on $P(\mathbf{q}_k|\mathbf{z}_k)$ to $P(\mathbf{q}_{k+1})$ as described in Eq. 3.11.

3.3.3. Adaptive Filtering with Covariance Matching

In this work we take advantages from the adaptive filtering theories which is originally developed for the Kalman filter and its variants. Following the instruction given by Mehra [65], we employ the covariance matching method, which is more suitable in our case than the others provided known process uncertainty and easy-to-calculate actual covariance.

Covariance Matching

The basis of covariance matching technique is to make the covariance of the actual residual (also denoted as *innovation* in some literature [67, 71]) consistent with its theoretical counterpart. The residual can be computed by

$$
\begin{aligned}
\Sigma^{\varepsilon} &= E(\varepsilon^2) - E(\varepsilon)^2 \\
&\approx \frac{1}{m}\sum_{i=1}^{m}\varepsilon_i\varepsilon_i^T
\end{aligned}
\tag{3.14}
$$

where ε is the actual residual gotten from $\varepsilon = z - h(\mathbf{q})$ and m is experientially chosen for giving statistical smoothing.

Measurement Uncertainty

One noticeable point is that $\mathbf{R}_k$ in Eq. 3.12 cannot be simply treated as the covariance of measurement noise. This is because the measurement model as denoted in Eq. 3.9 is a pseudo-measurement and hence the important stochastic theory [45, 72] as described in *Proposition 1* should be used.

Proposition 1. Denote $\mathbf{m} \in \mathbb{R}^m$; $\mathbf{n} \in \mathbb{R}^n$; $\mathbf{x} \in \mathbb{R}^l$ and the linear matrix mapping $\mathbf{G}(\cdot) : \mathbb{R}^l \to \mathbb{R}^{m \times n}$. Define $\mathbf{m} = \mathbf{G}(\mathbf{x}) \cdot \mathbf{n}$; then the uncertainty of $\mathbf{m}$ can be written as

$$\Sigma^{\mathbf{m}} = \mathbf{G}(x)\Sigma^{\mathbf{n}}\mathbf{G}(x)^T + \mathbf{N}(\Sigma^{\mathbf{n}} \otimes \Sigma^{\mathbf{x}})\mathbf{N}^T; \tag{3.15}$$

where $\Sigma^{[\cdot]}$ is the covariance of $\cdot$; $\otimes$ is the Kronecker product and $\mathbf{N}$ is defined as

$$\mathbf{N} := [\mathbf{N}_1, ..., \mathbf{N}_n] \in \mathbb{R}^{m \times nl}, \tag{3.16}$$

where $\mathbf{N}_i$ is defined according to

$$\mathbf{N}_i\mathbf{x} = \mathbf{G}(\mathbf{x})\mathbf{e}_i, \tag{3.17}$$

where $\mathbf{e}_i$ is the unit vector with 1 at position i and 0 otherwise.

Rewrite Eq. 3.10 into the form of $\mathbf{z} = \mathbf{H}\mathbf{q} = \mathbf{G}(\mathbf{q})\mathbf{r}$ in which

$$\begin{aligned} \mathbf{r} &= [\mathbf{a}_1^T, \mathbf{a}_2^T, \mathbf{b}_1^T, \mathbf{b}_2^T]^T; \\ \mathbf{G}(\mathbf{q}) &= [\mathbf{G}_1, -\mathbf{G}_1, \mathbf{G}_2, -\mathbf{G}_2], \end{aligned} \tag{3.18}$$

where $\mathbf{G}_1$ and $\mathbf{G}_2$ are denoted as

$$\mathbf{G}_1 = \begin{bmatrix} -{\mathbf{q}_\mathbf{v}}^T \\ -{\mathbf{q}_\mathbf{v}}^\times + q_0\mathbf{I}_3 \end{bmatrix}, \quad \mathbf{G}_2 = \begin{bmatrix} {\mathbf{q}_\mathbf{v}}^T \\ -{\mathbf{q}_\mathbf{v}}^\times - q_0\mathbf{I}_3 \end{bmatrix}. \tag{3.19}$$

By applying *Proposition 1* to Eq. 3.18, Eq. 3.15 can consequently be parameterized as that $\Sigma^{\mathbf{r}}$ is the original measurement noise and $\Sigma^{\mathbf{q}}$ is the processing uncertainty obtained from the Bingham distribution of $\mathbf{q}$. Therefore, we now have derived the mapping from the original measurement noise to the modeled one.

Adaptive Adjustment

Applying the covariance matching technique to the Bingham distribution based filter consequences the residual covariance in the form of

$$\Sigma^{\varepsilon} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{0} - \mathbf{H}_i\mathbf{q}_i)(\mathbf{0} - \mathbf{H}_i\mathbf{q}_i)^T, \tag{3.20}$$

where we determine the sample size m by using the Cochran sampling theory [73] by setting the margin of error to 5% and the confidence level to 80%. Also in practice we set $m \to m-1$ to get the unbiased estimation.

Equalize the theoretical covariance of $P(\mathbf{z}|\mathbf{q})$ and Σ^{ε} we can derive that

$$-[2\mathbf{M}(\mathbf{Z} + c\mathbf{I})\mathbf{M}^T]^{-1} = \Sigma^{\varepsilon}, \tag{3.21}$$

which can be rewritten into

$$\mathbf{M}\mathbf{Z}\mathbf{M}^T = -(2\Sigma^{\varepsilon})^{-1} - c\mathbf{I}, \tag{3.22}$$

where c should ensure that $-(2\Sigma^{\varepsilon})^{-1} - c\mathbf{I}$ is non-positive definite to meet the property of Bingham distribution and hence can be set as the maximum entry in $-(2\Sigma^{\varepsilon})^{-1}$. Then by importing the $\mathbf{D}_2$ in Eq. 3.12 to Eq. 3.22, $\mathbf{R}$ can be represented as

$$\mathbf{R} = -\frac{1}{2}\mathbf{H}[-(2\Sigma^{\varepsilon})^{-1} - c\mathbf{I}]^{-1}\mathbf{H}^T. \tag{3.23}$$

Now we can start to calculate the raw noise covariance. And depending on how the data is capture by the sensor system, the covariance representation may also vary. Hereby two common cases are explored.

Single coefficient

Within general cases the original measurement noise $\Sigma^{\mathbf{r}}$ can be written in the form of $\rho\mathbf{I}$, where ρ is the noise coefficient. Then by taking Eq. 3.15 into account, Eq. 3.23 can be expanded into the form of

$$\begin{aligned} &\mathbf{G}(\mathbf{q})\rho\mathbf{I}_{12}\mathbf{G}(\mathbf{q})^T + \mathbf{N}(\rho\mathbf{I}_{12} \otimes \mathbf{\Sigma}^{\mathbf{q}})\mathbf{N}^T = \mathbf{R}, \\ \Rightarrow \\ &\rho\left\{\mathbf{G}(\mathbf{q})\mathbf{G}(\mathbf{q})^T + \mathbf{N}(\mathbf{I}_{12} \otimes \mathbf{\Sigma}^{\mathbf{q}})\mathbf{N}^T\right\} = \mathbf{R}. \end{aligned} \tag{3.24}$$

By importing the details of $\mathbf{G}(\mathbf{q})$ from Eq. 3.18 and computing the Kronecker product, Eq. 3.24 can be further rewritten as

$$\rho \cdot (2\mathbf{A} + \mathbf{B}) = \mathbf{R}, \tag{3.25}$$

where $\mathbf{A}$ and $\mathbf{B}$ are denoted as

$$\begin{aligned} \mathbf{A} &= [\mathbf{G}_1\mathbf{G}_1^T + \mathbf{G}_2\mathbf{G}_2^T]; \\ \mathbf{B} &= \mathbf{N} \begin{bmatrix} \Sigma^{\mathbf{q}} & & \\ & \ddots & \\ & & \Sigma^{\mathbf{q}} \end{bmatrix} \mathbf{N}^T, \end{aligned} \tag{3.26}$$

where $\mathbf{N}$, $\Sigma^{\mathbf{q}}$, $\mathbf{G}_1$ and $\mathbf{G}_2$ are as defined in Eq. 3.16 and Eq. 3.19.

Double coefficients

In some applications such as filter-based ICP algorithms for point clouds registration or the filter-based multi-camera system calibration, the two corresponding point sets may be coupled with different noise amplitudes and hence the noise coefficients are also varied. In such a condition, the measurement noise $\Sigma^{\mathbf{r}}$ can be written in the form of $diag(\rho_1\mathbf{I}_6, \rho_2\mathbf{I}_6)$, where ρ_1 and ρ_2 are the noise coefficients for point sets $\mathbf{a}$ and $\mathbf{b}$ respectively. And similar to the derivation of Eq. 3.24, in double coefficients case Eq. 3.23 can be rewritten into

$$\begin{bmatrix} \rho_1\mathbf{I}_4 & \rho_2\mathbf{I}_4 \end{bmatrix} \cdot (2\mathbf{A}' + \mathbf{B}') = \mathbf{R}, \tag{3.27}$$

where $\mathbf{A}'$ and $\mathbf{B}'$ are in the forms of

$$\begin{aligned} \mathbf{A}' &= \begin{bmatrix} \mathbf{G}_1\mathbf{G}_1^T \\ \mathbf{G}_2\mathbf{G}_2^T \end{bmatrix}; \\ \mathbf{B}' &= \begin{bmatrix} \mathbf{N}_1\Sigma^{\mathbf{q}}\mathbf{N}_1^T + \cdots + \mathbf{N}_6\Sigma^{\mathbf{q}}\mathbf{N}_6^T \\ \mathbf{N}_7\Sigma^{\mathbf{q}}\mathbf{N}_7^T + \cdots + \mathbf{N}_{12}\Sigma^{\mathbf{q}}\mathbf{N}_{12}^T \end{bmatrix}, \end{aligned} \tag{3.28}$$

where $\mathbf{N}_i$ $(i = 1, ..., 12)$ are as shown in Eq. 3.17.

By solving Eq. 3.25 and Eq. 3.27 in each iteration, the covariance of measurement noise is adaptively self-adjusted and updated. When the computed ρ is not a real number, we set it to 0 for skipping this iteration.

In the above the single coefficient and double ones cases are used for demonstration. However, it is also noticeable that if more prior knowledge of the measurement noise is given, it can be set to a more general form that $\Sigma^{\mathbf{r}} = diag(\rho_1 \dots \rho_{12})$ where ρ_i is the respective noise coefficient coupled on the 3 dimensions of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1, \mathbf{b}_2$. Probable applications can be found when the data set is coupled with anistropic noise.

3.4. Experiments

The proposed method is evaluated on both the numerical simulation and the real-world data. Both the single coefficient model (abbr. ABF) and double coefficients one (abbr. ABFD) are tested. Without loss of generality, within all the experiments we assume that neither the initial pose nor the measurement noise is known previously. We compare our adaptive Bingham filter with its prototype (Bingham filter, BF) [46] and the dual quaternion based Kalman filter (DQF) [45] considering that 1) the three filters employ the same model to formulate the pose estimation problem so there is no relative modeling error; 2) the BF and DQF can present the state-of-the-art performances within the best of our knowledge.

3.4.1. Tests on Simulation

For setup we manually create two corresponding point sets $\mathbf{a}_i$ and $\mathbf{b}_i$ and register them with the proposed algorithm, which is similar to the one used in [45, 46]. In practice, the points set $\mathbf{a}_i$ is first randomly generated in the range of $[-200, 200]$. Then the noiseless $\mathbf{b}_i$ is generated by applying a transformation on $\mathbf{a}_i$ with a randomly set quaternion and a randomly generated translation within $[-70, 70]$. Varied noises are respectively added to $\mathbf{b}_i$ in different experiments for testing and comparisons. All the filters are initialized to $\mathbf{q}_0 = (1, 0, 0, 0)$, which subscribes to the Bingham distribution with $\mathbf{M}_0 = \mathbf{I}_4$ and $\mathbf{Z}_0 = diag(0, 0, 0, 0)$ for representing high uncertainty [1].

Randomly set ρ

One of the main advantages of the adaptive Bingham filter is the robustness on determining the coefficient ρ of the measurement noise. Hence hereby we first simulate the manual tuning procedure by assigning randomly selected value from $[0, 200]$ to ρ for all the filters.

In Expt. 1, no noise is added; in Expt. 2 and Expt. 3, the noises uniformly drawn from $[-3, 3]$ and $[-10, 10]$ are respectively added; and in Expt. 4 and Expt. 5, the Gaussian noises $N(\mathbf{0}, 3\mathbf{I}_3)$ and $N(\mathbf{0}, 10\mathbf{I}_3)$ are used. Within each noise condition the

[1] All the algorithms are tested with C++ implementation on a Thinkpad X240 laptop with 8GB RAM and an Intel i5-4200U CPU.

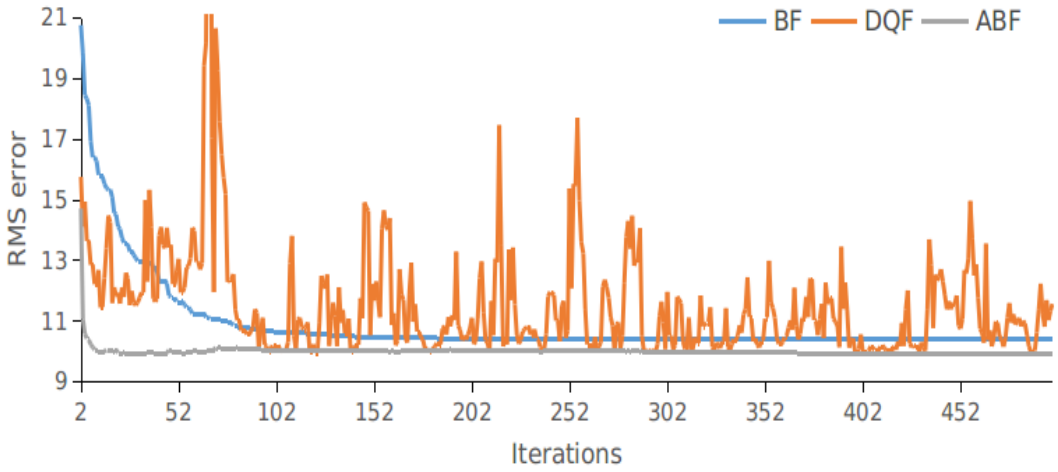


Figure 3.2.: Convergence curves in $[-10, 10]$ uniform noise condition (Expt. 3), where randomly set ρ is used. The result after the first iteration is omitted for better visualization.

Table 3.1.: RMS error in simulation with randomly set ρ

	Expt. 1 RMSE	Expt. 2 RMSE	Expt. 3 RMSE	Expt. 4 RMSE	Expt. 5 RMSE	Time (ms)
BF	0.00	3.59	12.57	5.67	20.73	19.32
DQF	0.00	3.21	10.26	5.65	29.05	9.38
ABF	0.00	2.98	9.97	5.27	17.34	31.04
ABFD	0.00	2.98	9.94	5.25	17.25	40.32

tests are repeated 100 times on varied $\mathbf{a}_i$ and $\mathbf{b}_i$; and the final root-mean-square error (RMSE) of the apart distances and time elapse shown in Table 3.1 are computed as the average values of them.

Another index to evaluate the performance is the rate of successful convergences (the estimated $\mathbf{q}$ converges to a reasonable error) as shown in Table 3.2. With Expt. 3 and Expt. 5 it is clear that the Kalman filter is more easy to suffer from divergence than the Bingham ones in large noise conditions.

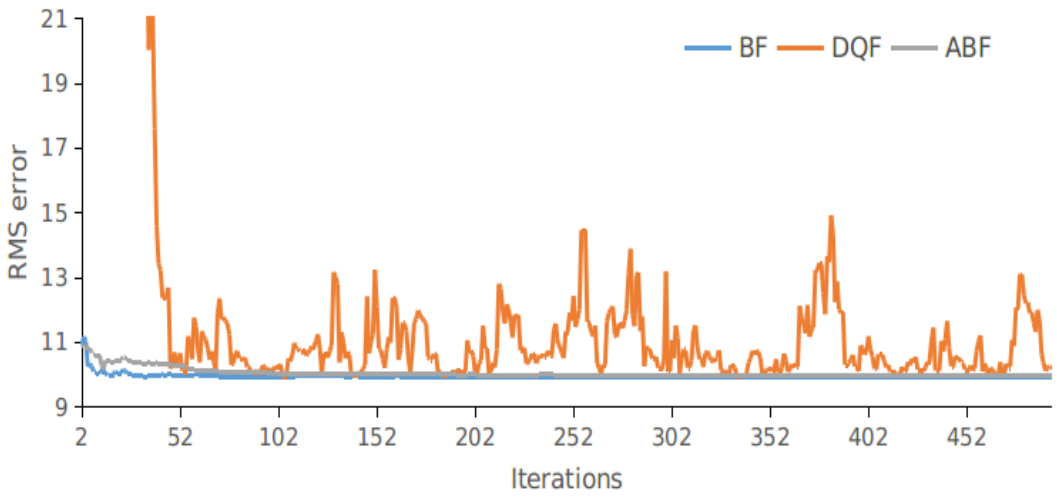


Figure 3.3.: Convergence curves in $[-10, 10]$ uniform noise condition (Expt. 3), where exhausted-searched ρ is used. The result after the first iteration is omitted for better visualization.

Table 3.2.: Success rate in silulation with randomly set ρ

	Expt. 1 SR	Expt. 2 SR	Expt. 3 SR	Expt. 4 SR	Expt. 5 SR
BF	100%	100%	100%	100%	100%
DQF	100%	99%	94%	99%	76%
ABF	100%	100%	100%	100%	100%
ABFD	100%	100%	100%	100%	100%

Optimal ρ

In this section we would like to give the BF and DQF the largest chances to win and hence we first process an exhausted searching to find the optimal ρ for them. Then the experiments described above are re-processed and the results are shown in Table 3.3. With the optimal ρ all the three filters reach 100% success rate.

Table 3.3.: RMS error in simulation with optimally selected ρ

	Expt. 1 RMSE	Expt. 2 RMSE	Expt. 3 RMSE	Expt. 4 RMSE	Expt. 5 RMSE
BF	0.00	2.98	9.94	5.18	17.25
DQF	0.00	3.17	10.37	5.24	21.27
ABF	0.00	2.99	9.97	5.19	17.26
ABFD	0.00	2.98	9.94	5.18	17.25

Result Discussion

As shown in Table 3.1, without any preliminaries of the measurement noise, the ABF is more accurate than the other past methods; and the ABFD performs slightly better than ABF. And with Table 3.3 it is clear that the ABF can achieve almost the same accuracy with the fine-tuned BF. For further analysis the convergence curves of Expt. 3 are also plotted. As shown in Fig. 3.2 and Fig. 3.3, the ABF can converge with less iterations when ρ is randomly set and has similar performance with the fine-tuned BF.

3.4.2. Tests on Realistic Data

For testing the proposed algorithm in the real world, we consider to take the camera pose estimation problem as an example. The experiments are processed on several sequences of the TUM RGB-D benchmark [74]. Within each sequence the pose between every 5 frames is estimation. Specifically, we extract SIFT feature points from the color images of two RGB-D frames and coherently match them. As shown in Fig. 3.4, the matched feature points can be projected into the 3D space with the information from the depth images for registration. Both the filters and the PnP algorithm [26] are used for comparisons.

When the $SE(3)$ transformation is estimated, all the pixels on the two frames can be corresponded and we take the RMSE as the pixel value differences among them. The filters are iterated $\frac{n}{2}$ times where n is the amount of extracted corresponding points. The final results of both the RMSE and the time elapse are represented as the means of them. As shown in Table 3.4, the ABF and ABFD can achieve more accurate results compared with the BF and DQF, which is just slightly worse than the batch-based

method (PnP). What's more, the ABF is also faster than the PnP.

Table 3.4.: RMS error in real world

		Seq. 1	Seq. 2	Seq. 3	Seq. 4	Seq. 5
PnP	RMSE	18.14	20.98	15.92	22.47	13.82
	Time (ms)	4.27	5.36	3.31	3.02	3.73
BF	RMSE	20.85	23.49	18.72	24.29	17.67
	Time (ms)	1.57	1.27	1.96	1.79	1.84
DQF	RMSE	62.35	57.56	62.14	99.46	51.86
	Time (ms)	1.23	1.02	1.56	1.37	1.45
ABF	RMSE	18.77	21.24	16.26	22.74	13.86
	Time (ms)	2.59	2.06	3.22	2.97	3.04
ABFD	RMSE	18.68	21.19	16.20	22.78	13.88
	Time (ms)	3.50	2.72	4.35	3.92	4.03

(a) (b)

Figure 3.4.: Point cloud representation of two RGB-D frames captured by the Kinect. The $SE(3)$ motion between them can be estimated by using the overlapped points.

3.5. Summary

In this chapter we have introduced an adaptive Bingham distribution based filter to estimate the $SE(3)$ elements. In each iteration, a pair of corresponding points is input and the rotation is estimated. Also the noise covariance is autonomously updated for next iteration. Besides the inherent merits of the unique properties of the Bingham distribution and the true-linear model developed in [45], we also employ adaptive filtering technique and consequently free the filter from manual parameter tuning, which is more robust, labor-saving and precise.

As demonstrated in the simulation and real world experiments, although the Bingham filter possesses the ability to present outstanding performance than the traditional Kalman filter, it requires manual tuning of the covariance coefficient (as shown in Table 3.1 and Fig. 3.2) and hence heavily rely on the fine-tuning of measurement noise coefficient. On the other hand, our adaptive approach can obtain high accuracy without manual adjustment.

4. Conclusion and Discussion

4.1. Conclusion

In this thesis we focus on precise 3D point clouds based 3D reconstruction problems and explore it from two perspectives: how to achieve selective reconstruction and how to enhance the efficiency and accuracy. And two algorithms are proposed respectively.

The first one is an RGB-D inapinting framework which enables to remove undesired objects. The annoyances can be removed by either manual selection or semantic segmentation. And we jointly use other frames belonging to the same sequence, exemplar based approach and propagation based one to fill in the holes. Our algorithm can be effective maintain the texture consistency and will not suffer from the scene missing triggered by camera undistortion. Also, although our framework is designed for RGB-D sequences, it can be easily modified to deal with RGB ones since the color image and the depth one are separately inpainted. However, based on the local homography algorithm, our method is easy to suffer from the sparseness of feature points when handling large baseline conditions.

The second algorithm is related to online pose estimation, in which we equip the Bingham filter with the covariance matching adaptive filtering technique, which effectively achieves higher accuracy. Besides, our proposal frees the users from burdensome manual tuning and hence more easy to use. Same as other filtering based approaches, however, although the adaptive filtering technique is robust, it still has the risk to suffer from being trapped in local minima provided large initialization error, which can be solved by the global optimization techniques such as branch-and-bound [75] or other heuristic methods [76, 77].

4.2. Future Work

For the inpainting algorithm, other methods like those employing grid optimization can be explored to deal with large baseline conditions [78] in the future. Also more suitable source selection strategy could be designed to avoid the current demand on weights adjustment. Another line of interest is that high-performance MRF [79] solutions can be used in order to enhance the time efficiency.

For the pose estimation algorithm, we plan to extend the current filter to a robust version since the current one still lacks the ability to deal with outliers, where robust filtering technique is a reasonable choice [80]. Another thread of the future work is about the time efficiency. Specifically, since the current filter calculates the error between $\mathbf{q_i}$ and $\mathbf{q_{i-1}}$ with an matrix-inverse metric in each iteration, whose time complexity is $O(n^3)$, some metric approximations [81] would be preferred to enhance the efficiency.

References

[1] Y. Zeng, Y. Hu, S. Liu, J. Ye, Y. Han, X. Li, and N. Sun, "Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3434–3440, 2018.

[2] V. Chougule, A. Mulay, and B. Ahuja, "Development of patient specific implants for minimum invasive spine surgeries (miss) from non-invasive imaging techniques by reverse engineering and additive manufacturing techniques," *Procedia Engineering*, vol. 97, pp. 212–219, 2014.

[3] F. López, P. Lerones, J. Llamas, J. Gómez-García-Bermejo, and E. Zalama, "A review of heritage building information modeling (h-bim)," *Multimodal Technologies and Interaction*, vol. 2, no. 2, p. 21, 2018.

[4] S. Caccamo, E. Ataer-Cansizoglu, and Y. Taguchi, "Joint 3d reconstruction of a static scene and moving objects," *arXiv preprint arXiv:1802.04738*, 2018.

[5] E. Ataer-Cansizoglu and Y. Taguchi, "Object detection and tracking in rgb-d slam via hierarchical feature grouping," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 4164–4171, 2016.

[6] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. on Graphics*, vol. 28, no. 3, p. 24, 2009.

[7] N. Kawai, T. Sato, and N. Yokoya, "Image inpainting considering brightness change and spatial locality of textures and its evaluation," in *Proc. of Pacific-Rim Symp. on Image and Video Technology (PSIVT)*, pp. 271–282, 2009.

[8] M. Ebdelli, O. Le Meur, and C. Guillemot, "Video inpainting with short-term windows: application to object removal and error concealment," *IEEE Trans. on Image Processing*, vol. 24, no. 10, pp. 3034–3047, 2015.

[9] F. Klose, O. Wang, J.-C. Bazin, M. Magnor, and A. Sorkine-Hornung, "Sampling based scene-space video processing," *ACM Trans. on Graphics*, vol. 34, no. 4, p. 67, 2015.

[10] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2881–2890, 2017.

[11] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[12] J. H. Lee, I. Choi, and M. H. Kim, "Laplacian patch-based image synthesis." in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2727–2735, 2016.

[13] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt, "Background inpainting for videos with dynamic objects and a free-moving camera," in *Proc. of European Conf. on Computer Vision (ECCV)*, pp. 682–695, 2012.

[14] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 1993–2019, 2014.

[15] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Trans. on Graphics*, vol. 26, no. 3, p. 4, 2007.

[16] O. Whyte, J. Sivic, and A. Zisserman, "Get out of my picture! internet-based inpainting." in *Proc. of British Machine Vision Conference (BMVC)*, p. 5, 2009.

[17] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," *ACM Trans. on Graphics*, vol. 23, no. 3, pp. 294–302, 2004.

[18] S.-H. Baek, I. Choi, and M. H. Kim, "Multiview image completion with space structure propagation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 488–496, 2016.

[19] T. Thonat, E. Shechtman, S. Paris, and G. Drettakis, "Multi-view inpainting for image-based scene editing and rendering," in *Proc. of Int. Conf. on 3D Vision (3DV)*, pp. 351–359, 2016.

[20] J. Philip and G. Drettakis, "Plane-based multi-view inpainting for image-based rendering in large scenes," in *Proc. of ACM SIGGRAPH Symp. on Interactive 3D Graphics and Games (I3D)*, pp. 1–11, 2018.

[21] D. Miao, J. Fu, Y. Lu, S. Li, and C. W. Chen, "Texture-assisted kinect depth inpainting," in *Proc. of IEEE Int. Symp. on Circuits and Systems (ISCAS)*, pp. 604–607, 2012.

[22] A. Atapour-Abarghouei and T. P. Breckon, "Depthcomp: real-time depth image completion based on prior semantic scene segmentation," in *Proc. of British Machine Vision Conference (BMVC)*, 2017.

[23] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving dlt," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2339–2346, 2013.

[24] P. C. Ng and S. Henikoff, "Sift: Predicting amino acid changes that affect protein function," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3812–3814, 2003.

[25] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[26] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *Int. journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[27] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[28] K. Bahrami and A. C. Kot, "A fast approach for no-reference image sharpness assessment based on maximum local variation," *IEEE Signal Processing Letters*, vol. 21, no. 6, pp. 751–755, 2014.

[29] L. Ying, Z. Li, and C. Zhang, "No-reference sharpness assessment with fusion of gradient information hvs filter (in chinese)," *Journal of Image and Graphics*, vol. 20, no. 11, pp. 1446–1452, 2015.

[30] H. Liu, G. Zhang, and H. Bao, "Robust keyframe-based monocular slam for augmented reality," in *Proc. of IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pp. 1–10, 2016.

[31] J.-M. Morel and G. Yu, "Asift: A new framework for fully affine invariant image comparison," *SIAM journal on imaging sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[32] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[33] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?" *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.

[34] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[35] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. on graphics*, vol. 22, no. 3, pp. 313–318, 2003.

[36] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3607–3613, 2011.

[37] N. Kawai and N. Yokoya, "Image inpainting considering symmetric patterns," in *Proc. of Int. Conf. on Pattern Recognition (ICPR)*, pp. 2744–2747, 2012.

[38] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of Int. Conf. on Intelligent Robot Systems (IROS)*, 2012.

[39] L. Feiran, D. Ming, T. Jun, and O. Tsukasa, "Static 3d map reconstruction based on image semantic segmentation," in *Proc. of Int. Conf. on Ubiquitous Robots (UR)*, 2018.

[40] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: an open-source mesh processing tool." in *Proc. of Eurographics*, pp. 129–136, 2008.

[41] K. Brink and A. Soloviev, "Filter-based calibration for an imu and multi-camera system," in *Proc. of Position Location and Navigation Symp. (PLANS)*, pp. 730–739, 2012.

[42] L. El Hafi, K. Takemura, J. Takamatsu, and T. Ogasawara, "Model-based approach for gaze estimation from corneal imaging using a single camera," in *Proc. of IEEE/SICE Int. Symp. on System Integration (SII)*, pp. 88–93, 2015.

[43] M. H. Moghari and P. Abolmaesumi, "A novel incremental technique for ultrasound to ct bone surface registration using unscented kalman filtering," in *Proc. of Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 197–204, 2005.

[44] C. Bingham, "An antipodally symmetric distribution on the sphere," *The Annals of Statistics*, pp. 1201–1225, 1974.

[45] R. A. Srivatsan, G. T. Rosen, D. F. N. Mohamed, and H. Choset, "Estimating se (3) elements using a dual quaternion based linear kalman filter." in *Proc. of Robotics: Science and Systems (RSS)*, 2016.

[46] R. A. Srivatsan, M. Xu, N. Zevallos, and H. Choset, "Bingham distribution-based linear filter for online pose estimation," in *Proc. of Robotics: Science and Systems (RSS)*, 2017.

[47] R. Arun Srivatsan, M. Xu, N. Zevallos, and H. Choset, "Probabilistic pose estimation using a bingham distribution-based linear filter," *Int. Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1610–1631, 2018.

[48] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-d rigid body transformations: a comparison of four major algorithms," *Machine Vision and Applications*, vol. 9, no. 5-6, pp. 272–290, 1997.

[49] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, no. 5, pp. 698–700, 1987.

[50] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America*, vol. 4, no. 4, pp. 629–642, 1987.

[51] B. K. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America*, vol. 5, no. 7, pp. 1127–1135, 1988.

[52] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-d location parameters using dual number quaternions," *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 358–367, 1991.

[53] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, no. 4, pp. 376–380, 1991.

[54] A. Myronenko and X. Song, "On the closed-form solution of the rotation matrix arising in computer vision problems," *arXiv preprint arXiv:0904.1613*, 2009.

[55] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Proc. of Sensor Fusion IV: Control Paradigms and Data Structures*, pp. 586–607, 1992.

[56] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992.

[57] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2743–2748, 2003.

[58] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2010.

[59] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Proc. of European Conf. on Computer Vision (ECCV)*, pp. 558–569, 2004.

[60] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3d point clouds and meshes: a survey from rigid to nonrigid." *IEEE Trans. on Visualization and Computer Graphics*, vol. 19, no. 7, pp. 1199–1217, 2013.

[61] M. H. Moghari and P. Abolmaesumi, "Point-based rigid-body registration using an unscented kalman filter," *IEEE Trans. on Medical Imaging*, vol. 26, no. 12, pp. 1708–1728, 2007.

[62] G. Bishop, G. Welch *et al.*, "An introduction to the kalman filter," *Proc. of SIGGRAPH, Course*, vol. 8, no. 27599-3175, p. 59, 2001.

[63] R. Sandhu, S. Dambreville, and A. Tannenbaum, "Point set registration via particle filtering and stochastic dynamics," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1459–1473, 2009.

[64] J. Glover and L. P. Kaelbling, "Tracking 3-d rotations with the quaternion bingham filter," *Technical Report of Massachusetts Institute of Technology*, 2013.

[65] R. Mehra, "Approaches to adaptive filtering," *IEEE Trans. on Automatic Control*, vol. 17, no. 5, pp. 693–698, 1972.

[66] F. Janabi-Sharifi and M. Marey, "A kalman-filter-based method for pose estimation in visual servoing," *IEEE Trans. on Robotics*, vol. 26, no. 5, pp. 939–947, 2010.

[67] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation," in *Proc. of Power & Energy Society General Meeting*, pp. 1–5, 2017.

[68] H. Zhou, H. Huang, H. Zhao, X. Zhao, and X. Yin, "Adaptive unscented kalman filter for target tracking in the presence of nonlinear systems involving model mismatches," *Remote Sensing*, vol. 9, no. 7, p. 657, 2017.

[69] G. Kurz, I. Gilitschenski, S. Julier, and U. D. Hanebeck, "Recursive bingham filter for directional estimation involving 180 degree symmetry," *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 90–105, 2014.

[70] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck, "Unscented orientation estimation based on the bingham distribution," *IEEE Trans. on Automatic Control*, vol. 61, no. 1, pp. 172–177, 2016.

[71] Y. Meng, S. Gao, Y. Zhong, G. Hu, and A. Subic, "Covariance matching based adaptive unscented kalman filter for direct filtering in ins/gnss integration," *Acta Astronautica*, vol. 120, pp. 171–181, 2016.

[72] D. Choukroun, I. Y. Bar-Itzhack, and Y. Oshman, "Novel quaternion kalman filter," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 174–190, 2006.

[73] W. G. Cochran, *Sampling Techniques*, 3rd ed., 1977.

[74] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 573–580, 2012.

[75] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.

[76] J. Kennedy, "Particle swarm optimization," *Encyclopedia of Machine Learning*, pp. 760–766, 2010.

[77] S. P. Brooks and B. J. Morgan, "Optimization using simulated annealing," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 44, no. 2, pp. 241–257, 1995.

[78] C.-C. Lin, S. U. Pankanti, K. Natesan Ramamurthy, and A. Y. Aravkin, "Adaptive as-natural-as-possible image stitching," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1155–1163, 2015.

[79] D. Thuerck, M. Waechter, S. Widmer, M. von Buelow, P. Seemann, M. E. Pfetsch, and M. Goesele, "A fast, massively parallel solver for large, irregular pairwise Markov random fields," in *Proc. of High Performance Graphics (HPG)*, 2016.

[80] L. Chang, B. Hu, G. Chang, and A. Li, "Robust derivative-free kalman filter based on huber's m-estimation methodology," *Journal of Process Control*, vol. 23, no. 10, pp. 1555–1561, 2013.

[81] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.

A. Appendix

A.1. Quaternion

Quaternion is outstanding for representing 3D rotation than other methods such as Euler angles and rotation matrix for its compact formation (4 versus 9 parameters in the rotation matrix) and non-singularity (does not suffer from Gimbal lock as Euler angles do).

A quaternion is written in the form of

$$\mathbf{q} = (q_0, q_1, q_2, q_3), \quad \mathbf{q} \in \mathbb{R}^4, \tag{A.1}$$

where q_0 is the scalar part and (q_1, q_2, q_3) is the vector part. A quaternion is unit if conditioned to $\|\mathbf{q}\| = 1$.

Operations on Quaternion

While the addition of quaternion works in the same way as that on the real numbers, the multiplication of two quaternions is denoted as

$$\mathbf{p} \odot \mathbf{q} = \begin{bmatrix} p_0 & -\mathbf{p_v}^T \\ \mathbf{p_v} & \mathbf{p_v}^\times + p_0\mathbf{I}_3 \end{bmatrix} \mathbf{q}, \tag{A.2}$$

where $\odot$ is the Hamilton product operator; $\mathbf{p_v}$ and $\mathbf{q_v}$ stand for the vector parts of the quaternions; and $[\ ]^\times$ is the skew-symmetric matrix representation of vector.

The conjugation $\mathbf{q}^*$ of a given quaternion $\mathbf{q}$ is denoted as $\mathbf{q}^* = (q_0, -q_1, -q_2, -q_3)$. For unit quaternion, the inverse $\mathbf{q^{-1}}$ is the same as $\mathbf{q}^*$.

Rotation Representation

An *SO*(3) rotation can be represented by a unit quaternion in the form of

$$\mathbf{q} = (cos(\frac{\theta}{2}), \mathbf{v}sin(\frac{\theta}{2})), \tag{A.3}$$

where the vector $\mathbf{v}$ stands for the rotation axis and $\theta \in [-\pi, \pi]$ is the rotation angle around it. Therefore, it is reasonable to confirm that $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation since rotating θ around axis $\mathbf{v}$ is the same as $-\theta$ around $-\mathbf{v}$.

The rotation between a given pair of 3D points $\mathbf{a}$ and $\mathbf{b}$ can be represented as

$$\mathbf{b} = \mathbf{q} \odot \mathbf{a} \odot \mathbf{q}^{-1}, \tag{A.4}$$

where $\mathbf{a}$ and $\mathbf{b}$ are respectively written into the quaternion forms $(0, \mathbf{a})$ and $(0, \mathbf{b})$.