修 士 論 文 概 要 書

Summary of Master's Thesis

Date of submission: 07/24/2018

専攻名 (専門分野) Department	Computer Science and Communications Engineering	氏 名 Name	Song Mengcheng	指 導 教 員 Advisor	印 Hiroshi Watanabe Seal
研究指導名 Research guidance	Audiovisual Information Processing	学籍番号 Student ID number	CD 5116FG15-1		
研究題目 Title	Research on Structure from Motion for Omni-directional Images Based on Modification Combined with Moving Objects Elimination Based on Mask R-CNN				

1. Introduction

Structure from Motion (SfM) is a visual based method which can recover the camera poses as well as sparse 3D structure from a set of corresponding unstructured images. It has been widely researched in conventional perspective image and fish-eye image cases for applications such as augmented reality (AR), automotive, video stabilization, scene roaming, etc. At present, with the appearance of Ricoh Theta, which is a portable omni-directional (OD) camera covering Field of View (FoV) of 360 degrees, more and more people tend to share and upload OD images to social networks. We can expect to see the fast growth of overwhelming amount of OD vision data in the future. Because of the large FoV and increasing amounts of OD images, it is worth researching on SfM by OD images. However, the projection model of conventional perspective image and fish-eye image is different with OD image, which leads to widely researched SfM pipeline not suitable anymore. Additionally, equi-rectangular (EQR) image, as the most common OD format we could search on the Internet, is associated with non-uniform distortion which makes the matched keypoints not reliable by conventional feature detection/matching algorithms. In this work, we propose a modified SfM system for OD images with preprocessing steps to address the problem of non-distortion.

On the other hand, the performance of SfM would suffer from the moving obstacles (such as pedestrians and vehicles) existing on used images since these pixels would bother the feature correspondences as well as motion estimation. Thus, we further combine the proposed SfM with moving objects elimination based on Mask R-CNN [1], which is a state-of-the-art deep Convolutional Neural Network base model for object detection.

The results show that the proposed SfM system could work well for OD images to recover a sparse 3D structure as well as ego-motion between corresponding images. Besides, by combining moving objects elimination, we can accelerate the process of RANSAC [2] which resulting in more accurate motion parameters.

2. Related Technologies

The SfM part of this work is based on modification of widely researched classical SfM pipeline [3][4], which first detect and match feature points by SIFT to obtain corresponding 2D keypoints between images, then recover the motion parameters by RANSAC with epipolar constraint, after that, 3D coordinates of the keypoints are calculated by 3D triangulation, finally bundle adjustment (BA) are utilized for non-linear refinement of initially recovered motion parameters and 3D points.

For combining elimination of moving obstacles, we first need to detect these moving objects. Here, we utilize Mask R-CNN model, it is proposed by Facebook Research in 2017. Mask R-CNN improved the performance of Faster R-CNN by building a multi-task model which is trained with the loss function not only for bounding boxes regression and classification, but also for a binary mask prediction.

3. Proposed Approach

Figure.1 shows the proposed system for OD images by combing the modification of conventional SfM pipeline with moving objects elimination based on Mask R-CNN

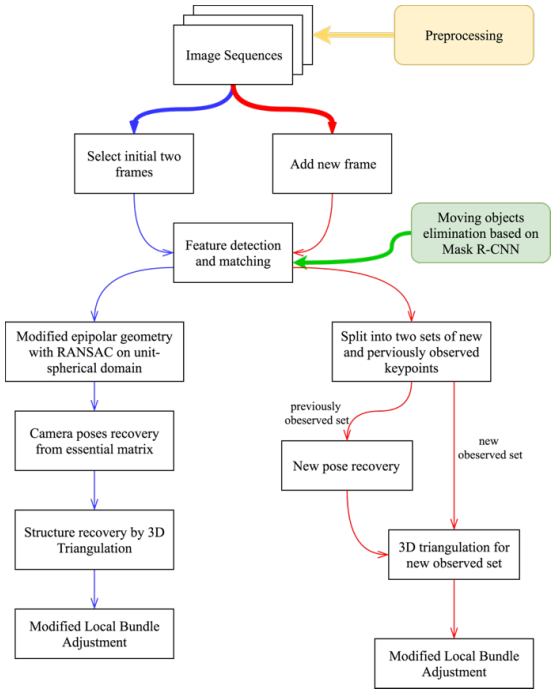


Figure.1. Proposed SfM system for OD images

3.1 Unit-spherical model

$$\begin{cases} u = (\pi - \phi) * \dfrac{width}{2\pi} \\ v = \theta * \dfrac{height}{\pi} \end{cases} \quad (1)$$

$$p' = [cos\theta cos\phi, sin\theta, cos\theta sin\phi] \quad (2)$$

As shown in Figure.2. From Eq. (1,2), we build the projection model which project 2D pixel on EQR image to 3D unit-spherical model, further processing steps are all based on the 3D unit-spherical coordinates of each point.

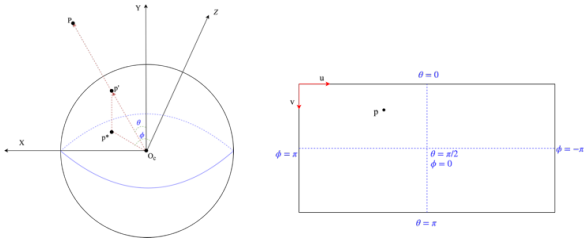


Figure.2. Projection from 3D to 2D of EQR image

3.2 Preprocessing

To address the non-uniform distortion problem described before, we applied cubic mapping.

3.3 RANSAC on unit-spherical domain

The RANSAC method for building fundamental matrix in classical SfM designed the error function on 2D pixel domain since there is a linear intrinsic matrix K could be used to describe the projection in perspective images. For EQR image, we define the error function as angular error L:

$$L = \cos^{-1}(p_2'^T \cdot E p_1') \quad (3)$$

We set the thresholding value as L> 85°, which means reprojected point on 3D domain should be perpendicular to the normal vector of epipolar plane. Note that in our system, RANSAC is the process for modeling the essential matrix E, which also remove some outliers.

3.4 Pose Estimation

In conventional perspective image case, the four possible combination of R and T ambiguity is addressed by judging the recovered depth value of 3D point. For OD image, there is no constraint of positive depth value, so we defined the angular error between reprojected point and detected point on 3D domain with angle α which should satisfy $\alpha < 5°$.

$$\alpha = \cos^{-1}(p_2'^T \cdot P_{c2}^T) \quad (4)$$

3.5 Bundle Adjustment

In conventional perspective image case, the error function for BA is designed as the 2D pixel distance between detected feature and reprojected feature, here we define it in 3D unit-spherical domain as:

$$\min_{R,T,P} \|p - p'\|^2 = \min_{R,T,P} \left\| \begin{bmatrix} x(R,T,P) \\ y(R,T,P) \\ z(R,T,P) \end{bmatrix} - \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \right\|^2 \quad (5)$$

4. Experiment and Results

The proposed SfM system is implemented in Python, we perform SIFT algorithm by using open-source library OpenCV, and we utilize the Mask R-CNN architecture implemented by [5]. The model is trained on COCO dataset. Additionally, image data are divided into 8 groups, group (1-3) are synthetic image pairs, group (4-8) are real images captured in two different scenes.

We evaluate the important steps of proposed SfM including feature correspondences with calculating the repeatability α:

$$\alpha = \frac{\#\ of\ matched\ keypoints\ after\ each\ stage}{\#\ of\ detected\ keypoints} = \frac{k_i}{K} \quad (6)$$

The results are shown in Figure.3. We can see that after cubic mapping, the correct matched points are increased in all the three steps since the non-uniform distortion has been addressed, besides, the process of RANSAC is accelerated because of higher repeatability.

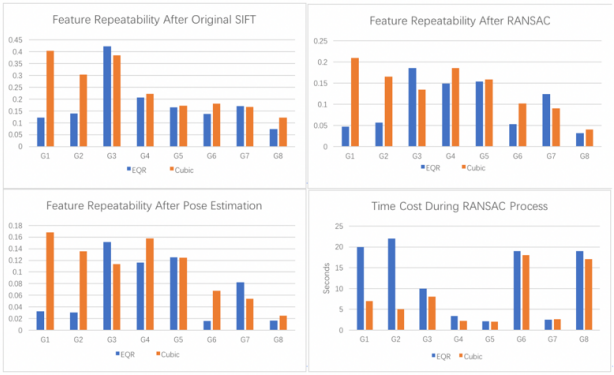


Figure.3. Comparison between EQR and Cubic image on performance of feature correspondences

Group (8,9) are captured in crowded scene with vehicles and pedestrians showing on the images. The intuition of detected results by Mask R-CNN is shown in Fig.4. The performance of combining moving objects elimination is illustrated in Fig.5, which shows it can accelerate the RANSAC process since obvious outliers are removed, besides, more correctly matched points are preserved which leads to more robust estimation of camera pose. We test the SfM system on synthetic image data. The results of recovered 3D scene and camera poses corresponding to each frame are shown in Fig.6 and Table.1. Our proposed system can work well for OD images. Note that, frame 1 is set as the reference of world coordinates system, with $R = I, T = 0$.

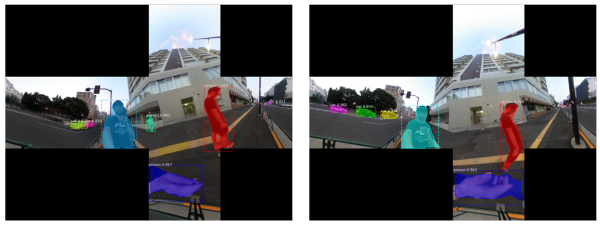


Figure.4. Performance of moving objects detection on cubic image (example taken from group 8)

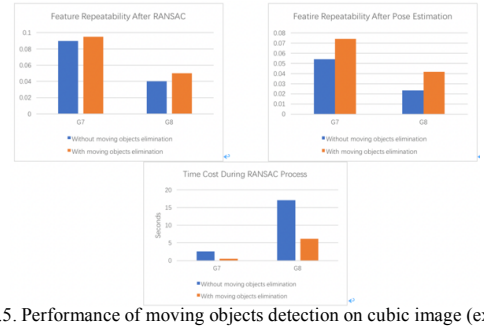


Figure.5. Performance of moving objects detection on cubic image (example taken from group 8)

	Tx	Ty	Tz	α [rad]	β [rad]	γ [rad]
Frame 2 RMSE	0.001	6.8 e-4	0.005	0.00	0.002	0.001
Frame 3 RMSE	0.045	1.3 e-4	0.084	-0.02	0.010	0.003

Table.1. Motion estimation by proposed SfM

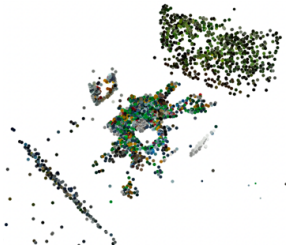


Figure.6. Visualizing reconstructed 3D points by proposed SfM

5. Conclusion

In this work, we proposed a novel SfM which is suitable to EQR images generated by OD camera since the conventional SfM pipeline is associated with perspective image which has different projection model with OD image. We evaluated our system for not only two-view but also multi-view SfM. The results of recovered structure and camera poses showed that proposed system works well with OD image.

We address the non-uniform distortion problem of EQR image by preprocessing step, which is proved to be effective for obtaining more reliable corresponding feature points.

Finally, we combined SfM with elimination of moving obstacles based on Mask R-CNN, the results show that SfM system could benefit from this combination which removed those obvious outliers, thus accelerates the process of RANSAC.

6. References

[1] He, K., et al. Mask r-cnn. in Computer Vision (ICCV), 2017 IEEE International Conference on. 2017. IEEE

[2] Fischler, M.A. and R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 1981. 24(6): p. 381-395.

[3] Snavely, N., S.M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. in ACM transactions on graphics (TOG). 2006. ACM

[4] Hartley, R. and A. Zisserman, Multiple view geometry in computer vision. 2003: Cambridge university press

[5] Abdulla, W. Mask R-CNN for object detection and instance segmentation on Keras and Tensorflow. 2017; Available from: https://github.com/matterport/Mask_RCNN

Research on Structure from Motion for Omni-directional Images Based on Modification Combined with Moving Objects Elimination Based on Mask R-CNN

A Thesis Submitted to the Department of Computer Science and Communications Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University
in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering

July 24 th , 2018.

Song Mengcheng

(5116FG15-1)

Advisor: Prof. Hiroshi Watanabe

Research guidance: Research on Audiovisual Information Processing

Acknowledgements

Firstly, I would love to express my sincere heartfelt thanks to my supervisor Prof. Hiroshi Watanabe for his patience and trust, he gave me so many precious advices during my two years of student life. This thesis would not have been possible without the continuous support from him. Besides, I am really grateful to Mr. Junichi Hara for his valuable suggestions in Omni-directional group.

I really appreciate the enormous help by all my friends in Watanabe lab. My research life in Japan would have been much harder without them.

Most important of all, I am extremely thankful to my parents who were always there for me when I felt depression and frustrated. I would not stick to my research without their supports.

Abstract

Structure from Motion (SfM) is a visual based method which can recover the camera poses as well as sparse 3D structure from a set of corresponding unstructured images. It has been widely researched in conventional perspective image and fish-eye image cases for applications such as augmented reality (AR), automotive, video stabilization, scene roaming, etc. At present, with the appearance of Ricoh Theta, which is a portable omni-directional (OD) camera covering Field of View (FoV) of 360 degrees, more and more people tend to share and upload OD images to social networks. We can expect to see the fast growth of overwhelming amount of OD vision data in the future. Because of the large FoV and increasing amounts of OD images, it is worth researching on SfM with OD images. However, the projection models of conventional perspective image and fish-eye image are different with OD image, which leads to widely researched SfM pipeline not suitable anymore. Additionally, equi-rectangular (EQR) image, as the most common OD format we could search on the Internet, is associated with non-uniform distortion which makes the matched keypoints not reliable by conventional feature detection/matching algorithms. In this work, we propose a modified SfM system for OD images with preprocessing steps to address the problem of non-distortion.

On the other hand, the performance of SfM would suffer from the moving obstacles (such as pedestrians and vehicles) existing on used images since these pixels would bother the feature correspondences as well as motion estimation. Thus, we further combine the proposed SfM with moving objects elimination based on Mask R-CNN, which is a state-of-the-art deep Convolutional Neural Network base model for object detection.

The results show that the proposed SfM system could work well for OD images to recover a sparse 3D structure as well as ego-motion between corresponding images. Besides, by combining moving objects elimination, we can accelerate the process of RANSAC which resulting in more accurate motion parameters.

Keywords: Structure from Motion; Omni-directional image; Mask R-CNN; Feature detection/matching; Moving object elimination;

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

Recovering camera poses and corresponding 3D structure is of great significance for AR, automotive, scene roaming, etc. According to [1, 2], the researches for this task could generally be divided into two groups: the first group can be regarded as monocular visual based methods, that is, only utilize images captured by a single camera; the second group is based on associating various sensors as complements, for example, GPS, IMUs, Lidar, Laser and RGB-D camera. However, the latter group has the limitations such as: expensive, limited range of estimated depth, low resolution and somehow suffering from severe conditions (under water, aerial scene). Thus, we focus on the monocular visual based method by using a single camera which is portable and low cost. Further, the monocular based approaches could be divided into another two groups: the first group is focusing on real-time applications, such as VO (Visual Odometry) [1] and SLAM (simultaneous localization and mapping)[3, 4], they are performed on consecutive sequence of frames which can be regarded as structured data; the second group, in contrary, is focusing on off-line applications [5-7] which is a more general method also known as SfM (Structure form Motion) since it is performed on unstructured image data. Thus, in this thesis, we focus on using SfM to recover the structure. SfM, as a feature-based approach, has been widely researched in last two decades for images taken by conventional pin-hole camera with narrow Filed of View (FoV). However, it usually suffers from the problem of large movement between the images, the tracked feature points might be missed.

Recently, there has been a lot of solutions of camera systems, such as GoPro Fusion and Ricoh Theta, which can produce the omni-directional (OD) vision for immersive experience. Furthermore, YouTube and Facebook have already realized the support of OD vision streaming media, more and more people share and upload this kind of 360° images and videos in social network. We can expect to see that with the growth of overwhelming amount of OD vision data, it would become a main trend of the future development of image and video applications.

Object detection is a computer technology to detect instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. As a main task of computer vision, it has been widely researched in various applications like image retrieval and video surveillance. With the rapid development of computation capability of GPU (Graphics Processing Unit) and image dataset, the performance of object detection has been dramatically improved by utilizing Deep Convolutional Neural Network (Deep CNN) in last few years [8-12]. On the other hand, since the SfM for 3D reconstruction involves the use of images of a stationary scene, the moving objects such as vehicles and pedestrians would apparently bother its performance when images are captured in a

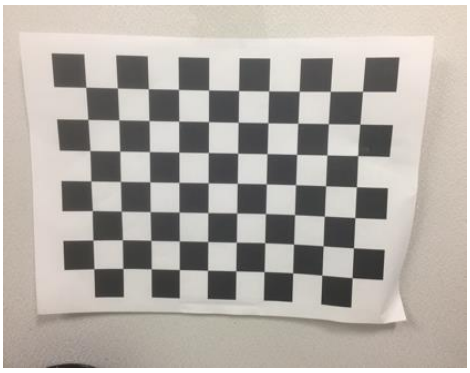
crowded scene. [13] proposed a VO system combined with pedestrian elimination, while the conventional Machine Learning based approach like Histogram of Oriented Gradients (HOG)[14] and Support Vector Machine (SVM)[15] are used for extraction of pedestrians to minimize the effect on reliability and accuracy of camera poses estimation.

As a result, this thesis focuses on modifying conventional SfM and combining it with objects segmentation for 3D reconstruction with OD images in consideration of two opportunities: first, 3D reconstruction and camera poses estimation in scenario with moving obstacles existed would benefit from OD images and objects segmentation, which is shown in Chapter 4; second, less number of OD images are enough to be used for reconstruction due to it covers 360 degree scene information which is far greater than the information in conventional images.

1.2 Problem Statement

3D reconstruction and ego-motion estimation with wide-angle camera has been paid increasing attention to over past few years with its characteristics of portability as well as the large Field of View (FoV). [16] demonstrated that using a wide-angle camera, such as fisheye camera, has an obvious advantage that larger FoV could improve the accuracy and robustness of ego-motion estimation because that the corresponding two images have large overlapping area. Two typical approaches are researched for dealing with this kind of radially distorted images. The first one is based on correction of distortion by calibrating the corresponding fisheye camera, then conventional widely researched SfM pipeline could be used directly like in[6]. This is the most popular way for commercial users or researchers to deal with different distorted projection models. Basically, a generic camera model [17] is utilized to cover various fisheye projections (e.g. stereographic, equidistance, equisolid angle and orthogonal projection), then a full model including radial and tangential distortion is optimized to obtain an intrinsic matrix which describe how 3D ray is projected to 2D pixel domain. Successively, they correct and convert the distorted image into perspective image. However, this kind of undistortion approach would lead to information loss since it introduces the undesirable “pin-cushion” shape [18] appearing at the corrected image boundary which are removed by cropping in most cases. The second approach is based on a relatively direct way. [19]used a polynomial fisheye model for SfM but the performance is not good due to the conventional method of feature correspondences is used. Omni LSD-SLAM [20] is an extension of LSD-SLAM for so-called omni-directional camera, but in fact fisheye camera was used. Besides, as described before, LSD-SLAM is a direct approach for simultaneously 3D mapping and localization rather than a feature-based approach in which feature correspondences should be considered, while the feature-based approach is what we pay attention to in this thesis. [21] proposed a method dealing with SfM for 3D reconstruction from feature correspondences in circular images taken by the cameras equipped with fish-eye lenses Nikon FC-E8

(183°) or Sigma 8mm-f4-EX (180°). Several researches made attempts to detect and match feature points directly on the radially distorted images such as sRD-SIFT [22] and MDBRIEF[23]. They proposed modifications to the SIFT [24] algorithm and BRIEF [25] algorithm that improve the detection repeatability and matching performance under radial distortion, while preserving the original characteristics of those detectors and descriptors. Again, these methods only considered the fisheye images with radial distortion but not 360° images with non-uniform distortion.



(a) Perspective image captured by conventional camera

(b) Full-circle fisheye image taken by fish-eye camera

(c) Omni-directional image taken by Ricoh Theta

Figure 1.1: Comparison of different camera

Unlike other wide-angle images, real OD image as depicted in Figure 1.1 covers surrounding 360° FoV along with horizontal direction and 180° FoV along with vertical direction. Equi-rectangular (EQR) image, as the most common format of OD images we could search on the Internet, is the object of our research. As shown in Figure 1.1, the projection model of EQR image is different with it of fish-eye image and so are the type of distortion (large distortion near south/north pole and less distortion near equator, respectively). Therefore, it introduces new challenge for researches dealing with the EQR image. For addressing the non-uniform distortion, [26] chose to rotate the EQR image for several times then apply conventional SIFT detection and matching on the less-distorted central area of each rotated image. This approach makes sense, but due to the location-dependent projection, the operation of rotation for EQR image costs large computation to calculate corresponding rotated

position of each pixel. On the other hand, [27] try to directly detect and describe feature points on EQR image via a sphere model. They project surrounding patch for each feature points onto a tangent plane to tackle the problem of distortion while preserving the rectangle kernel filter in the process of conventional SIFT algorithm. However, similar as in[26], the tangent projection for each patch increases the computation cost dramatically.

Objects segmentation in OD images is another task that we try to perform. As mentioned in section 1.1, moving obstacles have bad effect on the performance of SfM since they would contribute to wrong estimation and error propagation of camera pose. In [13], HOG is used for extracting feature descriptors which are then fed into SVM for classification, then the area of pedestrians detected in the image will be eliminated when processing feature detection and matching. [28] proposed a similar idea with us several weeks ago, they combined the segmentation with ego-motion estimation in visual based SLAM in order to obtain robust performance in tracking feature points. However, all of them cannot deal with detection or segmentation on OD images since the existing publicly-available datasets for training a conventional machine learning model (like HOG+SVM) or a Deep CNN (Convolutional Neural Network) based model are composed by a great number of manually labeled perspective images. The non-uniform distortion of OD images will dramatically affect the capacity of these methods. The specific reason will be explained in the following chapter.

The main contributions of this thesis are:

1. For obtaining camera poses and a sparse reconstruction of the scene, we propose the modified system to make the conventional Structure from Motion (SfM), which was designed for perspective images, suitable to the omni-directional images.

2. We combine moving objects segmentation in omni-directional images with proposed SfM to accelerate convergence and improve the performance of RANSAC and Bundle Adjustment, hence more accurate results of sparse and dense reconstruction could be achieved. Moreover, two proposals for objects segmentation on EQR images are proposed, which can be used not only in our SfM system but also in other computer vision tasks associated with detection or segmentation.

1.3 Outline

The outline of this thesis is organized as follows:

- Chapter 1: We describe the background and motivation of this research. Also, we explore the related works including their limitations and challenges to introduce the problem statement of this thesis. After that, the important contributions of our works are explained.

- Chapter 2: We introduce Omni-directional (OD) vision, the corresponding Equi-rectangular (EQR) projection and related knowledge about conventional SfM pipeline. Additionally, Mask R-CNN is also introduced in this chapter as it is the state of the art in deep learning-based instance segmentation approaches. For introducing our proposed system in later chapter in consideration of completeness and validity, we mathematically elaborate those related knowledge. Besides, the problems of directly applying conventional methods mentioned above to non-uniformly distorted EQR images are fully discussed.

- Chapter 3: Proposed methods are explained in this chapter for 3D reconstruction by OD images. We detail our modifications on conventional approaches in SfM for sparse reconstruction, resulting in 3D reconstruction directly achieved by OD images. Also, we elaborate the proposals for making conventional Mask R-CNN model work well on non-uniformly distorted EQR images to produce masks of moving objects (vehicles and pedestrians) in images. Those masks for eliminating bothering pixels are combined with feature correspondences to obtain more robust estimation.

- Chapter 4: In this chapter, we first explain the experiment setting and parameters we used for programming implementation. Next, subjective and objective evaluations of the performance of our methods with synthetic and real images are presented, which shows that our system works well in 3D reconstruction using OD images. Apart from that, the results of feature correspondences prove that our methods produce more correctly matched feature points and eliminate moving objects to avoid bothering the convergence of RANSAC and BA, which improves the accuracy of estimation of camera poses as well as final reconstructed model.

- Chapter 5: Finally, we summarize the proposed system in this chapter where the conclusions and deficiencies of our works are discussed. Potential future works are provided for extension of this research.

Chapter 2 Related Technologies

2.1 Omni-directional vision

Omni-directional (OD) vision, also known as 360-degree vision, is recorded by an omni-directional camera as depicted in Figure 2.1. It has been used recently because of rapid advancements in digital image/video technologies and photographic equipment. With the 360-degree vision, users could in demand choose which angle to view for enjoying the immersive experience. Nowadays, the popular solution of OD camera system is utilizing capture equipment such as six GoPro cameras to take pictures separately, followed by processing step of stitching to generate single panoramic image. Recently, with the appearance of Ricoh Theta and GoPro Fusion, which are handy and portable OD cameras embedded with two wide-angle fish-eye lenses (larger than $180°$ FoV) on both front and back sides, the focus has been changed to OD vision in last two years. Due to the potential of the emergence of a large amount of OD data, YouTube and Facebook has already realized the support of OD streaming media. More and more people tend to share and upload the $360°$ images and videos in social network due to the demand of immersive experience.

(a) Gopro Fusion[29] (b) Six Gopro Hero 4[30] (c) Ricoh Theta[31]

Figure 2.1: Three types of Omni-directional camera

2.1.1 Unit-spherical model

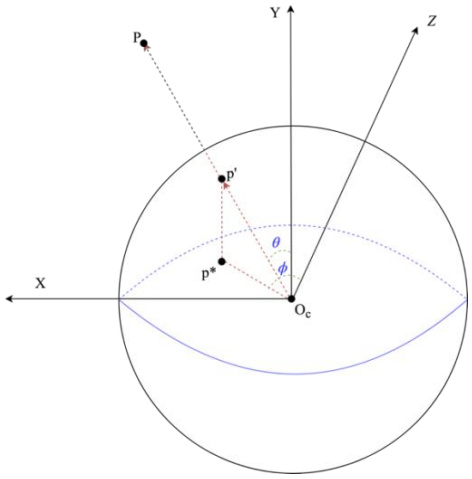
Here, we introduce a unit-spherical model to illustrate the relationship between 3D scene point and optical center of camera, as shown in Figure 2.2(a). It could be considered as a generic model for producing $360°$ images by OD cameras like GoPro Fusion and Ricoh Theta. We set a right-hand coordinate system with respect to the OD camera as Figure 2.2(a) where Z axis stands for the depth. Suppose there is a unit-sphere of radius 1 and a 3D scene point $P = [X_P, Y_P, Z_P]^T$, then it is projected to $p' = [cos\theta cos\phi, sin\theta, cos\theta sin\phi]^T$, where p' is the intersection of the sphere surface with the ray connected by center O_c and point P. So, we have the equation:

$$p' = \frac{P}{\|P\|} \tag{2.1}$$

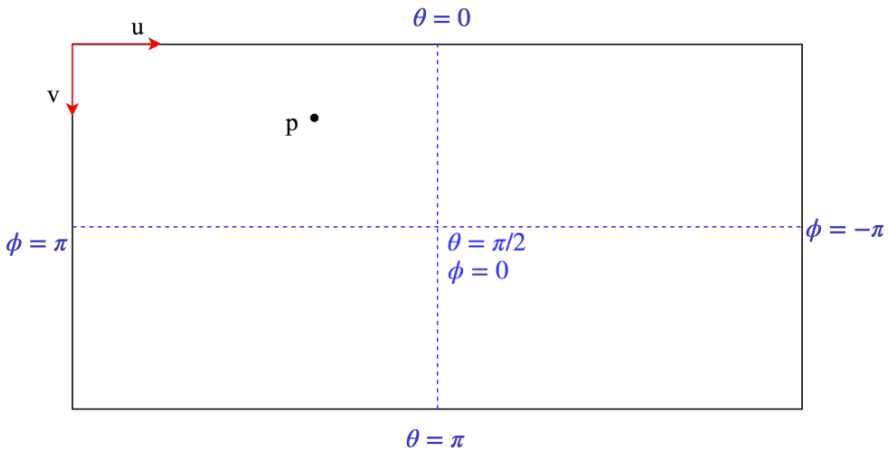
where $\|P\| = \sqrt{X_P^2 + Y_P^2 + Z_P^2}$, $\theta = \angle p' O_c Y$ which represents latitude and $\phi = \angle Z O_c p^*$ which represents the longitude given in radians, respectively. For calculating θ and ϕ from 3D point P by math toolbox, we have:

$$\begin{cases} r = \sqrt{X_P^2 + Z_P^2} \\ \theta = arctan2(r, Y_P) \\ \phi = arctan2(X_P, Z_P) \end{cases} \tag{2.2}$$

where $\theta \in [0, \pi]$ and $\phi \in [-\pi, +\pi]$.



(a) Unit-spherical model



(b) Equi-rectangular projection

Figure 2.2: Projection from 3D to 2D of Omni-directional camera

2.1.2 Equi-rectangular image

Essentially, OD vision is like a spherical vision which projects spatial 3D points on the sphere surface. However, for achieving the capability of storage, interchange, editing and presentation, it is necessary to map this kind of spherical vision to a flat 2D plane. In 2016, Omnidirectional Media Application Format (MPEG-OMAF) specified 4 projections of omni-directional media including equi-rectangular, cylinder, cube, and platonic solid. Since the mapping operation from 3D spherical domain into 2D domain would inevitably introduce interpolation which produces a large number of pixels without valid information and leads to redundant bitrate. As a result, some researches focusing on compression of OD image and video are proposed[32, 33]. Facebook proposed mapping from EQR image to cubic image which reduced redundant information to 25%, [32] compared the sampling density of different projections and proposed a tile segmentation scheme. [33] performed a flatten representation of a geodesic division sphere based on an icosahedron. Unlike those researches described above, our work is focusing on 3D reconstruction with OD images. Since Equi-rectangular (EQR) image, as displayed in Figure 2.2(b), is the most common format of OD image uploaded and shared on the Internet, this thesis is mainly focusing on dealing with EQR images.

Here, we explain the relation of mapping between sphere surface described above and EQR form in detail. As shown in Figure 2.2(b), suppose p is the corresponding 2D pixel of 3D point p' projected on EQR image. Let $p = [u, v]^T$, which is related to longitude ϕ and latitude θ by:

$$\begin{cases} u = (\pi - \phi) * \dfrac{width}{2\pi} \\ v = \theta * \dfrac{height}{\pi} \end{cases} \tag{2.3}$$

where,

$$p' = [cos\theta cos\phi, sin\theta, cos\theta sin\phi] \tag{2.4}$$

here, $width$ and $height$ stand for the horizontal pixels and vertical pixels of resolution of the EQR image, for example, Ricoh Theta produces low-quality EQR image with resolution $2048 * 1024$. From Eq. (2.2) and Eq. (2.3), we can easily implement the projection from 2D plane to normalized coordinates without intrinsic matrix obtained by calibration and vice versa.

Figure 2.3 illustrates the underlying principal of how OD cameras, such as Ricoh Theta and GoPro Fusion, produce EQR image. As mentioned before, The Ricoh Theta camera is actually embedded with two fish-eye lenses on each side of the camera body, each lens captures FoV slightly larger than $180°$. Once the button of shutter pressed, two fish-eye images are recorded with the nature of radial distortion. Then the camera handles stitching process internally so that the two fish-eye images could be combined together into one EQR image by either using the geometry or by detecting feature points on the periphery area of each circle and blending them together.

However, this process will introduce two problems that we have to consider:

1. The non-uniform distortion on EQR image, especially those area near the north and south pole, makes accurate and robust feature correspondences become suffering when we directly utilize traditional algorithm of feature correspondences, mainly because they have no invariance to this location-dependent non-uniform distortion.

2. Considering that the fish-eye image is distorted from the center to the periphery area, and the stitching from two fish-eye images is performed on the outermost area of the circles, so that it will lead to the problem of low quality (i.e., less details) at border area on generated EQR image.

Both of the problems described above would have bad effect on the performance of SfM as well as objects segmentation in our system. In order to address them, we propose an approach which will be explained in section 3.1.1.

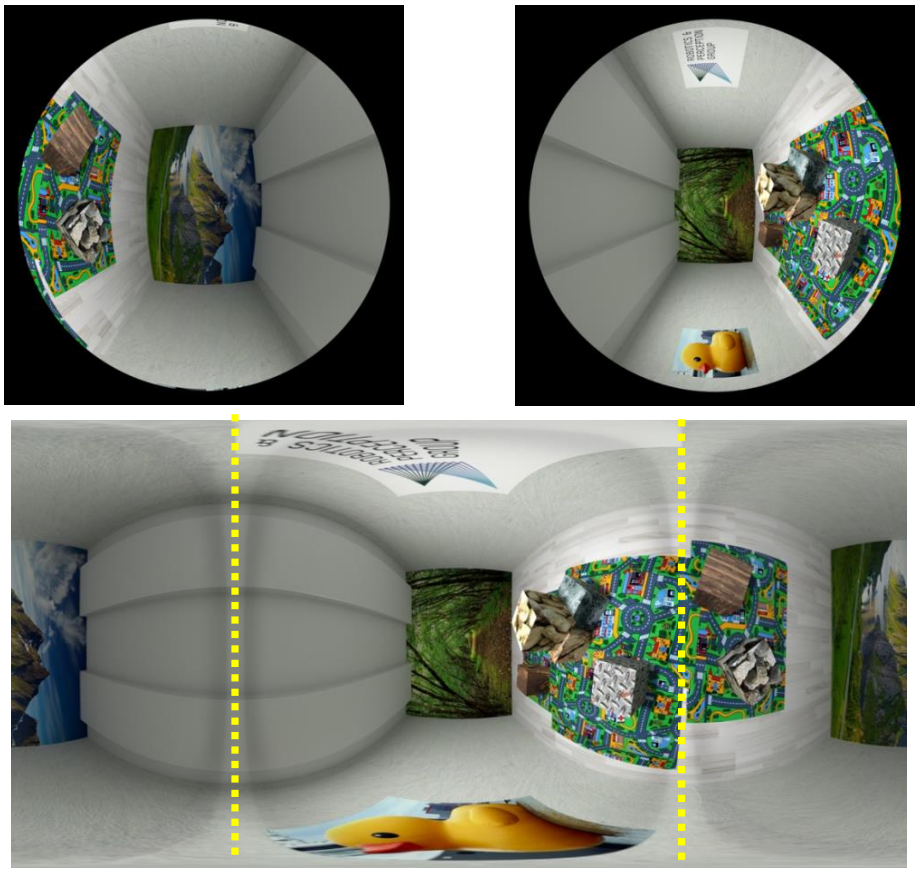


Figure 2.3: illustration of process in generating EQR image from two fish-eye images

2.2 Structure from Motion

Structure from Motion (SfM) is a typical task in Computer Vision to recover the 3D structure of a stationary scene from a set of structured or unstructured 2D images by geometric relationships. We could compute 3D scene points as well as camera locations from feature matches with images captured at different viewpoints in the same scene. In this section, we explore the main steps of classical SfM approach [34, 35]with conventional perspective images, also, the problems of applying it to EQR images are discussed for each step.

Figure 2.4 indicates the pipeline of classical SfM. This approach is beginning with collecting images and calibrating them with pin-hole projection model to obtain the intrinsic matrix. Next, feature detection and matching are performed to gather the robust corresponding points between images. Then, camera pose (extrinsic matrix) could be estimated from a set of pairs of feature points associated with outlier removal. After that, the 3D coordinates of those matched feature points could be calculated by 3D triangulation with camera poses estimated before. Finally, Bundle Adjustment (BA) is applied to nonlinear refinement for the parameters of all the associated images.

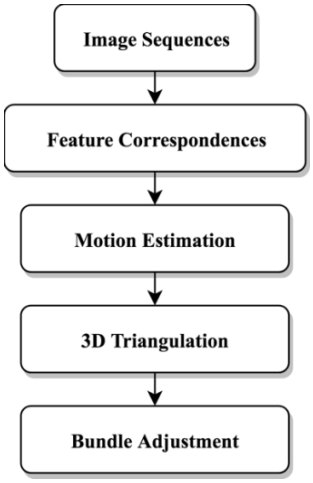


Figure 2.4: Classical Structure from Motion pipeline with perspective images

2.2.1 Pin-hole camera model

Pin-hole camera model, also known as perspective projection model, describes the relationship between 3D coordinates of a point in the scene and its projection onto the 2D image plane of an ideal pin-hole camera without distortion. As shown in Figure 2.5, the pin-hole model involves with the transformation between 3 coordinates system: World Coordinates, Camera Coordinates and Image Coordinates.

Suppose there is a point P in 3D space, let $P_W = [X_W, Y_W, Z_W]^T$ and $P_C = [X_C, Y_C, Z_C]^T$

where P_W and P_C are the coordinates of P with respect to world coordinates and camera coordinates respectively. Then rotation matrix R and translation matrix T could be used to describe the relationship between P_W and P_C as:

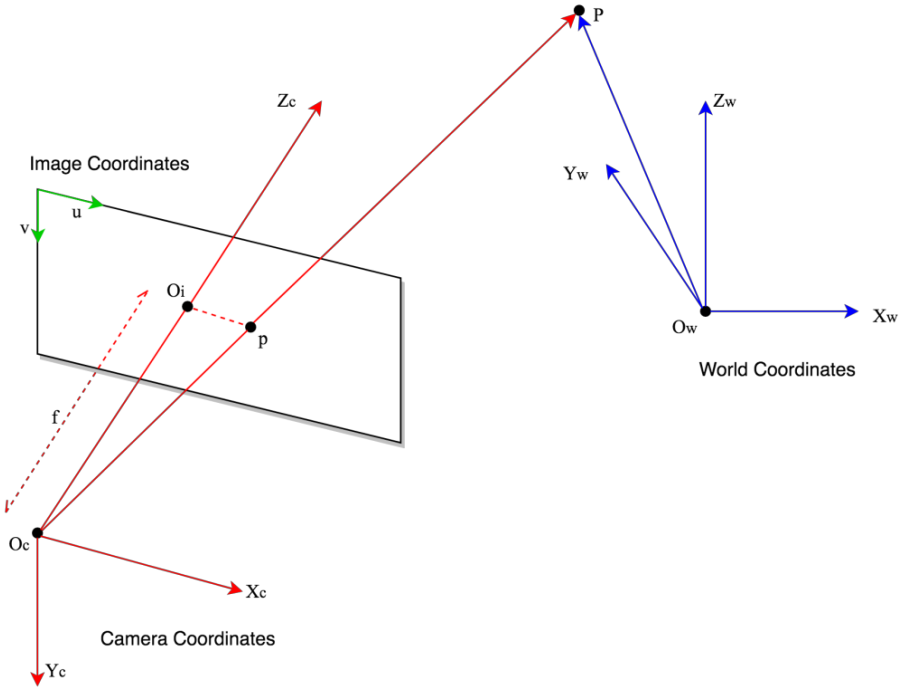


Figure 2.5: Projection from 3D to 2D of pinhole camera

$$P_C = RP_W + T \tag{2.5}$$

specifically,

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} + \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix} \tag{2.6}$$

where $T = [t_X, t_Y, t_Z]^T$ is a 3x1 matrix standing for the coordinates of the origin of world space with respect to camera space. R is a 3x3 orthogonal matrix representing the rotation from world space to camera space with the constraints:

$$\begin{aligned} r_{11}^2 + r_{12}^2 + r_{13}^2 &= 1 \\ r_{21}^2 + r_{22}^2 + r_{23}^2 &= 1 \\ r_{31}^2 + r_{32}^2 + r_{33}^2 &= 1 \end{aligned} \tag{2.7}$$

Sometimes we need present them in homogenous coordinates for simplification (the relation could be described as a single transformation matrix M):

$$\begin{bmatrix} P_C \\ 1 \end{bmatrix} = M \begin{bmatrix} P_W \\ 1 \end{bmatrix} \tag{2.8}$$

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{2.9}$$

So, we combine the rotation matrix R and translation matrix T as the extrinsic parameters which is the so-called ego-motion between cameras as our task.

As we can see in Figure 2.5, p is the intersection of the virtual image plane with the ray connected by camera optical center O_C and 3D point P, f is the focal length. Let $p_C = [x, y, f]^T$ which is the coordinates of p with respect to camera space, we have:

$$\begin{cases} x = f * \dfrac{X_C}{Z_C} \\ y = f * \dfrac{Y_C}{Z_C} \end{cases} \tag{2.10}$$

by normalizing Eq. (2.10) in homogenous coordinates, it can be written as:

$$Z_C \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \tag{2.11}$$

after that, we further transform it to pixel domain in image coordinates. Suppose $p_i = [u, v]^T$ is the pixel coordinates of p, $O_i = [u_0, v_0]^T$ is the pixel coordinates of the origin in image plane. Then p_i and p_C have the relation:

$$u - u_0 = \frac{x}{dx}, v - v_0 = \frac{y}{dy} \tag{2.12}$$

where dx and dy represent the physical size of single pixel in image sensor, also Eq. (2.12) could be rewritten in homogenous coordinates and matrix as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \dfrac{1}{dx} & 0 & u_0 \\ 0 & \dfrac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.13}$$

From Eq. (2.9), (2.11) and (2.13), we could see that: on the premise of the known camera parameters, if we know the 3D coordinates P_W of a point P with respect to the world space, then, the corresponding 2D pixel coordinates p_i could by calculated by:

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \dfrac{f}{dx} & 0 & u_0 \\ 0 & \dfrac{f}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} = K[R, T] \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} \tag{2.14}$$

we define that $K = \begin{bmatrix} \frac{f}{dx} & 0 & u_0 \\ 0 & \frac{f}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$, $[R, T] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{bmatrix}$, where K is the intrinsic matrix of a certain camera, the combination of rotation matrix R and translation matrix T is the extrinsic

matrix $[R,T]$ which describe the orientation and position of a moving camera.

To obtain the intrinsic parameters of K, it is necessary to calibrate the camera by an approach proposed by [36]. This approach is the most used method and implemented by OpenCV as a toolbox we can easily utilize.

	Type	Scale	Rotation	Affine transformation
Harris	corner	x	x	x
FAST	corner	x	x	x
ORB	corner	√	√	x
SIFT	blob	√	√	x
SURF	blob	√	√	x
Affine-SIFT	blob	√	√	√

Table 2.1: Comparison of properties of different detectors

2.2.2 Feature correspondences

Finding corresponding feature points is of utmost importance in SfM since the estimation of ego-motion is derived by a set of pairs of corresponding feature points among the input images. Therefore, the methods for feature detection and matching should generate enough robust matched feature points.

There are many feature detectors have been proposed in last two decades, which can be mainly divided by two groups: 1) the corner detectors, e.g., Harris[37], FAST[38] which is based on machine learning, and ORB[39] which is the extension of FAST by detecting FAST corners on different scale level; 2) the blob-like detectors, e.g., SIFT[24], SURF[40] which improve the computation efficiency of SIFT by using box filters to approximate the Gaussian derivatives, and Affine-SIFT [41] which apply the affine transformation space sampling technique to improve SIFT and aims to achieve affine invariance. The choice of the appropriate feature detectors in different tasks is based on the trade-off between computation-cost and repeatability (whether the detector is robust to scale, rotation invariance). We summarize the properties of these detectors in Table 2.1. Our work is aiming at an off-line 3D reconstruction system, we care more about the repeatability rather than computation-cost. Therefore, SIFT-based approach is used in our experiment, and it is basically composed of 3 main steps: 1) Detection: Identify the local feature points by SIFT detector; 2) Description: Extract a robust vector descriptor of the image content surrounding each detected point; 3) Matching: Comparing the similarities of the descriptors to match corresponding points.

2.2.2.1 SIFT detector

Scale Invariant Feature Transform (SIFT) was proposed by David Lowe in 2004 to detect and describe

local features in images. It is the most popular approach for feature correspondences due to the property of invariance to scaling, rotation and translation.

For explaining SIFT detector, we first introduce scale space in which stable features are searched across all the possible scales to achieve invariance to scale change. Let $I(x,y)$ be an input image, and $G(x,y,\sigma)$ be the 2-D Gaussian function with a variable scale. Then the scale space of an image I could be represented by a function $L(x,y,\sigma)$ which is obtained from the convolution of Gaussian function with the input image:

$$L(x,y,\sigma) = I(x,y) * G(x,y,\sigma) \tag{2.15}$$

where $*$ stands for the convolution operation and Gaussian kernel could be derived by:

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{2.16}$$

Scale space is generated by convolving input image with Gaussian kernel to get more and more blurred images as different scales. Then, the DoG (Difference of Gaussian) pyramid is built by calculating the subtraction of adjacent blurred images as a close approximation to LoG (Laplacian of Gaussian):

$$DoG(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{2.17}$$

After that, each sample pixel in the DoG pyramid is compared with its neighbors at current and adjacent scales to detect local extrema (maxima or minima) in space and scale dimensions.

Since the local extrema detected before are discrete and unstable, the next step is to perform a better fit to the nearby pixels for location, scale and ratio of principal curvatures in order to accurately localize the position and scale of the keypoints.

2.2.2.2 SIFT descriptor

The SIFT detector could find the stable keypoints with corresponding scale level, the next step is computing descriptor for each keypoint.

First, we should assign a consistent orientation to each keypoint based on the information of local image patch to achieve invariance to image rotation. According to the obtained scale level, we select the corresponding blurred image $L(x,y)$ so that the subsequent computations are performed in a scale-invariant manner. Let $g(x,y)$ and $\theta(x,y)$ be the gradient magnitude and orientation of each pixel, respectively. Then we have:

$$g(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2} \tag{2.18}$$

$$\theta(x,y) = \arctan\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \tag{2.19}$$

Then, we determine the orientation of each keypoint by calculating an orientation histogram of its neighbors. Specifically, the orientation histogram is composed of 36 bins covering 360 degrees, and

each neighbor pixel added to the histogram by its Gaussian weighted gradient magnitude. Finally, the peak in the histogram is set to be the dominant direction of local gradients, with this, we could assign an orientation to the corresponding keypoint.

Through the above steps, we obtained the image location, scale level and orientation of each keypoint, the next step is to generate a robust descriptor for it. Focusing on an 8-pixel radius (i.e., a 16x16 window) around a keypoint in the scale level at which it is detected. Then, an 8-bin orientation histogram for each 4x4 region is calculated so that the final descriptor is assigned with the 128-dimensional vector containing the histogram values of 4x4 regions. For further details, see [24].

2.2.2.3 SIFT matching

Matching is the final step for feature correspondences where the descriptors of keypoints are matched together by comparing their vector similarities. The best candidate match for each keypoint is found by identifying the its nearest neighbor with minimal distance compared to all other candidates. There are two types of distance measurements are usually used depending on the vector type of different descriptors, for example, Euclidean distance (L2-Norm) is utilized for real-valued descriptors (SIFT and SURF), and Hamming distance is used for binary-strings descriptor (ORB).

According to [24], for rejecting incorrect matches to achieve robustness, we compare the distance of the closest neighbor with that of the second-closest neighbor. This method is relatively reliable since the correct matches need to have the closest neighbor which is significantly closer than the closest incorrect match. Empirically, the ratio of distance (closest/second-closest) is set to 0.8 as a thresholding value.

Feature correspondences by SIFT could handle the conventional perspective images very well. However, as we can expect from the theory of conventional SIFT explained above, simply applying this 2D local features to the EQR image is unreliable due to the location-dependent distortion. This could be observed in Chapter 4 where the repeatability of feature detection decreases with distortion. Also, the quality of the image could influence on the performance of detection and matching. The proposed approach will be elaborated in section 3.1.1 to address these problems.

2.2.3 Motion estimation and 3D triangulation

After corresponding 2D feature points between images are detected and matched, we could estimate the ego-motion between images and reconstruct the 3D coordinates via these matches as in [34]. In this section, we will describe how to make use of geometry relationship between multi-views in order to recover the camera pose and 3D structure, the main theories are explained mathematically.

2.2.3.1 Epipolar geometry

In general, if there are two views and corresponding features in the same scene, we could obtain several geometry constraints according to the characteristics of camera, the position of 3D scene points and relative positional relationship between cameras. [34] defined epipolar geometry to describe these constraints.

As we can see from Figure 2.6, suppose there are two cameras C_1 and C_2 with the corresponding camera coordinate system F_{C1} and F_{C2}. For simplification, we use subscription index 1 and 2 as substitution for C_1 and C_2 in the following instruction. There is a point P in 3D space, let $P_1 = [X_1, Y_1, Z_1]^T$ and $P_2 = [X_2, Y_2, Z_2]^T$ where P_1 and P_2 are the coordinates of P with respect to F_{C1} and F_{C2} respectively. Then, it is projected on two image planes with $p_1 = [u_1, v_1]^T$ and $p_2 = [u_2, v_2]^T$. According to Eq. (2.5), we have:

$$P_2 = RP_1 + T \tag{2.20}$$

Here, R and T stands for the relative rotation and translation between camera C_1 and C_2. More importantly, the space point P, the image points p_1 and p_2 and the camera optical centers O_{C1} and

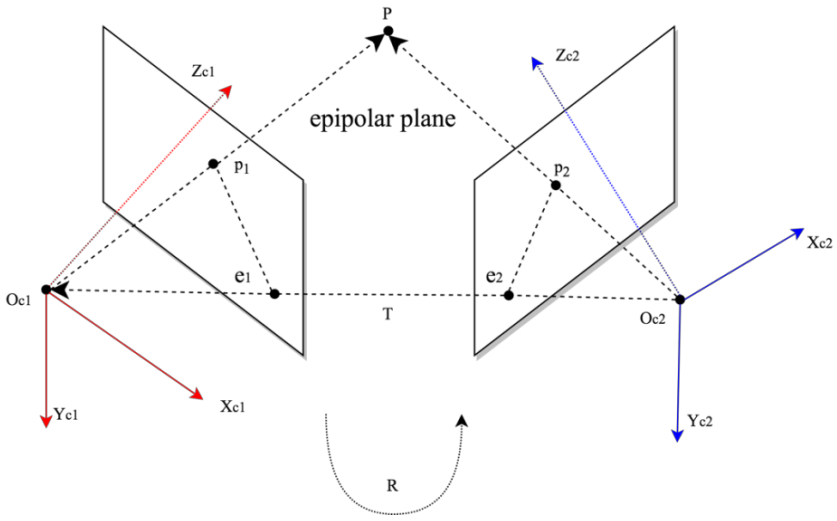


Figure 2.6: Epipolar geometry of conventional pinhole camera

O_{C2} are coplanar. This plane is denoted as epipolar plane. As shown in Figure 2.6, the normal vector of the epipolar plane can be obtained by cross product of $\overrightarrow{O_{C2}O_{C1}}$ (i.e. T) with $\overrightarrow{O_{C2}P}$ (i.e. P_2), and it should be perpendicular to the vector $\overrightarrow{O_{C1}P}$ (i.e. RP_1), so we get:

$$(RP_1)^T \cdot [T]_\times P_2 = 0 \tag{2.21}$$

Eq. (2.21) can be rewritten as:

$$P_2^T \cdot [T]_\times R \cdot P_1 = P_2^T E P_1 = 0 \tag{2.22}$$

where we define $E = [T]_\times R$ as the 3x3 essential matrix. After that, Eq. (2.22), which describes the relationship between two 3D rays, could be further transformed as the relationship between two image points according to Eq. (2.14):

$$\begin{bmatrix} p_2 \\ 1 \end{bmatrix}^T K^{-T} E K^{-1} \begin{bmatrix} p_1 \\ 1 \end{bmatrix} = \begin{bmatrix} p_2 \\ 1 \end{bmatrix}^T F \begin{bmatrix} p_1 \\ 1 \end{bmatrix} = 0 \tag{2.23}$$

where we define $F = K^{-T} E K^{-1}$ as the 3x3 fundamental matrix, K is the intrinsic matrix of the camera which is obtained by calibration. We should know that F is a matrix with 8 degrees of freedom since the scale is unknown.

For calculating fundamental matrix F, we introduce definition of epipolar line which gives another constraint to F: For any point p_1 in the left image, it will be projected to the right image on the corresponding epipolar line is $l_2 = F \begin{bmatrix} p_1 \\ 1 \end{bmatrix}$. Moreover, the epipolar line l_2 should always contain the epipole e_2, so e_2 satisfies $e_2^T F \begin{bmatrix} p_1 \\ 1 \end{bmatrix} = 0$ for any p_1. It means $e_2^T F = 0$, and e_2 is the left null-vector of F which gives the constraint that the rank of F is 2.

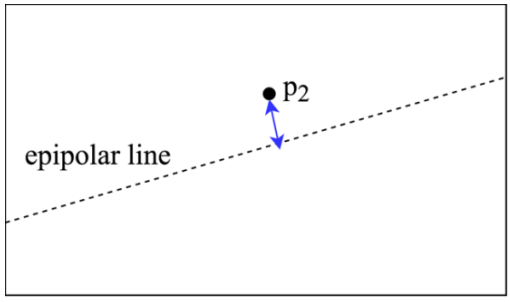


Figure 2.7: Euclidean distance on 2D pixel domain as error function for RANSAC in perspective case

2.2.3.2 RANSAC

RANSAC (Random Sample Consensus) is a classical iterative method proposed by Fischler and Bolles in 1981 [42] to estimate the parameters of a designed mathematical model from a large set of observed data containing outliers. The basic idea of RANSAC is: the observed data set is composed of inliers and noise, a reasonable model should be able to fit all the inliers and reject the those noise at the same time. Thus, we could simultaneously estimate the desired model and remove outliers by applying RANSAC.

In the case of SfM, the elements of matrix F is considered as the model we need to fit with observed data, and the observed data are those corresponding 2D keypoints that we matched before. As a result, we could design a RANSAC based method to compute the fundamental matrix F as well as remove outliers from matched SIFT feature points. The algorithm can be divided into following

steps:

Step 1: Define the computation method associating target model (i.e., fundamental matrix F) with observed data (i.e., feature points). Here, eight-points method [34] is used to calculate F. According to Eq. (2.23), one pair of feature points contributes to one degree of freedom, then 8 correspondences form 8 required equations which are enough to calculate F. Additionally, the previous section explained that the epipolar line introduces another constraint: the rank of F should be 2 (i.e., $detF = 0$). By calculating the SVD (Singular Value Decomposition) of F, we get:

$$F = UDV^T \tag{2.24}$$

where we set the last element of D to zero as D^*, then final F is cleaned up by $F = UD^*V^T$.

Step 2: Randomly sampling 8 pairs of matched keypoints to calculate the model F as described in Step 1.

Step 3: Define an error function L to judge if the other keypoints are satisfied to the calculated model F or not by an assigned thresholding value λ. As explained in previous section, for a keypoint p_1 in the left image, it should be projected to the right image on the corresponding epipolar line l_2 where $l_2 = F\begin{bmatrix} p_1 \\ 1 \end{bmatrix}$. Ideally, if F is a reliable model, then the corresponding point p_2 in the right image should lie on the epipolar line l_2. For simplification, the homogenous coordinates of epipolar line can be represented as $l_2 = [a, b, c]^T$. As shown in Figure 2.7, we define the Euclidean distance between $p_2 = [u_2, v_2]^T$ and l_2 as the error function L:

$$L = \frac{|au_2 + bv_2 + c|}{\sqrt{a^2 + b^2}} \tag{2.25}$$

Empirically, the epipolar error L should be smaller than thresholding $\lambda = 0.7089\ pixel$.

Step 4: Counting the number of inliers from all the other keypoints by calculating corresponding epipolar error L bellowing thresholding λ.

Step 5: Repeating step 2-4 to find the most reliable fundamental matrix F.

2.2.3.3 Estimation of poses and 3D points

After fundamental matrix F is obtained, we can compute the essential matrix E by Eq. (2.23) where the intrinsic matrix K is known. The next step is recovering rotation matrix R and translation matrix T from E. According to the constraint of epipole line explained in previous section, we know that $e_2^T E = 0$ where $T = e_2$, so the translation vector T is the left-null vector of E. By computing the SVD of E that:

$$E = U\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T \tag{2.26}$$

where T could be obtained from the third column of U since T is assumed as a unit-vector without scale information. There are four possible combinations of translation matrix and rotation matrix since

the translation $T = \pm u_3$ and $R = UYV^T$ or $R = UY^TV^T$, which results in 4 possible projection matrices $M = [R\ T]$. To resolve the 4-fold ambiguity problem, we first triangulate the 3D coordinates of P by all of the possible M, then check the depth value Z of P which should be positive so that pin-hole camera can observe. Then the correct combination of R and T can be obtained by ensuring the 3D point P is in front of two cameras. The 3D triangulation is derived by:

$$\begin{bmatrix} \begin{bmatrix} p_1 \\ 1 \end{bmatrix}_\times M_1 \\ \begin{bmatrix} p_2 \\ 1 \end{bmatrix}_\times M_2 \end{bmatrix} \begin{bmatrix} P \\ 1 \end{bmatrix} = A \begin{bmatrix} P \\ 1 \end{bmatrix} = 0 \tag{2.27}$$

Generally, the unknown 3D coordinate P could be obtained by a linear least squares method which compute the SVD of A.

2.2.4 Bundle Adjustment

After the rotation matrix R, translation matrix T and 3D coordinates of feature points are calculated by the methods described before, the next step is applying Bundle Adjustment (BA) to locally and globally optimize these parameters of camera poses and structure points. It is necessary to apply BA since we care more about optimal reconstruction under the assumptions regarding the noise pertaining to the observed images. BA was originally conceived in the field of photogrammetry during 1950s, which has been widely used by computer vision researchers during last two decades [34, 43, 44].

Unlike linear least squares problem which has a global optimal solution can be solved by simply calculating the normal equation, BA is a nonlinear least squares problem leading to many local minima. Therefore, the camera poses and structure points obtained before could be a good initialization in BA. Here, we introduce the derivation of how to solve this problem and the corresponding error function usually used in the case of conventional SfM with perspective images.

2.2.4.1 Nonlinear least squares

We define x as the parameters to be optimized, b as the value of observed data, $f(x)$ as the value of predicted data from x. Then the nonlinear least squares problem can be regarded as:

$$\min_x \|f(x) - b\|^2 = \min_x (f(x) - b)^T (f(x) - b) = \min_x f(x)^T f(x) - 2b^T f(x) \tag{2.28}$$

here we define L as the objective function to be minimized, then we have:

$$L = f(x)^T f(x) - 2b^T f(x) \tag{2.29}$$

then we could obtain a condition for the solution by calculating the partial derivative of L with respect to x and set it to 0:

$$\frac{\partial L}{\partial x} = 2\frac{\partial f(x)^T}{\partial x} f(x) - 2\frac{\partial f(x)^T}{\partial x} b = 0 \tag{2.30}$$

where we introduce the Jacobian matrix J as:

$$J = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{2.31}$$

Next, we take the Taylor expansion around x as:

$$f(x + \Delta x) \approx f(x) + \frac{\partial f(x)}{\partial x}\Delta x \tag{2.32}$$

Then substitute it into the Eq. (2.30), we get:

$$\frac{\partial L}{\partial x} = 2\frac{\partial f(x)^T}{\partial x}\left(f(x) + \frac{\partial f(x)}{\partial x}\Delta x\right) - 2\frac{\partial f(x)^T}{\partial x}b = 0 \tag{2.33}$$

which could be rewritten and organized by Jacobian matrix J as:

$$J^T J\Delta x = J^T\big(b - f(x)\big) \tag{2.34}$$

Note that, Δx in here can be regarded as direction of the next step for iteration to converge to the optimal solution as $x_{i+1} = x_i + \Delta x$, where:

$$\Delta x = (J^T J)^{-1} J^T\big(b - f(x)\big) \tag{2.35}$$

2.2.4.2 Error function

As we already know the process for optimize the parameters x as explained above, the next step is designing an error function $E = \big||f(x) - b|\big|^2$ to describe the relationship between observed value b and predicted value $f(x)$ of parameters x.

In the case of conventional SfM with perspective images, the error function is usually defined as a reprojection error on 2D pixel domain, as shown in Figure 2.8. The reprojection error is defined as the Euclidean distance between the observed keypoint $p' = [u, v]^T$ and predicted point $p = [u', v']^T$. p' stands for the keypoint originally matched by SIFT, while p represents the reprojection of reconstructed 3D point $P = [X_w, Y_w, Z_w]^T$ by Eq. (2.14) that:

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = K[R\ T] \begin{bmatrix} P \\ 1 \end{bmatrix} \tag{2.36}$$

Here K is the intrinsic matrix which is already known, so we can consider it as a constant value. R T and P are the rotation matrix, translation matrix and 3D points respectively, which is the target parameters that we want to optimize by BA. Thus, from Eq. (2.36) we can define u and v as a function:

$$u = \frac{x(R,T,P)}{z(R,T,P)}, v = \frac{y(R,T,P)}{z(R,T,P)} \tag{2.37}$$

Then our target of minimization can be regarded as:

$$\min_{R,T,P} \|p - p'\|^2 = \min_{R,T,P} \left\| \begin{bmatrix} x(R,T,P)/z(R,T,P) \\ y(R,T,P)/z(R,T,P) \end{bmatrix} - \begin{bmatrix} u' \\ v' \end{bmatrix} \right\|^2 \tag{2.38}$$

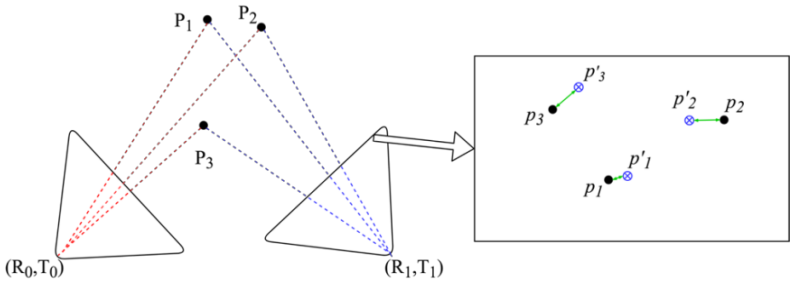


Figure 2.8: Reprojection error in BA defined on 2D pixel domain in perspective case

2.3 Mask R-CNN

Our target of applying object detection is to output bounding boxes and labels for moving objects (e.g., pedestrians and vehicles) which will bother the performance of motion estimation and structure reconstruction in SfM, and the bounding box represent the area we should eliminate when performing feature detection and matching.

Deep CNN based methods has dominated the task of object detection recently, the performance of object detection with conventional methods (e.g., HOG+SVM, DPM[45]) are outperformed by Deep CNN based methods. The main reason of why deep learning can outperform all the traditional methods is that: the former can fit a model which is trained to extract high-level features while the latter usually construct low-level features from using prior knowledge about problem domain. Generally, Deep CNN based object detection approaches could be divided into two groups: the first group can be regarded as two stage methods, which first generate several proposals, and then perform the classification through CNN, such as a series of works (R-CNN[8], Faster R-CNN[9] and Mask R-CNN[10]) proposed by Facebook Research; the second group can be regarded as one stage methods which pay more attention to the computation efficiency to achieve real-time application such as SSD[11], YOLO[12].

Mask R-CNN is the state-of-the-art method for both objects detection and instance segmentation on conventional perspective images, which extends the object detection task of Faster R-CNN by adding a branch of Fully Convolutional Network (FCN)[46] to parallelly provide pixel-level segmentation. Specifically, the backbone architecture combining ResNet-101[47] with Feature Pyramid Network (FPN)[48] is used to achieve strong scale-invariant feature extraction. Then, regions of interest produced by Region Proposal Network (RPN) mapped on extracted pyramid feature maps are passed through 3 branches for classification, bounding box regression and binary mask prediction respectively. The author demonstrated that this multi-task model can improve performance of

detection comparing with Faster R-CNN which only has two branches (classification and bounding box regression). Besides, ROIAlign is introduced for producing more accurate bounding boxes and masks. An example of the detection output of Mask R-CNN is shown in Figure 2.9.

Figure 2.9: An example output of Mask R-CNN trained on COCO dataset

2.3.1 Feature Pyramid Network

For detecting objects at different scale, Mask R-CNN combines ResNet-101 and Feature Pyramid Network (FPN) as the backbone architecture to extract strong scale-invariant features. To construct the pyramid, FPN involves a bottom-up pathway, a top-down pathway, and lateral connections.

The bottom-up pathway is performed by any classical backbone ConvNet, in which feature maps are downscaled by Conv layer and pooling layer again and again so that the feature map of deeper layer is stronger than it of lower layer. Specifically, using ResNet-101 as the backbone ConvNet, we define one pyramid level for each stage in ResNet-101 and choose the last layer of each stage as the reference set of feature maps. And the outputs of the stages are denoted as $\{C_2, C_3, C_4, C_5\}$.

The top-down pathway is to obtain strong features in lower layer. As we know, the lower layer corresponds to higher resolution but weaker feature maps. Thus, the top-down pathway upsamples the deeper layer feature map which is spatially coarser but semantically stronger, subsequently, it is enhanced with corresponding features from the bottom-up pathway by lateral connections. This process is iterated until the finest resolution feature map is generated. And the final set of constructed feature maps is denoted as $\{P_2, P_3, P_4, P_5\}$, corresponding to $\{C_2, C_3, C_4, C_5\}$.

2.3.2 Region Proposal Network

Region Proposal Network (RPN) was first proposed in Faster R-CNN to generate region proposals directly from the extracted features of the backbone ConvNet. Therefore, RPN and other branches

(classification, bounding box regression and binary mask prediction) can share the feature maps, which dramatically reduce the computation cost comparing with other methods for generating region proposals.

The basic idea of RPN is sliding a small network over the feature map output of the last shared conv layer. And this small network is connected to an $n \times n$ window of the feature map. To achieve the ability of detecting objects with different aspect ratios, k anchors (region proposals) at each sliding window location are predicted. Specifically, 3 scales and 3 aspect ratios are used which generate $k = 9$ anchors at each sliding position. Then, each anchor is mapped to a 256-dimensional vector and fed into 2 FC (Fully-connected) layers: one is called box-regression layer for predicting a rough bounding box for the region; the other one is called box-classification layer for classify this region contains possible object or not.

2.3.3 Problem in omni-directional case

The performance of current classification based Convolutional Neural Network (CNN) architecture on EQR image is not reliable because of two reasons:

1. For feature extraction, current CNN is constructed based on conventional Conv Layers with rectangle window filters. Because of downscaling feature with Pooling Layers, it can achieve invariance to scale, slight translation and rotation. However, the underlying projection model of CNN is perspective model, which cannot achieve the non-uniform distortion such as the location-dependent distortion introduced by the $360°$ EQR image.

2. A large scale labeled image dataset and data augmentation could make current CNN achieve invariance to large rotation, affine transformation, illumination change, etc. However, it is awkward that those current existing image datasets COCO[49] ImageNet[50] for training are perspective images as shown in Figure 2.10(c).

(a) Visualized filters of first layer of AlexNet[51]

(b) Visualized filters of first layer of ResNet-101[51]

(c) Labeled perspective images of COCO dataset[49]

Figure 2.10: Intuition of the problems of current CNN and dataset

Figure 2.10(a,b) illustrates the filters of Conv layer of AlexNet[52] and ResNet-101. We can see that even for the first Conv layer, it can only extract orientated edges, which is apparently not reliable for extracting features in EQR image especially for the large distortion near north pole and south pole. Figure 2.11 shows the result when we directly feed EQR image into Mask R-CNN trained on COCO dataset.

Figure 2.11: An example output of directly feeding EQR image into Mask R-CNN trained on COCO dataset, the performance is not good due to the non-uniform distortion (e.g., the book and TV are not detected)

Chapter 3 Proposed Approach

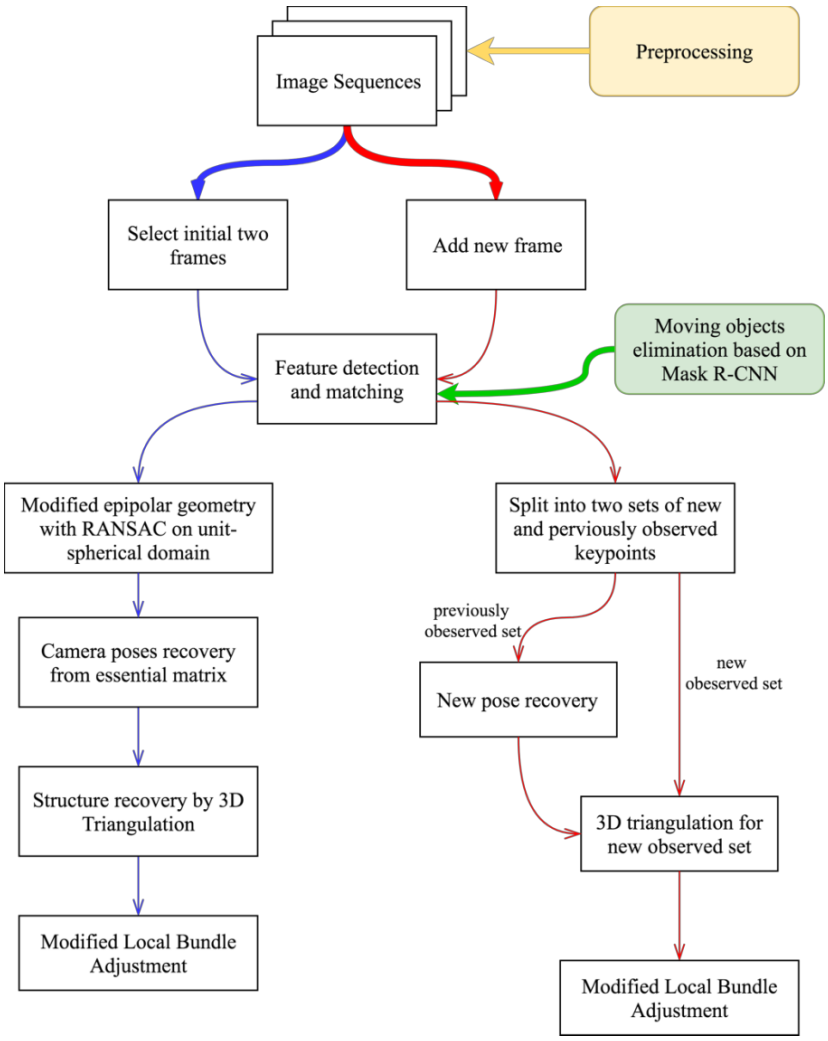


Figure 3.1: Proposed SfM pipeline for Omni-directional images

This chapter presents our proposed 3D reconstruction system with omni-directional images by combining modification of conventional SfM with moving objects detection based on Mask R-CNN, in which: we address the distortion problems of EQR images and make conventional SfM suitable to unit-spherical projection model described in section 2.1.

We start by giving a description of the proposed pipeline step by step. Then, we explain the details of the theory associated with each step in this pipeline, including how to make conventional SfM

suitable to non-uniform distorted EQR image and how to detect moving objects in this kind of image. Here, for object detection, we present two proposals based on Mask R-CNN to detect objects in various situations, which can be used not only in the proposed system for structure reconstruction but also in other computer vision applications. Additionally, we describe the way of plugging moving objects elimination into SfM.

As shown in Figure 3.1, the general pipeline can be abstracted as follows:

1. Gather the EQR images captured in the same scene together and perform preprocessing to address problems of distortion. The preprocessed images set are defined as I.

2. Select initial two frames i_1 and i_2 in set I, do feature detection and matching to obtain the corresponding 2D keypoints set S_{12}. Here, moving object detection is performed on both i_1 and i_2 to eliminating the bothering area in order to filter outliers away from S_{12}.

3. Estimate essential matrix E by RANSAC where outliers of keypoints are removed at the same time. After that, relative camera pose R_1 and T_1 matrices are recovered from E, and 3D points are reconstructed by 3D triangulation. Finally, all of the parameters are fed into BA for local nonlinear refinement.

4. Add the new image i_3 and do feature detection and matching with i_2 to obtain corresponding keypoints set S_{23}, then splitting S_{23} into two subsets as S_{old} (those keypoints observed before) and S_{new}(those new observed keypoints). Again, outliers are filtered away by moving object elimination.

5. Do PnP to directly recover the new pose R_2 and T_2 by S_{old} in which 3D coordinates have already been reconstructed in step 3. We then reconstruct the 3D points of S_{new} by 3D triangulation. Finally, all of the parameters are fed into BA for global nonlinear refinement.

6. Repeat steps 4-5 until all the images in set I are involved.

3.1 Structure from Motion by EQR images

In this section we explain our proposed SfM system in three parts corresponding to the pipeline in Figure 3.1: Preprocessing as the yellow path; Two-view SfM as the blue path; Multi-view SfM as the red path.

3.1.1 Preprocessing for feature correspondences

As mentioned in section 2.1.2, the EQR image generated by Ricoh Theta will introduce two problems:

non-uniform distortion and low quality at certain area. They would dramatically affect the performance of conventional feature detection and matching methods as we discussed in section 2.2.2, thus, in this thesis, we introduce two preprocessing steps performing on raw EQR images.

3.1.1.1 Cubic mapping

Cubic mapping has recently been utilized in preprocessing for compression of EQR images. It projects the EQR image to six tangent patches (each patch corresponds to $90°$ FoV) which are in a similar manner with images captured by perspective pin-hole camera in different point of view. Here we utilize cubic mapping to tackle the distortion problem by converting our raw EQR images into cube map for improving the reliability of SIFT feature correspondences. For implementing cubic mapping in our system, the process is explained as follows:

Suppose the raw EQR image I_e with resolution $width \times height$, then we define our target cubic image I_c with resolution $width \times \frac{3}{2} height$. Since artifacts will be produced by simply projecting each pixel of I_e to I_c, we adopt an inverse way by searching the corresponding location of each pixel in I_c on I_e. Specifically, we first calculate the corresponding 3D coordinates $P(X, Y, Z)^T$ of each pixel $p_c(u_c, v_c)$ on cubic image I_c, then we have:

$$\begin{cases} r = \sqrt{X^2 + Z^2} \\ \theta = arctan2(r, Y) \\ \phi = arctan2(X, Z) \end{cases} \tag{3.1}$$

Where θ and ϕ is the latitude and longitude as explained in section 2.1.1, then we get the corresponding floating coordinates of $p_e(u_e, v_e)$ on raw EQR image I_e by Eq. (2.3). As shown in Figure 3.2, we only know the RGB value of discrete pixel in I_e, and the floating pixel $p_e(u_e, v_e)$ is locating on the position surrounding with four neighboring discrete pixels $p_1(u_1, v_1), p_2(u_2, v_2), p_3(u_3, v_3)$ and $p_4(u_4, v_4)$. For obtaining the RGB value of p_e, we use the bilinear interpolation.

By calculating the weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ corresponding to the RGB values i_1, i_2, i_3, i_4 of 4 neighbor pixels p_1, p_2, p_3, p_4, the interpolated RGB value i_e can be obtained by:

$$i_e = \lambda_1 i_1 + \lambda_2 i_2 + \lambda_3 i_3 + \lambda_4 i_4 \tag{3.2}$$

The example cubic map from a raw EQR image is shown in Figure 3.3.

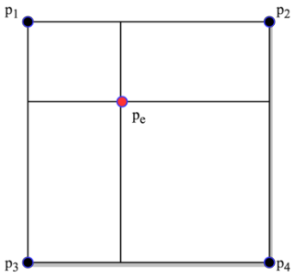


Figure 3.2: Bilinear interpolation used in cubic mapping

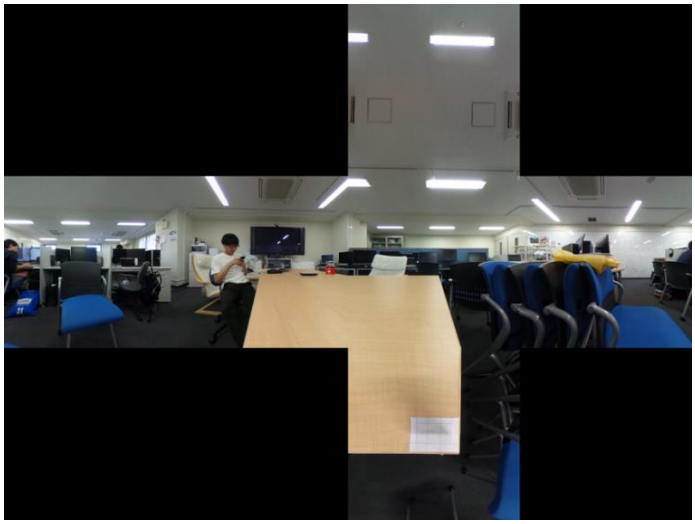


Figure 3.3: An example of proposed cubic mapping to address the problem of distortion

3.1.1.2 Merging from pure rotation

For addressing the problem of low quality area introduced by interpolation and stitching when EQR image is generated by OD camera, we design a system as illustrated in Figure 3.4. Each time we obtain two images I_1 and I_2 with pure rotation at one spot and compute the rotated angles by detecting the most reliable set of pairs of keypoints from SIFT, this can be realized by raising the ratio of distance between two nearest pairs of keypoints described in [24]. Here we select 5 most reliable pairs of feature points as a reference. Specifically, we define a pair of keypoints as $p_1(u_1, v_1)$ and $p_2(u_2, v_2)$. For those two frames with pure rotation, the difference in locations of p_1 and p_2 only occur with the horizontal u axis. We can simply use the pixel distance $du = u_2 - u_1$ to rotate I_2 to generate I_{2rot} and merge it with I_1, the merged EQR image is then mapped to a cubic image I_c as described in section 3.1.1.1. The example of merged image is shown in Figure 3.4.

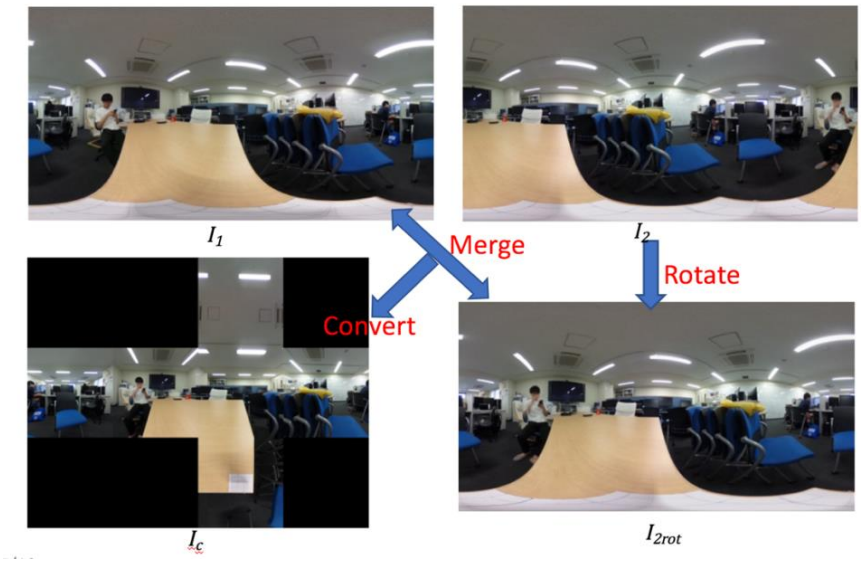
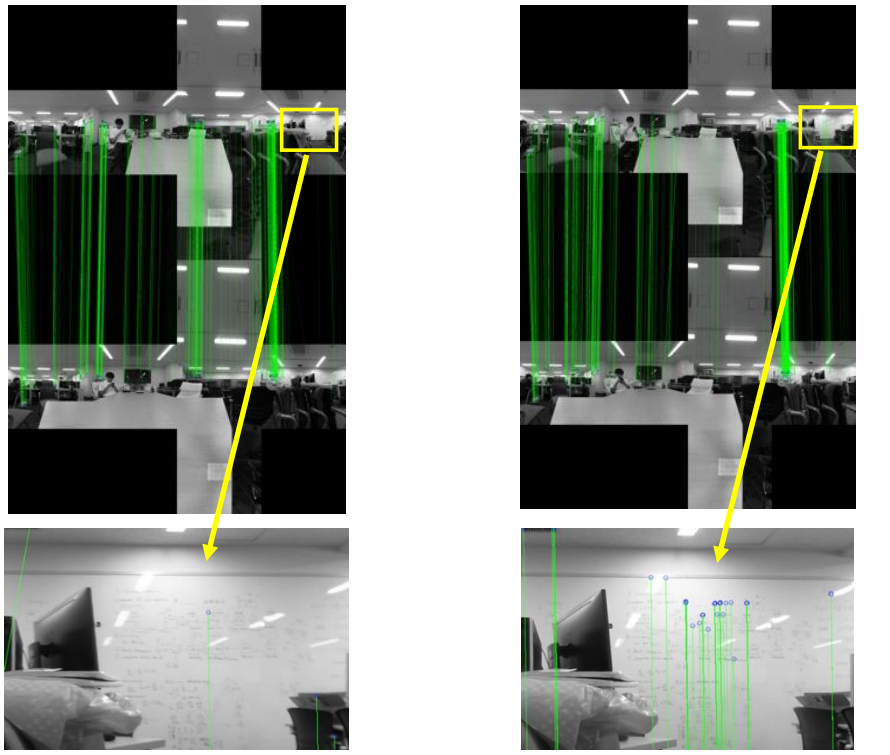


Figure 3.4: Proposed merging step to address the problem of low quality at certain area



Matched feature points by SIFT without merging

Matched feature points by SIFT With merging

Figure 3.5: Intuition of improvement on repeatability of SIFT by preprocessing.

3.1.1.3 Feature correspondences

After the preprocessing steps described above, we apply conventional SIFT algorithm for feature correspondences, the detail of SIFT algorithm are described in section 2.2.2 including keypoints detection, description and matching. Figure 3.5 shows the performance of SIFT with preprocessing steps. The further evaluation in Chapter 4 shows that by tackling the un-uniform distortion especially on the area near north pole and south pole of the raw EQR image and addressing the low quality at certain area can increase the repeatability of conventional SIFT method.

3.1.2 Two-view Structure from Motion

From initially selected two frames, we now get a set of pairs of matched keypoints by performing feature correspondences with preprocessing steps described in previous section. Next, we need to estimate the relative pose R and T from these corresponding 2D keypoints. As described in section 2.2.3, in the situation of perspective images, this motion estimation problem can be solved by calculating fundamental matrix F from epipolar constraint with Eq. (2.23). However, this is not suitable to EQR images since we cannot describe the unit-spherical projection model (introduced in section 2.1.1) with a linear intrinsic matrix K. In unit-spherical domain, the epipolar geometry could be illustrated as Figure 3.6. According to Eq. (2.3) and (2.4), we get:

$$p_1' = [x_1, y_1, z_1]^T = [cos\theta_1 cos\phi_1, sin\theta_1, cos\theta_1 sin\phi_1]^T \quad (3.3)$$

$$p_2' = [x_2, y_2, z_2]^T = [cos\theta_2 cos\phi_2, sin\theta_2, cos\theta_2 sin\phi_2]^T \quad (3.4)$$

Where p_1' and p_2' are the unit-spherical coordinates corresponding to a pair of keypoints $p_1(u_1, v_1)$ and $p_2(u_2, v_2)$. According to Eq. (2.22), the epipolar constraint in unit-sphere domain could be regarded as:

$$p_2'^T E p_1' = 0 \quad (3.5)$$

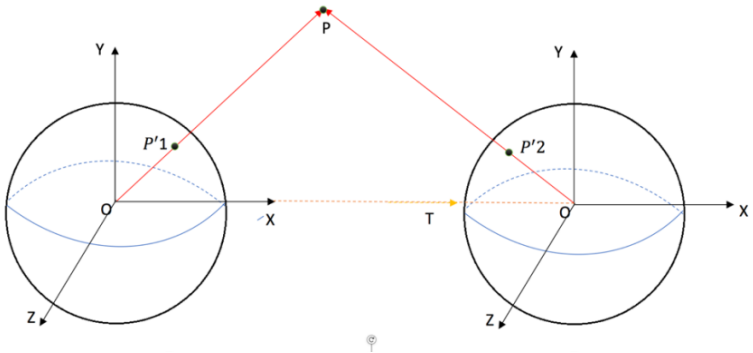


Figure 3.6: Epipolar geometry in unit-spherical domain

3.1.2.1 RANSAC on unit-sphere domain

Different with RANSAC for computing fundamental matrix F in the situation of perspective images. Here, we define the RANSAC in EQR images as directly computing essential matrix E according to Eq. (3.5). The process of the RANSAC algorithm explained in section 2.2.3.2 can still be used in here, but the error function L should be redesigned since we cannot define L as Euclidean distance on 2D pixel domain as described in Eq. (2.25).

As we know from the concept of epipolar plane presented in section 2.2.3.1, Ep_1' can be regarded as the normal vector of the epipolar plane, and ideally, p_2' as a unit vector should lie on the epipolar plane, which means the dot product of Ep_1' and p_2' should be 0, that is, the angle between Ep_1' and p_2' should be $90°$. Thus, we define the error function as angular error:

$$L = \cos^{-1}(p_2'^T \cdot Ep_1') \tag{3.6}$$

Where we should also set the thresholding value λ, in our experiment, $\lambda = 85°$ could be a relatively reliable value, and those pairs of keypoints with $L > \lambda$ are counted as inliers. The results of the matched feature points after removing outliers by RANSAC with epipolar constraint are shown in Chapter 4.

3.1.2.2 Pose estimation in EQR images

After RANSAC, we get the essential matrix E, the next step is recovering R and T by the algorithm described in section 2.2.3.3. Same as in the situation of perspective images, there are four possible combinations of R and T. Note that we cannot tackle this ambiguity by simply judging the depth value of reconstructed 3D point. In conventional SfM, this depth value should be positive since all of the points observed by perspective image should be consistently in front of the camera. However, this constraint is not suitable to OD camera which covers 360 FoV. For solving this ambiguity in spherical model, we propose a different method. Suppose we get 4 possible combination of R_i and T_i, where $i \in [1,2,3,4]$. From each pair of them, we obtain the reconstructed 3D point $P_w = [X_w, Y_w, Z_w]^T$ by 3D triangulation as:

$$\begin{bmatrix} [p_1'^T]_\times [I \;\; 0] \\ [p_2'^T]_\times [R_i \; T_i] \end{bmatrix} \begin{bmatrix} P_w \\ 1 \end{bmatrix} = A \begin{bmatrix} P_w \\ 1 \end{bmatrix} = 0 \tag{3.7}$$

where we set the coordinates reference of first camera as the world coordinates. So the rotation matrix and translation matrix for first camera are identity matrix I and 0, respectively. Then we get the corresponding coordinates of P_w in second camera as P_{c2} which is presented as:

$$P_{c2}^T = [R_i \; T_i] \begin{bmatrix} P_w \\ 1 \end{bmatrix} \tag{3.8}$$

According to section 2.1.1, we know that ideally:

$$p_2'^T = \frac{P_{c2}^T}{\|P_{c2}^T\|} \tag{3.9}$$

Thus, for each combination R_i and T_i, we calculate all the 3D points by Eq. (3.7) and then projected to the coordinates reference of the second camera by Eq. (3.8). After that, we check the angle α between p_2' and P_{c2} as:

$$\alpha = \cos^{-1}(p_2'^T \cdot P_{c2}^T) \tag{3.10}$$

Here, we set a thresholding value $\lambda = 5°$, which is shown to be a reliable value in our experiment. In a certain combination of R_i and T_i, if a pair of keypoints associating with $\alpha < \lambda$, then we vote once for this combination. As a result, the correct R_i and T_i should have the most votes. Besides, the benefit of this procedure is that we can remove those outliers which satisfies the epipolar constraint in RANSAC (i.e., reprojected ray lie on the epipolar plane) but not satisfies $\alpha < 5°$. The evaluation and results of the matched points after recovering camera pose are shown in Chapter 4.

3.1.2.3 Bundle Adjustment for EQR images

As described in section 2.2.4, Bundle Adjustment is a method to perform nonlinear refinement for optimizing the parameters of camera poses and reconstructed 3D points. In conventional perspective image case, according to Eq. (2.38), the error function is designed on 2D pixel domain as the Euclidean distance between observed keypoints (the original keypoints obtained by SIFT algorithm) and corresponding projected pixels from reconstructed 3D points. However, this error function is not reliable for EQR image case due to the location-dependent non-uniform distortion. Here, we introduce an error function on 3D unit-spherical domain to make BA suitable to EQR images.

Suppose we have $P_c = [X_c, Y_c, Z_c]^T$ as the coordinates of reconstructed 3D point P with respect to the coordinate reference of camera C. According to Eq. (3.8), we have the relationship:

$$P_c^T = [R \;\; T] \begin{bmatrix} P \\ 1 \end{bmatrix} \tag{3.11}$$

Then calculate the normalized coordinates of P_c as $p = [x, y, z]$, where:

$$p = \frac{P_c}{\|P_c\|} \tag{3.12}$$

Meanwhile, we know the unit-spherical coordinates of detected keypoint as $p' = [x', y', z']^T$ which can be calculated from Eq. (2.3) and (2.4). Thus, the error function of Eq. (2.38) can be rewritten as:

$$\min_{R,T,P} \|p - p'\|^2 = \min_{R,T,P} \left\| \begin{bmatrix} x(R,T,P) \\ y(R,T,P) \\ z(R,T,P) \end{bmatrix} - \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \right\|^2 \tag{3.13}$$

3.1.3 Multi-view Structure from Motion

The process of two-view SfM can reconstruct relative pose R_1, T_1 and a set of 3D coordinates of

matched 2D keypoints S_1 by the steps described above. When we add a new frame into the structure, first detect and match a set of feature points S_2 between new frame and previous one, then split S_2 into two subsets: first set S_2^{old}contains those keypoints observed before; second set S_2^{new}contains new observed keypoints. This can be considered as a perspective-n-point (PnP) problem [34] to estimate the new camera pose relative to the current estimated structure. Here we propose the modification on conventional PnP by utilizing unit-spherical coordinates.

Let $M = [R_2\ T_2]$ as the new camera pose we want to estimate. $P_w = [X_w, Y_w, Z_w, 1]$ is homogenous coordinate of the reconstructed 3D point in current structure. $p = [x, y, z]^T$ is the unit-spherical coordinates corresponding to the detected keypoint in S_2^{old}. According to Eq. (3.11) and (3.12), we have:

$$\lambda p = MP_w \tag{3.14}$$

$$\lambda \begin{bmatrix} x \\ y \\ z \end{bmatrix} = MP_w \tag{3.15}$$

Where λ represents an unknown scale. Then Eq. (3.15) could be rewritten as an equation with cross product:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\times} MP_w = 0 \tag{3.16}$$

Where $\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\times}$ is a skew-symmetric matrix which can be written as a 3x3 matrix with rank=2:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\times} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \tag{3.17}$$

Combine Eq. (3.16) and (3.17), we have:

$$\begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} P_w^T & 0_{1\times4} & 0_{1\times4} \\ 0_{1\times4} & P_w^T & 0_{1\times4} \\ 0_{1\times4} & 0_{1\times4} & P_w^T \end{bmatrix} \begin{bmatrix} M_1^T \\ M_2^T \\ M_3^T \end{bmatrix} = 0 \tag{3.18}$$

$$A \begin{bmatrix} M_1^T \\ M_2^T \\ M_3^T \end{bmatrix} = 0 \tag{3.19}$$

According to Eq. (3.18) and (3.19), we have a 3x12 matrix A with rank=2 which contributes to 2 degrees of freedom in a 12x1 unknown matrix $\begin{bmatrix} M_1^T \\ M_2^T \\ M_3^T \end{bmatrix}$, thus, for computing the elements of M, we need 6 pairs of corresponding points. Eq. (3.19) is a linear least squares problem which can be simply solved by SVD.

After we got the camera pose M, the next step is reconstructing new 3D points corresponding to S_2^{new} by 3D triangulation described in Eq. (3.7). Finally, we perform the global BA for all the camera

poses and 3D points.

3.2 Moving objects detection in EQR images

To improve the performance of object detection in EQR image, we propose two approaches based on preprocessing steps. The basic idea is transforming them into perspective-like images. After that, we combine the moving objects detection with proposed SfM system by eliminating the pixel area of detected pedestrians and vehicles to remove the outliers matched by SIFT.

3.2.1 Proposal 1

As shown in Figure 3.7, in pre-processing stage, we apply cubic mapping same as in section 3.1.1.1 to obtain a cubic image which project the EQR image to six perspective patches, which then be fed into Mask R-CNN at inference phase. Therefore, we can get the output cubic image with masks, bounding boxes, class labels, confidences of detected objects. The post-processing step is remapping it to EQR projection with same size as input image.

We test on 10 raw images captured by Ricoh Theta with different position and viewpoints, Figure 3.8 shows the performance of baseline (directly feed raw EQR image into Mask R-CNN) and proposal 1 intuitively. By adjusting the thresholding value λ, which stands for the lower bounding of the confidence (probabilities) for detecting objects. We get the precision-recall curve as Figure 3.9.

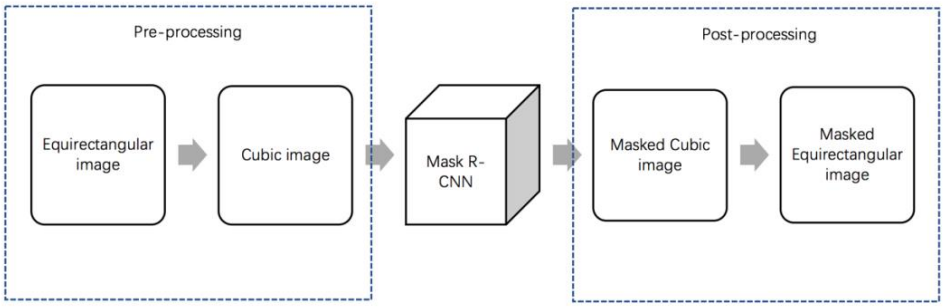


Figure 3.7: Proposal 1 for object detection on EQR image

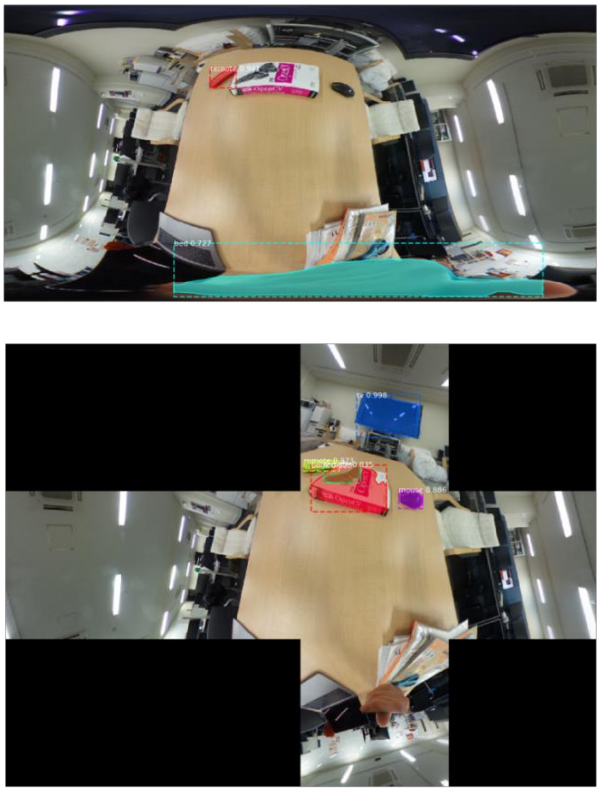


Figure 3.8: Intuition of object detection by baseline and proposal 1

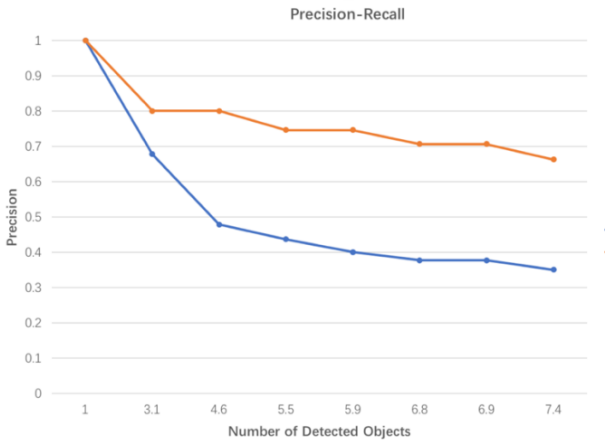


Figure 3.9: Performance of object detection by baseline and proposal 1 (red line represents the precision-recall curve for proposal 1, and blue line for baseline)

3.2.2 Proposal 2

The limitation of proposal 1 is the performance would be not reliable when there exists object projected across the border area of the six faces of cubic map as shown in Figure 3.10. Note that, the shape of the book, which is projected on the yellow circled area, is distorted by cubic mapping.

Figure 3.10: Limitation of proposal 1

To tackle the limitation of proposal 1, we propose a method which repeatedly generate tangent planes as the same manner with the projected six faces in cubic mapping. As shown in Figure 3.11, the process of our proposal is as follows:

Step 1: Sampling a set of center points as candidates. Figure 3.12 shows the typical characteristic of EQR image that the shape and size is location-dependent when projecting an object to this longitude-latitude image. So, it can be utilized to design our sampling strategy for reducing redundant information. The detail of our sampling will be presented later.

Step 2: Projecting EQR image on the unit-sphere surface. This converting process are described in section 2.1.

Step 3: Consider the center of the sphere as the optical center of a virtual perspective camera with narrow FoV, then direct the camera to the sampled center points obtained from step 1.

Step 4: Projecting surrounding patch of each sampled center point to a tangent plane with the projection model of designed virtual perspective camera. The process of tangent plane projection is as

the same manner with cubic mapping explained before, that is, searching the interpolated RGB value on EQR image for each pixel of target tangent plane.

Step 5: Feeding all the tangent plane candidates into Mask R-CNN.

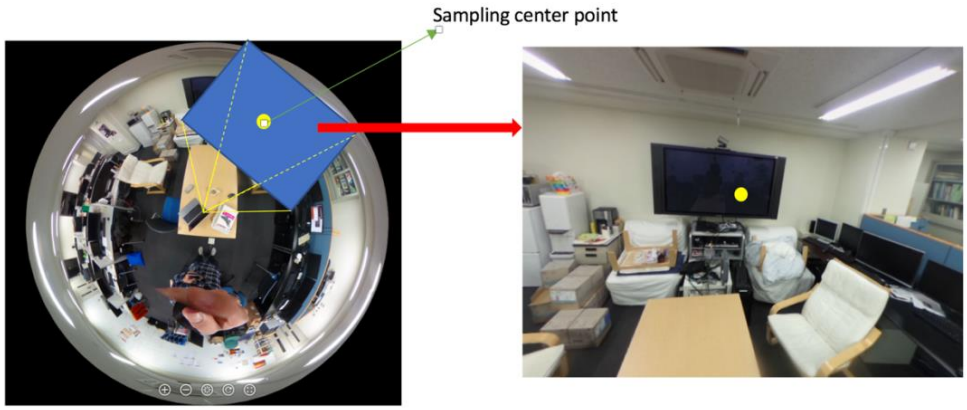


Figure 3.11: Intuition of proposal 2

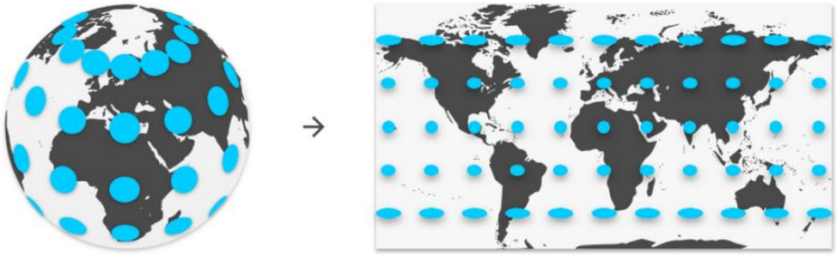


Figure 3.12: Sampling strategy based on location-dependent projection of EQR image

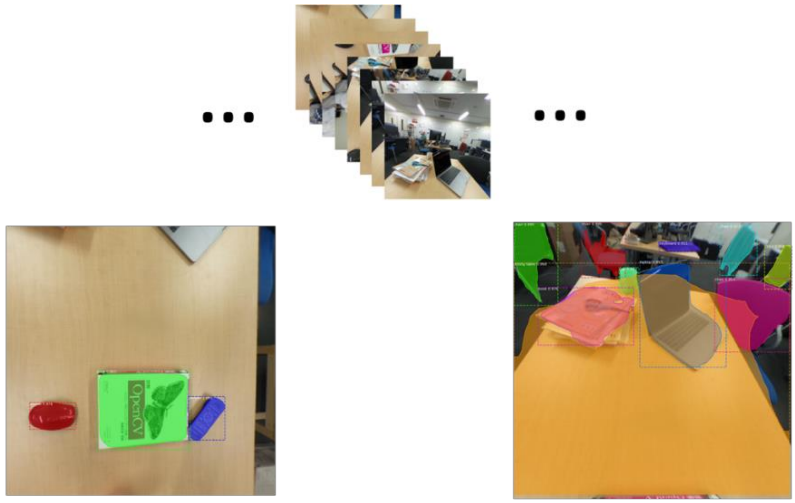


Figure 3.13: Performance of object detection by Proposal 2

In our experiment, the EQR image with resolution 2048x1024 is used. We define the sampling center point on unit-sphere domain as $p(\theta, \phi)$, where θ and ϕ represent the latitude and longitude respectively. For different intervals along with θ, we sample a set of center points p with $\{\theta = \pm 90°, \phi = 0°\}$, $\{\theta = \pm 60°, \phi = 0°, 120°, 240°\}$, $\{\theta = \pm 40°, \phi = 0°, 40°, 80°, 120°, \dots, 320°\}$, and $\{\theta = 0° and \pm 20°, \phi = 0°, 20°, 40°, 60°, \dots, 340°\}$. We set the FoV of virtual perspective camera as $90°$, and the resolution of candidate tangent plane as 1000x1000. The performance of object detection by proposal 2 is shown in Figure 3.13. By comparing with the results by proposal 1 in Figure 3.10, we can see that the mouse, book, and remote which are projected across the border area in proposal 1 can be well detected by proposal 2.

Chapter 4 Experiments and Results

In this chapter, we evaluate the important steps of proposed SfM system described in Chapter 3 including feature correspondences with calculating the repeatability, Two-view SfM with the experiments for depth estimation, and Multi-view SfM with evaluating the reconstructed structure and camera poses.

The proposed SfM system is implemented in Python from the scratch by implementing the algorithm in a naive manner without optimization, thus we cannot ensure the efficiency of our program. Note that, there are two parts we utilized from open-source library: SIFT provided by a third-party package in OpenCV [53], Mask R-CNN implemented by [54].

We conduct our experiments on two groups of EQR images: first, real world images captured in different scene with different direction, the EQR image resolution is 2048x1024 produced by Ricoh theta; second, synthetic images with resolution 2400x1200 rendered by Blender. Object detection model is trained on COCO dataset.

4.1 Feature correspondences

In this experiment. We compare the feature repeatability in certain three steps associated with feature correspondences to evaluate the performance of preprocessing, here we name it as CubicSIFT for EQR image. Specifically, we compare the repeatability in three stages: after original SIFT matching, after our RANSAC (remove outliers which are not satisfied to epipolar plane), and after our pose estimation (remove the remaining outliers which satisfied to epipolar plane). The detail of these steps is described in Chapter 3. The repeatability after each step in our experiment is defined as:

$$\alpha = \frac{\#\ of\ matched\ keypoints\ in\ each\ stage}{\#\ of\ detected\ keypoints} = \frac{k_i}{K} \tag{4.1}$$

Where K is a constant value which is set manually, it stands for the upper bound of the number of detected feature points by original SIFT, in our experiment, we set K as 5000. k_i represents the matched feature points after each step.

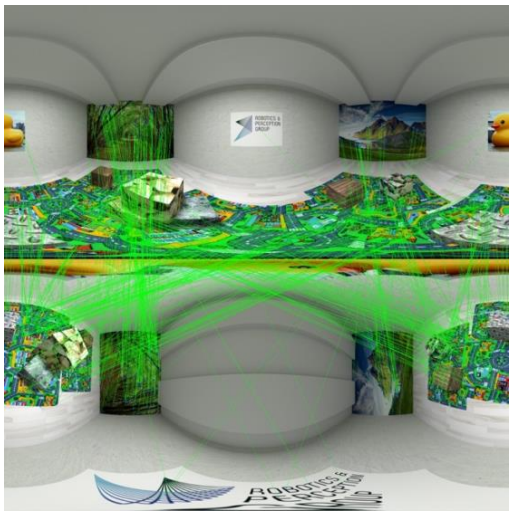
The image pairs for our experiments are divided into 9 groups, group (1-3) are synthetic image pairs, group (4-8) are real images captured in two different scenes, all of the image pairs are captured in different directions. We show the example EQR images of our image set with the counterpart cubic images by preprocessing in Figure 4.1.

Figure 4.1: Example images taken from synthetic scene and real scene, left column shows raw EQR images, right column shows the counterpart preprocessed cubic images.

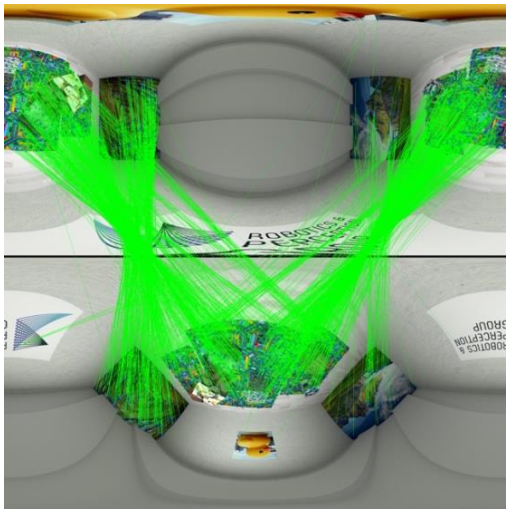
From the results shown in Figure 4.2, we can see that by cubic mapping, the repeatability of SIFT will dramatically increase, because the large distortion on EQR images will affect the performance of detector and descriptor of original SIFT. We consider that the main reason is because the non-uniform distortion makes the detected location of keypoint not accurate anymore, moreover, the surrounding patch for generating feature descriptors are distorted which makes them been weakened. By preprocessing of cubic mapping, the original SIFT can achieve the invariance to this non-uniform distortion. Note that, for group (3,5,7) the performance of feature correspondences on original EQR images is slightly better than on cubic counterparts, this is because the rich texture regions are

projected on central area of EQR image with less distortion, showing that SIFT is robust to slight distortion, meanwhile, the cubic map will suffer from the interpolation by image transformation. The intuition of the above explanation in our evaluation is shown in Figure 4.3. Besides, the results show that the time cost during RANSAC process is also reduced by cubic mapping, it is proved that RANSAC process benefits from our pre-processing step. The main reason is that original SIFT algorithm on cubic map can detect and match more inliers.

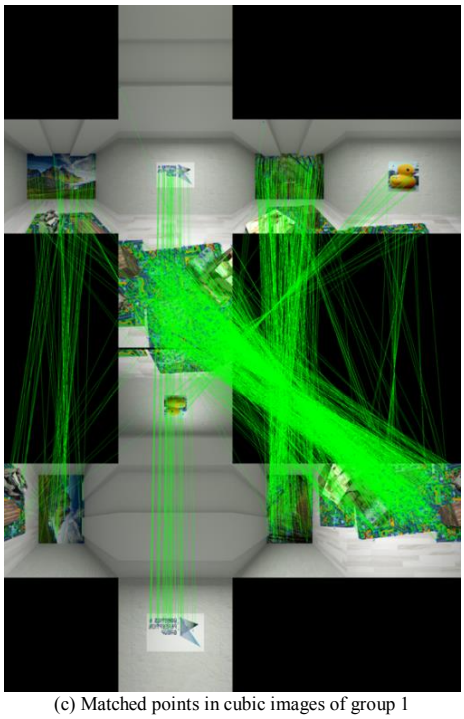
Figure 4.2: Comparison between EQR and Cubic image on performance of feature correspondences.



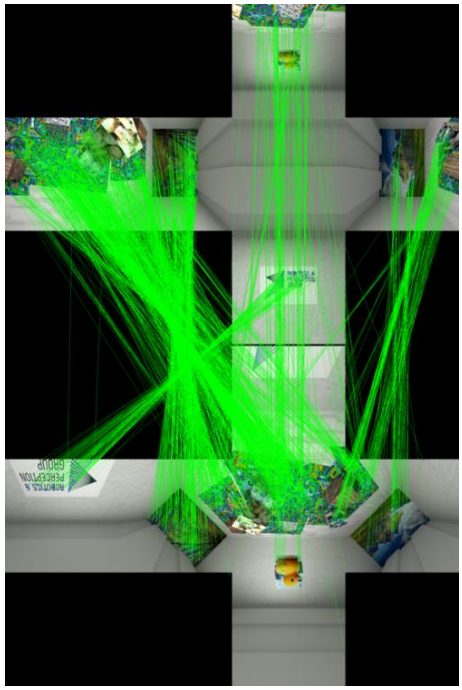
(a) Matched points in EQR images of group 1



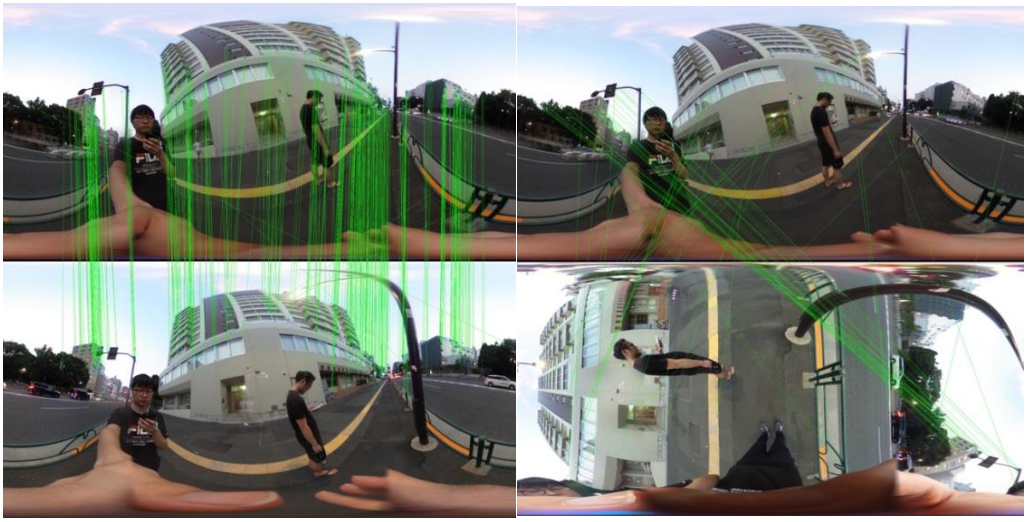
(b) Matched points in EQR images of group 3



(c) Matched points in cubic images of group 1

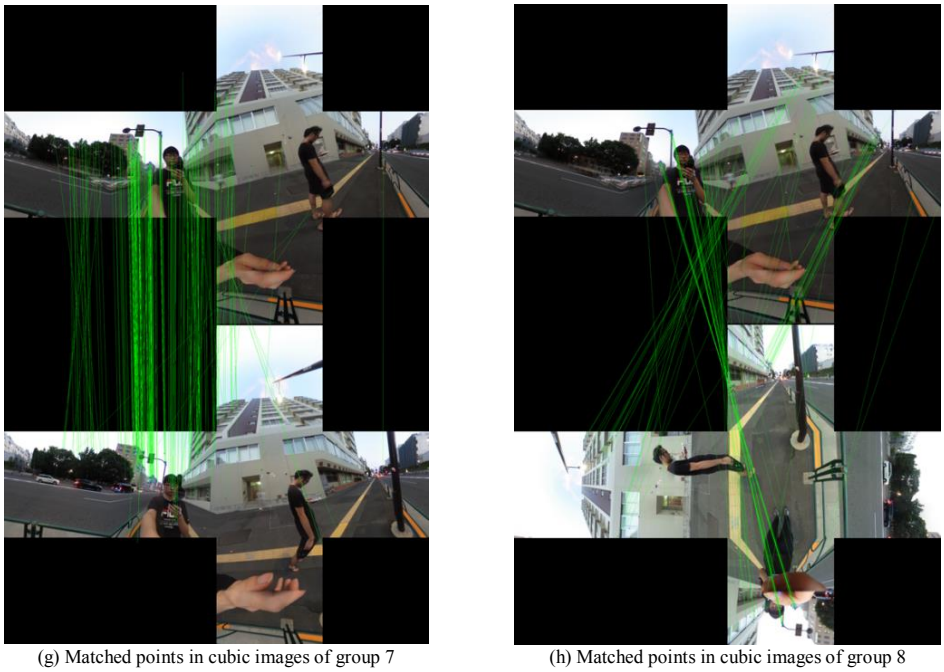


(d) Matched points in cubic images of group 3



(e) Matched points in EQR images in group 7

(f) Matched points in EQR images in group 8



(g) Matched points in cubic images of group 7

(h) Matched points in cubic images of group 8

Figure 4.3: Performance of feature correspondences in two situations ((b,d,e,g) shows with slight distortion, the feature correspondences does not benefit from our preprocessing step; (a,c,f,h) shows large distortion does affect the performance on raw image but can be addressed by preprocessing.)

4.2 Moving objects detection

In this experiment, we perform moving objects detection on group 7 and group 8, which are captured on the street with pedestrians and vehicles as moving objects. We extract the bounding boxes of

moving objects by Mask R-CNN, Figure 4.4 shows that all the possible moving objects are correctly detected with accurate bounding boxes. After that, we remove those detected keypoints (generated by first stage, matched points after original SIFT) located on moving objects, then perform RANSAC and pose estimation for removing outliers. As we know, those matched keypoints located on moving obstacles can be obviously considered as outliers for motion estimation because the fundamental constraint for this task is based on static points. Thus, if those outliers are assumed as inliers in motion estimation, then there will be a great error in the estimated model. We compare the performance of feature correspondences similar in previous experiment, the results are shown in Figure 4.5. Clearly, we can see that by elimination of pedestrians and vehicles, all of the obvious outliers of keypoints are removed as shown Figure 4.6, as a result, it can accelerate the process of RANSAC which resulting in a more accurate motion model been estimated.

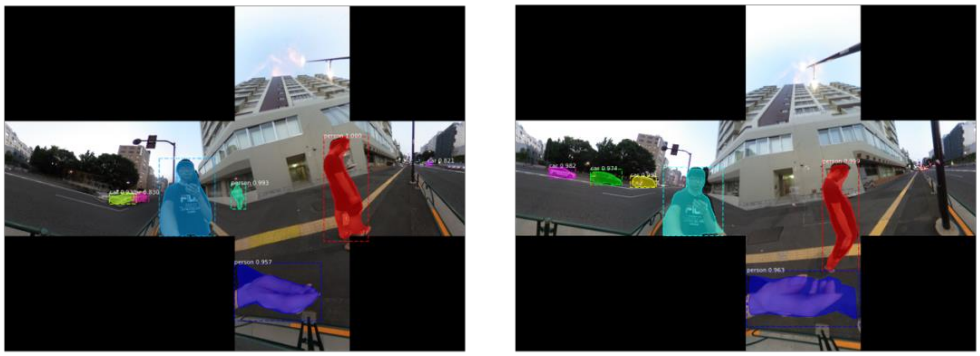


Figure 4.4: Performance of moving objects detection on cubic image (example taken from group 8)

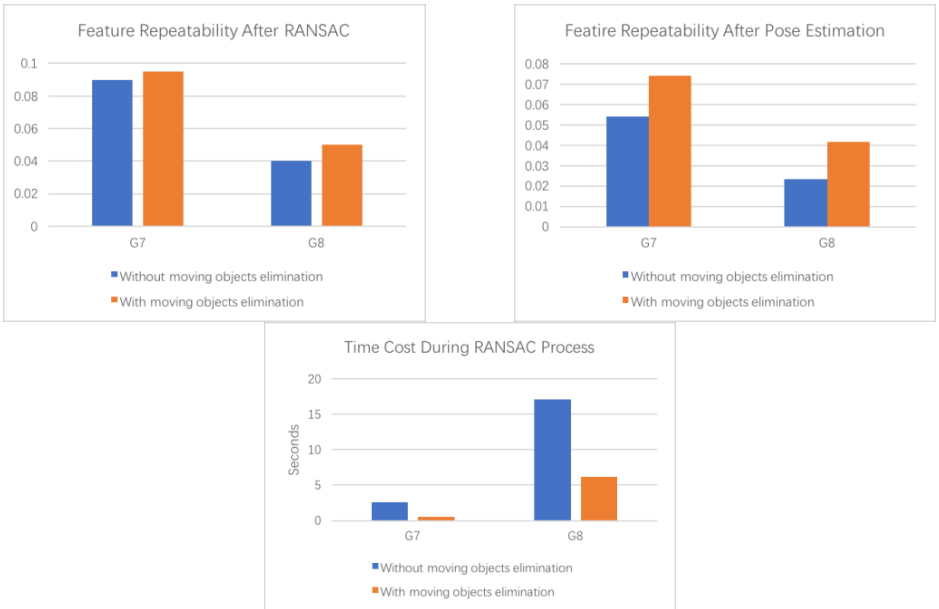


Figure 4.5: Improvement by elimination of moving objects

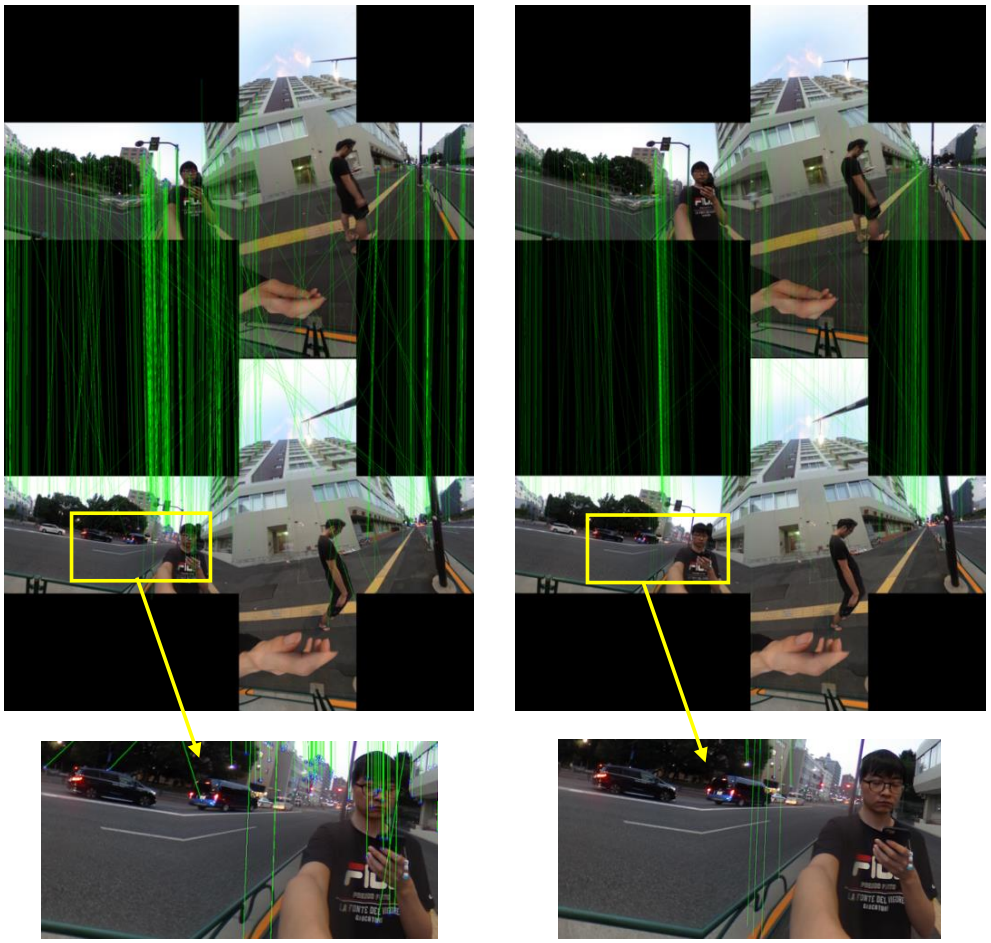


Figure 4.6: Intuition of improvement by elimination of moving objects (example taken from group 8)

4.3 Structure from Motion

4.3.1 Two-view SfM

In this experiment, we validate our modified SfM system in two-view SfM manner, which is similar as depth estimation from stereo camera.

The experiment is set up as shown in Figure 4.7: Placing the camera at two spots with known baseline R, T. Put 11 red markers on the different place with known distance to camera as ground truth. We first capture two raw EQR images then transform them into cubic map. By conducting SIFT feature correspondences and modified two-view SfM system described in Chapter 3, we can recover the relative camera pose R and T, after that, the 3D coordinates of 11 red markers could be calculated by 3D triangulation. The results are shown in Table 4.1. Note that marker P_{12} is placed on the baseline, which cannot be triangulated correctly. Apart from P_{12}, all the markers are reconstructed with error

lower than 4%, which could be reduced by the further nonlinear optimization of BA. Figure 4.8 illustrates the reconstructed 11 markers and camera pose in 3D space.

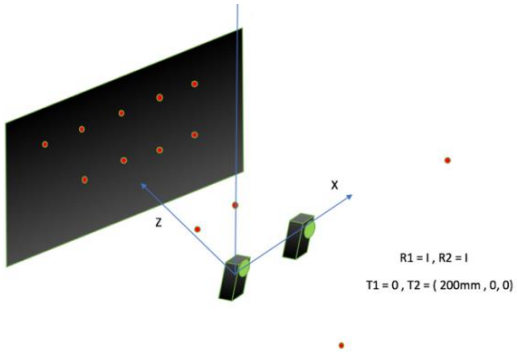


Figure 4.7: Experiment setting for validating two-view SfM

	Estimated_X (dm)	Estimated_Y (dm)	Estimated_Z (dm)	Real_Z (dm)	error
P1 (whiteboard)	-23.29741427	9.34565167	17.28468467	18	3.97%
P2 (whiteboard)	-14.75660424	9.427751749	17.43500841	18	3.14%
P3 (whiteboard)	-5.875507245	9.38854451	17.37849596	18	3.45%
P4 (whiteboard)	2.906190215	9.506031339	17.60431446	18	2.19%
P5 (whiteboard)	11.60279123	9.360921179	17.46213184	18	2.98%
P6 (whiteboard)	-14.75716705	-0.664885324	17.68390698	18	1.75%
P7 (whiteboard)	-5.797492221	-0.691453736	17.48054691	18	2.88%
P8 (whiteboard)	2.86507601	-0.673939487	17.60927556	18	2.17%
P9 (whiteboard)	11.65819371	-0.651355563	17.6700973	18	1.83%
P10 (Desk)	0.093485067	-4.979875875	8.002723373	8	0.03%
P11 (Display)	4.615549642	-0.44612974	7.049298961	7	0.73%
P12 (Right side)	55.73078258	-2.175920543	1.727667707	1	72.76%
P13 (Back)	0.028739201	-1.298988064	-12.29575198	-12.4	0.84%

Table 4.1: The result depth estimation from proposed SfM

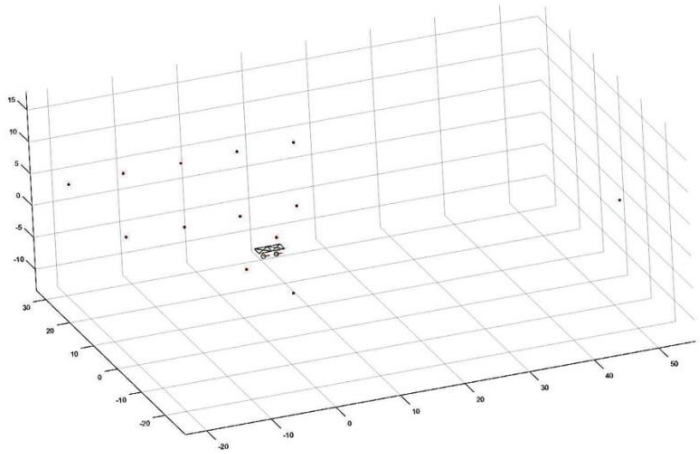


Figure 4.8: Visualizing reconstructed marker points in 3D space

4.3.2 Multi-view SfM

In this experiment, we validate our modified SfM system in multi-view SfM manner. Note that, the estimated translation matrix is presented as $T = [T_x, T_y, T_z]$, the rotation matrix is presented as $R = [\alpha, \beta, \gamma]$ which represents the 3 Euler angles corresponding three axes in 3D space. We set (R_1, T_1) of first frame as (0, 0) which represents the reference system of world coordinates. The test images are taken from synthetic group, the results of motion estimation and 3D reconstruction by our proposed SfM system are shown in Table 4.2 and Figure 4.9, respectively, which proves that our SfM system can work well for omni-directional images.

	Tx	Ty	Tz	α [rad]	β [rad]	γ [rad]
Frame 2	0.999	6.8 e-4	0.005	0.00	0.002	0.001
Ground truth	1.000	0.000	0.000	0.00	0.000	0.000
Frame 3	1.955	1.3 e-4	0.084	-0.02	0.010	0.003
Ground truth	2.000	0.000	0.000	0.00	0.000	0.000

Table 4.2: Motion estimation by proposed SfM

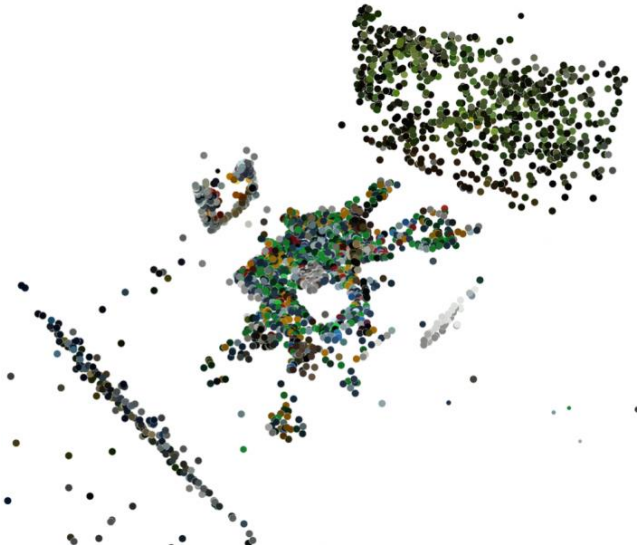


Figure 4.9: Visualizing reconstructed 3D points by proposed SfM

Chapter 5 Conclusion

In this work, we proposed a novel Structure from Motion (SfM) based system to recover the sparse 3D structure and ego-motion from Omni-directional (OD) images with elimination of moving obstacles based on deep CNN model Mask R-CNN. The problems of non-uniform distortion introduced by Equi-rectangular (EQR) images are discussed in detail: first, it can dramatically affect the repeatability of conventional algorithm for feature detection and matching, especially in a pair of images with large rotation, which leads to unreliable performance in further processing; second, it also shows unsatisfactory performances on object detection since current CNN architecture and existing perspective images based datasets cannot achieve invariance this kind of location-dependent distortion. In order to address them, we proposed several approaches based on image transformation which are proved to be effective.

Furthermore, we modified the conventional SfM to make it suitable to OD images since the conventional system is associated with perspective image which has totally different projection model with OD image, we evaluated our system for not only two-view SfM (which also can be used in depth estimation from stereo OD camera) but also multi-view SfM, the results of recovered structure and camera poses showed that proposed system can works well with OD images.

Finally, we combined SfM with elimination of moving obstacles (pedestrians and vehicles) based on Mask R-CNN, the results show that SfM system could benefit from this combination since those moving obstacles located on large pixel area in EQR image would interfere the operation of feature detection and matching.

However, our system is associated with large computation cost in image transformation (such as cubic mapping) which still can be improved in future work. Moreover, this work is involved with frequent processing of mapping and interpolation which leads to information loss. For future work, the dense reconstruction for OD images could be conducted with the estimated camera poses as well as the sparse structure reconstructed by our proposed SfM system.

Chapter 6 Appendix

6.1 List of academic achievements

[1] Mengcheng Song, Junichi Hara and Hiroshi Watanabe: “Instance Segmentation on Omni-directional Images Based on Mask R-CNN,” IEICE General Conference, BS-2-13, Mar. 2018

[2] Mengcheng Song, Junich Hara and Hiroshi Watanabe: “Robust 3D Reconstruction with Omni-directional Camera based on Structure from Motion,” International Workshop on Advanced Image Technology (IWAIT2018), No.105, pp.1-4, Jan. 2018

Bibliography

1. Scaramuzza, D. and F. Fraundorfer, *Visual odometry [tutorial]*. IEEE robotics & automation magazine, 2011. **18**(4): p. 80-92.
2. Özyeşil, O., et al., *A survey of structure from motion**. Acta Numerica, 2017. **26**: p. 305-364.
3. Mur-Artal, R., J.M.M. Montiel, and J.D. Tardos, *ORB-SLAM: a versatile and accurate monocular SLAM system.* IEEE Transactions on Robotics, 2015. **31**(5): p. 1147-1163.
4. Engel, J., T. Schöps, and D. Cremers. *LSD-SLAM: Large-scale direct monocular SLAM*. in *European Conference on Computer Vision*. 2014. Springer.
5. Agarwal, S., et al. *Building rome in a day*. in *Computer Vision, 2009 IEEE 12th International Conference on*. 2009. IEEE.
6. Snavely, N., S.M. Seitz, and R. Szeliski. *Photo tourism: exploring photo collections in 3D*. in *ACM transactions on graphics (TOG)*. 2006. ACM.
7. Wu, C., *VisualSFM: A visual structure from motion system*. 2011.
8. Girshick, R., et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
9. Ren, S., et al. *Faster r-cnn: Towards real-time object detection with region proposal networks*. in *Advances in neural information processing systems*. 2015.
10. He, K., et al. *Mask r-cnn*. in *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017. IEEE.
11. Liu, W., et al. *Ssd: Single shot multibox detector*. in *European conference on computer vision*. 2016. Springer.
12. Redmon, J., et al. *You only look once: Unified, real-time object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
13. H.Watanabe, Y.Z.a., *Trajectory of Ego-Motion Videos with Pedestrian Based on Monocular Visual Odometry and Machine Learning*, in *IEICE General Conference*. 2017.
14. Dalal, N. and B. Triggs. *Histograms of oriented gradients for human detection*. in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. 2005. IEEE.
15. Hearst, M.A., et al., *Support vector machines.* IEEE Intelligent Systems and their applications, 1998. **13**(4): p. 18-28.
16. Zhang, Z., et al. *Benefit of large field-of-view cameras for visual odometry*. in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. 2016. IEEE.
17. Kannala, J. and S.S. Brandt, *A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses*. IEEE transactions on pattern analysis and machine intelligence, 2006. **28**(8): p. 1335-1340.
18. Hughes, C., et al., *Validation of polynomial-based equidistance fish-eye models.* 2009.

19. MORITA, T., K. TERABAYASHI, and K. UMEDA, *魚眼カメラを用いた EPI 解析による 3 次元環境復元.*
20. Caruso, D., J. Engel, and D. Cremers. *Large-scale direct SLAM for omnidirectional cameras*. in *IROS*. 2015.
21. Micusik, B. and T. Pajdla, *Structure from motion with wide circular field of view cameras.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006. **28**(7): p. 1135-1149.
22. Lourenço, M., J.P. Barreto, and F. Vasconcelos, *sRD-SIFT: keypoint detection and matching in images with radial distortion.* change, 2012. **4**: p. 5.
23. Urban, S., M. Weinmann, and S. Hinz, *mdBRIEF-a fast online-adaptable, distorted binary descriptor for real-time applications using calibrated wide-angle or fisheye cameras.* Computer Vision and Image Understanding, 2017. **162**: p. 71-86.
24. Lowe, D.G., *Distinctive image features from scale-invariant keypoints.* International journal of computer vision, 2004. **60**(2): p. 91-110.
25. Calonder, M., et al. *Brief: Binary robust independent elementary features*. in *European conference on computer vision*. 2010. Springer.
26. 井上優希, et al., *全球パノラマ画像を用いた SfM による 3 次元復元と自己位置・方位推定への応用.* 精密工学会誌, 2015. **81**(12): p. 1173-1179.
27. Cruz-Mota, J., et al., *Scale invariant feature transform on the sphere: Theory and applications.* International journal of computer vision, 2012. **98**(2): p. 217-241.
28. Kaneko, M., et al. *Mask-SLAM: Robust Feature-Based Monocular SLAM by Masking Using Semantic Segmentation*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018.
29. *GoPro Fusion*. Available from: https://jp.shop.gopro.com.
30. *Gopro Omni*. Available from: https://www.bhphotovideo.com/c/product/1247866-REG/gopro_mhdhx_007_omni_rig_only.html.
31. *Ricoh Theta*. Available from: https://fotocentreindia.com/product/ricoh-theta-s-spherical-vr-digital-camera-in-mumbai-india/.
32. Li, J., et al. *Novel tile segmentation scheme for omnidirectional video*. in *Image Processing (ICIP), 2016 IEEE International Conference on*. 2016. IEEE.
33. Li, S. and Y. Hai. *A full-view spherical image format*. in *Pattern Recognition (ICPR), 2010 20th International Conference on*. 2010. IEEE.
34. Hartley, R. and A. Zisserman, *Multiple view geometry in computer vision*. 2003: Cambridge university press.
35. Szeliski, R., *Computer vision: algorithms and applications*. 2010: Springer Science & Business Media.
36. Zhang, Z., *A flexible new technique for camera calibration.* IEEE Transactions on pattern analysis and machine intelligence, 2000. **22**.

37. Derpanis, K.G., *The harris corner detector.* York University, 2004.
38. Rosten, E. and T. Drummond. *Machine learning for high-speed corner detection*. in *European conference on computer vision*. 2006. Springer.
39. Rublee, E., et al. *ORB: An efficient alternative to SIFT or SURF*. in *Computer Vision (ICCV), 2011 IEEE international conference on*. 2011. IEEE.
40. Bay, H., T. Tuytelaars, and L. Van Gool. *Surf: Speeded up robust features*. in *European conference on computer vision*. 2006. Springer.
41. Morel, J.-M. and G. Yu, *ASIFT: A new framework for fully affine invariant image comparison.* SIAM journal on imaging sciences, 2009. **2**(2): p. 438-469.
42. Fischler, M.A. and R.C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.* Communications of the ACM, 1981. **24**(6): p. 381-395.
43. Triggs, B., et al. *Bundle adjustment—a modern synthesis*. in *International workshop on vision algorithms*. 1999. Springer.
44. Lourakis, M.I. and A.A. Argyros, *SBA: A software package for generic sparse bundle adjustment.* ACM Transactions on Mathematical Software (TOMS), 2009. **36**(1): p. 2.
45. Felzenszwalb, P., D. McAllester, and D. Ramanan. *A discriminatively trained, multiscale, deformable part model*. in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. 2008. IEEE.
46. Long, J., E. Shelhamer, and T. Darrell. *Fully convolutional networks for semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
47. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
48. Lin, T.-Y., et al. *Feature Pyramid Networks for Object Detection*. in *CVPR*. 2017.
49. Lin, T.-Y., et al. *Microsoft coco: Common objects in context*. in *European conference on computer vision*. 2014. Springer.
50. Deng, J., et al. *Imagenet: A large-scale hierarchical image database*. in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. 2009. Ieee.
51. Fei-Fei, L. *Visualizing and Understanding*. Available from: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf.
52. Krizhevsky, A., *One weird trick for parallelizing convolutional neural networks.* arXiv preprint arXiv:1404.5997, 2014.
53. Bradski, G. and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. 2008: " O'Reilly Media, Inc.".
54. Abdulla, W. *Mask R-CNN for object detection and instance segmentation on Keras and Tensorflow*. 2017; Available from: https://github.com/matterport/Mask_RCNN.