Robust Pose Estimation for Spherical Panorama

by

Shuai Wang

May 14, 2024

A thesis submitted to the
faculty of the Graduate School of
the University at Buffalo, The State University of New York
in partial fulfillment of the requirements for the
degree of

Master of Science

Department of Electrical Engineering

Abstract

Spherical panoramas have emerged as a critical component in various domains, including computer vision, robotics, and immersive experiences. Their applications span Virtual Reality (VR), Augmented Reality (AR), Mixed Reality (MR), Virtual Tourism (VT), and robot navigation. In virtual travel scenarios, akin to navigating cities via Google Street View, the demand for seamless transitions from outdoor to indoor environments has surged. Achieving accurate pose estimation becomes paramount for enabling smooth scene-to-scene navigation.

This research tackles the inherent challenges posed by diverse panoramic scenes. Single keypoint detection and matching algorithms often encounter limitations, such as insufficient matching points and susceptibility to high noise levels. To address these issues, we propose an innovative algorithm that synergistically combines multiple keypoint detection and matching methods. By doing so, we mitigate the scarcity of matching points and enhance robustness.

Furthermore, we address noise influence by transforming the matching points onto a unit sphere and subsequently grouping them. This grouping strategy effectively reduces outlier rates within specific data clusters. Notably, our approach operates in scenarios where traditional single-view methods struggle due to weak textures, varying lighting conditions, and limited overlap.

In addition, we introduce a novel reprojection error function for evaluating the essential matrix. Unlike conventional threshold-based approaches, our function directly computes the reprojection error using camera coordinates, eliminating the need for manual parameter tuning.

Extensive testing across a diverse dataset of over 300 high-resolution spherical panoramas validates the stability and accuracy of our method. Even in challenging scenarios, such as scenes with minimal overlap or high error rates due to incorrect matches, our approach consistently delivers reliable results.

Finally, we extensively test our approach on over 400 pairs of high-resolution spherical panoramas across more than 10 series. Our method consistently delivers accurate results, even in challenging scenarios such as weak textures, varying lighting conditions, and limited overlap or high error rates due to incorrect matches.

Acknowledgement

Firstly, I sincerely want to thank my advisor, Dr. Chen Wang, for giving me the opportunity to work with him. Whenever I faced challenges, he always provided valuable advice. Working in his lab was truly an unforgettable experience.

I also express my gratitude to all my colleagues in the lab. Our weekly meetings enriched my understanding and knowledge. Additionally, I appreciate the friends who contributed to memorable experiences during my time studying in University at Buffalo.

Lastly, I extend my truly thanks to my family for their unwavering support and assistance in every decision I made.

CONTENTS

List of Tables

List of Figures

Chapter 1 Introduction

Spherical panoramas, also known as 360 ° or omnidirectional panoramas, capture the entire environment in all directions[1, 2]. Unlike perspective images, which focus on a narrow field of view, spherical panoramas allow viewers to look in any direction. Based on this feature, they provide the viewers a sense of immersion, allowing them to explore the scene as if they were physically present. These make it widely used in Virtual, Augmented, and Mixed Reality (VR, AR, MR)[3]. In the area of Tourism and Marketing, they are used in travel websites, real estate listings, and promotional materials. They can also be used for Astrophysical Data Visualization.

In the realm of virtual travel experiences, Google Street View has revolutionized remote exploration by allowing users to immerse themselves in distant cities without leaving their homes. Beyond serving armchair travelers, it also provides valuable tourist guidance for those planning physical visits. However, Google Street View predominantly captures street-level scenes, leaving gaps in its representation of off-road locations, such as restaurants and businesses. Over the past decade, advancements in optical, mechanical, and electronic technologies have led to the proliferation of panoramic cameras. Simultaneously, various panorama stitching software tools have emerged (e.g., PtGui, AutoPano, RealViz and Panorama Factory)[4], resulting in an increasing availability of spherical panorama images on the web. Both outdoor and indoor immersive virtual navigation have gained significant attention.

In the context of virtual immersive navigation, panoramic images contain far more information than regular images. Camera movements between two panorama frames can be substantial (large baseline), resulting in minimal overlap in many cases—such as transitioning from outdoor to indoor environments or moving from one room to another. Furthermore, the environmental diversity encountered is vast.

Relative pose estimation plays a pivotal role in enabling seamless transitions from one panoramic scene to another. Without accurate pose estimation, users risk becoming disoriented within the panorama. In the fields of computer vision and robotics, pose estimation between two views has been extensively studied for decades, but primarily focusing on perspective images. DeMoN [5], BA-Net [6], and RegNet [7] propose differentiable pose optimization techniques to obtain well-aligned poses. However, those approaches are designed for perspective images. OpenVSLAM [8] and 360VO [9] performed correctly on equirectangular videos, which has small baseline. LF-VISLAM [10] designed to the cameras with extremely Large FoV with loop closure. But it may not be suitable for low-light environments due to the limited light entering the panoramic annular lens camera.

For spherical panoramic images, since spherical panoramic images have high distortion, [11] leverages normalization to mitigate the impact of uneven keypoint distributions and the correspondence of outliers. However, it was only tested for outlier rates below 20%, and despite its faster execution, it did not significantly outperform Random Sample Consensus (RANSAC)

[12] with strict thresholds. [13] involves manual feature selection for pose estimation. However, relying on manual feature selection is impractical in an era emphasizing automation. Additionally, [14] simplifies camera motion to a planar model, reducing computational complexity but imposing strict constraints on camera movements.

Those above reasons motivated us to design a robust pose estimation approach specifically for spherical panoramic images, which have various environments, large baseline, small overlap, high distortion.

The typical pose estimation pipeline for spherical panoramic images involves keypoint detection and matching, essential matrix estimation using epipolar geometry constraints, and subsequent decomposition by singular value decomposition (SVD) [15, 16] to obtain relative poses. While spherical panoramas share similarities with perspective images, they lack intrinsic camera matrices, exhibit distinct epipolar geometry, and suffer from significant distortions due to their equirectangular projection (ERP)[3, 17, 18] onto a rectangular image.

Feature matching task draw attention for many years. Hand-crafted features like SIFT, ORB [19, 20] have been widely used for a long time. Specialized algorithms tailored for spherical panoramas (e.g., SPHORB, SSIFT) [21] [22] have also emerged. However, those hand-crafted methods are struggling to the appearance or viewpoint changes. To address these challenges, learning-based methods have been developed. Like the detector-based methods SuperPoint [23], D2-Net [24], and DISK [25]. Since the detector-based local feature matching algorithms only produced sparse keypoints, to achieve pixel-wise dense matching, detector-free methods were proposed, such as DRC-Net [26], LoFTR [27] , COTR [28] and 3DG-STFM[29]. Networks are trained and tested in different datasets and reach the state-of-the-art performances in different field. However, the data-driven nature still requires more data to adapt to various environments.

In[30] , the authors evaluated various keypoint matching and pose estimation methods in different environments, and showed that the performance of a descriptor can be biased to a scenario or dataset. This inspired us to explore the possibility of using multi- matching methods with a selection mechanism to improve the pose estimation performance. With multi-methods and proper selection mechanism, may result in a system that combine the advantages of the methods without taking huge effort to train the models.

Our contributions are as follows:

1. Hybrid Keypoint Detection and Matching:

   We address the limitations of single keypoint detection and matching algorithms by combining multiple methods. A simple mechanism dynamically selects the most suitable matching algorithm based on the environment.

2. Noise Reduction via Normalization and Grouping:

In real-world scenarios, matching points often suffer from noise (outliers) due to lighting variations, occlusions, repetitive patterns, and texture scarcity. To mitigate this, we normalize the matching points onto a unit sphere and group them. By clustering the points, we significantly reduce the outlier rate within specific data clusters.

3. Reprojection Error Function:

Instead of directly removing outliers during RANSAC, we introduce a novel reprojection error function. This function computes the reprojection error using camera coordinates, eliminating the need for manual threshold tuning.

4. Extensive Testing:

We evaluate our method on a dataset comprising over 400 pairs of high-resolution spherical panoramas across ten series. We compare the results with pure SIFT + KNN(only SIFT for matching), pure SuperPoint + LightGlue, and no preprocessing stage. The results showed that multi-matching methods outperform single method, and the designed preprocessing improve the pose estimation performance. Overall, our approach consistently delivers stable and accurate results, even in challenging environments with weak textures and substantial noise.

Our method provides precise directional cues for virtual immersive navigation, ensuring a seamless transition from one scene to another.

Chapter 2 Related Work

The market has witnessed a growing number of spherical panorama capture devices. These devices can be broadly categorized into three main types. First is the stitching-based panoramic imaging system. This category includes both single-camera rotational scanning and multi-camera stitching systems. In single-camera rotational scanning, a camera is mounted on a rotating mechanical system to capture images from all angles, which are then stitched together[31]. Multi-camera stitching involves simultaneous capture from different directions using multiple cameras, followed by stitching. In [32], they discuss using six visible light cameras for field-of-view stitching. Second is fisheye panoramic imaging systems. It adopts fisheye lenses[33-35], with their wide-angle capabilities, allow 360° panoramas to be stitched using only two lenses. Last is catadioptric panoramic imaging system. This system uses reflective optical elements to reflect 360° light rays into subsequent lens assemblies for panoramic imaging[36-38].

Additionally, emerging technologies such as monocentric panoramic system [39], hyper-hemispheric lens imaging [40], and panoramic annular lenses [41] have made acquiring panoramic images increasingly accessible. Furthermore, commercial software tools like PtGui, AutoPano, RealViz, and Panorama Factory [4] contribute to the development of spherical panorama imaging through advanced stitching techniques.

Pose estimation between two views has been extensively studied for decades. DeMoN [5], BA-Net [6], and RegNet [7] propose differentiable pose optimization techniques to obtain well-aligned poses. In [42], they significantly improve pose estimation performance by connecting feature matching and pose optimization in an end-to-end trainable approach. However, those approaches are designed for perspective images. OpenVSLAM [8] is correctly performed both equirectangular video captured outdoors and indoors. 360VO [9] takes the advantage of a 360-degree camera for robust localization and mapping. However, those methods use the equirectangular video which has small baseline. LF-VISLAM [10] designed to the cameras with extremely Large FoV with loop closure. But it may not be suitable for low-light environments due to the limited light entering the panoramic annular lens camera.

For two spherical panoramic images, relative pose estimation mainly involves three key steps: Keypoint Detection and Matching, Essential Matrix Estimation and Outlier Detection and Removal.

Traditional methods include SIFT for generating scale-invariant features, and ORB[20] for quickly creating feature vectors from keypoints in images. There are also specialized methods like SSIFT [22] and SPHORB [21] designed specifically for spherical panoramic images. However, in environments with weak texture, it can be challenging to achieve good results using these methods. Apart from the traditional feature matching, learning-based methods were employed recently. SuperPoint [23] uses self-supervised learning for training interest point detectors and descriptors. The original paper showed that it outperforms ORB, SIFT, and LIFT [43]. D2-Net [24] proposed a method that a single convolutional layer plays a dense feature

descriptor and a feature detector simultaneously. DISK [25] leverages principles from Reinforcement Learning (RL) and optimizes end-to-end for a high number of correct feature matches. However, those several learning-based methods and the traditional methods are detector-based local feature matching algorithms, and only produced sparse keypoints. To achieve pixel-wise dense matching, detector-free methods were proposed. DRC-Net [26] proposes a coarse-to-fine approach to produce dense matches with higher accuracy. LoFTR [27] was proposed to learn global consensus between image correspondences by leveraging transformers. COTR [28] adopt attention and a coarse-to-fine approach to achieve high-quality matches. For the local feature matching task, 3DG-STFM[29] is the first student-teacher learning method and achieves state-of-the-art performances on several image matching on indoor and outdoor datasets. But it needs the depth information. Networks are trained and tested in different datasets and reach the state-of-the-art performances in different field. However, the data-driven nature still requires more data to adapt to various environments.

Additionally, there are numerous studies on keypoint matching. LightGlue [44] presents a matching method based on Transformer's attention mechanism, simulates human feature matching. It exhibits good robustness to weak texture, lighting variations, and changes in viewpoint.

Epipolar constraints in epipolar geometry [45] are widely used in perspective images. However, due to the different projection geometry in spherical panoramic images compared to the pinhole model, there is a distinct epipolar geometry for spherical panoramas [46]. In perspective image pairs, estimating the essential matrix requires knowledge of the intrinsic camera parameters (represented by the intrinsic matrix). However, in the case of spherical panoramas, the spherical model doesn't involve intrinsic parameters [47]. Several methods have been proposed to find the essential matrix, including the Eight-Point Algorithm (8-PA) [48, 49] , Five-Point Algorithm (5-PA) [50], Non-linear least-squares optimization (NLR) [51], and SK Non-linear optimization (SK) [11]. Among these methods, the five-point (5-PA) and eight-point (8-PA) algorithms are the most widely applied. While the former uses fewer matching points to estimate the essential matrix, its implementation often relies on a polynomial approximation with multiple solutions. In contrast, the 8-PA is a linear approach that yields an unambiguous result. Moreover, the 8-PA is commonly used for spherical panoramic images [8, 47, 52, 53] due to its simplicity and stability in large field-of-views.

Commonly used methods for outlier detection and removal include RANSAC [12] and error functions [54]. For spherical images, [55] addresses error function evaluation for the essential matrix. It proposes three types of reprojection errors to evaluate the essential matrix for spherical panoramas. However, all these methods require image rectification, resulting in significant computational overhead. Our study introduces a novel reprojection error function that directly computes Euclidean distances between camera coordinates. This approach yields excellent results.

In general, the effectiveness of keypoint detection and matching directly influences the accuracy of pose estimation. It is highly affected by external factors such as lighting conditions and lack of textures. In [48], a strategy of normalizing input data is used to reduce the impact of noise. However, this approach is specifically designed for perspective cameras.

The [11] uses normalization to mitigate the effects of unevenly distributed key features in spherical panoramic images and their corresponding outliers. However, it was only tested with an outlier rate below 20%. Compared to using RANSAC with a strict threshold, the method offers faster processing but does not show significant advantages. [13] estimates pose by manually selecting keypoints, but this manual process can be time-consuming and labor-intensive. In [56], the authors demonstrate that for essential matrix estimation in spherical panoramic images, normalizing image coordinates to unit vectors is necessary. Therefore, in this paper, we transform image correspondences onto a unit sphere and divides them into two to three groups based on their distribution on the sphere. This approach reduces the proportion of noise within a specific group, thereby improving the accuracy and robustness of pose estimation.

Chapter 3 Background and Spherical Geometry

3.1 Panoramic Imaging

Panoramic images provide a comprehensive view of a scene. Unlike pin-hole cameras, they require very few images to cover an entire scene. This allows panoramic images to accommodate more scene information in a smaller size with higher resolution. Additionally, panoramic images free operators from the constraints of fixed field of view inherent in pin-hole cameras. With technological advancements, various panoramic imaging techniques have emerged. Currently, widely used methods include single-camera rotation scanning, multi-camera stitching, fisheye panoramic imaging and catadioptric panoramic imaging. Over the past decade, emerging optical technologies such as freeform surfaces, thin-plate optics and metasurfaces[57], along with artificial intelligent (AI) applications, have led to expectations of panoramic imaging systems with higher resolution, no blind spots, miniaturization, and multidimensional intelligent perception. These features enable panoramic imaging to play a more powerful role in various fields.

3.1.1 Stitching Panoramic Imaging

Stitching panoramic imaging stitch multiple images to achieve large field of view (FoV)[58]. At the inception of panoramic imaging technology, the pursuit of a wider FoV led to capturing multiple images using standard cameras and then stitching, merging, and synthesizing them to create panoramic scenes. Stitching-based panoramic imaging typically involves two methods: single-camera rotation scanning and multi-camera combination stitching.

In single-camera rotation scanning system, a single camera captures images from different directions by rotating at a high speed and frame rate. To meet strict geometric conditions, a high-precision mechanical rotating stage is used, as shown in Fig. 3.1(a). This method is straightforward and yields high-resolution panoramic images. However, it may not meet real-time requirements in certain situations.

The multi-camera combination stitching system compensates for the limitations of single-camera scanning. Multiple cameras simultaneously capture images from various directions, as shown in Fig. 3.1(d), resulting in a complete scene image. Multi-camera stitching can achieve real-time performance while maintaining high resolution. However, the equipment structure is more complex, and the use of multiple cameras increases the overall cost.

Regardless of whether it’s single-camera rotation scanning or multi-camera stitching, the goal is to expand the field of view by capturing multiple images that contain information about the entire scene and then synthesizing them into a panoramic image.

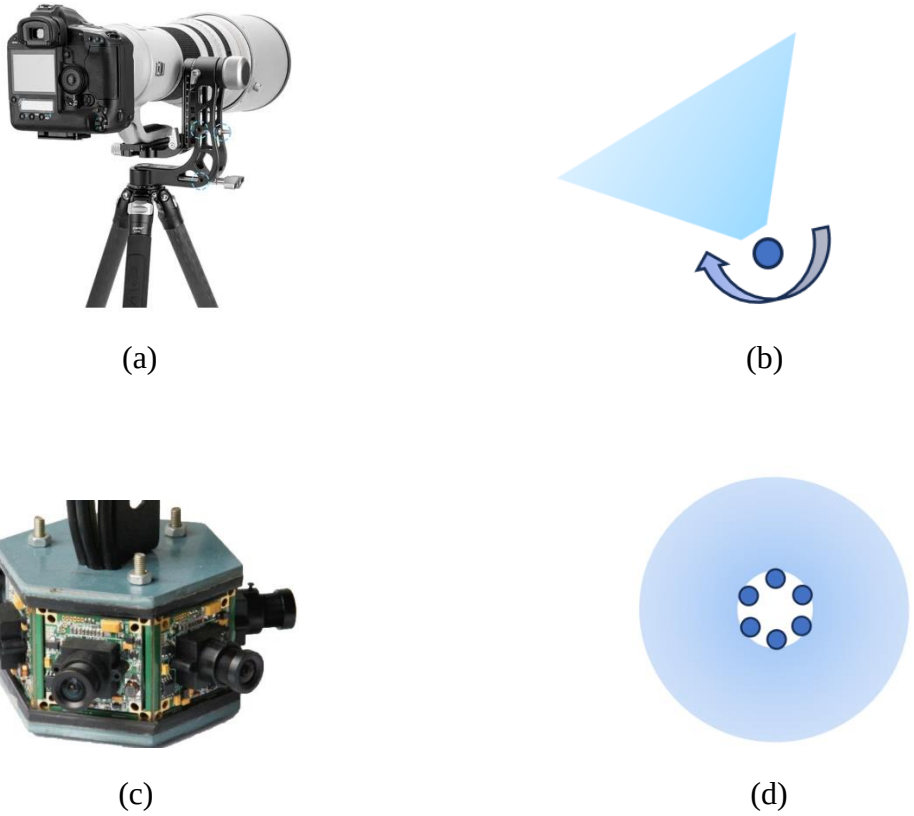


Fig. 3. 1. The technology of stitching panoramic imaging. (a) A case of stitching panoramic imaging system using single-camera with a tripod and head. (b) Principle of single-camera scanning shooting. (c) A case of stitching panoramic imaging system using six visible light cameras. (d) Principle of multiple cameras shooting.

3.1.2 Fisheye Panoramic Imaging

Fisheye panoramic imaging system utilizes fisheye lenses to achieve panoramic imaging. A fisheye lens is an ultra-wide-angle lens [59] with a visual effect like observing objects on the water's surface through a fisheye, as shown in Fig. 3.2(a). Fisheye panoramic imaging system typically consists of two or three negative meniscus lenses as the front optical group. These lenses compress the large field of view of the captured scene to match the field of view required by conventional lenses. Subsequently, a subsequent lens group corrects for aberrations.

Compared to stitching panoramic imaging, fisheye panoramic imaging allows capturing nearly 180° field of view without the need for stitching. It provides a hemispherical view of the scene. In practical applications, fisheye panoramic imaging systems can use two fisheye lenses for image stitching, resulting in an even larger panoramic field of view. However, images captured through fisheye lenses exhibit significant barrel distortion [60]. In this distortion, the central area of the image remains unchanged, while other objects that should be horizontal and vertical radiate outward from the center in various directions, creating the characteristic fisheye effect, as shown in Fig. 3.2(c).

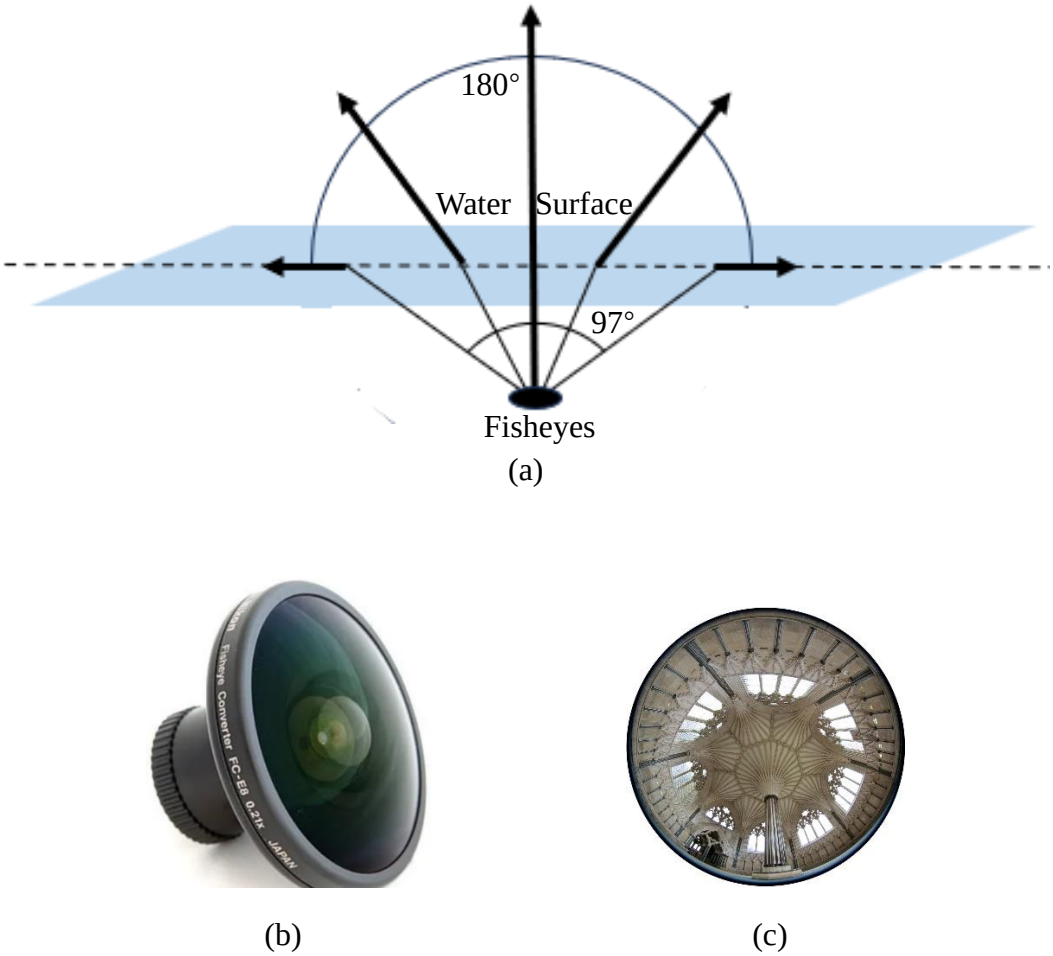


Fig. 3. 2. Fisheye panoramic imaging. (a) A fish's field of view in the water. (b) A case of fisheye lens. (c) A panoramic image captured by a fisheye lens. Images from Wikipedia.

3.1.3 Catadioptric Panoramic System

Catadioptric panoramic imaging is an imaging technique that combines conventional imaging devices with reflective optical elements. It primarily consists of two parts: reflective optical components and refractive optical components. The reflective optical elements here typically refer to various types of mirrors, such as spherical, parabolic, ellipsoidal, and hyperbolic mirrors. Unlike fisheye panoramic imaging systems, catadioptric panoramic systems use mirrors to reflect 360-degree light rays into subsequent lens groups, as shown in Fig. 3.3(a). Due to the characteristics of catadioptric panoramic imaging, this optical system self-images at the center of the image, which can also be considered a blind spot or a non-interesting area.

(a) (b) (c)

Fig. 3. 3. Catadioptric panoramic imaging. (a) Principle of catadioptric panoramic imaging system. (b) Single mirror catadioptric panoramic system. (c) An image captured by single mirror catadioptric panoramic system. Images from [57].

3.1.4 Other Panoramic Imaging Systems and Stitching Software

In the past decade, with the rapid development of high-speed communication and AI technology, panoramic imaging has become an intelligent solution for next-generation environmental perception and measurement due to its wide field of view. As attention to panoramic imaging grows, and with the emergence of technologies like freeform surfaces and metasurfaces, the landscape of panoramic imaging systems has undergone significant transformation.

In addition to well-established and widely used methods such as single-camera scanning, multi-camera stitching, fisheye, and catadioptric panoramic imaging systems, new architectural designs have emerged. These include monocentric panoramic systems [39], hyper-hemispheric lens imaging [40], and panoramic annular lens systems [41]. These diverse architectures enrich the field of panoramic imaging and enhance their effectiveness in various application domains.

Furthermore, Commercial software tools like PtGui, AutoPano, RealViz, and Panorama Factory [4] also play a crucial role in advancing spherical panorama imaging through sophisticated stitching techniques. Overall, panoramic imaging continues to evolve, offering powerful solutions for diverse applications.

3.2 Spherical panoramic image projection model

A spherical panoramic image is an image that covers the entire scene. It provides a complete 360 × 180-degree environmental view. To obtain an undistorted, high-quality spherical panoramic image, certain strict geometric conditions must be met. Specifically, specialized tripods and spherical heads designed for panoramic imaging are used to align the line-of-sight centers of the

images (this is the typical setup for capturing spherical panoramas). Stitching software compensates for radial distortion and residual horizontal and vertical offsets, resulting in the final panoramic image with minimal distortion [61].

Once the panoramic image is obtained, it is projected onto an equirectangular mapping (also known as cylindrical equidistant projection) according to [62]. In this projection, meridians (lines of longitude) and parallels (lines of latitude) are represented by equidistant vertical and horizontal lines, respectively. A spherical panoramic image provides a 360-degree horizontal view and a 180-degree vertical view. In other words, its width is exactly twice its height, as shown in Fig. 3.4.

Fig. 3. 4. Example Spherical Panorama of an indoor scene

The geometric expression for spherical panoramic imaging is fundamental and serves as the basis for pose estimation. Unlike pinhole cameras, which have intrinsic parameters, spherical cameras do not possess such intrinsic parameters. However, there are similarities between the two. In the pinhole camera model, projection involves scaling light rays by a certain factor to map them onto the image plane. The resulting points on the image are in 2D. In the spherical imaging model, projection normalizes light rays onto a unit sphere. Consequently, points on a spherical panoramic image are 3D points with a magnitude of 1.

3.2.1 Coordinate systems

Spherical panoramic image coordinates can be divided into several coordinate systems: image coordinates, spherical coordinates, camera coordinates, and world coordinates.

1) World Coordinates:

The world coordinate system, also known as the measurement coordinate system, is a 3D Cartesian coordinate system representing absolute coordinates in the 3D world. It serves as a reference for describing the spatial positions of cameras and objects of interest. The origin of the

world coordinate system can be freely determined based on practical considerations. Points in the world coordinate system are represented as P_W $(X_W, Y_W, Z_W)^T$.

2) Camera Coordinates:

The camera coordinate system is 3D Cartesian coordinate system. Unlike the pin-hole camera model, where the camera coordinate system has its origin at the camera's optical center, the spherical panoramic camera coordinate system places its origin at the center of the sphere. The X-axis points forward, the Y-axis rotates counterclockwise 90° from the X-axis, and the Z-axis points upward. Points in the camera coordinate system are represented as P_C $(X_C, Y_C, Z_C)^T$.

3) Spherical Coordinates:

The spherical coordinate system is a 3D space coordinate system with its origin overlapping the camera coordinate system's origin. The axes align with those of the camera coordinate system, but points are represented using radius r and angles (θ and φ). Given a point, the line connecting it to the origin is called the radial line. The length of this radial line is the radius r. The angle between the radial line and its projection onto the X-Y plane is the polar angle (θ), and the projection's angle with respect to the X-axis is the azimuth angle (φ). Points in the spherical coordinate system are represented as $(r, \theta, \varphi)^T$. Since all the spherical coordinates are on the sphere, it can also be represented as p $(x, y, z)^T$.

4) Image Coordinates:

The image coordinate system is a 2D coordinate system with its origin at the top-left corner of the image plane. Image coordinates are generated from spherical coordinates using equirectangular mapping (also known as cylindrical equidistant projection). Points in the image coordinate system are represented as $(u, v)^T$.

3.2.2 Spherical panoramic projection model

Given the diverse environments we encounter in our task (including various indoor and outdoor scenes), and considering that we only have uncalibrated spherical panoramic images without additional information such as depth, GPS, or focal length, we can only obtain a relative pose up to scale. However, this relative pose is sufficient for connecting and navigating these panoramic images. For simplicity and without affecting subsequent results, we can assume that the sphere is a unit sphere.

Mapping a 3D point P_W $(X_W, Y_W, Z_W)^T$ from the world coordinates to the 2D image coordinate system $(u, v)^T$ can be broken down into five steps, as shown in Fig. 3.5. Spherical panoramic projection model is shown in Fig. 3.6.

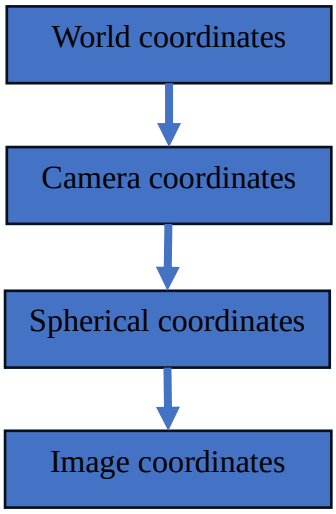


Fig. 3. 5. The spherical panoramic imaging process

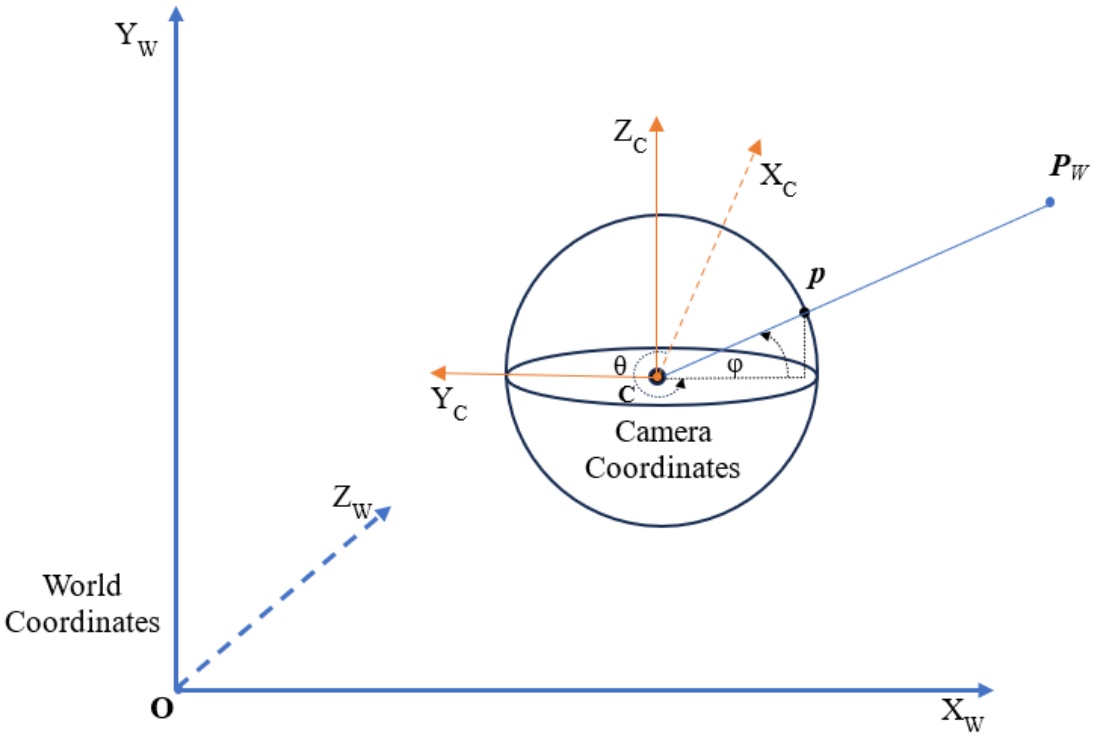


Fig. 3. 6. Spherical panoramic projection mode. The camera frame C is at the center of the sphere, with the X-axis pointing forward ($\theta = 0$), the Y-axis pointing left ($\theta = \pi/2$), and the Z-axis pointing upwards ($\varphi = \pi$).

1) Transformation from world coordinates to camera coordinates

The transformation from world coordinates to camera coordinates involves rigid body transformations. Converting a 3D Cartesian coordinate system from one reference frame to another can be achieved through a combination of rotation and translation. A single rotation aligns the axes of the two coordinate systems, while a translation moves one origin to coincide with the other. Specifically, if we have a point P_W $(X_W, Y_W, Z_W)^{\mathrm{T}}$ in the world coordinate system, its corresponding point as P_C $(X_C, Y_C, Z_C)^{\mathrm{T}}$ in the camera coordinate system can be expressed as follows:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} + \begin{bmatrix} t_0 \\ t_1 \\ t_2 \end{bmatrix} \qquad (3-1)$$

The rotation matrix is an orthogonal matrix with the determinant 1. Let:

$$R = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}, \quad t = \begin{bmatrix} t_0 \\ t_1 \\ t_2 \end{bmatrix}$$

We can simplify the formula (3-1). The relationship between camera coordinates and world coordinates is the following:

$$\boldsymbol{P_C} = \boldsymbol{R}\,\boldsymbol{P_W} + \boldsymbol{t} \qquad (3-2)$$

2) Transformation from camera coordinates to spherical coordinates

Unlike pin-hole cameras, the camera center of a spherical panoramic image is located at the center of the sphere. The camera coordinate system is projected onto a unit sphere[14], and the projected point intersects the sphere's center and its surface, denoted as point p $(x, y, z)^{\mathrm{T}}$.

$$\boldsymbol{p} = \frac{\boldsymbol{P_C}}{\|\boldsymbol{P_C}\|} = \frac{\boldsymbol{R}\,\boldsymbol{P_W} + \boldsymbol{t}}{\|\boldsymbol{R}\,\boldsymbol{P_W} + \boldsymbol{t}\|} \qquad (3-3)$$

This model is quite general—it applies to catadioptric cameras, as well as spherical panoramas obtained through stitching [63]. The model abstracts away from the specific details of how the input images were acquired.

The point $\boldsymbol{p}$ is represented in Cartesian coordinates. φ is the angle between the projection of $\overrightarrow{pC}$ and the $\boldsymbol{Z_C}$ axis, while θ is the angle between the projection and X_C axis, in a counterclockwise direction, as shown in Fig. 3.6. The p $(x, y, z)^{\mathrm{T}}$ can be represented by $(r, \theta, \varphi)^{\mathrm{T}}$, where r is 1.

$$\begin{cases} \mathrm{x} = \cos\varphi \cos\theta \\ \mathrm{y} = \cos\varphi \sin\theta \\ \mathrm{z} = \sin\varphi \end{cases} \qquad (3-4)$$

And

$$\begin{cases} \varphi = \sin^{-1} z \\ \theta = \tan^{-1} \dfrac{y}{x} \end{cases} \tag{3-5}$$

3) Transformation from spherical coordinates to image coordinates

The image pixel coordinate axes are defined by $(u,\ v)^{\mathrm{T}}$, with u going horizontally and v vertically. The image coordinates $(u,\ v)^{\mathrm{T}}$ correspond directly to points on the viewing sphere $(\theta,\ \varphi)^{\mathrm{T}}$ as shown in Fig. 3.7. The relationship between sphercial coordinates and image coordinates is the following:

$$\begin{cases} u = \dfrac{W\ (\pi\ -\ \theta)}{2\pi} \\ v = \dfrac{H\ (\pi\ -\ 2\varphi)}{2\pi} \end{cases} \tag{3-6}$$

Where *W, H* are the width and height of the image in pixel respectively and *W = 2H*.

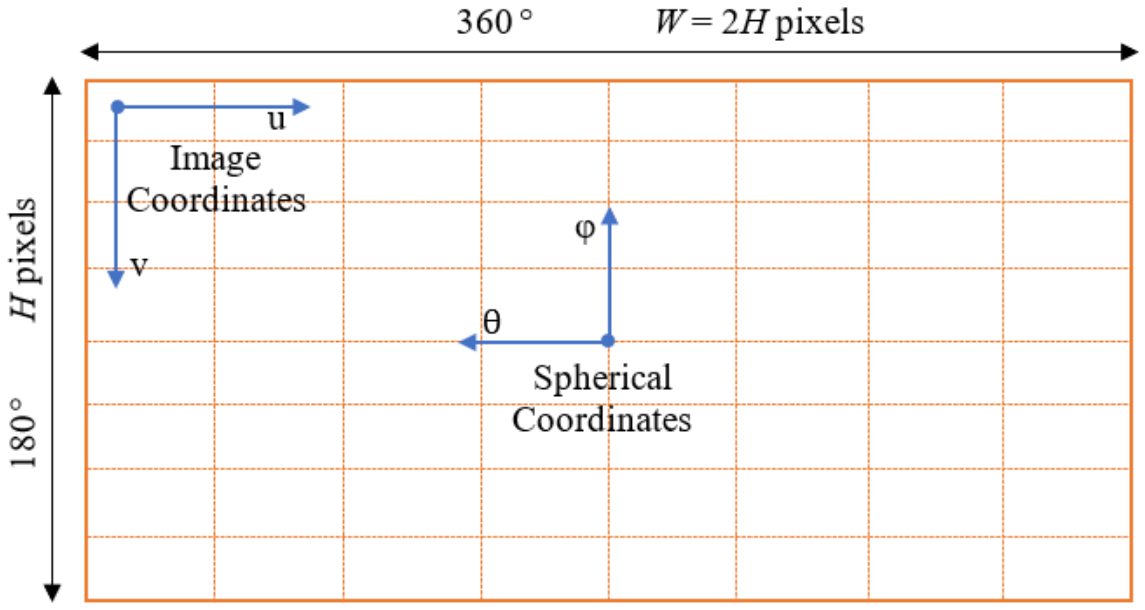


Fig. 3. 7. The relationship between sphercial coordinates and image coordinates

3.3 Epipolar Geometry of Spherical Panoramic System

In pose estimation, epipolar geometry plays a crucial role. In the pin-hole camera model, epipolar constraints significantly simplify image matching. Similarly, in spherical panoramic systems, epipolar geometry fully describes the relationship between two spherical panoramas, as shown in Fig. 3.8.

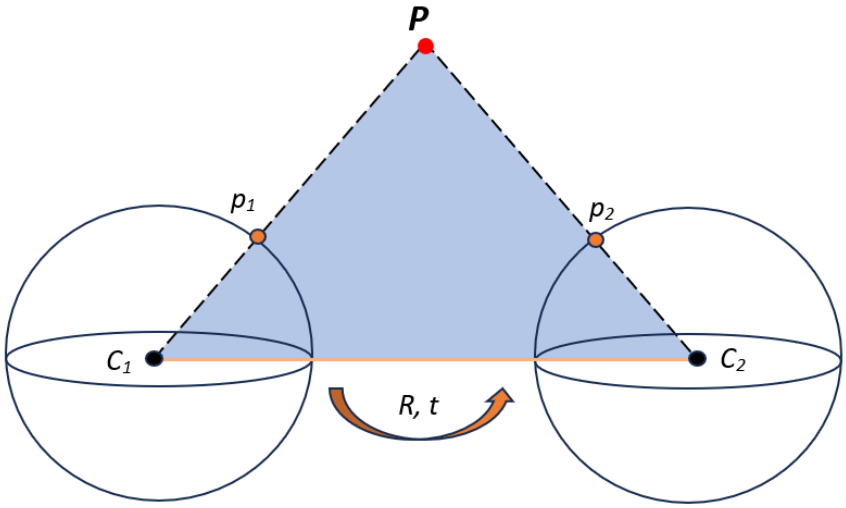


Fig. 3. 8. The general setup of epipolar geometry

For two cameras, p_1 and p_2 are the projection of P_W on the two unit sphere respectively, as shown in Fig. 3.8. Acoording equation 2-3, there are two equations:

$$\begin{cases} \boldsymbol{p_1} = \dfrac{\boldsymbol{R_1 P_W} + \boldsymbol{t_1}}{\|\boldsymbol{R_1 P_W} + \boldsymbol{t_1}\|} \\ \boldsymbol{p_2} = \dfrac{\boldsymbol{R_2 P_W} + \boldsymbol{t_2}}{\|\boldsymbol{R_2 P_W} + \boldsymbol{t_2}\|} \end{cases} \qquad (3-7)$$

Let the right camera be the original coordinate system. This means

$$\begin{aligned} R_2 &= I_{3\times 3} \\ t_2 &= 0_{3\times 1} \end{aligned} \qquad (3-8)$$

Substitue (3-8) into (3-7), $\boldsymbol{p_2}$ can be represented as $\boldsymbol{p_2} = \frac{\boldsymbol{P_W}}{\|\boldsymbol{P_W}\|}$, Assum $s = \|P_W\|$. The world coordinate $\boldsymbol{P_W}$ with respect to spherical coordinate $\boldsymbol{p_2}$ is:

$$\boldsymbol{P_W} = \boldsymbol{s}\,\boldsymbol{p_2} \qquad (3-9)$$

Substitute (3-9) into the above formula (3-7).

$$\boldsymbol{p_1} = \frac{\boldsymbol{R_1 s\, p_2} + \boldsymbol{t_1}}{\|\boldsymbol{R_1 s\, p_2} + \boldsymbol{t_1}\|} \qquad (3-10)$$

Cross product by $\boldsymbol{t_1}$ on both sides,

$$\boldsymbol{t_1} \times \boldsymbol{p_1} = \frac{\boldsymbol{t_1} \times \boldsymbol{R_1 s\, p_2} + \boldsymbol{t_1} \times \boldsymbol{t_1}}{\|\boldsymbol{R_1 s\, p_2} + \boldsymbol{t_1}\|} \qquad (3-11)$$

Since the cross product by itself is 0, organize the formula (2-11) and multiply by $\boldsymbol{p_1^T}$ on both sides,

$$\boldsymbol{p_1^T t_1} \times \boldsymbol{p_1} = \frac{\boldsymbol{p_1^T t_1} \times \boldsymbol{R_1 s\, p_2}}{\|\boldsymbol{R_1 s\, p_2} + \boldsymbol{t_1}\|} \qquad (3-12)$$

Assume $\boldsymbol{p_1} = [x \quad y \quad z]^T$, $\boldsymbol{t_1} = [t_x \quad t_y \quad t_z]^T$, it can be proved that the $\boldsymbol{p_1^T t_1} \times \boldsymbol{p_1} = \boldsymbol{0}$

$$\boldsymbol{p_1^T t_1} \times \boldsymbol{p_1} = [x \quad y \quad z] \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0$$

Rewitten the formula (3-12)

$$\frac{\boldsymbol{p_1^T t_1} \times \boldsymbol{R_1 s\, p_2}}{\|\boldsymbol{R_1 s\, p_2} + \boldsymbol{t_1}\|} = \boldsymbol{0} \qquad (3-13)$$

The *s* is the distance between P_W and the center of the right camera, $\|\boldsymbol{R_1 s\, p_2} + \boldsymbol{t_1}\|$ is the distance between P_W and the center of the left camera. Since the point lies outside of the cameras, we can assume both of the distances can not be 0. This means that

$$\boldsymbol{p_1^T t_1} \times \boldsymbol{R_1\, p_2} = \boldsymbol{0} \qquad (3-14)$$

Let $t = t_1$, $R = R_1$, the essential matrix is defined as

$$\boldsymbol{E} = \boldsymbol{t_\times R} \qquad (3-15)$$

Where the $t_\times$ is a skew-symmetric matrix. Substitute (3-15) into (3-14), the epipolar constraint is the following.

$$\boldsymbol{p_1^T E\, p_2} = \boldsymbol{0} \qquad (3-16)$$

Where $\boldsymbol{E}$ represents the essential matrix $\in R^{3\times3}$ with rank of 2.

Chapter 4 Proposed Methodology

4.1 Experimental Design

The main objective of this research is to achieve accurate and stable relative pose estimation between two spherical panoramic images. The goal is to provide an automated direction recognition solution for indoor navigation, like Google Street View. Given the complexity and diversity of the encountered scenes, standard pose estimation methods alone cannot meet the requirements. Therefore, this study proposes a hybrid approach that combines traditional and learning-based pose estimation methods.

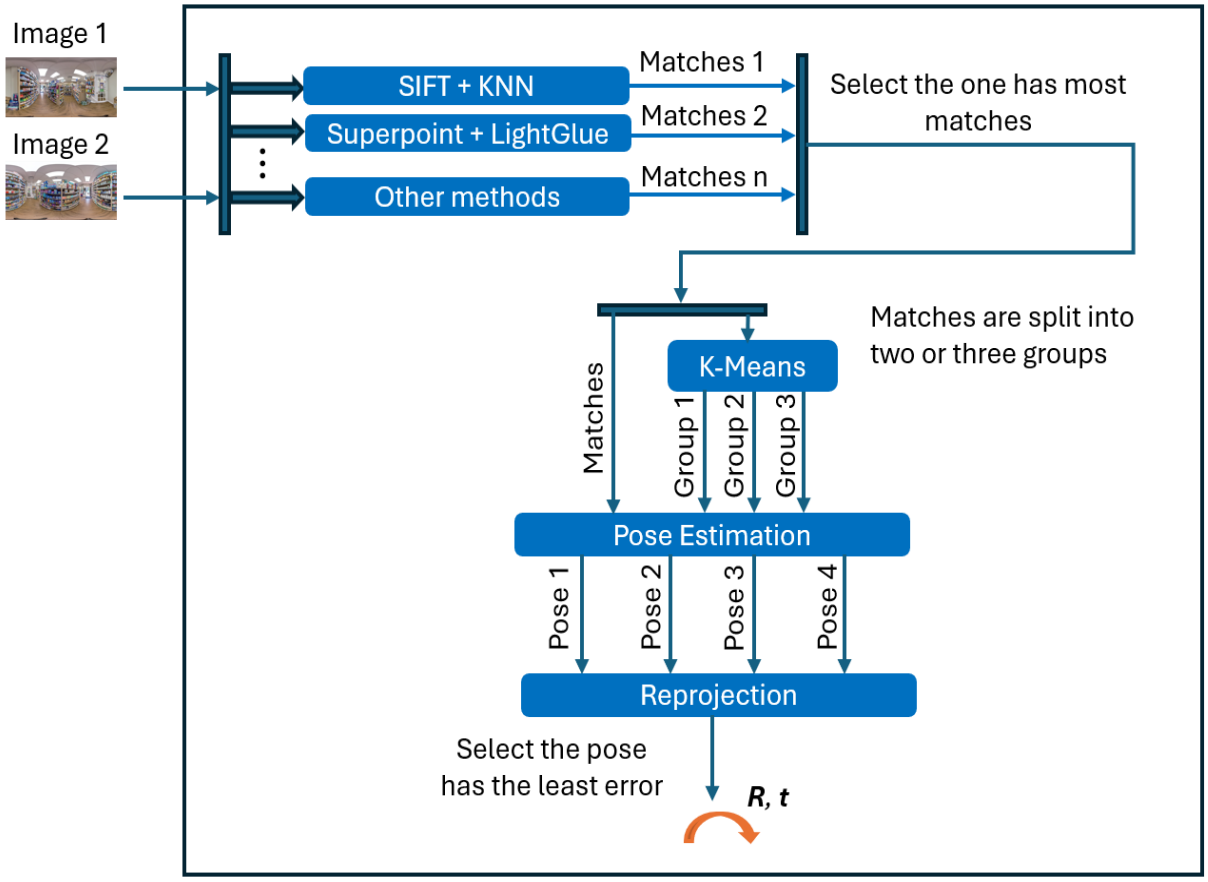


Fig. 4. 1. The pipeline of the pose estimation

The pipeline of the pose estimation algorithm is shown as Fig. 4.1. For two spherical panoramic images, we extract features using both traditional SIFT with KNN and learning based SuperPoint with LightGlue (or other high-quality keypoint detection and matching algorithms). Since it's challenging to find a keypoint detection and matching algorithm that adapts well to all environments, combining multiple methods is essential.

4.2 Spherical Keypoints Detection and Matching

Keypoints detection and matching have always been fundamental and critical techniques in various computer vision applications. Image matching allows us to identify corresponding or similar structures/content between two or more images. This field has been continuously evolving, and over the past few decades, researchers have proposed an increasing number of methods. These techniques demonstrate good accuracy across different application environments.

4.2.1 Keypoints Detection and Matching methods

Keypoints or points of interest are specific locations in an image, such as corners, edges, and regions. However, due to variations in shooting angles, distances, lighting conditions, and other factors between two images, these seemingly robust points can change. Therefore, researchers have designed various stable image features to achieve robust matching of the same points.

1) Some keypoints detection methods

Scale-Invariant Feature Transform (SIFT) is a widely used descriptor in the field of image processing. It exhibits scale invariance and can detect keypoints in an image. SIFT is a local feature descriptor that identifies extrema points in spatial scales and extracts their position, scale, and rotation invariants. David Lowe introduced this algorithm in 1999 and further refined it in 2004. Its applications span various domains, including object recognition, robot map perception and navigation, image stitching, 3D model construction, gesture recognition, image tracking, and action matching. SIFT essentially searches for keypoints (feature) at different spatial scales and computes their orientations. The keypoints identified by SIFT are robust against factors such as lighting variations, affine transformations, and noise, including corner points, edge points, bright points in dark regions, and dark points in bright regions.

Oriented Fast and Rotated Brief (ORB) is a method for quickly creating feature vectors for keypoints in an image. It uses the FAST algorithm to find keypoints and generates descriptors using BRIEF. ORB is somewhat insensitive to viewpoint changes and image transformations (such as rotation and scaling). Its standout feature is its high speed.

Spherical ORB（SPHORB）is an improved version of ORB specifically designed for detecting keypoints and extracting binary features in spherical panoramic images. SPHORB features exhibit some scale and rotation invariance.

SuperPoint is a feature detection and descriptor extraction method based on self-supervised training. Its fully convolutional model operates on full-sized images and simultaneously computes pixel-level interest point locations and associated descriptors in a single forward pass. SuperPoint introduces Homographic Adaptation, a multi-scale, multi-homography method that enhances interest point detection repeatability and enables cross-domain adaptation.

2) Some matching methods

K-Nearest Neighbor (KNN) is one of the simplest machine learning algorithms. It is theoretically well-established and can be used for both classification and regression tasks. KNN is a supervised learning algorithm. The basic idea is as follows: if a sample's K most similar (i.e., nearest) neighbors in feature space predominantly belong to a certain class, then that sample is also assigned to that class. In other words, KNN makes classification decisions based on the class labels of the nearest neighbors.

LightGlue is an upgraded version of SuperGlue. It leverages attention mechanisms based on Transformers to simulate human feature matching. Compared to traditional methods, deep learning-based image matching approaches offer several advantages, such as robustness against texture variations caused by weak textures, lighting changes, or viewpoint variations.

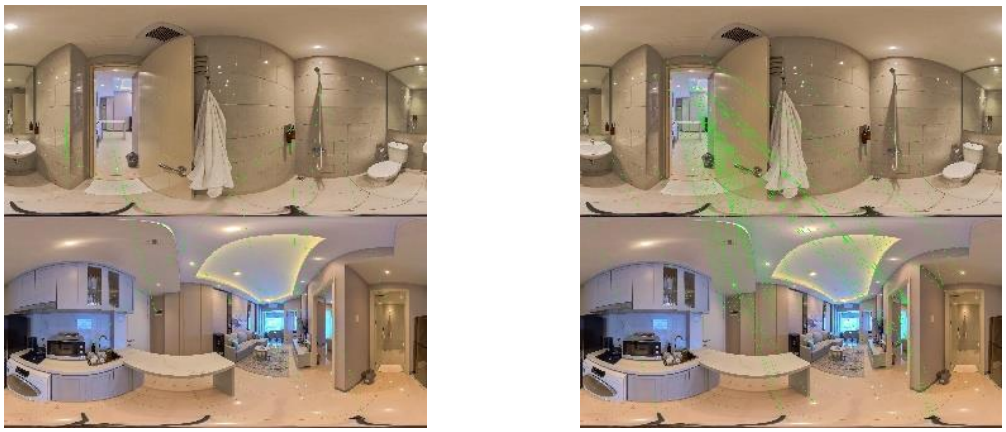
4.2.2 Method selection

In this section, different methods for detecting image feature points and performing feature matching are discussed. For the task of pose estimation in this context, a crucial prerequisite is to obtain enough correct correspondences in diverse environments. Considering the diversity of environments, there isn't a single method that performs well universally. Therefore, this article proposes using multiple feature extraction and matching algorithms simultaneously, while also providing an interface for adding more such algorithms.

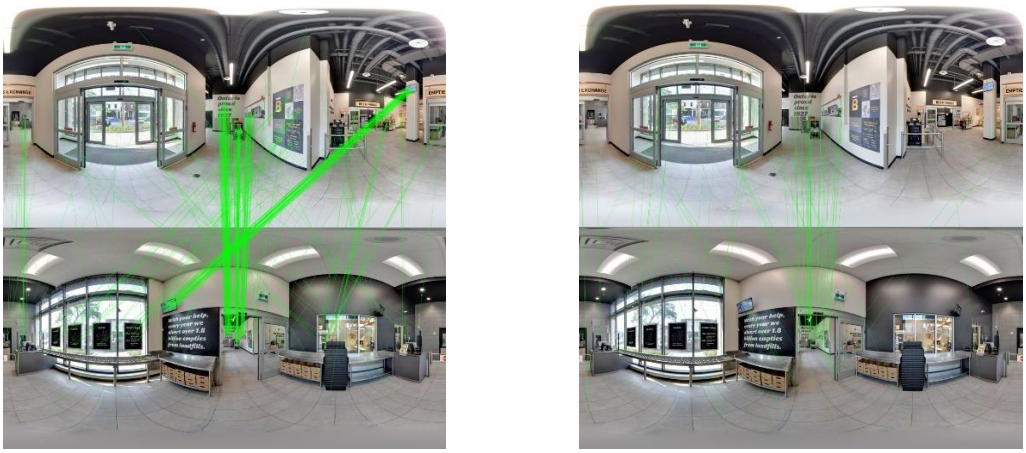
After testing various corner cases, the article selects two feature extraction methods with strong adaptability: SIFT and SuperPoint. For matching algorithms, KNN is used in conjunction with SIFT, while LightGlue is used alongside SuperPoint.

By combining these methods, the research aims to improve pose estimation results across different environments. This approach allows for flexibility and customization based on specific user requirements and scenarios.

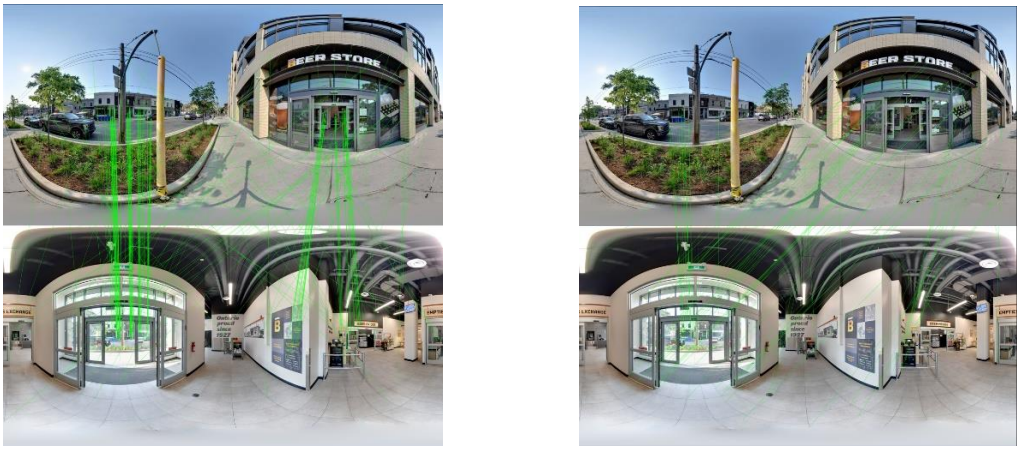
(a)



(b)



(c)



(d)

Fig. 4. 2. The matching results of two methods in different environments. Left is SIFT + KNN and right is SuperPoint + LightGlue.

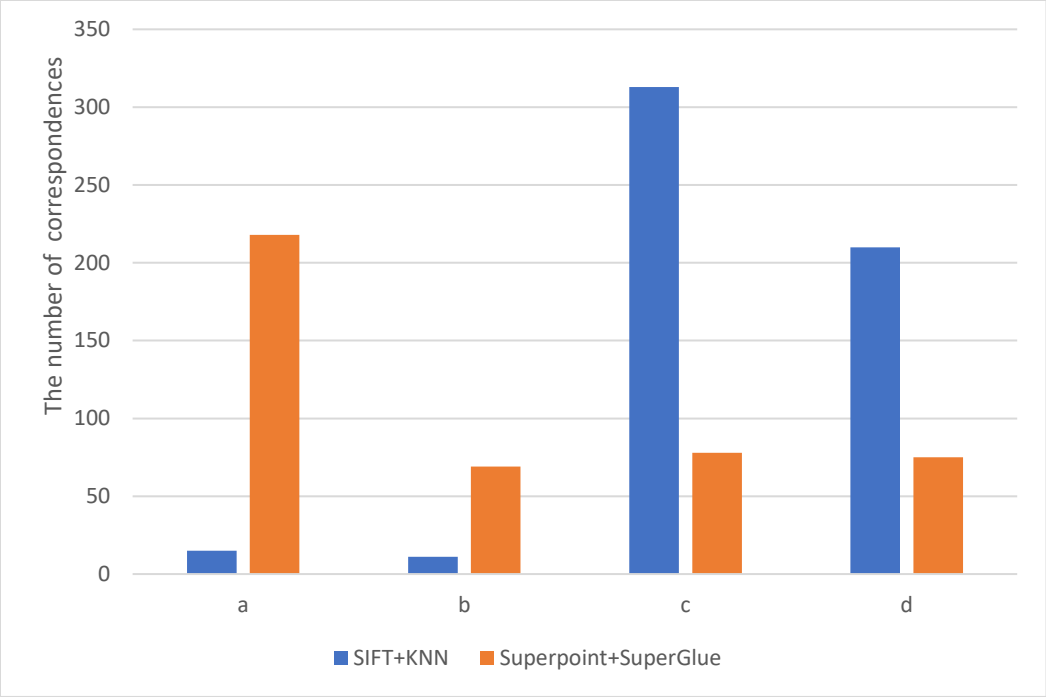


Fig. 4. 3. The number of correspondences in different environments

As shown in Fig. 4.2, (a) are the matching results of two methods in a same room with lighting change. (b) are the matching results in two weak texture rooms. (c) are the matching results in different rooms with lighting change. (d) are the matching results of two views including an outdoor scene and an indoor scene. The number of correspondences in different environments is shown in Fig. 4.3. Different methods perform differently in various environments, and each has its own strengths and weaknesses. SIFT + KNN can only obtain very few matching points in environments with weak textures, while LightGlue is sensitive to lines and can still provide sufficient matching points even in relatively smooth environments. Conversely, LightGlue does not perform as well as SIFT in environments with strong textures. However, in all the tested cases, using the method with more matching points consistently yielded correct results. Through this testing, combining both methods and selecting the one with more matching points effectively leveraged their complementary advantages and achieved good results.

4.3 Robust Pose Estimation for Spherical Panorama

In Chapter 2, we studied the epipolar geometry of spherical panoramas and identified geometric constraints. Unlike perspective geometry, spherical geometry does not involve intrinsic parameters, and we can consider the essential matrix and the fundamental matrix to be the same. Using epipolar constraints, we estimate the essential matrix using the 8-point method. Pose estimation primarily involves three steps: 1） Find enough distinct matching points. These points are then mapped onto the unit spheres. 2) 8-PA solve the essential matrix. 3) Recover pose from essential matrix by performing SVD on it.

4.3.1 Pose Estimation

1) Estimation of the essential matrix

As shown in Fig. 3.8, where $\boldsymbol{p_1}$, $\boldsymbol{p_2}$ are corresponding spherical projection points on the two-unit spheres. Given $\boldsymbol{p_1} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{y_1} \\ \boldsymbol{z_1} \end{bmatrix}$ and $\boldsymbol{p_2} = \begin{bmatrix} \boldsymbol{x_2} \\ \boldsymbol{y_2} \\ \boldsymbol{z_2} \end{bmatrix}$, according to the epipolar constraint, we can write the equation (3-16) as follows.

$$[x_1 \quad y_1 \quad z_1] \begin{bmatrix} e_{00} & e_{01} & e_{02} \\ e_{10} & e_{11} & e_{12} \\ e_{20} & e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = 0 \qquad (4-1)$$

Reorganize the above equation,

$$[x_1 x_2 \quad x_1 y_2 \quad x_1 z_2 \quad y_1 x_2 \quad y_1 y_2 \quad y_1 z_2 \quad z_1 x_2 \quad z_1 y_2 \quad z_1 z_2] \begin{bmatrix} e_{00} \\ e_{01} \\ e_{02} \\ e_{10} \\ e_{11} \\ e_{12} \\ e_{20} \\ e_{21} \\ e_{22} \end{bmatrix} = 0 \qquad (4-2)$$

For n matches, we will obtain a set of linear equations of the form.

$$\begin{bmatrix} x_1^1 x_2^1 & x_1^1 y_2^1 & x_1^1 z_2^1 & y_1^1 x_2^1 & y_1^1 y_2^1 & y_1^1 z_2^1 & z_1^1 x_2^1 & z_1^1 y_2^1 & z_1^1 z_2^1 \\ x_1^2 x_2^2 & x_1^2 y_2^2 & x_1^2 z_2^2 & y_1^2 x_2^2 & y_1^2 y_2^2 & y_1^2 z_2^2 & z_1^2 x_2^2 & z_1^2 y_2^2 & z_1^2 z_2^2 \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ x_1^n x_2^n & x_1^n y_2^n & x_1^n z_2^n & y_1^n x_2^n & y_1^n y_2^n & y_1^n z_2^n & z_1^n x_2^n & z_1^n y_2^n & z_1^n z_2^n \end{bmatrix} \begin{bmatrix} e_{00} \\ e_{01} \\ e_{02} \\ e_{10} \\ e_{11} \\ e_{12} \\ e_{20} \\ e_{21} \\ e_{22} \end{bmatrix} = 0 \qquad (4-3)$$

Simply the formula (4-3) as

$$\boldsymbol{Ae} = \boldsymbol{0} \qquad (4-4)$$

Obviously, the equation (4-4) above has a total of 9 unknowns (the essential matrix item). If $rank(A) = 9$, then there is only a 0 solution. However, if $rank(A) < 9$, the equation will have a non-zero solution. For a system of homogeneous linear equations with *n* unknowns, the fundamental system of solutions contains $n - rank(A)$ vectors. Since we can only know the essential matrix up to scale, we need at least eight points match to determine the essential matrix, which is n >=8. In practice, to reduce the impact of noisy measurements, it is often better to use more than 8 correspondences and create larger $\boldsymbol{A}$ matrix.

Since scale is indifferent, multiplying both sides of the formula (4-4) by a constant *k* does not change the result.

$$Ae = A\,[ke] = 0 \qquad (4-5)$$

It means that e and ke describe the same epipolar geometry. For convenient, we can impose another constraint.

$$\|e\| = 1$$

The solution vector e found in this way. We want Ae as close to 0 as possible.

$$\text{Minimize } \|Ae\|^2 \quad s.t. \|e\|^2 = 1 \qquad (4-6)$$

The solution to this system can be solved by Singular Value Decomposition (SVD) or other methods. It's a constrained linear least squares problem, here are two solutions: First, $\boldsymbol{e}$ is an eigenvector of $\boldsymbol{A^T A}$ associated to its smallest eigenvalue. Second, $\boldsymbol{e}$ is the singular vector corresponding to the smallest singular value of $\boldsymbol{A}$, that is, the last column of V in the SVD

$$\boldsymbol{A = UD\,V^T}$$

Both solutions will get the same result. Rearrange the vector $\boldsymbol{e}(9 \times 1)$ to get the fundamental matrix $\boldsymbol{E}(3 \times 3)$.

We know the true essential matrix $\boldsymbol{E}$ has the rank 2. However, due to the noise, the above solutions will give us an estimate of the essential matrix $\boldsymbol{\hat{E}}$, which may have full rank.

The usual approach is to enforce rank 2 of $\boldsymbol{E}$, we can decompose it as SVD, put $\sigma_3 = 0$ and recompose.

$$\hat{E} = \boldsymbol{U\,D\,V^T} \qquad (4-7)$$

Where $\boldsymbol{D}$ *is a* (3×3) diagonal matrix, and the diagonal elements are $\sigma_1, \sigma_2, \sigma_3$.

$$\boldsymbol{D} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$$

Note $\sigma_1 \geq \sigma_2 \geq \sigma_3$. Then, put $\sigma_3 = 0$

$$\boldsymbol{D_{new}} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4-8)$$

Finally, substitute (4-8) into (4-7), the best rank-2 approximation is found by.

$$\boldsymbol{E} = \boldsymbol{U}\, \boldsymbol{D_{new}}\, \boldsymbol{V^T} \quad (4-9)$$

2) Decomposition of the essential matrix

Essential matrix is defined by (3-15). It contains the camera pose *R, t*, so we need to decompose the essential matrix. According (3-15), $t_\times$ is a skew symmetric matrix. The part of the properties of the *E* inherited from the skew symmetric matrix. It' known that *A* 3×3 *matrix is an essential matrix if and only if two of its singular values are equal, and the third is zero*[15].

According to the above result, wo could have

$$\boldsymbol{E} = \boldsymbol{U}\, \text{diag}\,(1, 1, 0)\, \boldsymbol{V^T} \quad (4-10)$$

Note: The essential matrix is up to scale. Given a property of skew symmetric matrix, the 4 × 3 skew-symmetric matrix $t_\times$ can be decomposed into

$$\boldsymbol{t_\times} = \boldsymbol{k}\, \boldsymbol{U}\, \boldsymbol{Z}\, \boldsymbol{U^T} \quad (4-11)$$

Where $\boldsymbol{k}$ is a constant, $\boldsymbol{U}$ is orthogonal, *and* $\boldsymbol{Z}$ is skew-symmetric.

$$\boldsymbol{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Introduce an orthogonal matrix $\boldsymbol{W}$.

$$\boldsymbol{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

It's noticed.

$$\boldsymbol{Z}\, \boldsymbol{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = -\,\text{diag}\,(1, 1, 0)$$

$$\boldsymbol{Z}\, \boldsymbol{W^T} = \text{diag}\,(1, 1, 0)$$

Using the previous notation and property, up to scale, $\boldsymbol{E}$ is reconstructed as follows.

$$\boldsymbol{E} = \boldsymbol{U}\, \text{diag}\,(1, 1, 0)\, \boldsymbol{V^T} = \boldsymbol{U}\, \boldsymbol{Z}\, \boldsymbol{W^T}\, \boldsymbol{V^T} = \boldsymbol{U}\, \boldsymbol{Z}(\boldsymbol{U^T U})\, \boldsymbol{W^T V^T} \quad (4-12)$$

Note $\boldsymbol{U^T U} = I$. In (4-12), $\boldsymbol{UZU^T}$ is a skew-symmetric matrix, and $\boldsymbol{U\,W^T V^T}$ is an orthogonal matrix, as shown in Fig. 4.4.

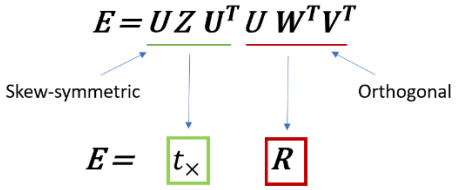


Fig. 4. 4. The decomposition form of the essential matrix

Up to sign, there are four possible solutions:

$$(1)\quad \boldsymbol{t_\times = UZU^T},\ \boldsymbol{R = UW^T V^T}$$

$$(2)\quad \boldsymbol{t_\times = UZU^T},\ \boldsymbol{R = U\,WV^T}$$

$$(3)\quad \boldsymbol{t_\times = UZ^T U^T},\ \boldsymbol{R = U\,W^T V^T}$$

$$(4)\quad \boldsymbol{t_\times = UZ^T U^T},\ \boldsymbol{R = U\,WV^T}$$

The rotation matrix does not need to further decompose. The translation vector t can be decomposed from $\boldsymbol{t_\times}$. Since the cross product by itself is 0, for $\boldsymbol{t_\times = U\,Z\ U^T}$

$$[\boldsymbol{t_\times}]\, t = \boldsymbol{U\ Z\ U^T}\, t = 0$$

Multiply both sides of the above equation by $\boldsymbol{U^T}$.

$$\boldsymbol{U^T U\ Z\ U^T} t = \boldsymbol{U^T} 0 \qquad (4-13)$$

Sine $\boldsymbol{U}$ is orthogonal matrix, $\boldsymbol{U^T U} = I$. Substitute it into (3-13),

$$\boldsymbol{Z}\,(\boldsymbol{U^T} t) = 0 \qquad (4-14)$$

Assume $\boldsymbol{z} = [0, 0, 1]^{\boldsymbol{T}}$, then $\boldsymbol{z_\times = Z}$, whereas $\boldsymbol{z_\times\, z} = 0$, the solution of $(\boldsymbol{U^T} t)$ can be extracted from (4-14),

$$\boldsymbol{U^T} t = z \qquad (4-15)$$

Multiply both sides of the above equation by $\boldsymbol{U}$, **the** $\boldsymbol{t}$ **is solved**.

$$\mathrm{t} = \boldsymbol{U}\, z = \boldsymbol{U}\ [0, 0, 1]^{\boldsymbol{T}} = \boldsymbol{U.}\ col\ (3)$$

Same way, for $\boldsymbol{t_\times = U\ Z^T\ U^T}$,

$$\mathrm{t} = \boldsymbol{U}\ [0, 0, -1]^{\boldsymbol{T}} = -\,\boldsymbol{U.}\ col\ (3)$$

3) Find the correct pose

After decomposition of the essential matrix, there are four possible combinations of translation vector and rotation matrix. In the perspective case, if the triangulated point $\boldsymbol{P}$ is in front of the two cameras, it is the correct set of solutions. However, for the spherical case, we introduce a new method to find the correct pose. The relationship between camera coordinates $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ in terms of $\boldsymbol{R}$, $\boldsymbol{t}$ is the following.

$$\boldsymbol{P_1} = \boldsymbol{R}\boldsymbol{P_2} + \boldsymbol{t} \tag{4-16}$$

According to (2-3), the spherical coordinates $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ can be represented as

$$\begin{cases} \boldsymbol{p_1} = \dfrac{\boldsymbol{P_1}}{\|\boldsymbol{P_1}\|} \\ \boldsymbol{p_2} = \dfrac{\boldsymbol{P_2}}{\|\boldsymbol{P_2}\|} \end{cases} \tag{4-17}$$

Substitute (4-17) into (4-16),

$$\boldsymbol{p_1} * \|\boldsymbol{P_1}\| = \boldsymbol{R}\boldsymbol{p_2} * \|\boldsymbol{P_1}\| + \boldsymbol{t} \tag{4-18}$$

Note: $\|\boldsymbol{P_1}\|$ and $\|\boldsymbol{P_2}\|$ should greater than 0. Reorganize the formula (4-18)

$$\boldsymbol{p_1} * \|\boldsymbol{P_1}\| - \boldsymbol{R}\boldsymbol{p_2} * \|\boldsymbol{P_1}\| = \boldsymbol{t} \tag{4-19}$$

Transfer the above equation to matrix form,

$$[\boldsymbol{p_1} \quad -\boldsymbol{R}\boldsymbol{p_2}] \begin{bmatrix} \|\boldsymbol{P_1}\| \\ \|\boldsymbol{P_2}\| \end{bmatrix} = t \tag{4-20}$$

This is an overdetermined system of linear equations. We can use the pseudo inverse get the least squares solution. Assume $\boldsymbol{A} = [\boldsymbol{p_1} \quad -\boldsymbol{R}\boldsymbol{p_2}]$, $\boldsymbol{x} = [\|\boldsymbol{P_1}\|, \|\boldsymbol{P_2}\|]^{\boldsymbol{T}}$, The equation (4-20) can be rewritten as follows.

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{t} \tag{4-21}$$

Where $\boldsymbol{A}$ and $\boldsymbol{t}$ are known.

$$\boldsymbol{x} = (\boldsymbol{A^T A})^{-1}\boldsymbol{A^T t} \tag{4-22}$$

The correct pair of *R* and *t* should have the solution that both greater than 0 ($x > 0$). For all the correspondences, the correct pair of *R* and *t* should have the most correct solutions.

4.3.2 Result improvement

In practice, image matching points often contain a significant amount of noise (outliers). Without removing these outliers, it is challenging to obtain accurate and stable results. In typical pose estimation algorithms, RANSAC is commonly used to achieve robust estimates. However, to ensure robustness across various environments, this paper goes beyond using RANSAC alone.

We introduce an approach to preprocess the correspondences by grouping them based on their distribution on a sphere. By doing so, the goal is to reduce the proportion of outliers in the data.

1）New error function for RANSAC

RANSAC is an iterative algorithm used to accurately estimate mathematical model parameters from a set of data that contains outliers. These “outliers” typically refer to noise in the data, such as mismatches in feature matching or outliers in the estimated curve. Therefore, RANSAC can also be considered an outlier detection algorithm. RANSAC is a probabilistic algorithm. It produces results with a certain probability, which increases as the number of iterations grows. The more iterations performed, the higher the likelihood of obtaining a correct result.

RANSAC typically requires two components: fitting Model and error function. The fitting model generates a hypothesis based on a random subset of data points. The error function determines which points are considered inliers (consistent with the model) and which are outliers. The process involves fitting the model to a random subset, evaluating the error, and determining inliers and outliers based on a threshold. After multiple iterations, the model with the most inliers is selected as the final estimate.

In RANSAC, setting an appropriate threshold for error function is crucial during outlier removal. If the threshold is too large, valid inliers may be incorrectly classified as outliers. If the threshold is too small, true outliers may be considered inliers, leading to incorrect results. To reduce this effect, The study introduces a novel check model for RANSAC. This model computes the scaling factor of spherical mapping points by substituting $\boldsymbol{R}$ and $\boldsymbol{t}$ (rotation and translation) parameters. The reprojected error is then computed. Finally, the pose estimate is chosen based on the average reprojected error, aiming for the smallest value.

The error function computes the reproject error in camera coordinates according to equation (4-18).

$$\boldsymbol{error} = \boldsymbol{p_1} * \|\boldsymbol{P_1}\| - (\boldsymbol{R p_2} * \|\boldsymbol{P_1}\| + \boldsymbol{t}) \qquad (4-23)$$

For every **R** and **t**, estimate its mean error for all the correspondences.

$$\textbf{Mean error} = \frac{\boldsymbol{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=0}}^{\boldsymbol{n}} (\boldsymbol{p_1} * \|\boldsymbol{P_1}\|) - (\boldsymbol{R p_2} * \|\boldsymbol{P_2}\| + \mathbf{t}) \qquad (4-24)$$

The pose that has the least mean error is selected.

Argmin (Mean errors)

The steps for pose estimation algorithm is as follows.

1: Randomly selected at least 8 points.

2: Solve the parameters of compute essential matrix (E) by 8-PA.

3: Decompose the E to 4 possible combinations of rotation and translation (Rs, ts).

4: Compute the scaling factors $\|P_1\|$ and $\|P_2\|$ based on the (Rs, ts).

5: If the $\|P_1\|$ and $\|P_2\|$ of one correspondence are both positive, count one vote. Select the (R, t) has the most votes.

6: Compute the reproject error based on R, t, $\|P_1\|$ and $\|P_2\|$,

7: Remove the points with errors in the top 10% and recalculate the reproject error.

8: Repeat steps 1 through 7 until the iterations is reached.

9: Select the pose has the least error.

To mitigate the impact of extreme outliers on the results, this algorithm excludes points with significant errors (approximately 10%) when calculating the average reprojection error, as shown in Fig. 4.5 (excludes the error in red rectangle). Experimental results demonstrate that using this approach eliminates the need for predefining a threshold to distinguish inliers from outliers, reducing the influence of human factors and facilitating future parameter tuning.

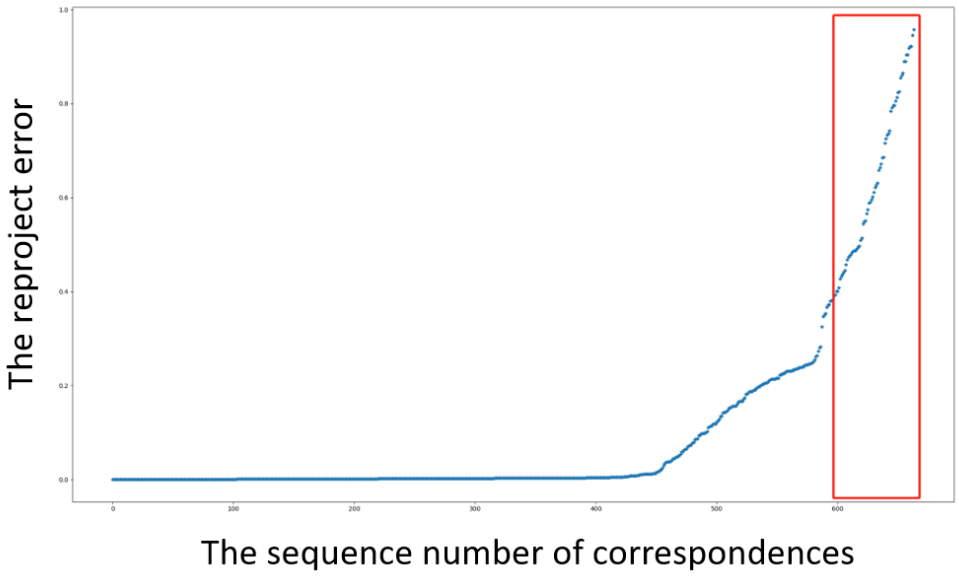
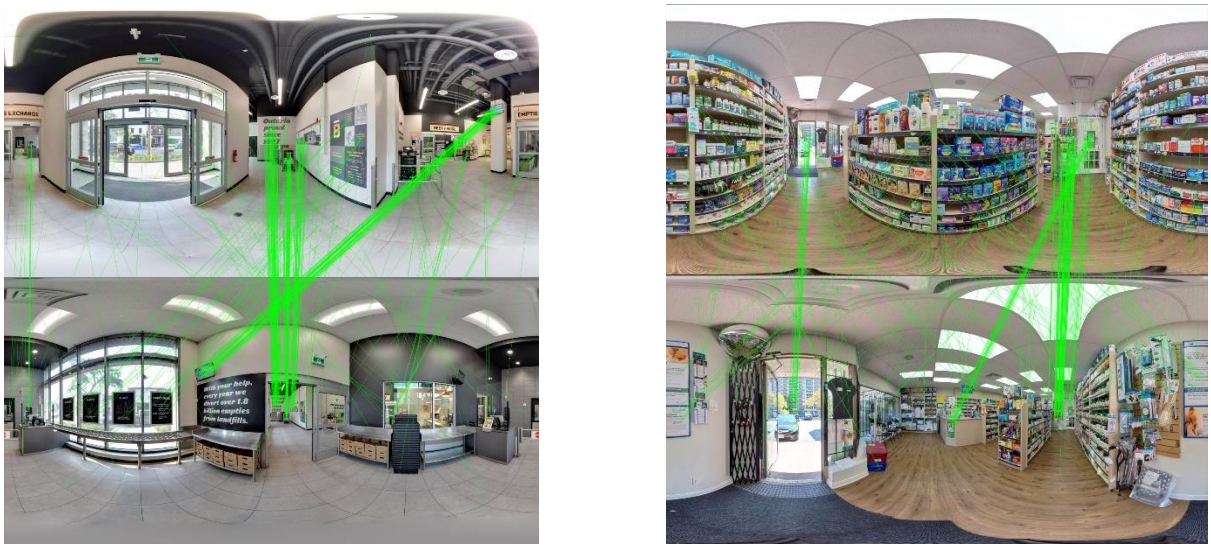


Fig. 4. 5. The reproject error from small to large.

2）Reduce the outlier's rate

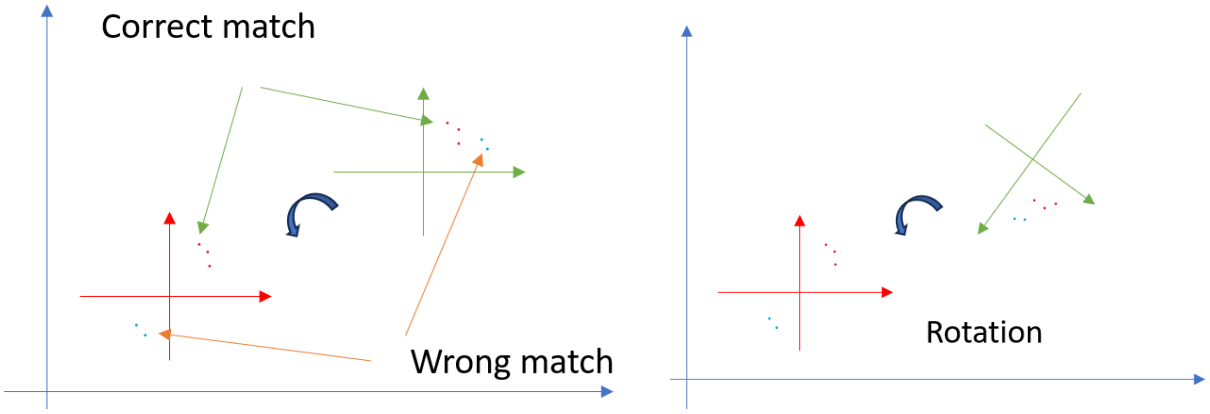
While the previous designed pose estimation algorithm has achieved good results, in scenarios with significant noise, obtaining accurate results becomes challenging due to a low inlier rate. This issue is particularly pronounced when dealing with incorrect matches. As shown in Fig. 4.6, the same scene in different monitor and the similar poster in different position result in incorrect match.



(a) (b)

Fig. 4. 6. Matching result. (a) Wrong match due to the same scene in different monitor. (b) wrong match due to the similar poster in different position.

The occurrence of incorrect matches due to similar patterns is a common and challenging issue in the field of image processing. In pose estimation, this can also lead to erroneous estimation results. Analyzing the reprojection process (illustrated using a planar coordinate system) reveals the underlying reasons for these errors.

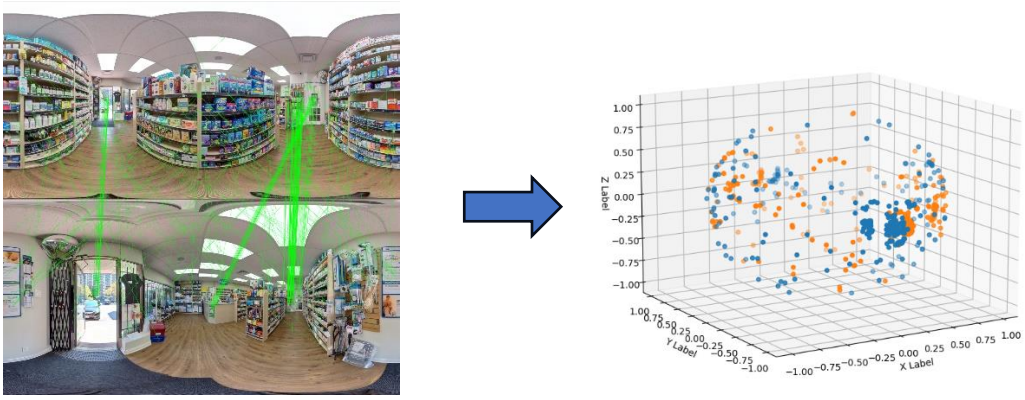


Step1 step2

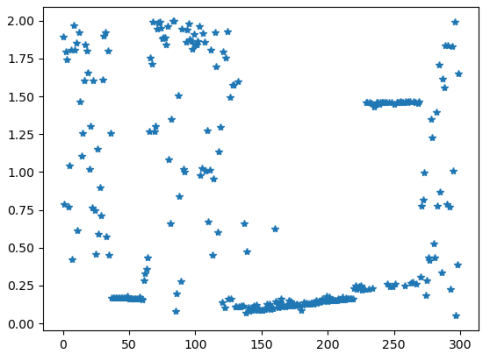
Step3 step4

Fig. 4. 7. The process of the reprojection algorithm

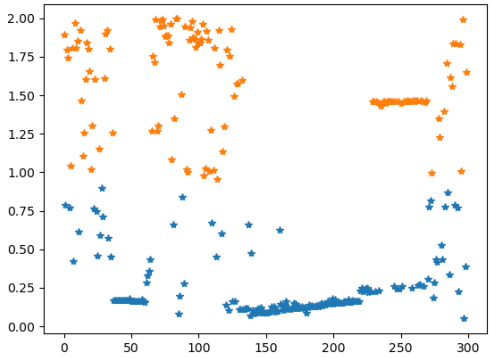
In the pose estimation algorithm proposed in this study, it can be observed that although incorrect matching points are 180° apart from the correct matching points, both the erroneous and correct correspondences achieve perfect reprojection after translation and scaling, as shown in Fig. 4.7. However, this leads to incorrect results. To address this issue, we analyze the distribution of matching points and propose a method of grouping them using K-means clustering. By doing so, we not only resolve this specific problem but also reduce the proportion of outliers in the data to some extent, allowing pose estimation to remain robust even in noisy environments. The steps are as follows: Map the image matching points onto a sphere. Calculate the distances between the correspondences. Apply K-Means clustering and spilt the correspondences into 2 to 3 groups, as shown in Fig. 4.8.



(a) (b)



(c)



(d)

Fig. 4. 8. The process of split the correspondence into groups. (a) The matching result of the two images. (b) The distribution of matches in spherical coordinate. (c) The distance between correspondences. (d) Split into two groups by K-Means with K = 2.

Chapter 5 Results and Discussion

5.1 Evaluation Datasets

We've tested more than 10 series of spherical panoramas, each containing 7 to 40 ultra-high-definition spherical images (with resolutions of up to 14,000 × 7,000 pixels). The spherical panoramas include various locations such as coffee shops, beer stores, pharmacies, restaurants, apartments, and houses. Some capture the street views, while others consist entirely of green pathways. The data comes from the website GoThru, a new company dedicated to indoor virtual navigation and immersive content. It aims to allow seamless transitions from outdoor to indoor spaces, providing comprehensive navigation. These data also offer relevant panoramic images from outside to inside, which rarely see in other datasets.

For each pair of panoramas, they provided the latitude, longitude and heading degree. We compute the ground truth for direction of the motion by using the GPS data and the heading degree.

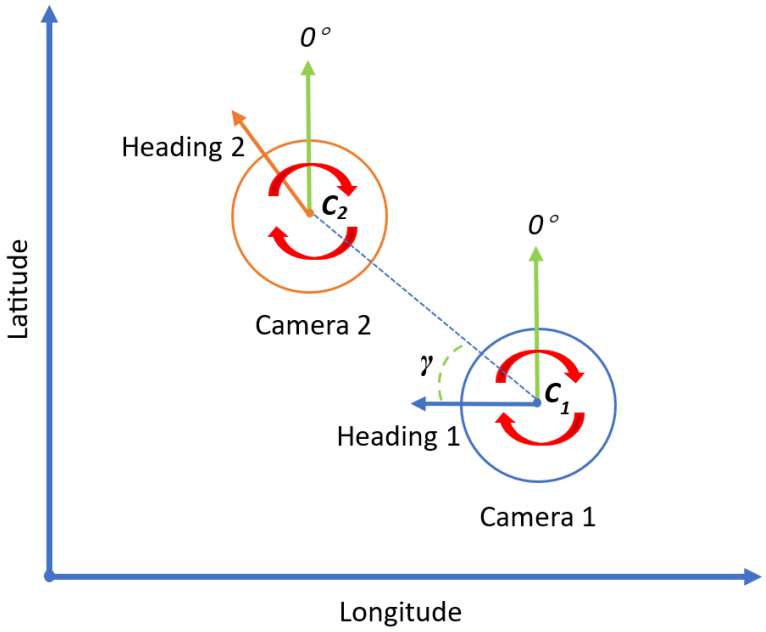


Fig. 5. 1. The illustration of the direction of the motion between the two cameras.

As shown in Fig. 5.1, the green lines toward the *0°*(*360°*), where the degree is increased clockwise. C_1 (x_1, y_1) and C_1 (x_2, y_2) are the center of the cameras. Assume the heading degree of the camera 1 is *h*, then the direction of the motion from the camera 1 to camera 2 is computed as the following:

$$\gamma = h - (360^\circ - (\tan^{-1}\frac{y_2 - y_1}{x_2 - x_1} - 90^\circ)) \qquad (5-1)$$

Simplify the above equation, the direction of the motion is,

$$\gamma = h - 450^{\circ} + \tan^{-1}\frac{y_2 - y_1}{x_2 - x_1} \qquad (5-2)$$

During image capture, tripods were used to ensure consistent heights. This consistency is evident from the estimated poses: the z of the translation vector is nearly zero, indicating minimal height variation between the two capturing devices. Consequently, we can simplify the motion to a planar translation, as shown in Fig. 5.2.

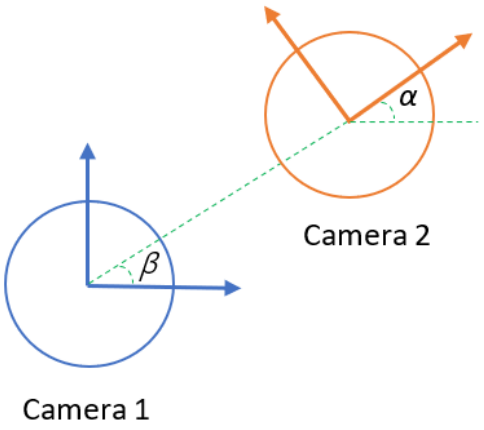


Fig. 5. 2. The projection of the camera frames on the X-Y plane

As shown in Fig. 5.2, the β is the direction of the motion, the α represented the rotation angle in X-Y plane. According to the equation (4-16), the direction of motion from P_1 to P_2 can be represented as

$$\beta = \tan^{-1}\frac{t_y}{t_x} \qquad (5-3)$$

Transfer β to image coordinate, where W is the width of the image in pixel.

$$pixel = \frac{W}{2} - \frac{\beta \times W}{360} \qquad (5-4)$$

From the equation we can get the pose of camera 2 relative to camera 1, their relationship can be defined as

$$P_2 = R^{-1}\,P_2 - R^{-1}t$$

Assume $t' = -R^{-1}t$. The direction of motion from P_2 to P_1 can be represented as

$$\beta' = \tan^{-1}\frac{t'_y}{t'_x}$$

Transfer to image coordinate uses the equation (5-3). After obtaining the coordinates of the two motion directions in the image, drawing this direction onto the image allows for a visualobservation of the results. For compare with the direction of the motion (γ) the dataset provided, as shown in Fig. 5.1, the error between the estimation and the ground truth is.

$$error = \gamma - \beta \qquad (5-5)$$

5.2 Result and Discussion

In this study, four methods were tested on 453 pairs of spherical panoramic images. Among these, 74 pairs were taken indoors, and 379 pairs were taken outdoors. The indoor spherical panoramic images included multiple scenes, while the outdoor ones depicted park pathways. The four methods evaluated were: the proposed method, a combination of multiple matching methods without preprocessing the correspondences, SIFT + KNN without preprocessing, and SuperPoint + LightGlue without preprocessing. Table 1 shows the accuracy of different algorithms across all datasets. Table 2 displays the accuracy of different algorithms on indoor datasets, and Table 3 presents the accuracy on outdoor datasets.

Table 1. Accuracy (%) for different acceptance thresholds in various environments.

Method	5° ↑	10° ↑	15° ↑	20° ↑	25° ↑
SIFT + KNN	25.61	47.90	58.28	64.24	67.77
SuperPoint + LightGlue	29.58	52.10	66.45	74.83	80.13
Without preprocessing	30.68	55.63	69.09	78.15	84.77
Our	**56.73**	**56.73**	**71.96**	**81.68**	**86.76**

Table 2. Accuracy (%) for different acceptance thresholds in various indoor environments.

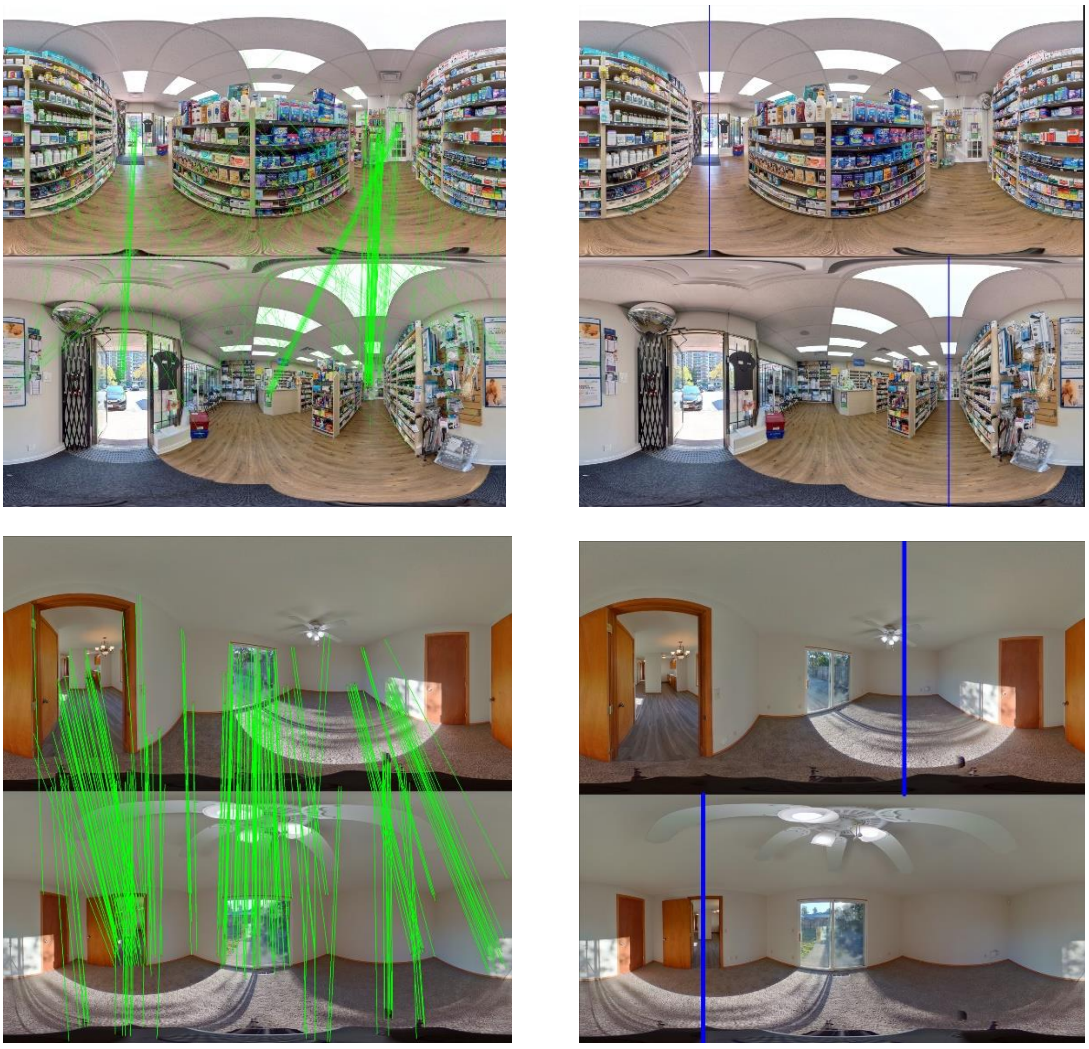
Method	5° ↑	10° ↑	15° ↑	20° ↑	25° ↑
SIFT + KNN	39.19	90.54	93.24	94.59	94.59
SuperPoint + LightGlue	41.90	70.27	77.02	79.73	79.73
Without preprocessing	43.24	89.19	91.90	91.89	93.24
Ours	43.24	94.59	95.95	95.95	95.95

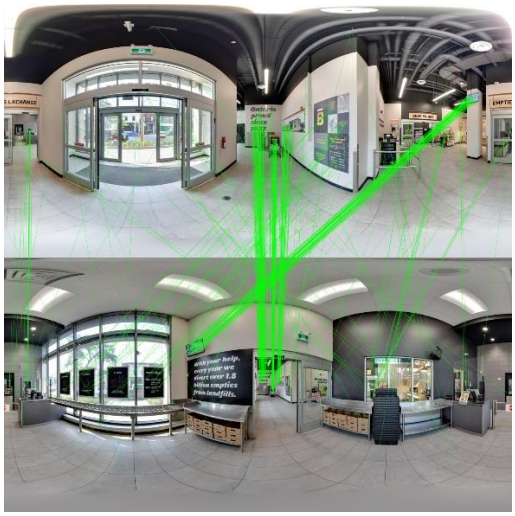
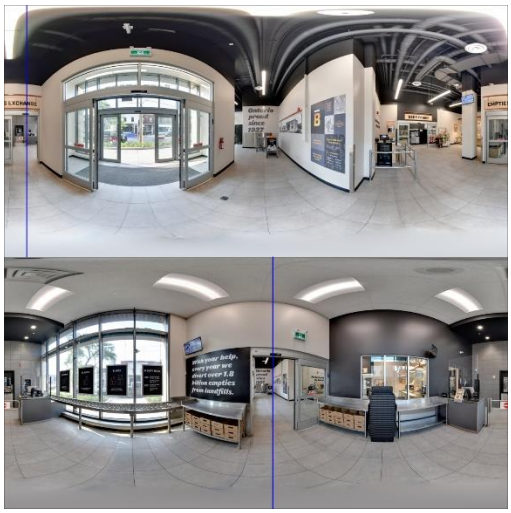
Table 3. Accuracy (%) for different acceptance thresholds in various outdoor environments.

Method	5° ↑	10° ↑	15° ↑	20° ↑	25° ↑
SIFT + KNN	22.96	39.58	51.45	58.31	62.53
SuperPoint + LightGlue	27.18	48.55	64.38	73.88	80.21
Without preprocessing	28.23	49.08	64.64	75.46	83.11
Ours	30.61	49.34	67.28	78.89	84.96

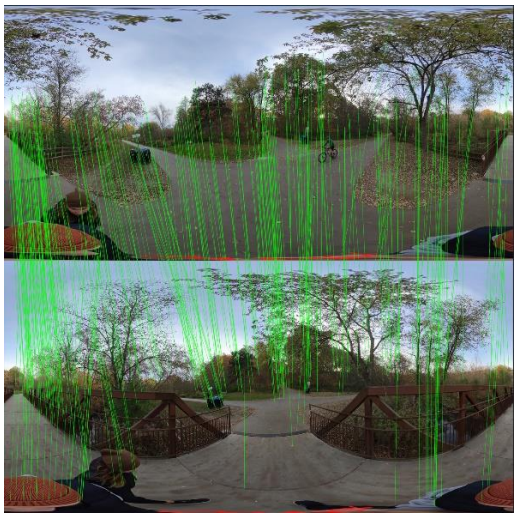
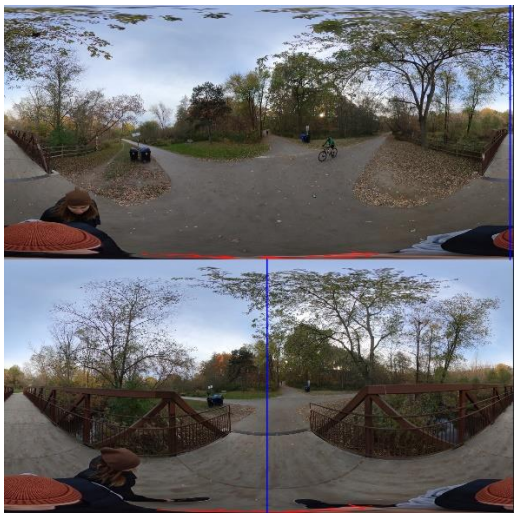
The results indicate that using individual matching methods has its advantage and disadvantage: SIFT performs better indoors, while SuperPoint excels outdoors. However, regardless of whether it's indoor or outdoor environments, combining both methods yields better performance than using them individually. Additionally, all results demonstrate that performance improves after preprocessing. Overall, in the various environments, multi-matching and preprocessing delivered pretty good performance than the single matching and without preprocessing.

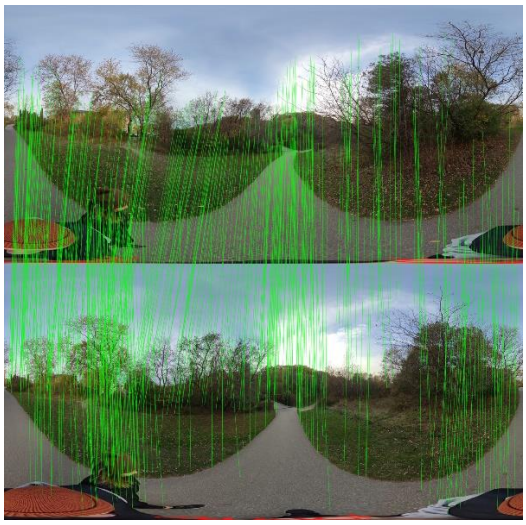
As shown in Fig. 5.3, there are some of the match and test results, where the direction of motion is marked with a blue line. In the results, some cases do not get good results in the matching process (big noise), but still have the pretty accurate and stable results for pose estimation.

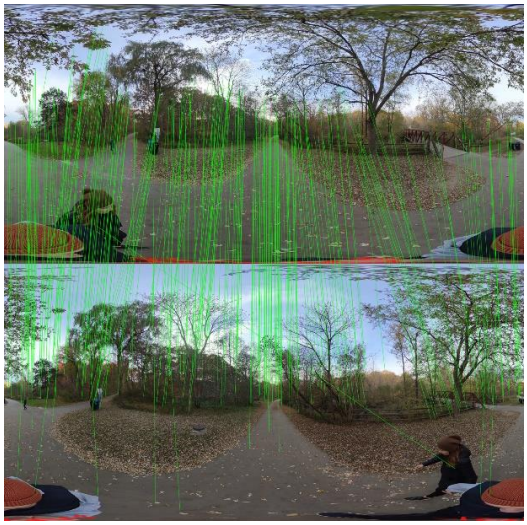
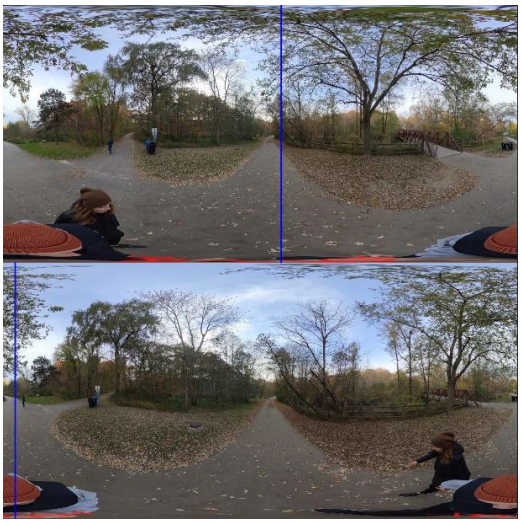


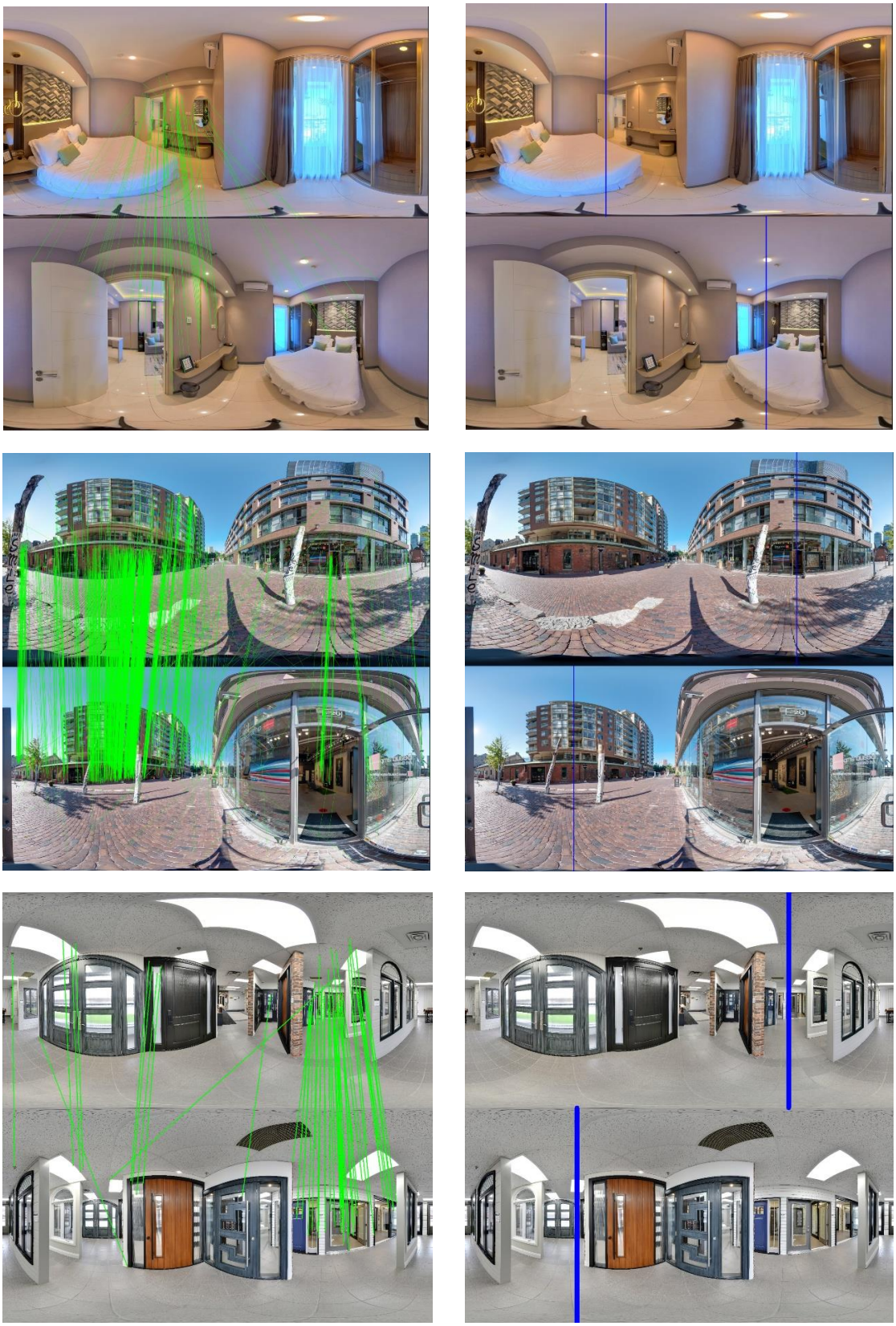



Champions of Beer
NEW ARRIVALS

Champions of Beer
NEW ARRIVALS







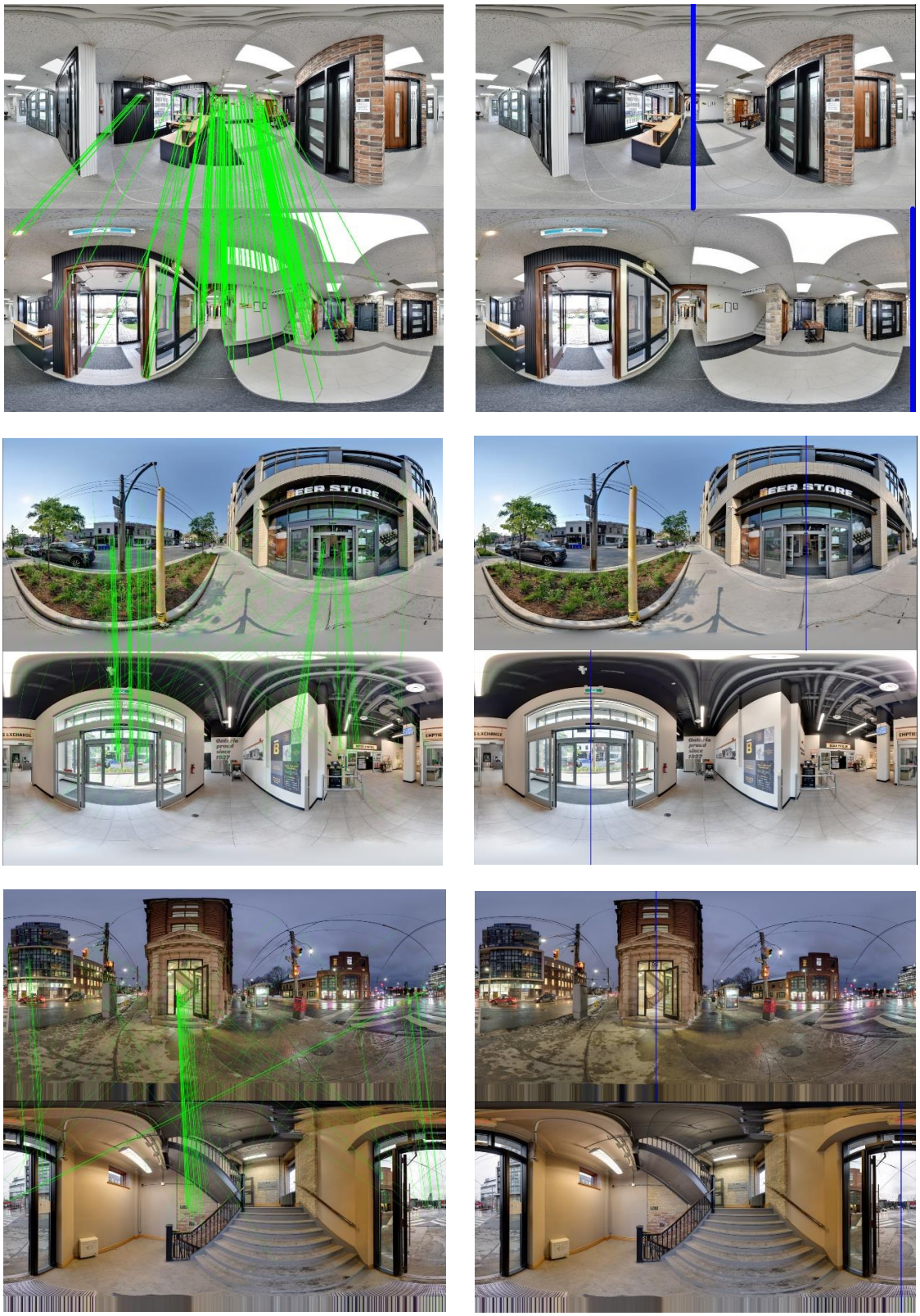


Fig. 5. 3. Some results of the pose estimation. Left is the match result between two spherical panoramas. Right shows the direction of the motion (blue lines).

During the testing process, different scenes were associated with different matching methods. In environments with strong textures (such as buildings on the sides of streets), SIFT with KNN was frequently used. However, in scenarios like tree-lined pathways or smooth bathrooms, Superpoint with LightGlue performed well. This highlights the benefits of using multiple contemporary matching algorithms to achieve better adaptability (see Table 1, 2 and 3). Currently, the choice of matching algorithm is based on the number of correspondences, but in the future, other selection mechanism could be explored to invoke more matching algorithms.

In some scenes, the number of incorrect correspondences exceeded the correct ones, as shown in Fig. 5.3. However, by categorizing the correspondences, it's possible to significantly reduce the proportion of outliers within a specific category, thereby enhancing the algorithm's robustness. The results also demonstrate the effectiveness of this approach.

Due to the adoption of multiple matching algorithms and separate pose estimation for categorized matching points, the overall computation time has increased significantly. While it performs well in tasks without real-time requirements, further research is needed to address the real-time aspects for broader applications.

Chapter 6 Conclusion

In this paper, we propose a novel architecture for robust spherical panorama pose estimation. We integrate multiple matching algorithms to adapt to diverse environments. By studying the distribution of matching points, we map these points onto a unit sphere and compute their Euclidean distances. Finally, we use K-Means clustering to group them, reducing the presence of outliers in the dataset. This method significantly enhances the algorithm's robustness, allowing it to produce accurate results even when the matching algorithms perform poorly. When using RANSAC, we design an error function that does not require setting a threshold to distinguish inliers from outliers, thereby reducing human-induced uncertainty. In summary, our approach demonstrates excellent adaptability and robustness, even in noisy scenarios, providing a strong foundation for generating virtual navigation maps in the future.

References

1. Su, Y.-C. and K. Grauman. *Learning Spherical Convolution for Fast Features from 360° Imagery*. in *Neural Information Processing Systems*. 2017.
2. Im, S., et al., *All-Around Depth from Small Motion with a Spherical Panoramic Camera*. Vol. 9907. 2016.
3. Silveira, T.L.T.d. and C.R. Jung, *Dense 3D Scene Reconstruction from Multiple Spherical Images for 3-DoF+ VR Applications.* 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2019: p. 9-18.
4. Fangi, G. and C. Nardinocchi, *Photogrammetric Processing of Spherical Panoramas.* The Photogrammetric Record, 2013. **28**.
5. Ummenhofer, B., et al., *DeMoN: Depth and Motion Network for Learning Monocular Stereo.* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: p. 5622-5631.
6. Tang, C. and P. Tan, *BA-Net: Dense Bundle Adjustment Network.* ArXiv, 2018. **abs/1806.04807**.
7. Han, L., et al., *RegNet: Learning the Optimization of Direct Image-to-Image Pose Registration.* ArXiv, 2018. **abs/1812.10212**.
8. Sumikura, S., M. Shibuya, and K. Sakurada, *OpenVSLAM: A Versatile Visual SLAM Framework.* Proceedings of the 27th ACM International Conference on Multimedia, 2019.
9. Huang, H. and S.-K. Yeung, *360VO: Visual Odometry Using A Single 360 Camera.* 2022 International Conference on Robotics and Automation (ICRA), 2022: p. 5594-5600.
10. Wang, Z.-y., et al., *LF-VISLAM: A SLAM Framework for Large Field-of-View Cameras With Negative Imaging Plane on Mobile Agents.* IEEE Transactions on Automation Science and Engineering, 2022.
11. Solarte, B., et al., *Robust 360-8PA: Redesigning The Normalized 8-point Algorithm for 360-FoV Images.* 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021: p. 11032-11038.
12. Fischler, M.A. and R.C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.* Commun. ACM, 1981. **24**: p. 381-395.
13. Chen, B. and C. Peng, *Interactive Relative Pose Estimation for 360° Indoor Panoramas through Wall-Wall Matching Selections.* SIGGRAPH Asia 2023 Posters, 2023.
14. Aly, M.A. and J.-Y. Bouguet, *Street view goes indoors: Automatic pose estimation from uncalibrated unordered spherical panoramas.* 2012 IEEE Workshop on the Applications of Computer Vision (WACV), 2012: p. 1-8.
15. Hartley, R. and A. Zisserman, *Multiple View Geometry in Computer Vision*. 2 ed. 2004, Cambridge: Cambridge University Press.
16. Scaramuzza, D. and F. Fraundorfer, *Visual Odometry [Tutorial].* IEEE Robotics & Automation Magazine, 2011. **18**: p. 80-92.
17. Eder, M., et al., *Tangent Images for Mitigating Spherical Distortion.* 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019: p. 12423-12431.
18. Guan, H. and W. Smith, *BRISKS: Binary Features for Spherical Images on a Geodesic Grid.* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: p. 4886-4894.
19. Lowe, D.G., *Distinctive Image Features from Scale-Invariant Keypoints.* International Journal of Computer Vision, 2004. **60**: p. 91-110.

20. Rublee, E., et al., *ORB: An efficient alternative to SIFT or SURF.* 2011 International Conference on Computer Vision, 2011: p. 2564-2571.
21. Zhao, Q., et al., *SPHORB: A Fast and Robust Binary Feature on the Sphere.* International Journal of Computer Vision, 2014. **113**: p. 143 - 159.
22. Cruz-Mota, J., et al., *Scale Invariant Feature Transform on the Sphere: Theory and Applications.* International Journal of Computer Vision, 2011. **98**: p. 217 - 241.
23. DeTone, D., T. Malisiewicz, and A. Rabinovich, *SuperPoint: Self-Supervised Interest Point Detection and Description.* 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017: p. 337-33712.
24. Dusmanu, M., et al., *D2-Net: A Trainable CNN for Joint Description and Detection of Local Features.* 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019: p. 8084-8093.
25. Tyszkiewicz, M.J., P. Fua, and E. Trulls, *DISK: Learning local features with policy gradient.* ArXiv, 2020. **abs/2006.13566**.
26. Li, X., et al., *Dual-Resolution Correspondence Networks.* ArXiv, 2020. **abs/2006.08844**.
27. Sun, J., et al., *LoFTR: Detector-Free Local Feature Matching with Transformers.* 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021: p. 8918-8927.
28. Jiang, W., et al., *COTR: Correspondence Transformer for Matching Across Images.* 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021: p. 6187-6197.
29. Mao, R., et al. *3DG-STFM: 3D Geometric Guided Student-Teacher Feature Matching*. in *European Conference on Computer Vision*. 2022.
30. Murrugarra-Llerena, J., T.L.T.d. Silveira, and C.R. Jung, *Pose Estimation for Two-View Panoramas based on Keypoint Matching: a Comparative Study and Critical Analysis.* 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022: p. 5198-5207.
31. Wagner, D., et al., *Real-time panoramic mapping and tracking on mobile phones.* 2010 IEEE Virtual Reality Conference (VR), 2010: p. 211-218.
32. Yuan, H., et al., *A Novel Method for Geometric Correction of Multi-cameras in Panoramic Video System.* 2010 International Conference on Measuring Technology and Mechatronics Automation, 2010. **1**: p. 248-251.
33. Thibault, S. *New generation of high-resolution panoramic lenses*. in *SPIE Optical Engineering + Applications*. 2007.
34. Thibault, S. *Panoramic lens an historical perspective: from sky lens to consumer wide angle freeform optics*. in *International Optical Design Conference*. 2021.
35. Yan, Y. and J.M. Sasián, *Photographic zoom fisheye lens design for DSLR cameras.* Optical Engineering, 2017. **56**.
36. Chahl, J.S. and M.V. Srinivasan, *Reflective surfaces for panoramic imaging.* Applied optics, 1997. **36 31**: p. 8275-85.
37. Kweon, G.-i., et al., *Folded catadioptric panoramic lens with an equidistance projection scheme.* Applied optics, 2005. **44 14**: p. 2759-67.
38. Rigelsford, J.M., *Panoramic Vision: Sensors, Theory and Applications.* Sensor Review, 2002. **22**.
39. Wang, Y., et al. *The design of miniaturization and super wide angle monitor lens based on monocentric lens*. in *International Symposium on Advanced Optical Manufacturing and Testing Technologies (AOMATT)*. 2019.

40. Pernechele, C. *Hyper-hemispheric and bifocal panoramic lenses*. in *Optics/Photonics in Security and Defence*. 2013.
41. Wang, J., Y. Liang, and M. Xu, *Design of panoramic lens based on ogive and aspheric surface.* Optics express, 2015. **23 15**: p. 19489-99.
42. Roessle, B. and M. Nießner, *End2End Multi-View Feature Matching with Differentiable Pose Optimization.* 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2022: p. 477-487.
43. Yi, K.M., et al. *LIFT: Learned Invariant Feature Transform*. in *European Conference on Computer Vision*. 2016.
44. Lindenberger, P., P.-E. Sarlin, and M. Pollefeys, *LightGlue: Local Feature Matching at Light Speed.* 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2023: p. 17581-17592.
45. Faugeras, O., *Three-dimensional computer vision: a geometric viewpoint.* MIT press, 1993.
46. Svoboda, T., T. Pajdla, and V. Hlaváč. *MOTION ESTIMATION USING CENTRAL PANORAMIC CAMERAS*. 1998.
47. Guan, H. and W. Smith, *Structure-From-Motion in Spherical Video Using the von Mises-Fisher Distribution.* IEEE Transactions on Image Processing, 2017. **26**: p. 711-723.
48. Hartley, R.I., *In Defense of the Eight-Point Algorithm.* IEEE Trans. Pattern Anal. Mach. Intell., 1997. **19**: p. 580-593.
49. Longuet-Higgins, H.C., *A computer algorithm for reconstructing a scene from two projections.* Nature, 1981. **293**: p. 133-135.
50. Nistér, D., *An efficient solution to the five-point relative pose problem.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004. **26**: p. 756-770.
51. Pagani, A. and D. Stricker, *Structure from Motion using full spherical panoramic cameras.* 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011: p. 375-382.
52. Fujiki, J., A. Torii, and S. Akaho. *Epipolar Geometry Via Rectification of Spherical Images*. in *International Conference on Computer Vision/Computer Graphics Collaboration Techniques*. 2007.
53. Taira, H., et al., *Robust Feature Matching for Distorted Projection by Spherical Cameras.* IPSJ Trans. Comput. Vis. Appl., 2015. **7**: p. 84-88.
54. Brückner, M., F. Bajramovic, and J. Denzler. *Experimental Evaluation of Relative Pose Estimation Algorithms*. in *International Conference on Computer Vision Theory and Applications*. 2008.
55. Fujiki, J., *Three types of reprojection error on spherical epipolar geometry.* 2008.
56. Pajdla, T., T. Svoboda, and V. Hlaváč. *Epipolar geometry of central panoramic catadioptric cameras*. 2001.
57. Gao, S., et al., *Review on Panoramic Imaging and Its Applications in Scene Understanding.* IEEE Transactions on Instrumentation and Measurement, 2022. **71**: p. 1-34.
58. Gurrieri, L.E. and E. Dubois, *Acquisition of omnidirectional stereoscopic images and videos of dynamic scenes: a review.* Journal of Electronic Imaging, 2013. **22**.
59. Martin, C. *Design issues of a hyperfield fisheye lens*. in *SPIE Optics + Photonics*. 2004.

60. Hwang, D.-H., K. Aso, and H. Koike, *Toward human motion capturing with an ultra-wide fisheye camera on the chest.* 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2019: p. 1524-1526.
61. Pöntinen, P. *ON THE GEOMETRICAL QUALITY OF PANORAMIC IMAGES*. 2004.
62. Snyder, J.P. *Flattening the Earth: Two Thousand Years of Map Projections*. 1994.
63. Technical, C., B. Micus, and T. Pajdla. *Para-catadioptric camera auto-calibration from epipolar geometry*. 2004.