東京大学
THE UNIVERSITY OF TOKYO

博士論文

Doctoral Thesis

Development of Image Generation Systems using Top-view Camera for Crane Operation Assistance

（クレーン操作支援のための吊下型カメラを用いた画像生成システムの開発）

指導教員　　鈴木 宏正 教授

東京大学大学院 工学系研究科 精密工学専攻

37-167269

王　宇

Wang Yu

2020

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Background

A crane is a machine that has a very long history for centuries. The crane has always been playing a vital role in construction, transportations and other industries [62]. In recent years, many types of cranes are invented to meet all kinds of different construction and transportation requirements. Various forms of the cranes can be seen everywhere no matter in the cities or the countrysides.

Despite there are various forms of cranes, the necessary components of the crane are a rope and several sheaves. The crane can have many different forms such as rough crane and tower crane, but basically the crane can be categorized into the fixed crane and the mobile crane according to their mobility. Figure 1.1 illustrates four different types of cranes. In this figure, 1.1a and 1.1b are two forms of the mobile crane. 1.1c and 1.1d are two forms of the fixed crane.

The basic functionality of the crane is to lift up and down materials from one position to another position. The types of movements can be different for different forms of cranes. For example, the crawler crane and the rough crane shown in Figure 1.1a and 1.1b, the movements to lift the materials are pitching, rotation and hoisting. For the level luffing crane shown in Figure 1.1d, it is more convenient to do a horizontal movement for the lifted object in the pitching process. And for the tower crane shown in Figure 1.1c, it can not make the movement of pitching. Instead, a trolley in the boom can make a horizontal movement.

As the crane plays a more and more important role in contemporary society, safety becomes a major problem in the construction field and industry. The complex, dynamic, and continually changing nature of construction work has been recognized as an important

(a) Crawler Crane [4]

(b) Rough crane

(c) Tower crane [60]

(d) Level luffing crane [6]

Fig. 1.1: Various forms of cranes

contributor to the high rates of injuries and fatalities in industry [53]. In construction operations, the crane is playing a central role and associated with a large number of accidents in construction. According to statistics of the survey, the crane is involved in up to one-third of all construction and maintenance fatalities [53].

One of the reasons for the accidents is considered to be the very complex working environment of the crane. Except for some initial construction work, the working environment is always congested with many objects and buildings in different height. In addition, the workers in the crane's working environment is also a concern for crane operation. It is prohibited for the people standing under the boom and the hanging object to keep safety. Furthermore, while in the operation of the crane, the operator needs to notice the crane status from the operation cabin. Many numeric data from the display panel is associated

with the safety problem. To sum up, for a crane operator, he needs to know the things:

- Congested working environment: The working environment of the crane is highly congested with different kinds of objects and buildings. In the operation of a crane, the operator needs to avoid the collision.
- The workers: In the construction field, there are many people. The crane operation should avoid the worker and the situation that the worker is under the boom of the crane should never happen.
- The crane: The status of the crane in working should also be noticed by the crane operator. Many status information in the cabin's display is important for operation safety.

In addition, the operation with safety and efficiency is the major concern for the crane in construction. The problems encountered by the crane operator are vital for ensuring working efficiency and safety. As can be seen, assistance systems which can help the crane operator have a better vision and notification about crane's working environment is required. And in recent years, many assisting systems have been planned and applied to assist crane operation. The systems can provide more information of the crane's working environment to the crane operator.

1.2 Challenges

The problems encountered by the crane operators are mainly the congested working environment and vision limitations. The working efficiency and safety are the major concerns for the construction work conducted by the crane. To ensure the working efficiency under the problems, it is considered to be the most important to provide the crane operators with a clear view of the congested working environment around the crane.

For the crane operation, it is always done by an operator sitting in a small cabin, as shown in Figure 1.2.

Fig. 1.2: The operation cabin of a rough crane

To have a better understanding of the crane's behaviors, the working status information of the crane is displayed in the small cabin as shown in Figure 1.3. It is named an automatic moment limiter (AML) in this thesis. The current crane's working condition can be checked from the display with the boom length, pitching angle, working radius and etc.

The construction work should concern both the crane status and the working environment. And the display of AML is not intuitive and hard for the crane operator to recognize the operation process. Sitting in the small operation cabin, a lack of vision for the working environment is also a problem for the crane operator. In some cases, the

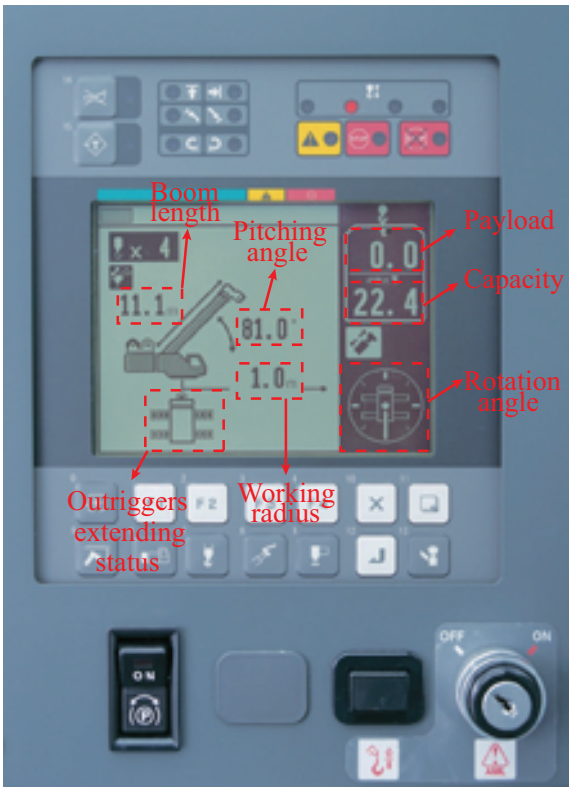


Fig. 1.3: Autmomatic moment limiter (AML) [1]

crane operators can not see the load they are moving. A cooperative work needs to be made to lift the load to where it should be lifted to. This scene is shown in Figure 1.4 where the crane operator can not see the moving object because of a huge obstacle. To move the object to the target location, the signalman will give signals to the tagman. The operator will make the lifting according to the guidance of the tagman. The lifting work has to be made by the cooperative work of these three people.

The limited information, congested working environment and limited vision make the crane operation quite difficult. To help the crane operator to have more efficient and safe control of lifting work, a very intuitive and concise idea is to provide the crane operator with more views about the working environment with the camera. The video system depending on the video camera can give the crane operator extra vision of the crane and its working environment. However, these video systems can only provide a narrow vision to the crane operator. And it lacks accurate distance information which is very important sometimes. For example, a video camera system [47] was trying to provide the top-view vision by attaching a video camera to the tower-crane trolley. It is very useful to solve

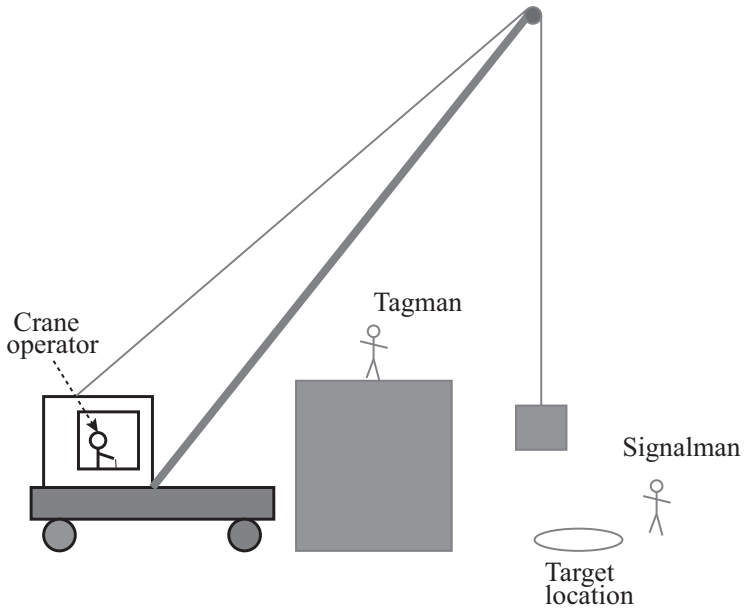
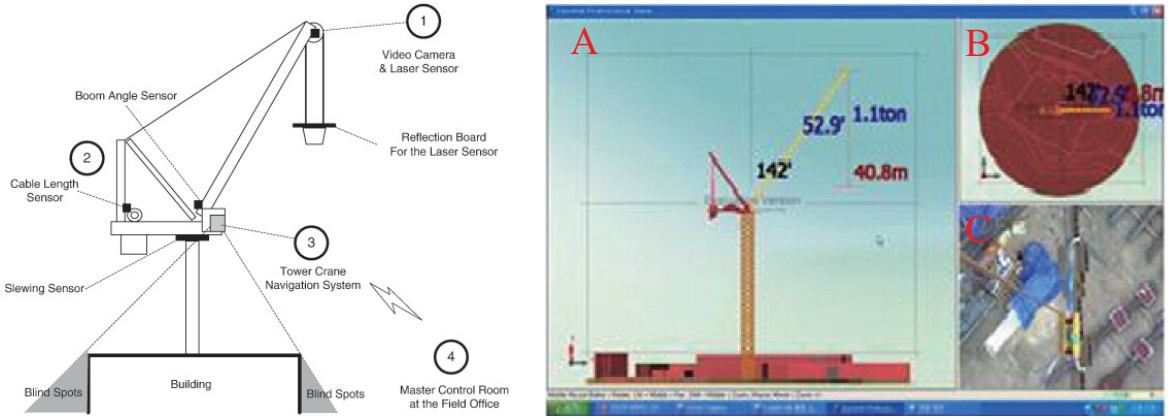


Fig. 1.4: A cooperative work of three people for a lifting

the problem of vision block which is shown in Figure 1.4. But the top-view camera can not obtain an overall view of the crane's working environment and can not give the crane operator a complete sense of all the working environment. The system also can not have a performance for the case of tall buildings because a lack of height information. By providing more cameras, the problem can be alleviated [46]. But it is not applicable to find the right locations for cameras because of the congested working environment.

Based on the camera systems, more sensors and information can be integrated to have a better assistance for the crane operator. A building information management (BIM) system integrated with a laser sensor and several encoder sensors is installed at the tower crane [46]. The schematic drawing of the system is shown in Figure 1.5. The schematic drawing of the system is shown in Figure 1.5a. With the data from sensors and BIM database, the system can know the 3D information of the working environment and the relationship between the crane and the environment. As shown in Figure 1.5b, the display is split into three views. The view A and view B is a virtual display about the current crane's relationship with the working environment. The view C is a display by a top-view camera. Compared with the video camera system mentioned above, the BIM system can fully navigate the crane operation with some lifting path planning algorithms because of the known working environment provided by BIM and the relationship between the crane

and the environment.



(a) BIM based navigation system [46] (b) A split view of the display [46]

Fig. 1.5: A BIM based navigation system and display panel

However, in most cases, there will not be a BIM database for usage. The use of BIM is still very limited to advanced construction companies and its use for crane operation is not even common at all.

Researches for obtaining the 3D information are conducted to recover the working environment of the crane. There are various sensors can be used for reconstructing the crane's working environment. The sensors mainly include radar, Global Positioning System [43, 44, 78], 3D laser scanners [22, 70], radio frequency(RF) [50] and infrared detection [80]. The above sensors can be used to explore the environment of the worksite and identify the objects' positions. The reconstruction of worksite-based on the sensors mentioned is acquired directly by the hardware structure of the sensor. Among these sensors, the laser scanner has accuracy in millimeter to centimeter and can obtain hundreds of thousands of points every second. It can make a good reconstruction of crane's worksite. Figure 1.6 is the scanning process of laser scanner. With scans from multiple positions in the worksite, an overall survey of the environment can be made with many slices of the point cloud. Along with a digital camera, these point cloud slices can be fused with the captured RGB images. By registration the slices of fused point cloud at different positions [35], the whole reconstruction of the worksite can be reconstructed.

Figure 1.7 shows an example of reconstruction with a 3D laser scanner. The point cloud of a mobile crane is established with a ground-based laser scanner. With the point

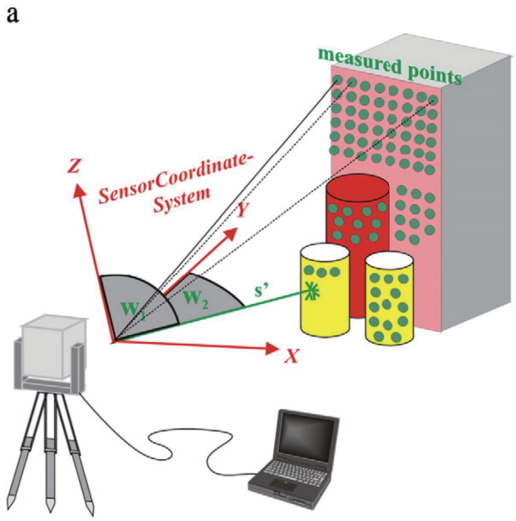
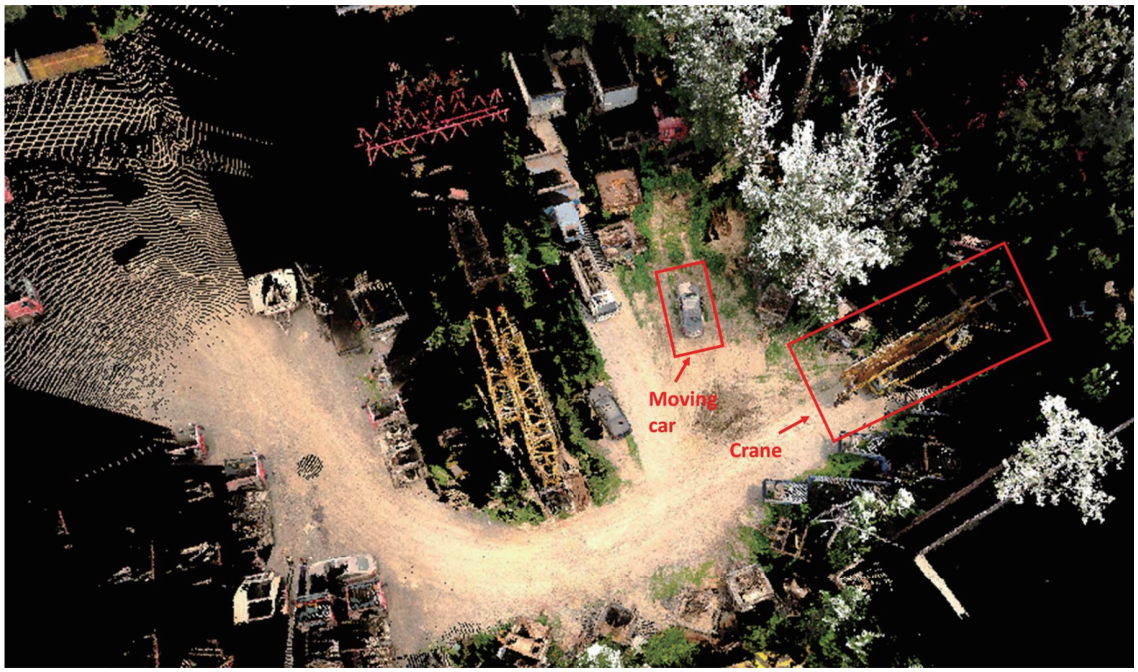


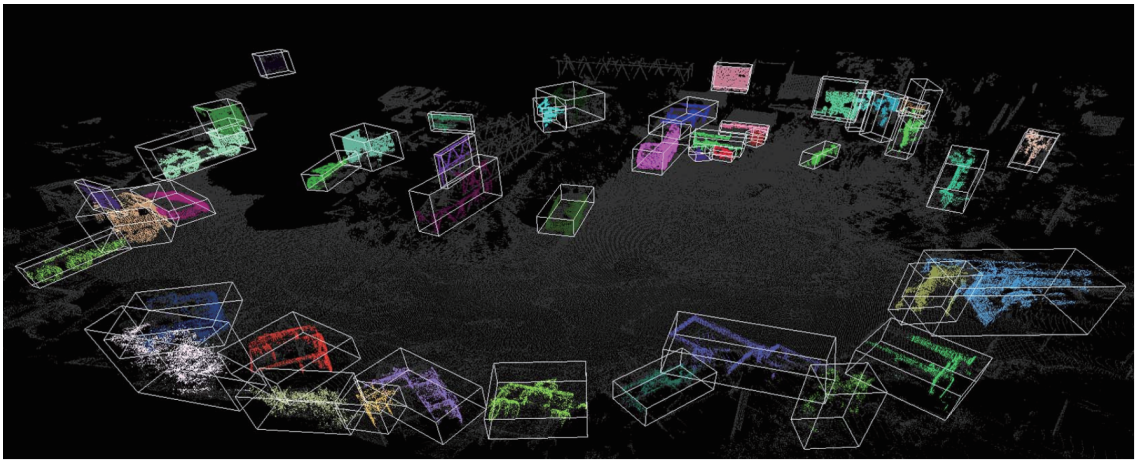
Fig. 1.6: Laser scanning process [72]

cloud registration, Figure 1.7a is the top-down view of the point cloud. The clustered result with bounding boxes of the point cloud is represented in 1.7b. These bounding box representing the objects on the ground can be used for obstruction detection in crane lift path planning and real-time hazard analysis [20]. With the reconstructed 3D information, assistance can be applied to help the crane operation.

Except for the sensors with depth-sensing capabilities, only with a camera, the environment can also be recovered in 3D through visual simultaneously location and mapping (VSLAM) and structure from motion (SFM). And in recent years, many commercial softwares have been invented to achieve the function of VSLAM. The reconstruction of the environment with SLAM is a hot topic in computer vision. Generally, a sequence of images is required for reconstruction. The images should cover all the crane's working environment. With SLAM, the environment can be recovered into 3D with a sparse or dense point cloud. The density of the 3D point cloud depends on the SLAM methods. The feature-based VSLAM will give a sparse point cloud, while for the direct VSLAM, the point cloud is always a dense point cloud. In research [51], a drone has taken a video of a construction site and a reconstruction of the environment is achieved with Autodesk Recap [61]. This example is shown in Figure 1.8. Figure 1.8a shows the drone taking a camera. The drone will fly over the construction site and take a video. After that, a sequence of images will be picked out from the recorded video by the drone and sent to



(a) A top-down point cloud of view [20]



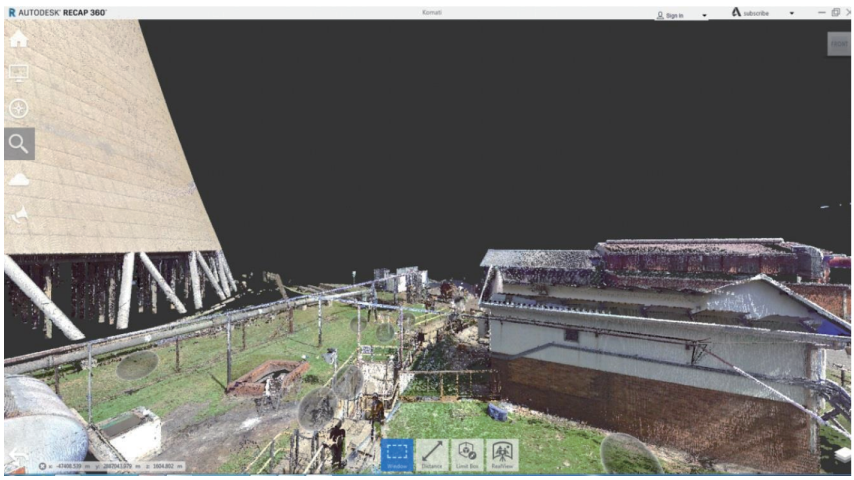
(b) Bounding box representation of environment objects [20]

Fig. 1.7: A point cloud generated with laser scanner

Autodesk Recap. The reconstructed 3D result is shown in Figure 1.8b. With the reconstructed 3D environment by VSLAM with the cameras, the application to navigate the crane operation with path planning like paper [46] can be made even without any BIM information.

As can be seen, the video camera and other depth sensors are promising tools used in crane-related applications to provide more information for the crane working in the congested environment. The assistance by the camera is useful by giving extra views of

(a) An drone with a RGB camera [51]



(b) Reconstruction result with Autodesk ReCap [51]

Fig. 1.8: 3D reconstruction with an UAV and Autodesk Recap

the working environment. But for some applications for assistance, the depth information is also very important. The top-view camera is a down-looking camera generally mounted on the boom head of a mobile crane or the hook (trolley) of a tower crane. Figure 1.9 shows a top-view camera mounted on the boom head of a rough crane. The example for the top-view camera of the tower crane can be seen in Figure 1.5a. Some applications also put the camera on the hook which can be lower down with the hook. With the development of computer vision technology, a top-view camera can both give an extra top-view of the environment and recover the 3D environment around the crane with VSLAM. The benefits of a top-view are listed below:

- Cheap: Compared to other sensors especially with the capability of depth-sensing, the cost of mounting a camera to the crane is cheap in cost.
- Convenient: The top-view camera is easy to be integrated into the crane. It is better than providing extra site views by planting cameras in other positions of the working environment. The top-view camera itself does not need to consider the working environment which means it is more applicable.

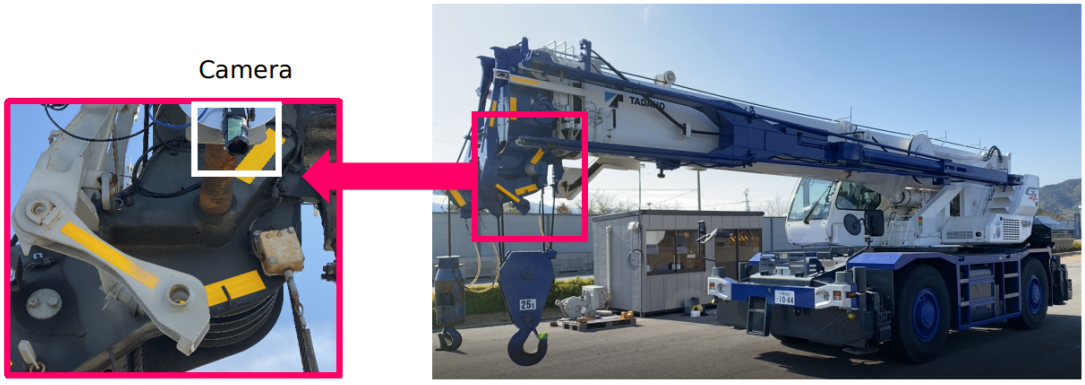


Fig. 1.9: The top-view camera mounted on the boom head of a rough crane.

For applications, the top-view camera has only been used to provide a top-view of the crane's working environment near the lifted object. Some examples can be seen in paper [29, 46, 47].

1.3 Objectives and Approaches

In this thesis, the objective is to develop an image generation system using a top-view camera to provide crane operation assistance. A simple display of an image captured with the top-view camera can not help the crane operator have a good sense while in operation.

The image captured with the top-view camera is a local area near the hook. It is very useful for the case of vision occlusion while in operation. But a local display means it lacks overall information about the crane's working environment to assist the crane operation.

The crane's working environment can be different at different stage phases of a construction project. For different construction stages, the assistance image should also be different.

As mentioned in the above sections, the construction site can be from simple very complex and congested. The assisting image for the crane operator should consider different conditions of the construction site. Generally, in the beginning of a construction project, the working environment for a crane is a very flat ground with only some construction materials distributed. However, after a period of construction, the working environment for a crane becomes congested and complex which contains many high objects. Taking these aspects into account, it is better to support different assisting images in different construction conditions. Therefore, the construction environment for providing assisting images to the crane operator is categorized into two types which are initial construction site and middle/last construction site. They are correspondent to two construction stages which are initial construction stage and middle/last construction stage for the crane. For different construction stages, the objectives are different. The details of definition and promising assisting images for the two construction stages are as follows:

- Initial construction stage: The working environment for a crane is an almost flat ground with a very low height variance. In the initial construction stage to lift an object from one position to another position, the lifted height of the object is not so important because the low height variance of the ground. The most important information is the safety related information and the location of lifted object in the construction site. A wide range image or an overall image which covers all the

working environment by image stitching is promising way to help solve the limited vision in the operation cabin. By providing the location of lifted object in the wide range image, the operator can know all the working environment, object location and the relationship between them.

- Middle and last construction stage: The working environment for a crane is contains many high objects which makes the height variance very big. In this construction stage, giving an overall image which showing all the environment and location can not be achieved by image stitching. Instead, a 3D spatial map be 3D reconstruction can provide useful visual information to the crane operator. Many types of information are related to the 3D spatial map of the crane's working environment because they are different for the areas with different height. For some important information, it is necessary to be displayed in the top-view image to remind the operator to notice the hidden dangers.

The two construction stages are classified by considering the height variance. It is because that for the middle and last construction stage, an overall map covering all the crane's working environment can not be generated is mainly because of the projective ambiguity which will be discussed in Chapter 4. In a word, the images which taken by a moving camera with translational movement for the environment with obvious depth variance, the image stitching can not generate high quality map. It is not strictly defined for now because it should consider the height of the camera. Generally, for a top-view camera at the height of 20 meters, it is better to take height under 4 or 5 meters as the initial construction stage. For the two different construction stages, we are aiming at developing two systems to provide the image based on the top-view camera to help crane operation. They will be described separately in the following two subsections.

1.3.1 2D Workspace Map Generation and Application

The first system based on the top-view camera is a system providing a 2D workspace map for the crane's working environment. It is for the initial construction work and some other work where the environment contains only the objects which are just several meters in height. In this kind of environment, the 2D workspace of the crane is important for providing the operator with an overall understanding of the environment. Because the height variance is not very significant under such an environment, the operator only needs to care about where the source location and target location for the object. With the 2D workspace map and the location of the lifted object on the map, the operator can achieve the operation of the crane. Furthermore, remote control for the working environment of dangers becomes possible and easier with the 2D workspace map.

To generate a clear workspace map, the image stitching with foreground detection by motion segmentation is used. In the image stitching process, two problems need to be clarified:

- **Reduction projection ambiguity**: The image stitching is only for the scene with low depth variance or the scene very far away from the pin-hole camera. For an object with obvious depth variance to the pin-hole camera, the image stitching will yield a panoramic image with a ghost on it. By putting the camera at a far position, the projection ambiguity effect will become small. So, in the process of generating a 2D workspace map, the captured image should be at a high position and the object on the ground of the working environment should be only at a low position. For instance, when the height of the top-view camera reaches 20 meters, then the object should be lower than four meters.
- **Removal foreground object**: In most cases, the image taken by the top-view camera contains a boom head and a hook. They are called foreground objects. The boom head and hook are shown in the image captured by the top-view camera will lead the stitched workspace map to be unclear. The ghost will appear on the stitched workspace map [13]. To make the image stitching result clear, detection and removal for the foreground objects need to be carried out first.

The approach of assisting crane operation with the 2D workspace map is from two stages. In the first stage, a clear 2D workspace map of the crane's working environment will be generated by image stitching with the foreground object detection and removal. In the second stage, the location of the boom head (hook) will be located in the 2D workspace map by matching the top-view image with the 2D workspace map. And for the continuous working of a crane operator, the lifting path can be computed and shown on the map. The path can be used for the evaluation and training of the crane operator. Figure 1.10 shows an illustration of the provided visual image to the crane operator. From the approach, a 2D workspace covering most area of the construction site will be generated. After that, for an image captured with the top-view camera, its position on the map can be displayed in the 2D workspace map. The illustration image shows that for a lifting job from the source location to the target location. The green dashed arrows are the best ways. In present application even it is not planned and shown directly to the crane operator. From the map and experience, the crane operator know how to lift the object and know where is now the hanging object located from the 2D workspace map with location.

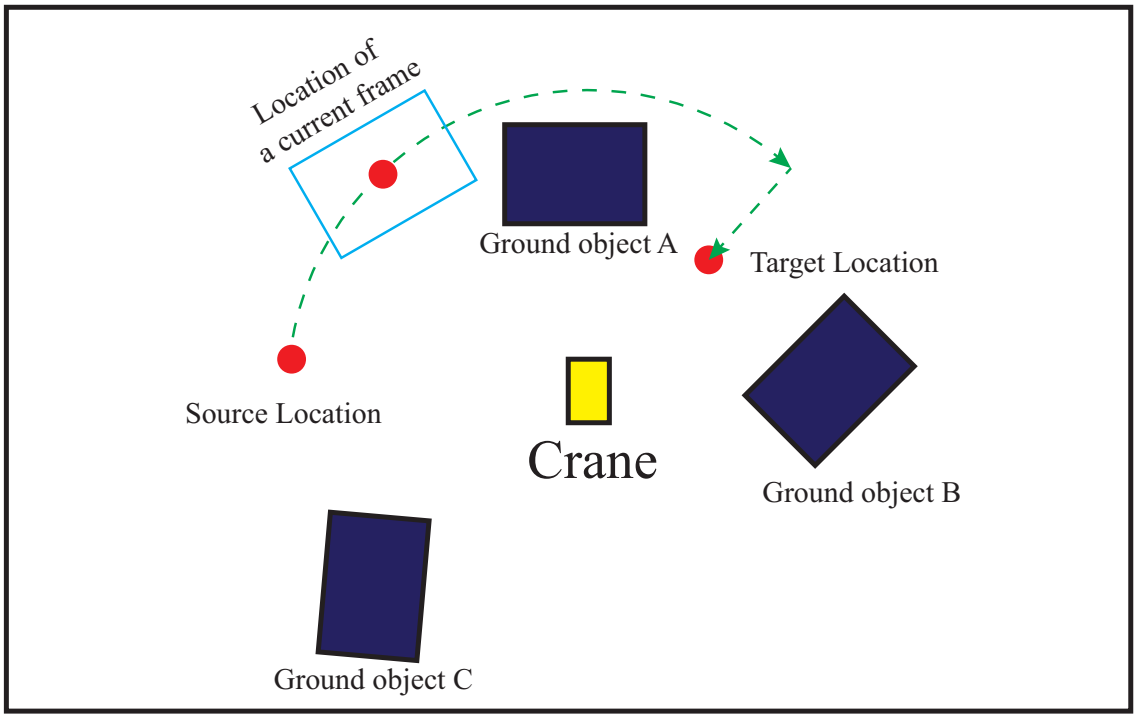


Fig. 1.10: An illustration of providing 2D workspace map with location to the crane operator.

To sum up, we have made the following contributions to develop the system:

- Clarify the conditions in which the system can be helpful. To reduce and avoid the projection ambiguity, the pre-shot condition for the clear 2D workspace map is clarified.
- A new approach for the detection of the foreground of a crane's top-view image is proposed by comparing the motion caused by 2D homography and dense optical flow. For a robust estimation, a detection pipeline is proposed.
- A system to generate a clear 2D workspace map is developed. By using the images with the removed foreground, a clear 2D workspace map can be generated. It mainly includes how to pick the keyframes and support frames in a video to detect the foreground of the keyframes.
- With the application to show the location of the boom head in the 2D workspace map and record the path of a lifted object over the 2D workspace map.

To ensure the system is applicable, the quality of the 2D workspace map is vital. As in the image stitching process, the transform is a projective transform that will change its metric property. The metric property means the property of similarity which means that the 2D workspace map is similar to the real world. Evaluations of errors for the 2D workspace map and application of location are necessary:

- **Stitching Error**: As mentioned, the projective transform will change the property of similarity. The ratios between the real length and imaged length or the ratios of width and height of some objects will be changed after image stitching. Only the 2D workspace map is of good metric property, it is applicable and safe to provide visual information to the crane operator. This will be checked by comparing the ratios of some objects' width and height both in the real world and in the 2D workspace map. Also, the length ratio of a real world object and an imaged object will be checked. It is better to have this difference by five percent. The error requirement of five percent for the metric property is not only proper but also strict. For displaying some information start from the crane's rotation axis such as a limit range for 20 meters, the safety threshold in this direction can ensure that inside 19 meters are safe. And also, for some application which only measure the distance of a local area within 1 meter such as for measuring, the error will be just 0.05 meters which will

not affect the functionality of the application.

- **Location Error**: The most concern for the application of location is the error of the estimated location in the 2D workspace map and the real position in the 2D workspace map. And with a discussion with a crane maker, it is preferred to be less than 0.2 meters. Except very precise lifting for direct assembling purpose, the location error less than 0.2 meters is capable for most cases of lifting work. For applicability, the value 0.2 meters of location error is a result discussed with a crane company.

Except for the errors which need to be ensured, the speed for the application also matters. For a new image captured by the top-view camera, a quick display for the location in the 2D workspace map is desired. As the crane moves slowly, a frame rate of 15 fps is enough for application. It means a 66 ms delay for an input top-view image to find its location on the 2D workspace map.

1.3.2 Top-view Image with Important Height-related Information

The second system based on the top-view camera is a system to generate height-related information in the top-view image. For the Middle and last stage construction which there are many tall objects, the 2D workspace map can not be generated with image stitching. And on the other hand, the height information becomes vital in this stage. Much height-related information is important for working efficiency and safety.

In this thesis, the working area limit line is displayed in the top-view image. It is a line separating the safe area and danger area and is affected by the depth of the camera to the environment.

The proposed approach includes a 3D reconstruction process of the crane's working environment and displaying the working area limit line in the top-view image process. The details are:

- A selection of the VSLAM approach for reconstruction of the crane's working environment. From many of state of the art technologies, the proper approach is selected out to reconstruct a precise and dense 3D point cloud of the environment. With the precise and dense 3D point cloud, many applications can be made such as collision detection and lifting path planning for the crane.
- Displaying the working area limit line in the top-view image. The final working area limit line in the top-view image is a projection from the dense 3D point cloud to the top-view image. So, it requires first drawing the working area limit line in the 3D point cloud. It includes finding the rotation axis in the 3D point by fitting the trajectory of the camera and drawing the working area limit line by checking the distance to the rotation axis. Then, with the estimation of the camera pose against the 3D point cloud, the final working area limit line is projected and displayed in the top-view image.

The image in the cabin display for the crane operator is a top-view image which containing the working area limit line, as shown in Figure 1.11. The illustration image shows that a working area limit line has separated the working area into safe working area and danger working area. And the working area limit line is affected by the high object. It is

different for the working area limit line in different heights. In a lifting job, the operator can know from the top-view image with such working area limit line which is affected by the 3D spatial map (height) at very beginning. It can prevent the accidents which are caused because of lifting the object out of the safe working area. As shown in this figure, the crane is lifting the green object through the path to the target location A or B. From the working area limit line in the ground, the operator will think that the target location is dangerous. However, the working area limit line on the high object is the small arc which correctly shows that the target location A is dangerous while B is safe.

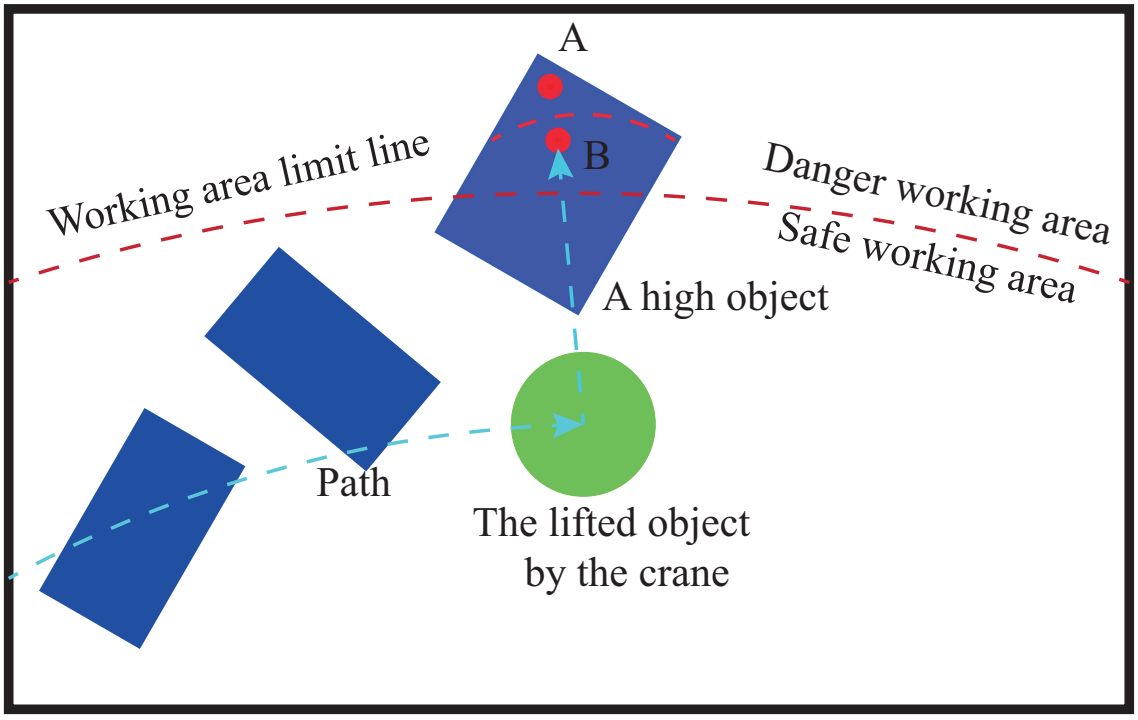


Fig. 1.11: An illustration of displaying the important working area limit line to the operator while the crane is lifting an object.

To ensure the system is applicable, the qualities for the reconstructed 3D dense point cloud and final projected error of the working area limit line are required.

For the reconstruction quality, generally, VSLAM approaches for such a long distance can not have a very good precision than the depth-sensors. But with different applications, the requirements also differ. In the application to show the working area limit line, the reconstruction error is assumed to be smaller than 0.5 meters in this study. Generally, the VSLAM approaches can not have a 3D reconstruction as precise as other depth-sensor based approaches. The reconstruction error requirement for 0.5 meters is mainly taking

consideration of ensuring that it can be used for application. Such as for the collision detection for crane, to ensure the collision free, the error value 0.5 meter means to lift up the object 0.5 meters. Another example is for the application of displaying depth-related information. Taking the boom head up to 20 meters and the height of 10 meters for a ground object, the error of 0.5 meters means 5 percent error which can ensure the accuracy of the displayed information.

The final projected working area limit line in the top-view image is affected by both the reconstruction error and the camera pose estimation error. The error should be compared with the display size. For a size of 1000 pixels in the display, the error less than 1 percent (10 pixels) is preferred. For a general camera with a field angle view about 45 degrees, for a camera above the ground about 20 meters in height, the covering range for 1000 pixels is about 16.5 meters. So, the error of 10 pixels means an error of 0.165 meters for the final projected working area limit line which is acceptable for application.

The last requirement for displaying is the speed of the application. The frame rate of a camera can be from 30 fps to 60 fps or more. It is an ideal result of displaying every frame with the information. However it also means a large more amount of computation. Considering that the crane moves slowly while in operation, it can allow a slower displaying. And a frame rate of 15 fps can be set as out objective.

1.4 Thesis Overview

The organization of this dissertation is as follows:

Chapter 1 has firstly made an introduction to the research background. Then, many related researches and systems for assisting the crane operation have been reviewed. And based on that, the objectives and approaches have been proposed and briefly stated. Two systems for two different working conditions are clarified.

Chapter 2 contains the fundamentals for understanding the later chapters. It can be separated into two parts. The first part containing four sections is mainly for chapter 3 and chapter 4. It first gives a brief introduction about the image features and explains two specific types of image features in detail. Also, the matching approach for the image feature is explained. Then the explanations for 2D image transformations including the homogeneous coordinate, rigid transformation and homogeneous transformation are made. Following that, the image view geometry including single and two views are explained. In the last, the introduction for optical flow and a DNN based dense optical flow approach FlowNet2 is made. The second part is for the understanding of chapter 5. It includes a short introduction to VSLAM.

Chapter 3 has proposed a method to do a dense motion segmentation for the image. By applying the dense optical flow and 2D homography, the foreground which consists of moving objects can be detected. Then a mask only with the background can be used in Chapter 4 for generating a clear workspace map.

Chapter 4 explains the process of generating a clear visual workspace map of the crane work site and how to applying it to assist operation systematically. It consists of a preprocessing stage and an operation stage. In the preprocessing stage, the images used for automatic image stitching will be selected out from a video to generate a workspace map with the detected background masks. And in the operation stage, for a new image captured with a top-view camera, its location and lifting path can be displayed on the generated workspace map.

Chapter 5 describes the approach of displaying the working area limit line in the top-view image. It also contains two stages in the pipeline. First, a 3D reconstruction with a current state of the art technology is used to have a precise and dense 3D point cloud for

the crane's working environment. Then, in the crane operation stage, the working area limit line is displayed in the top-view image to help the crane operator.

Finally, in Chapter 6, a summary of this thesis is made, followed by a discussion on future work.

The organization and an overview of this thesis is shown in Figure 1.12.

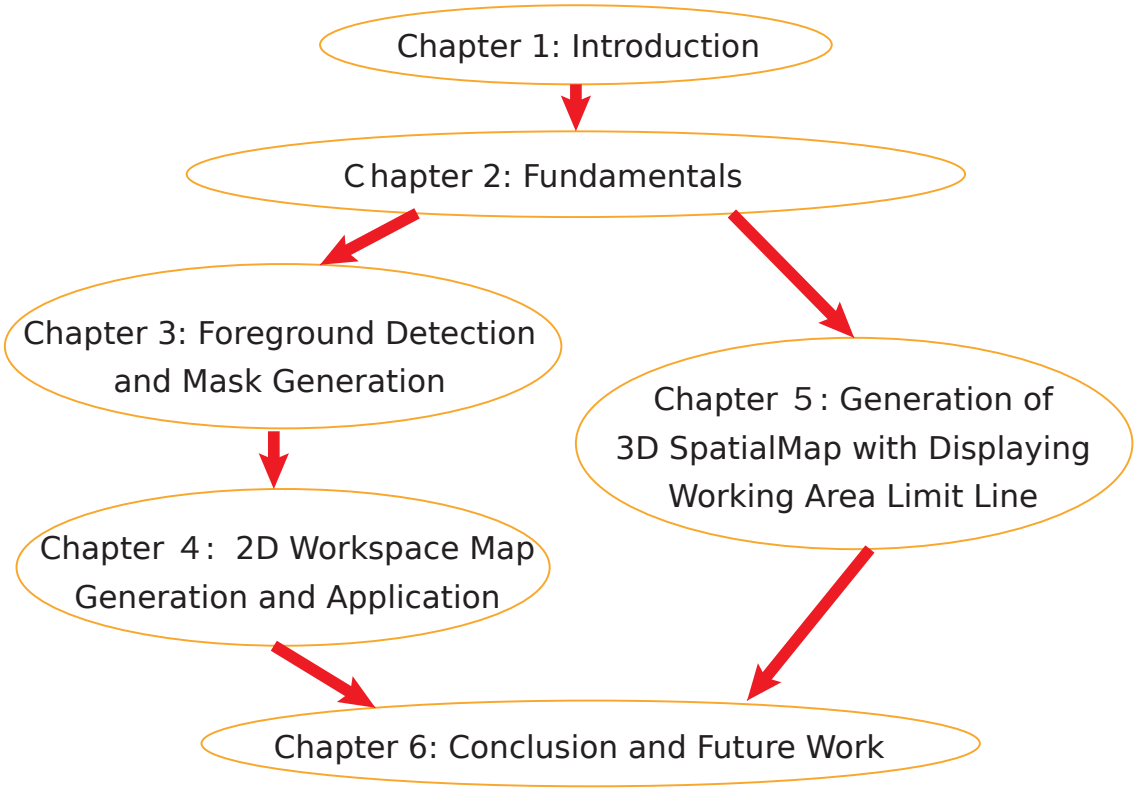


Fig. 1.12: Organization of this thesis

Chapter 2

Fundamentals

This chapter describes the basic concepts and technologies which are required for understanding the later chapters. This chapter is split into two parts. In the first part, basic concepts and technologies related to Chapter 3 and Chapter 4 are introduced which aiming at 2D workspace map generation, while in the second part those related to Chapter 5 are introduced.

For the first part, in the beginning, the image feature is explained. The image feature is an essential component of many computer vision applications. The explanation of the image feature includes a concept introduction and two specific types of image features. Following that, the matching approach for the image feature will be explained.

After that, the explanation comes to the 2D image transformation. There are various types of image transformations. Firstly, the homogenous coordinate is introduced. Based on it, the rigid transformation and homogenous transformation are explained in detail.

Following that, the image geometry is explained. The single view geometry and two view geometry are explained respectively. Many concepts such as camera pose and epipolar constraint are introduced in this section.

In the last section of the first part, the optical flow is explained briefly. A brief introduction to the optical flow is given at the beginning of this section. And following that, a DNN based approach FlowNet2 which is used in Chapter 3 is introduced.

Then in the second part, the introduction for visual SLAM is briefly given. In Chapter 5, a reconstruction pipeline is established based on the selection of the current visual SLAM approaches.

Related Basic Concepts and Technologies for 2D Workspace Map Generation

2.1 Image Feature Point

2.1.1 Introduction

For various computer vision applications and image processing techniques such as object recognition, structure from motion (SFM) and visual simultaneously location and mapping (VSLAM), the image feature detection and matching are always an essential component. For example, to stitch two images into a panoramic image, the first step is to detect the features in both images. And then, from one image to the other image, the detected features will be matched and many image feature correspondences will be established. There are many different types of image features. They can be divided into edges, corners, blobs and ridges [56].

The corner and blob are a feature of a local area of an image with representativeness. They are also called the image feature point which is always a prerequisite to compute camera pose, 2D homography for image stitching and some other applications. For the same scene, even the camera has some translational and rotational movement, the image feature point in the captured image should be detected out as usual.

For two different images, the image feature point usually needs to be detected respectively and matched. It means that for the image feature point there is always a detection pipeline and a matching pipeline.

In the image feature point detection pipeline, many state of the art detectors have been invented in the last decade such as Harris corner, scale-invariant feature transform (SIFT), sppeded up robust feature (SURF), features from accelerated segment test (FAST) and oriented FAST, rotated BRIEF (ORB) and KAZE. For a good detector, the following properties are required to detect the image feature point stably and robustly:

- Repeatability [79]: For the same image feature point of a region while in two or more different images, it can be detected out.
- Distinctiveness [79]: For different image feature point, the nearby region should have a description distinctively. The image feature point can have enough information

for matching.

- Efficiency [66]: In an image, the count of the image feature point should be far less than the pixels' count of the image.
- Locality: The image feature point is only related to a local region of an image.

For the image feature point, it always consists of two parts which are keypoint and descriptor.

The keypoint contains the information of where the image feature point's location in the image. It is a result of the image feature detection process. Different kinds of detectors define keypoint differently. For the images captured by a camera with different camera poses, the influence of illumination, distortion and texture is obvious. Besides, the scale difference is also a problem. These challenges require the detector to be robust and capable of these challenges, which means that even under different illumination and scale the image feature point can be detected out in a different image.

The descriptor is a description for a detected keypoint. Taking the detected image point as the center, the nearby region with pixel will be converted into a stable and compact descriptor. And the descriptor can be matched with the descriptors detected from other images.

With a reliable and robust matching algorithm and filtering algorithm, the descriptors from one image to the other image can be matched correctly in most cases. Thus, the image feature point can be detected and matched correctly.

2.1.2 FAST Corner

FAST (Features from Accelerated Segment Test) is a type of image feature point detection approach to search corners in the image. The main working principle of FAST is detecting the pixel of obvious intensity change locally.

FAST only cares about the changes in pixel's intensity while detecting the image feature point. It makes the detection for the image very fast. Compared to the other image feature point such as SURF and SIFT which cost 217.3 ms and 5228.7ms for 1000 image feature points respectively [8, 49], the detection of FAST corners only takes about 15.3 ms [63].

The principle of FAST corner detection is shown in Figure 2.1. Whether the pixel $\boldsymbol{p}$ shown in the image will be selected as a feature point depends on the nearby pixels on the dashed circle. It's a continuous sequence of pixels marked from 1 to 16. It is required that only a continuous N pixels' intensity is bigger than the pixel $\boldsymbol{p}$'s intensity with a threshold T_p the pixel $\boldsymbol{p}$ will be selected as a FAST corner. The radius of the circle is fixed as 3 pixels.

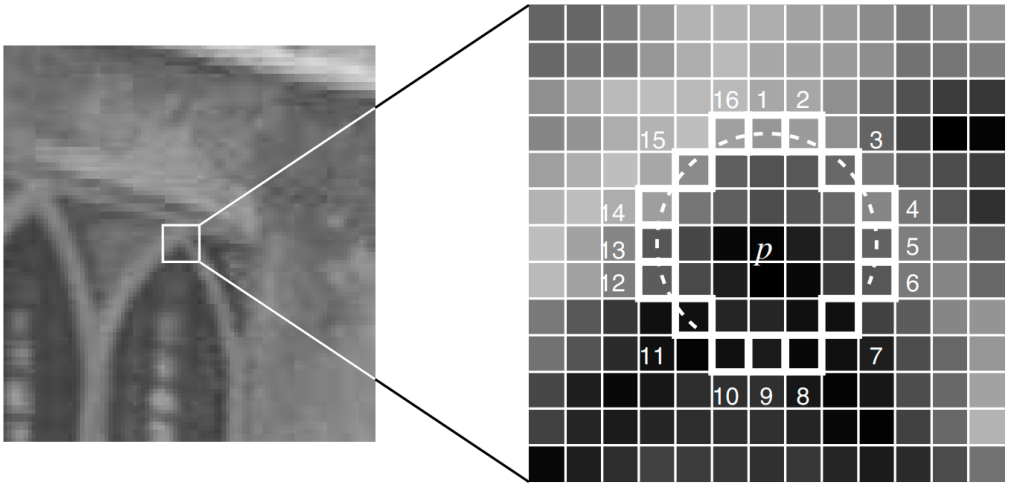


Fig. 2.1: FAST corner detection principle [63]. For the pixel $\boldsymbol{p}$, the nearby pixels on a dashed circle will be compared. Only a continuous N pixels' intensity is bigger than the pixel $\boldsymbol{p}$'s intensity with a threshold. The pixel $\boldsymbol{p}$ will be selected as a FAST corner.

In more detail, the algorithm of FAST corner detection can be explained as:

- The intensity of the pixel $\boldsymbol{p}$ is I_p.
- Setting a threshold T_p where $T_p = 0.2I_p$.
- Taking the pixel $\boldsymbol{p}$ as center, with a radius of 3 pixels, there will be 16 continuous

pixels on the circle. Check if a continuous N pixels are all bigger than $I_p + T_p$ or less than $I_p - Tp$ (Generally $N = 9, 11, 12$, and the correspondent FAST corner detection is called FAST-9, FAST-11 and FAST-12). There will be totally 16 pixels in the circle. For each pixel on the circle $x = 1, 2, \cdots, 16$ with intensity I_x, there will be three cases [63]:

$$\boldsymbol{S}_{\boldsymbol{p}\to\boldsymbol{x}} = \begin{cases} darker & I_x \leqslant I_p - T_p \\ similar & I_p - T_p \leqslant I_x \leqslant I_p + T_p \\ brighter & I_p + T_p \leqslant I_x \end{cases} \tag{2.1}$$

If a continuous of N pixels is darker or brighter, the pixel $\boldsymbol{p}$ is taken as a FAST corner.

- Do the above four steps for all the pixels in the image.

With the initial detection process mentioned above, the FAST corners in the image can be detected out. But there are always a lot of pixels in a very small region being detected as FAST corners. To make the detection more clean and reliable, generally, after the first time of detection, there will be a non-maximal suppression. In a small region, for the detected FAST corners in the first time, only the most significant one can be selected as the FAST corner.

To achieve the non-maximal suppression, a score function is defined as equation 2.2 [63].

$$V = \max\left(\sum_{x\in S_{\text{bright}}} |I_{p\to x} - I_p| - t, \sum_{x\in S_{\text{dark}}} |I_p - I_{p\to x}| - t\right) \tag{2.2}$$

For every detected corner, its score V can be computed. If a corner's score V is less than an adjacent corner's score V, the current corner will be removed.

2.1.3 Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) is an approach to transform image data into scale-invariant coordinates relative to local features [49]. It can provide a highly robust detection on feature point and good descriptor to describe the feature point. The main advantages of SIFT are as following:

- Stable and invariant: it is highly robust under rotation, scale and illumination conditions. To a certain extent, it can deal with cases such as image projective transformation and noise.
- Distinctiveness: The detected feature point with SIFT can be matched to a large database.
- Quantity: From a small image, there are many feature points that can be detected out with SIFT. A typical image of size 500x500 pixels will give rise to about 2000 stable features (although this number depends on both image content and choices for various parameters) [49].
- Efficiency: with optimized SIFT detection algorithm and GPU acceleration, the computation can be fast to even real-time.

As a type of image feature point, a SIFT feature also consists of a keypoint and a descriptor. To generate the SIFT features from an image, there are major four stages of computation [49]:

1. Scale-space extrema detection: To make a detection for SIFT features, firstly, the image will be used to construct an image pyramid consisting of many octaves. By applying the difference of Gaussian (DoG) to every octave, each octave will contain many images. The potential feature points are invariant to scale and orientation.
2. Keypoint localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability [49].
3. Orientation assignment: at each keypoint, the local gradient of the nearby region will be computed and one or more orientations are assigned to the keypoint. Later

operations are also binded with the orientation and thus the invariance of scale and rotation is ensured.

4. Keypoint descriptor: as mentioned in the above sections, the descriptor is used for feature matching. Measurement around the SIFT keypoint will be conducted.

The four stages of SIFT feature detection can be divided into the keypoint detection process and descriptor extraction process.

SIFT Keypoint Detection

The SIFT keypoint detection consists of a rough potential keypoint detection, a more precise approximation for the location of keypoint and the removal of unstable potential keypoint. First, a rough detection for the potential keypoint will be made. The potential keypoint in the image will be detected out in pixel precision. Following that, with the nearby data of a potential keypoint, a more detailed fit will be applied to have a more precise location in sub-pixel for the keypoint. In the last, the unstable keypoint with low contrast and edge responses will be removed.

The potential keypoint detection has to face the problem of scale. SIFT deal with the scale problem by constructing a scale space. The scale space of SIFT is an image pyramid with many octaves. And for different octave of the scale space, the difference of Gaussian will be conducted to generate many difference-of-Gaussian images as shown in equation 2.3 [49].

$$\begin{cases} G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(x^2+y^2\right)/2\sigma^2} \\ L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \\ D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \end{cases} \tag{2.3}$$

From the above process, it is clear that the scale space is consists of o octaves of the image pyramid and s levels generated with $G(x,y,\sigma)$. The process can be seen in Figure 2.2. The left side is an image pyramid. From the previous octave to the next octave, the image is resized to a half. And in each octave, with different Gaussian kernel, there are many levels. For each neighbor level, the difference will form many DoG images. The SIFT keypoint is detected in the DoG images.

The SIFT keypoint detection process is shown in Figure. For three adjacent DoG images, the pixel will be compared with its neighboring 26 pixels. If the pixels' intensity

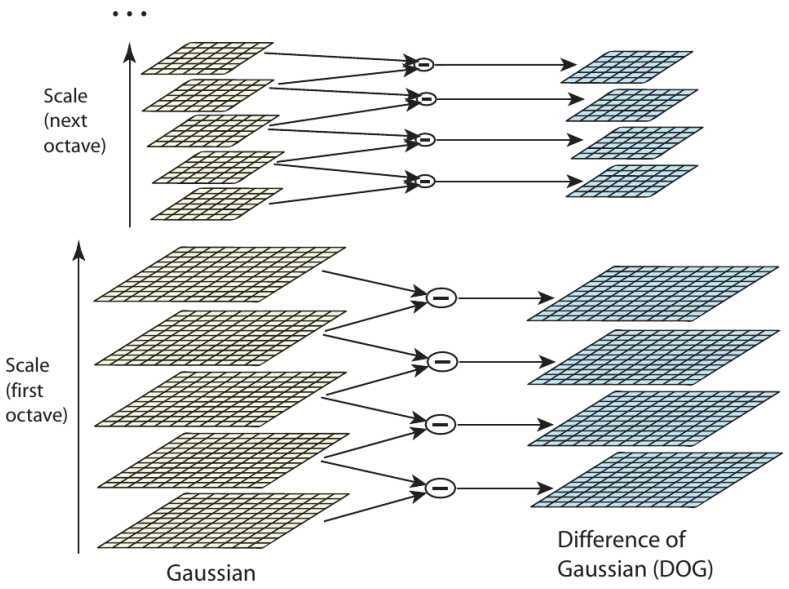


Fig. 2.2: The scale space [49]. The scale space consists of o octaves. In each octave, by a convolution operation with different Gaussian kernels, s levels will be generated. The DoG is the difference between two adjacent images. Then for each octave, there are $s-1$ DoG images will be generated.

is minima or maxima, it can be selected as a SIFT potential keypoint with the precision for pixel.

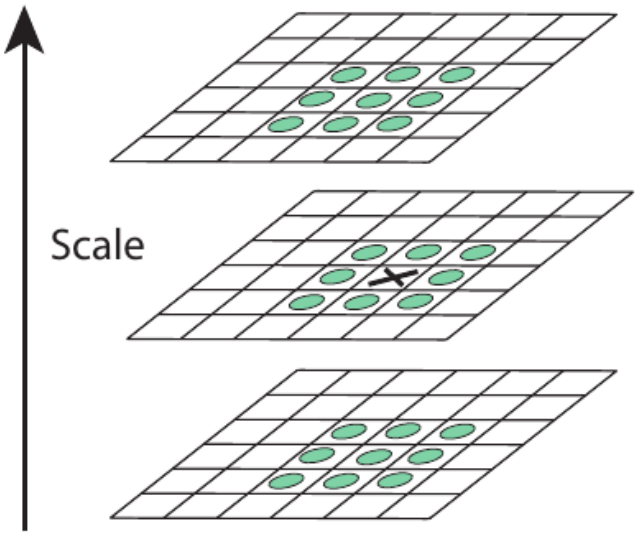


Fig. 2.3: The SIFT keypoint will be detected on three continuous DoG images2.3. If the pixel's intensity is bigger than or less than the other 26 neighbors' intensity, the pixel will be considered as a SIFT keypoint.

To have a more precise location of the keypoint, a local fitting with the nearby sample

points need to be made. The fitting is through a 3D quadratic function and it will compute the interpolated location of the maximum [12]. This fitting is achieved with Taylor expansion of the scale space $D(x, y, \sigma)$ which is shown in equation 2.4 [49].

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{2.4}$$

where the evaluation for the derivatives of D and itself are computed at the sample point. And $\mathbf{x} = (x, y, \sigma)$ denotes the potential keypoint's offset. By setting the equation 2.4 equals zero, the location can be estimated more precisely. It gives us:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}} \tag{2.5}$$

which represents the location of the potential keypoint with a precision up to sub-pixel.

The last step is to reject the bad potential keypoint. The removal of bad potential keypoint is by the low contrast and edge responses. By substituting equation 2.5 to equation 2.4, it gives the contrast value $D(\hat{\mathbf{x}})$ [49]. If $|D(\hat{\mathbf{x}})|$ is less than 0.03, the keypoint is a low contrast keypoint and should be removed.

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\hat{\mathbf{x}} \tag{2.6}$$

In the area of the edges, the DoG has a more strong response on the edge responses. The performance on edges will be affected by very small noise. In the DoG function, a poorly defined peak keeps a large principal curvature across the edges but a small one in the perpendicular direction [49]. For the area around the location of the potential keypoint of the scale, a 2×2 Hessian matrix $\mathbf{H}$ will be used to compute the principal curvatures. The matrix $\mathbf{H}$ [49] is defined in as:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{2.7}$$

where the derivatives are estimated numerically by the neighboring pixels. Assuming the α and β are the two eigen values with the larger magnitude and smaller magnitude respectively, the sum and product of the eigen values can be computed with the trace and determinant [49] respectively:

$$\begin{aligned} \mathrm{Tr}(\mathbf{H}) &= D_{xx} + D_{yy} = \alpha + \beta \\ \mathrm{Det}(\mathbf{H}) &= D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \end{aligned} \tag{2.8}$$

Let $r = \frac{\alpha}{\beta}$ be the ratio of two eigen values, from equation 2.8,we can have the equation [49]:

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{2.9}$$

Then the difference which is directly represented by the two eigen values can be denoted with the trace and determinant of the Hessian matrix. If the two eigen values are close, the r will increase. So the only thing we need to check becomes:

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r} \tag{2.10}$$

With equation 2.10, for the pricipal curvatures of the potential keypoint greater than r can be eliminated.

SIFT Descriptor Extraction

The keypoint descriptor consist of a 128 dimensional vector. It is a description of the keypoint's nearby region. To ensure its rotation invariant property, a nearby region of keypoint will be used to compute the main direction of the keypoint. The computation for the main direction is shown in equation 2.11 [49]. For the local region pixels, $m(x, y)$ denoted every pixel's gradient length and $\theta(x, y)$ denotes every pixel's gradient angle. Through the histogram analysis of the gradient, the main direction can be found. Better optimization can be made by an approximation with the three bins close to the main direction.

$$\begin{aligned} m(x,y) &= \sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \\ \theta(x,y) &= \tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y))) \end{aligned} \tag{2.11}$$

After that, the descriptor can be obtained with the nearby region of a keypoint. The local area of the keypoint will be rotated with the main direction mentioned above. As shown in Figure 2.4, a local region of 16×16 pixels are selected out to generate the descriptor of keypoint. The left one is a computation of the pixel gradient. It is separated into four 4×4 small region. Again a histogram for the gradient of the small region will be made. Each bin will represent 45 degrees. So, each small region will produce an 8-dimensional vector, and totally a 64-dimensional vector will be generated as the descriptor for keypoint. A larger size of 16×16 pixels can also be used to generate the descriptor with a 128-dimensional vector.

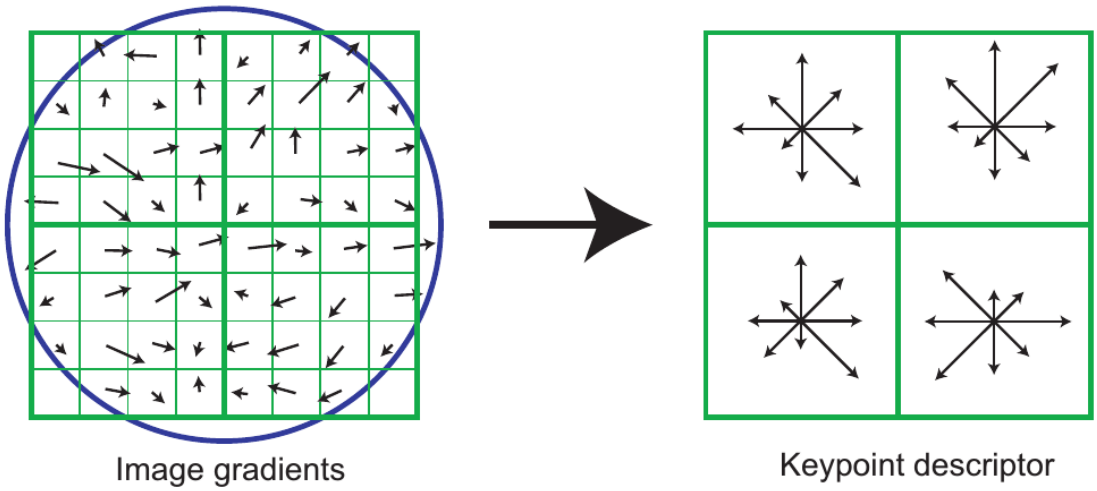


Fig. 2.4: The descriptor is computed around the keypoint [49]. A local region of 8×8 or 16×16 pixels can be chosen to describe the keypoint.

2.1.4 Feature matching

Feature matching is an important way to associate two images by establishing the corresponding relationship of the image feature points. There are various ways to find the image feature point's correspondent feature point on the other image such as the least-squares of intensity, optical flow and descriptor matching. With the matched feature points, many applications such as camera pose estimation and the geometry relationship between two images can be made.

For the feature points, they can be matched with the descriptors which are one component of the feature point. For different feature points' descriptors, they are different in representation. But generally, the descriptor is a vector with different representations. For example, the SIFT descriptor is a 128-dimensional vector and each component of the vector represents a gradient distribution of a local 4×4 pixels region.

Assuming that in the image $\boldsymbol{I}_t$, there are totally M feature points have been detected and their keypoints and descriptors are noted as $\boldsymbol{x}_{t,m}$ and $\boldsymbol{v}_{t,m}$ for $m = 1, 2, \cdots, M$ respectively. The feature points are also detected out in the image $\boldsymbol{I}_{t+1}$ as $\boldsymbol{x}_{t+1,n}$ and $\boldsymbol{v}_{t+1,n}$ for $n = 1, 2, \cdots, N$. For the similar feature points, the descriptors should also be similar, which means that the vector length should be similar. The matching process can have many algorithm such as brute-force algorithm and FLANN algorithm. Taking the SIFT descriptor for example, the length of i-th feature point's descriptor $\boldsymbol{v}_{t,i}$ is:

$$\left\| v_{(t,i)} \right\| = \sqrt{\sum_{k=1}^{128} v_k^2} \tag{2.12}$$

It is the same for the image $\boldsymbol{I}_{t+1}$. All the descriptor in the image $\boldsymbol{I}_{t+1}$ will be compared with the descriptor $\boldsymbol{v}_{t,i}$ by checking the length and the j-th descriptor which has the minimum difference:

$$\left| \left\| v_{(t,i)} \right\| - \left\| v_{(t+1,j)} \right\| \right| \leqslant \left| \left\| v_{(t,i)} \right\| - \left\| v_{(t+1,k)} \right\| \right| \tag{2.13}$$

for $k = 1, 2, \cdots, j-1, j+1, \cdots, N$. Then the i-th feature point in the image $\boldsymbol{I}_t$ is associated with the j-th feature point in the image $\boldsymbol{I}_{t+1}$. To do this for all the feature points in the image $\boldsymbol{I}_t$, the correspondent feature points can be found.

2.2 2D Image Transformation

2.2.1 2D Homogeneous Coordinate

Homogeneous coordinate is a representation for a n dimension point with a $n+1$ dimension representation. It is widely used in image transformation and projective geometry.

2D homogeneous coordinate is using a 3 dimensional vector to represent a 2D plane point. For a point $\mathbf{x} = (x, y)^\mathsf{T}$ lying on a line $\mathbf{l} = (a, b, c)^\mathsf{T}$ can be represented as:

$$ax + by + c = 0 \quad \Leftrightarrow \quad (x, y, 1)(a, b, c)^\mathsf{T} = 0 \tag{2.14}$$

which means that the plane point $(x, y)^\mathsf{T}$ has a representation as a 3-vector with a final coordinate of 1. For equation 2.14, with a non-zero constant k, the equation becomes:

$$(kx, ky, k)(a, b, c)^\mathsf{T} = 0 \tag{2.15}$$

which makes it natural to use the set of vectors (kx, ky, k) to represent the point (x, y) in a 2D plane.

For a 2D point's representation in homogeneous coordinate which is denoted with a 3 components vector $\mathbf{x} = (x_1, x_2, x_3)^\mathsf{T}$, it represents a 2D point with the coordinate of $(x_1/x_3, x_2/x_3)$.

It is the same to use homogeneous coordinate to represent a 2D line (ka, kb, kc) because for the point meet equation 2.14 will still meet the equation:

$$(x, y, 1)(ka, kb, kc) = 0 \tag{2.16}$$

With homogeneous coordinate for the points and lines in 2D, it is more convenient to do transformations and find the relationships. For example for two 2D point in a plane, the line which cross the two points can be obtained by the cross product of two homogeneous coordinates:

$$\mathbf{l} = (x_1, y_1, 1) \times (x_2, y_2, 1) \tag{2.17}$$

And it also the same for two lines which meet at the same point (kx, ky, k):

$$(kx, ky, k) = (a_1, b_1, c_1) \times (a_2, b_2, c_2) \tag{2.18}$$

The homogeneous coordinate can also represent a point at infinity by just changing the last component of the vector to zero. It is a very useful way for projective geometry and

image transformation. For a two dimensional plane point $(x, y) \in \mathbb{R}^2$, its homogeneous coordinate $(x, y, 1)$ is in $\mathbb{R}^3$. The set of all the homogeneous coordinate except for $(0, 0, 0)$ forms the perspective space $\mathbb{P}^2$ [36].

2.2.2 Rigid Transformation

The rigid transformation is a basic operation for the image transformations. It is a transformation with only the translation and rotation of the image. The rigid transformation is a Euclidean property preserving transformation. As it preserves the Euclidean properties after transformation, it is also called Euclidean transformation. For an image, the transformed image has the same measure as the original image. The angles and lengths of some objects in the transformed image will be the same as the original image.

The rigid transformation can be represented as:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos\theta & sin\theta & t_x \\ -\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2.19}$$

in 2D homogeneous coordinate. The (x', y') is the coordinate after the rigid transformation. And the (x, y) is the coordinate before the rigid transformation. The rigid transformation models the motion of a rigid object. They are by far the most important iosmetries(ios=same, metries=same measure) in practice [36]. The geometry properties for the rigid transformation can be more clearly represented with the block form of equation 2.19:

$$\boldsymbol{x}' = \boldsymbol{H}_R \boldsymbol{x} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^\mathsf{T} & 1 \end{bmatrix} \boldsymbol{x} \tag{2.20}$$

where $\boldsymbol{R}$ is a 2×2 rotation matrix (orthogonal matrix which makes $\boldsymbol{R}\boldsymbol{R}^\mathsf{T} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R} = \boldsymbol{I}$), $\boldsymbol{0}$ is a null 2 dimensional vector and $\boldsymbol{t}$ is a translation 2 dimensional vector.

The rigid transformation has three degrees of freedom. As shown in the equation 2.20, the block matrix $\boldsymbol{R}$ has one degree of freedom for rotation. And the vector $\boldsymbol{t}$ has two degrees of freedom for the translation along the x-axis and y-axis. The rigid transformation can be confirmed with two correspondent points between two images.

As mentioned, the invariants of the rigid transformation are clear. The invariants of rigid transformation are length (the distance of two points), angles (the angle between two lines) and area. A rigid transformation is a subgroup of higher transformations such as affine transformation and projective transformation.

2.2.3 2D Homography

2D homography is a 2D projective transformation for the plane. The projective transformation is defined as:

$$\boldsymbol{x}' = \boldsymbol{H}_P \boldsymbol{x} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ \boldsymbol{v}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ v_1 & v_2 & v \end{bmatrix} \tag{2.21}$$

where the matrix $\boldsymbol{H}_p$ is the 3×3 matrix representing the 2D projective transformation of a 2D plane.

As shown in equation 2.21, the projective transformation has four components which are the affine transformation $\boldsymbol{A}$, the translation $\boldsymbol{t}$ and the vector $\boldsymbol{v} = (v_1, v_2)$.

Generally, the 2D projective transformation is called 2D homograpohy. It is a mapping from $\mathbb{P}^2$ to $\mathbb{P}^2$ with a non-singular matrix $\boldsymbol{H}_P = \boldsymbol{H}_{3\times3}$, just as shown in equation 2.21. For each point in $\mathbb{P}^2$ which is represented as a three dimensional vector $\boldsymbol{x}$, a linear mapping will be applied to get its mapping point in $\mathbb{P}^2$ as $\boldsymbol{H}_P\boldsymbol{x}$. The process can be more clear in Figure 2.5.

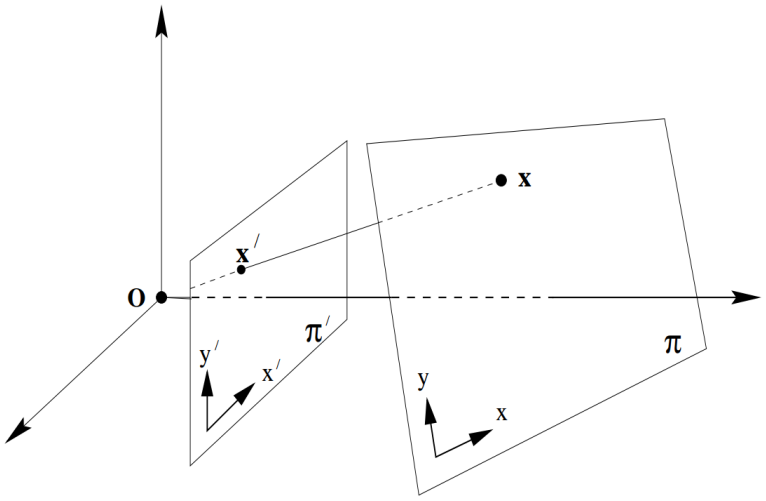


Fig. 2.5: Projective transformation [36]. A central projection takes the point $\boldsymbol{x}$ in the plane $\boldsymbol{\pi}$ to the point $\boldsymbol{x}'$ in the plane $\boldsymbol{\pi}'$. The mapping process is a linear mapping of homogeneous coordinate which is $\boldsymbol{x}' = \boldsymbol{H}_P\boldsymbol{x}$.

After the homography (projective transformation), there are distortions. The invariants after the projective transformation are only line preserving which means a line in the original image after the projective transformation is still a line.

The homography is the key for the image stitching and some other applications. To know the homography between two images, at least four correspondent points should be found. This is because the 3×3 projective matrix $\boldsymbol{H}$ has 9 entries which are only defined up to scale. So, it has only 8 degrees of freedom. Four correspondences of point can provide 8 constraints. It can be solved with the direct linear transformation (DLT) algorithm [36], the objective is trying to find the homography $\boldsymbol{H}$ with the known n ($n \geq 4$) pairs of 2D to 2D point correspondences $\{\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}'_i\}$.

2.3 Image View Geometry

An image is a mapping from the 3D world to a 2D plane. The mapping process can be different such as orthogonal projection (affine camera), central projection (pin-hole camera) and spherical projection (fish-eye camera). In this section, the pin-hole camera imaging process is explained. The single view geometric process of projection at different camera poses is explained. Following that, an explanation for two view geometry is made.

2.3.1 Single View Geometry

For the pin-hole camera, the image is a linear mapping from 3D environment to a 2D plane. The central projection will take the 3D point to a 2D plane, just as shown in Figure 2.6.

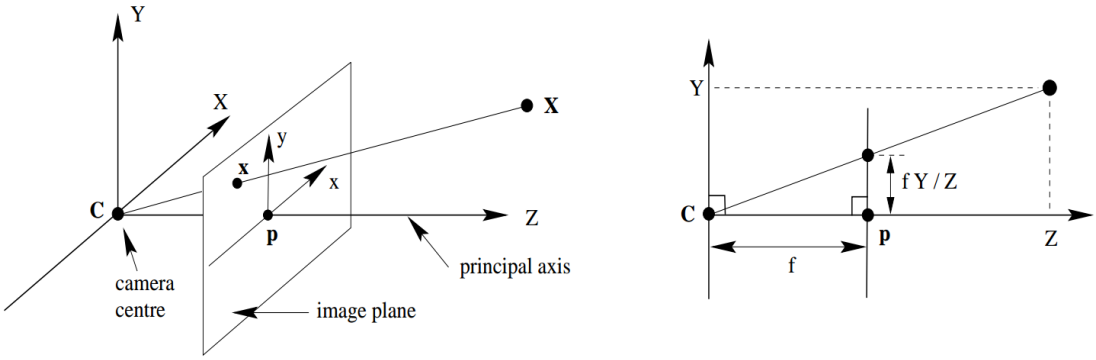


Fig. 2.6: Central projection of the pin-hole camera [36]. The 3D point **X** is projected to the image point **x**. The origin **C** is the camera center and the image origin **p** is the principal point.

From the right part of Figure 2.6, assuming that the coordinate for the 3D point is (X, Y, Z), the imaged point (x, y) on the plane $z = f$ is:

$$\begin{cases} x = \frac{fX}{Z} \\ y = \frac{fY}{Z} \end{cases} \tag{2.22}$$

which can be seen from Figure 2.6. The central projection can also be expressed using homogeneous coordinates. If the 3D point $\mathbf{X}$ and the imaged 2D point x are all using homogeneous coordinates, the central projection can be described as:

$$w\mathbf{x}_h = \mathbf{P}\mathbf{X}_h \leftrightarrow w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.23}$$

where the matrix $\boldsymbol{P}$ is called the projection matrix. And the left 3×3 of $\boldsymbol{P}$ can be denoted as $\boldsymbol{K}$ which is called the camera calibration matrix [36].

For the projected point, there are two more cases that need to be considered. One is for the offset which needs a translation for the projected point. The other one is that the image coordinate is in pixels that need a scale along the x-axis and y-axis.

For the offset, a translation can be added to the camera calibration matrix $\boldsymbol{K}$ to make it as:

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.24}$$

which can give a translation after the 3D point is projected to the image plane. The translations along x-axis and y-axis are p_x and p_y respectively.

The scale problem can be solved by a scale factor respectively to x-axis and y-axis. The camera calibration matrix can be changed as:

$$\boldsymbol{K} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.25}$$

where $\alpha_x = m_x f$, $\alpha_y = m_y f$, $x_0 = m_x p_x$ and $y_0 = m_x p_y$. The scale factor along x-axis and y-axis are m_x and m_y respectively. The camera calibration matrix can be obtained through chessboard camera calibration [24]. The projection process can then be denoted as:

$$\boldsymbol{x}_h = \boldsymbol{K}[\boldsymbol{I} \mid \boldsymbol{0}]\boldsymbol{X}_h \tag{2.26}$$

For the camera center which is not lying on the origin of the world coordinate frame, we need to consider the camera pose which consists of a camera translation and rotation. As shown in Figure 2.7 [36], the camera center $\boldsymbol{C}$ is away from the world coordinate frame center $\boldsymbol{O}$ and the principle axis is not toward the z-axis of the world coordinate frame.

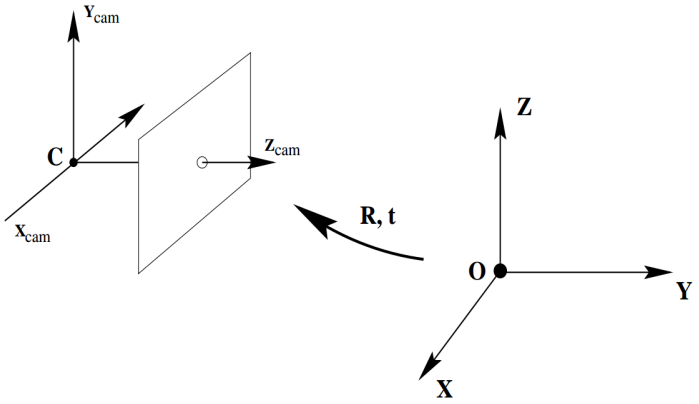


Fig. 2.7: A transformation from the world coordinate frame to the camera coordinate frame [36].

Assuming the camera coordinate in the world coordinate frame is $\boldsymbol{C}$ and a 3D point in the world coordinate frame is $\boldsymbol{X}_w$, the 3D point's coordinate in the camera coordinate

frame $\boldsymbol{X}_c$ can be expressed as:

$$\boldsymbol{X}_c = \boldsymbol{R}(\boldsymbol{X}_w - \boldsymbol{X}_c) \tag{2.27}$$

where $\boldsymbol{R}$ is a 3×3 matrix. And the camera pose is generally consists of the rotation $\boldsymbol{R}$ and the translation $\boldsymbol{t} = \boldsymbol{C}$. Then the final projection at a camera pose can be expressed with equation 2.28.

$$\boldsymbol{x}_h = \boldsymbol{P}\boldsymbol{X}_w = \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I} \mid -\boldsymbol{X}_c]\boldsymbol{X}_w \tag{2.28}$$

2.3.2 Two View Geometry

Given two images taken at different camera poses, the geometric relationship can be estimated such as the relative rotation and translation direction.

The epipolar constraint is a basic constraint for two images. The epipolar geometry between two views is essentially the geometry of the intersection of the image planes with the pencil of planes having the baseline as an axis (the baseline is the line joining the camera centers) [36]. The epipolar constraint can always be denoted with the fundamental matrix $\boldsymbol{F}$ and computed with the point correspondences of the two images.

Assuming that the same 3D point $\boldsymbol{X}$ is mapped to two different images as $\boldsymbol{x}$ and $\boldsymbol{x}'$ respectively with the same camera at different camera poses $\boldsymbol{C}$ and $\boldsymbol{C}'$, just as shown in Figure 2.8 [36], the imaged point $\boldsymbol{x}$ and $\boldsymbol{x}'$ should lie on the plane $\boldsymbol{\pi}(\boldsymbol{XCC}')$. And the plane $\boldsymbol{\pi}$ is called epipolar plane. In Figure 2.8b, the inverse projection line from $\boldsymbol{C}$ to $\boldsymbol{x}$ is correspondent to the line $\boldsymbol{l}'$ and the point $\boldsymbol{X}$ which has been projected to right image plane should be in the line $\boldsymbol{l}'$. The line $\boldsymbol{l}'$ is the epipolar line. The points $\boldsymbol{e}$ and $\boldsymbol{e}'$ are generated by the intersection of the two image plane and the line $\boldsymbol{CC}'$. They are called epipole.

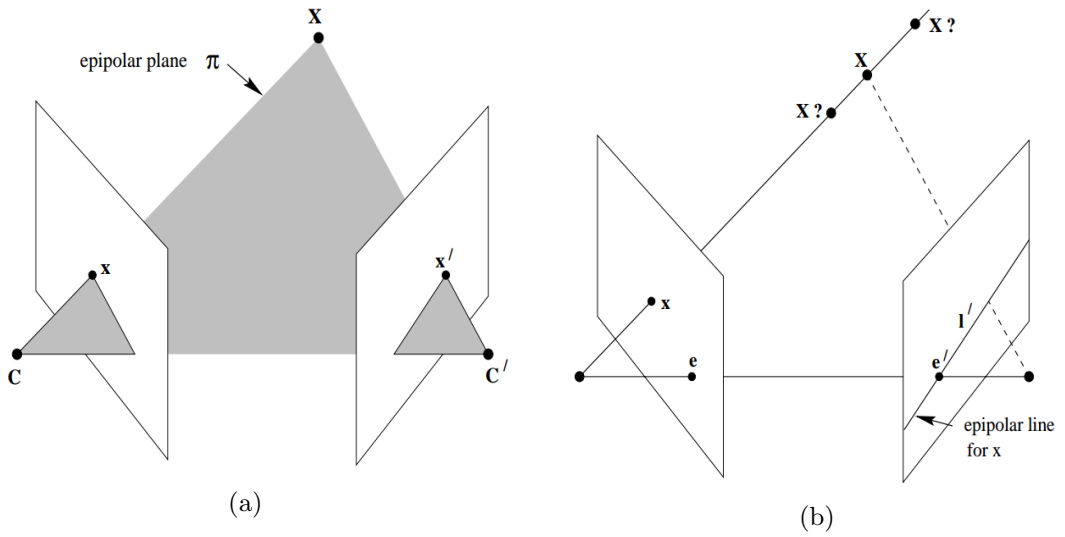


Fig. 2.8: Point correspondence geometry [36]. (a)The projection at two different camera poses $\boldsymbol{C}$ and $\boldsymbol{C}'$. The five points $(\boldsymbol{X}, \boldsymbol{x}, \boldsymbol{C}, \boldsymbol{C}', \boldsymbol{x}')$ are on the epipolar plane $\boldsymbol{\pi}$. (b)The projection line from $\boldsymbol{C}$ to the point $\boldsymbol{X}$ contains the point $\boldsymbol{X}$. And the projection line is correspondent to the epipolar line $\boldsymbol{l}'$. The projection of the point $\boldsymbol{X}$ on the right image should lie on the epipolar line $\boldsymbol{l}'$.

With only the two images, given only the point $\boldsymbol{x}$ and the epipolar line $\boldsymbol{l}'$ for the point, it can only be figured out that its correspondent point $\boldsymbol{x}'$ is on the epipolar line $\boldsymbol{l}'$. And the epipolar line can be computed with the point $\boldsymbol{x}$ and a fundamental matrix $\boldsymbol{F}$ between the two images.

As shown in Figure 2.8a, taking the camera center point $\boldsymbol{C}$ as the world coordinate frame, the point $\boldsymbol{X} = (X, Y, Z)$ is projected to $\boldsymbol{x} = (x, y, 1)$ and $\boldsymbol{x}' = (x', y', 1)$ respectively with:

$$\begin{cases} w_1\boldsymbol{x} = \boldsymbol{K}\boldsymbol{X} \\ w_2\boldsymbol{x}' = \boldsymbol{K}(\boldsymbol{R}\boldsymbol{X} + \boldsymbol{t}) \end{cases} \tag{2.29}$$

where $\boldsymbol{R}$ and $\boldsymbol{t}$ are the relative motion of the camera and $\boldsymbol{R}\boldsymbol{X} + \boldsymbol{t}$ takes the point $\boldsymbol{X}$ to the coordinate in the coordinate frame of the camera at position $\boldsymbol{C}'$. The process is a up to scale model which can make $w_1 = w_2 = w$. And it can be like:

$$\begin{cases} \boldsymbol{x} = \boldsymbol{K}\boldsymbol{X}/w \\ \boldsymbol{x}' = \boldsymbol{K}(\boldsymbol{R}\boldsymbol{X}/w + \boldsymbol{t}/w) \end{cases} \leftrightarrow \begin{cases} \boldsymbol{x} = \boldsymbol{K}\boldsymbol{X}_n \\ \boldsymbol{x}' = \boldsymbol{K}(\boldsymbol{R}\boldsymbol{X}_n + \boldsymbol{t}_n) \end{cases} \tag{2.30}$$

where $\boldsymbol{X}_n = \boldsymbol{X}/w$ and $\boldsymbol{t}_n = \boldsymbol{t}/w$. The coordinate can be transformed from the pixel to the normalized coordinate with:

$$\begin{cases} \boldsymbol{r} = \boldsymbol{K}^{-1}\boldsymbol{x} \\ \boldsymbol{r}' = \boldsymbol{K}^{-1}\boldsymbol{x}' \end{cases} \tag{2.31}$$

and taking it into equation 2.30, we will get:

$$\boldsymbol{r}' = \boldsymbol{R}\boldsymbol{r} + \boldsymbol{t}_n \tag{2.32}$$

multiplying with $\boldsymbol{t}_n^\wedge$ on both sides, the equation becomes:

$$\boldsymbol{t}_n^\wedge \boldsymbol{r}' = \boldsymbol{t}_n^\wedge \boldsymbol{R}\boldsymbol{r} \tag{2.33}$$

because of $\boldsymbol{t}_n^\wedge \boldsymbol{t}_n = \boldsymbol{0}$. For a three dimensional vector $\boldsymbol{t}_n = (t_x, t_y, t_z)$, $\boldsymbol{t}_n^\wedge$ is defined as:

$$\boldsymbol{t}_n^\wedge = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{2.34}$$

which can achieve the cross product with a three dimensional vector with a matrix form in computation. For equation 2.33, multiply $\boldsymbol{r}'^{\mathsf{T}}$, the equation becomes:

$$\boldsymbol{r}'^{\mathsf{T}} \boldsymbol{t}_n^\wedge \boldsymbol{R}\boldsymbol{r} = 0 \tag{2.35}$$

because $\boldsymbol{t}_n^\wedge \boldsymbol{r}'$ is perpendicular to both $\boldsymbol{t}_n^\wedge$ and $\boldsymbol{r}'$. By replacing the coordinate in equation 2.35 with the pixel coordinate, it will becomes:

$$\boldsymbol{x}'^{\mathsf{T}} \boldsymbol{K}^{-\mathsf{T}} \boldsymbol{t}_n^\wedge \boldsymbol{K}^{-1} \boldsymbol{x} = \boldsymbol{0} \tag{2.36}$$

In equation 2.35, the essential matrix is defined as $\boldsymbol{E} = \boldsymbol{t}_n^{\wedge}\boldsymbol{R}$. The essential matrix contains the relative camera pose $\boldsymbol{t}$ and $\boldsymbol{R}$. But as the constraint is only up to scale. The translation vector $\boldsymbol{t}$ can only gives the camera moving direction and gives a reconstruction up to scale. The fundamental matrix is defined as $\boldsymbol{F} = \boldsymbol{K}^{-\mathsf{T}}\boldsymbol{t}_n^{\wedge}\boldsymbol{K}^{-1}$. As can be seen, the fundamental matrix and the essential matrix only have a difference with the calibration matrix $\boldsymbol{K}$. With at least 7 pairs of correspondences, the fundamental matrix can be obtained with DLT algorithm [36].

2.4 Optical Flow and FlowNet2

Optical flow is a motion estimation for each pixels between two images. In the past decades, various traditional optical flow estimation approaches have been invented such as Lucas-Kanade method [14], Horn-Schunck method [38] and so on. The traditional approaches are trying to estimate each pixel's motion by minimizing the brightness or color difference between corresponding pixels. The constraint can be given as

$$\boldsymbol{I}(x, y, t) = \boldsymbol{I}(x + \delta x, y + \delta y, t + \delta t) \tag{2.37}$$

for a 2D+t dimensional representation of two images.

Except for the traditional approach, with the rapid development of deep learning, there are many DNN based optical flow estimation method. The traditional optical estimation approaches can have a good result when the pixel doesn't move very far. For a long-distance moving of a pixel, even a coarse-to-fine optical flow estimation can be made by building the Gaussian pyramid of the images can be made, the result can sometimes still not be very precise to use. In contrast, the optical flow generated with the DNN approaches can have a better result, especially in the training stage if there can be a lot of data that has large pixel motion. One of the representative is FlowNet2 [39].

FlowNet2 is a DNN approach for estimating the dense optical flow. It can estimate every pixel's motion of an image. The architecture of FlowNet2 is shown in Figure 2.9 [39]. FlowNet2 has made improvement by considering the small displacement compared with FlowNet [26]. It can have very good optical estimation result even compared with the best traditional state of the art technologies for the estimation of optical flow.

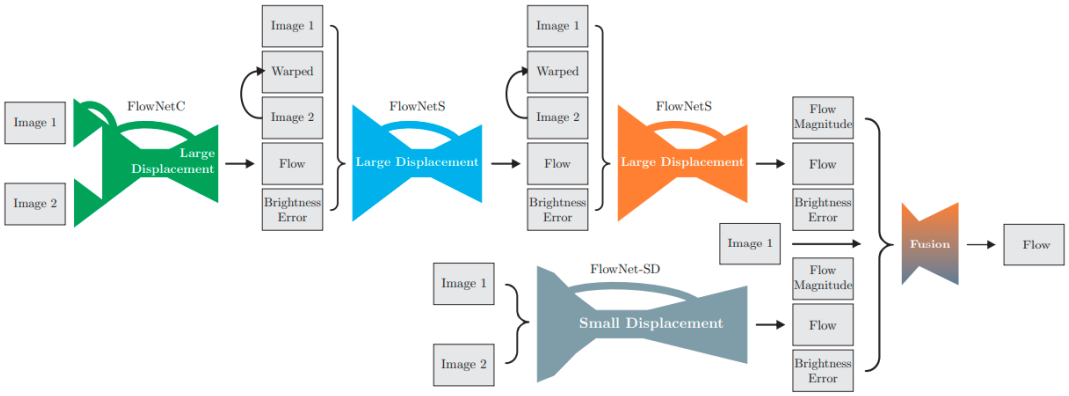


Fig. 2.9: The architecture of FlowNet2 [39]

Related Works for Chapter 5

2.5 Visual SLAM

To reconstruct the environment in 3D, various methods can be applied to acquire the 3D spatial map consisting of many 3D points. Based on whether the sensor has the capability of directly sensing the depth, the methods to reconstruct the environment can be divided into range sensor-based and algorithm-based.

The range sensor has the capability of sensing the depth information. There are many dierent types of range sensor based on dierent principles of physics, such as LiDar, Sonar and radar. In recent years, some researches have used LiDar to reconstruct the crane's working environment. Such as the researches [13, 25], the LiDar has been used to acquire the depth information of the environment. Also for the crane's working environment, the LiDar has been used for the reconstruction of a tower crane's working environment [5].

Visual SLAM is the algorithm based approach for 3D reconstruction with only one or several RBG cameras. The images taken by the cameras will be used to reconstruct the 3D environment with algorithms. Visual SLAM can both estimate the camera pose and reconstruct the environmental map simultaneously. Sometimes it is called tracking (tracking camera pose) and mapping (reconstruct environment map) (TAM).

In recent years, as the development of the convolutional neural network, many deep neural networks are developed to reconstruct the 3D environment and have the same functionalities as traditional visual SLAM approaches. They can recover the environment in 3D with one single image [67], two images [75] and a sequence of images [82]. Many DNNs have been used for camera pose estimation and map reconstruction. But these works in reconstruction will contain distortions in some cases and do not have a good performance for a large scale reconstruction and camera pose estimation.

The traditional visual SLAM is based on the multiple view geometry and many constraints to recover the camera pose and environment map. In the past decade, many state of the art technologies have been invented and have a good performance. In recent years, many commercial VSLAM software is developed for 3D reconstruction of the environment such as Autodesk Remake [61] and Pixe4D [57]. These commercial soft wares can make a reconstruction for the environment with a sequence of RGB images only. Autodesk Re-

make provides a cloud computation for reconstruction. As we tested it with our data, for about 80 images, it takes about 20 to 30 minutes for reconstruction and a dense 3D point cloud of the environment can be obtained. Pix4D also takes a long time in reconstruction. Through the processing feedback of Pix4D, it can be confirmed that the reconstruction with Pix4D is based on the image features. The reconstruction result is a sparse 3D point cloud and contains fewer 3D points than the result of Autodesk Remake. The algorithm-based approach for the 3D reconstruction is cheap and convenient to be integrated into the crane.

By the consideration of the dependence on image features, the traditional visual SLAM can be divided into feature-based VSLAM, semi-direct VSLAM and direct VSLAM.

The feature-based VSLAM needs to detect and match features for the images. Generally, it takes a lot of time for the detection and matching process. Many VSLAM approaches are feature-based methods. One of state of the art technologies is ORB-SLAM [65] which relies on ORB features. For the feature-based approach, the reconstruction result is decided by the feature amount detected from the images. The reconstructed environment can only be described with a sparse 3D point cloud consisting of thousands of points.

The semi-direct VSLAM also depends on the image features. But the feature detection is only required for a few images. Unlike the feature-based VSLAM, as there is almost no time consumption in the process of feature detection and matching, the semi-direct approaches can save a lot of time. One representative of this approach is semi-direct visual odometry (SVO) [31]. SVO can process 55 frames per second for the image size of 752×480 with an embedded platform, which means it is a real-time approach with a high frame rate. As the semi-direct approaches still depend on the image features. The reconstructed environment still consists of just thousands of 3D points.

For the direct VSLAM, it is not using image features such as dense tracking and mapping (DTAM) [54] and REMODE [58]. Instead, every pixel's intensity will be used to estimate the camera pose and reconstruct the environment. The computation for direct methods can be highly paralleled but is very sensitive to illumination changing. One state of the art approach is DTAM. In the result of the direct method, generally, every pixel of the image will be recovered to 3D. There will be millions of 3D points to describe the environment.

Chapter 3

Foreground Detection with Motion Segmentation

3.1 Background

Motion segmentation is an inevitable component for mobile robotic systems such as the case with robots performing SLAM and collision avoidance in dynamic worlds [52]. Generally, for a moving camera, the captured images will change apparently. The changes in these images are not only caused by the moving camera. Another cause of these changes is the moving objects in the scene. Often, it is a necessary component for many applications such as SLAM and automatic image stitching to segment moving objects in the image captured by a moving camera. With a moving camera, every pixel of the image moves. Except for the reason of the moving camera, the apparent pixel motion also has a relationship with the independent object motion. For a totally static scene, the moving camera is the only cause for pixel motion. And the structure of the static scene and the camera motion determine how every pixel of an image moves. But for the case of both camera motion and object independent motion, the pixel motion gets more complicated.

With a stereo sensor, the moving object can be segmented and extracted out more easily. The stereo sensor can have more constraints to find the independent object motion [2,21,28,71]. Nevertheless, the objective of detecting the independent motion in the image gets a lot more difficult while with only a single moving camera. There are many studies that are trying to achieve this objective for the images captured with monocular systems. Literature for this objective can be loosely divided into four categories [52].

The first category of methods concentrates on the estimation of the background motion model. Methods of this category will estimate a global parametric motion model for the

background. The estimation of background is usually made through 2D homography, layered motion model [59, 77].

The second category of methods is by using plane-parallax constrains [41,81] The plane-parallax constraint uses a residual displacement field which is called parallax to showing the scene structure. And this residual displacement filed is with respect to a 3D reference plane in the scene.

The third category of approaches is counting on multiple view constraints [36]. The constraint of 2 views, 3 or more views can be applied to achieve the motion segmentation. And two views constraint such as epipolar geometry constraint is one of the most used ways.

The last category of motion segmentation approaches is by using optical flow [19, 34] and pixel intensity [33].

3.2 Challenges and Objectives

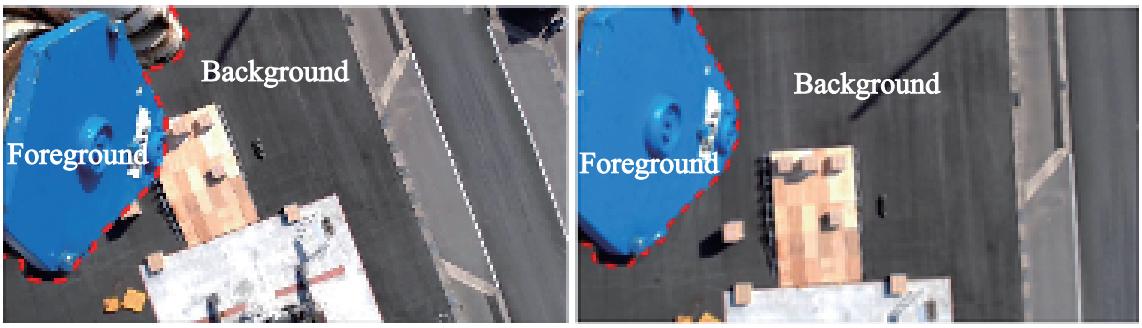
Image stitching is very common in our life. Nowadays, the smart phone also contains such functionalities to stitch the surrounding environment. But for traditional image stitching process such as in paper [13], the final panoramic image is a stitching result with several static images without any moving object on it. If the images used to make a panoramic image, the final result will contain many ghosts. In the conclusion part of paper [13], it mentions and shows the result about a very difficult stitching problem which includes many moving objects and large changes in brightness between images, many ghosts can be seen.

To produce a more clear panoramic image with automatic image stitching, the moving objects existed in the image should be detected. Then by removal of the foreground moving objects, the automatic image stitching with only the background can give us a good panoramic image.

In our case, the top-view camera mounted on the boom head will capture the image containing a part of the boom head and hook. The contained boom head and hook in the image is called foreground. If we do not remove the foreground, the stitched panoramic will contain the ghost. Figure 3.1 shows the ghost effect of stitching two images. In Figure 3.1, both Figure 3.1a and Figure 3.1b contains a blue hook. The blue hook is the foreground that needs to be detected and removed before automatic image stitching. The rest part shown in the image is named background. If we do not make the detection and removal first, after automatic image stitching, the panoramic with a ghost will appear, as shown in Figure 3.1c.

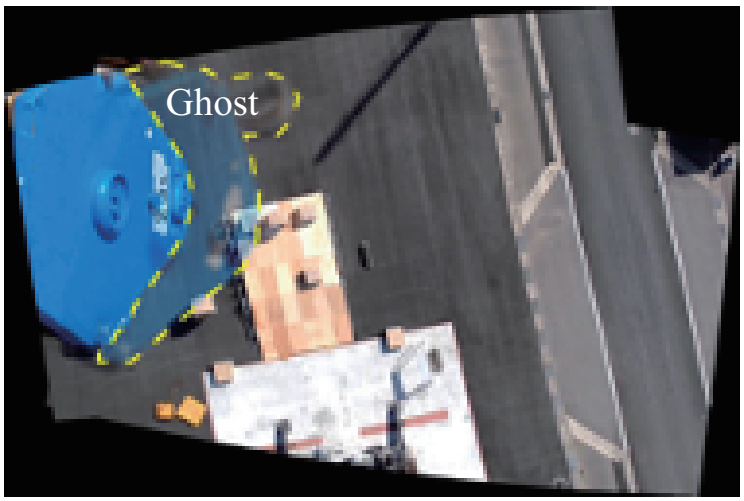
In addition, image features are a necessary component for image stitching. Sometimes there will be more image features in the foreground rather than in the background, it will lead to a totally wrong stitching result in which the foreground is aligned.

In a word, to detect and remove the foreground is an inevitable component to obtain a clear panoramic image for the crane work site. To achieve this objective, we have proposed a new approach and successfully detected the foreground of the image captured with a top-view camera.



(a) The first image

(b) The second image



(c) Stitched result

Fig. 3.1: A stitching result with ghost. (a) and (b) are two images captured with the top-view camera at different position. The foreground is a blue hook enclosed with dotted red lines. The rest is the background. Image (c) is the stitched result which contains ghost of the hook in the region enclosed by yellow lines.

3.3 Proposed Approach for Detecting Foreground Objects

The new approach proposed is considered as the third category of methods for motion segmentation as mentioned in Section 3.1. We should remove the foreground objects with a mask covering them as precisely as possible to reduce the ghost in the result. In addition, it can improve the stability of the stitching process. As our proposed new approach could be considered an improvement of Serajeh's method [69], we will first explain his method for better understanding.

3.3.1 Related work

Serajeh has proposed a method for this problem based on epipolar geometry and dense optical flow. The method is intended to extract moving objects from images captured with a hand-held moving camera. This paper considers the addressing of this problem using a structure from motion (SFM) technique.

First, with RANSAC algorithm [30], the epipolar geometry between two images is estimated to calculate the fundamental matrix. The fundamental matrix for two images will create an epipolar constraint for every pixel. Second, the dense optical flow is calculated to find the corresponding point in the second image for every pixel in the first image. Then the corresponding points in the second image that keep a significant distance to the epipolar lines are detected as moving objects in the scene. Through feature detection and matching, the epipolar plane π for two pixels in two correspondent images can be confirmed, as shown in Figure 3.2. Hence, the pixel x in the first image at camera position C is correspondent to the line $o'x'$ in the second image at camera position C'. The line $o'x'$ is called the epipolar line for the pixel x. As the depth for x is not known. The only thing that can be confirmed is that X is projected in the line ox and $o'x'$. With the estimation of optical flow, the pixel motion can be known. Because of error and moving objects, it can not just locate at the epipolar line $o'x'$. Without thinking of any error for this approach, for the background pixels, they should all lie on their correspondent epipolar lines. As shown in Figure 3.2, if there is no error exist and the pixel x is a background pixel, it should appear in its corresponding epipolar line $o'x'$, here noted as x' in this figure. But if the pixel x is a foreground pixel, because of independent motion existed, after optical estimation, it is located at $x_1^{'}$ in the second image which keeps a significant distance from the epipolar line $o'x'$. This process is applicable to a wide range of cases.

As shown in Figure 3.3, an experiment is conducted by the author. Figure 3.3a and 3.3b are two images by a moving camera at two position. The estimated optical flow is shown in Figure 3.3c with HSV color representation. With image shown in Figure 3.3a and the optical flow shown in Figure 3.3c, we can reconstruct the image shown in Figure 3.3d which is correspondent to the image shown in Figure 3.3b. Figure 3.3e shows some image features' motion estimated with the optical flow, and Figure 3.3f shows the

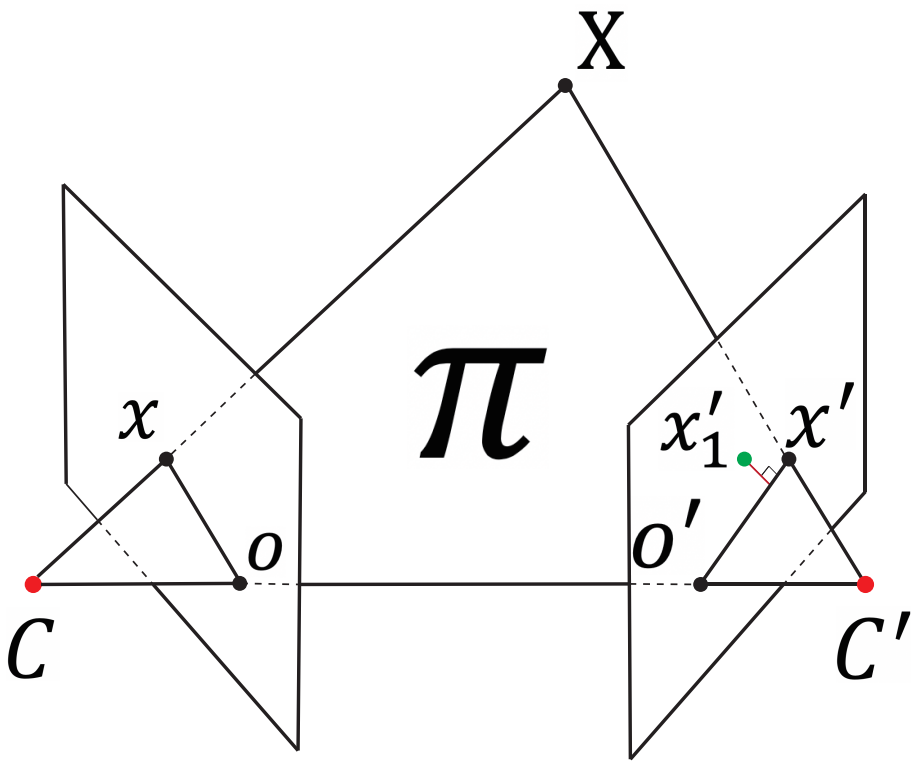


Fig. 3.2: Distance for a pixel to its correspondent epipolar line. The pixel x in the left image should lie on the epipolar line $o'x'$. Through the estimation of optical flow for the pixel x, its position in the right image is $x_1^{'}$, which keeps a significan distance from the line $o'x'$

pixels after motion on the epipolar lines. After estimation every pixel's distance to their correspondent epipolar lines, we can obtain a distance image, as shown in Figure 3.3g. Figure 3.3h shows the foreground detection result by overlapping the binarized image 3.3g on the image 3.3a. Basically, the foreground which contains a black glass case has been successfully detected out.

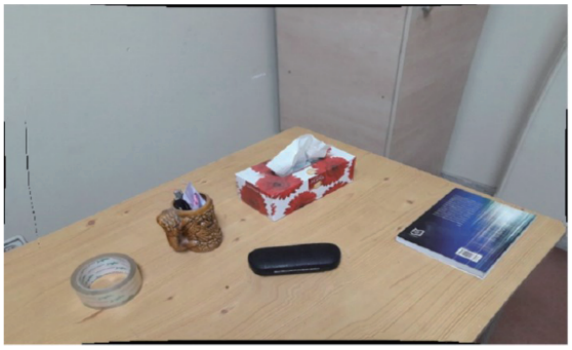
Under some conditions, however, the SFM technique will not yield satisfactory results. An example is that in which the movement of a moving object is complicated with both translational and rotational movements. In this case, some points on the moving object may lie on epipolar lines of the second image while the rest do not. One worst case is that the camera moves in one direction, and the object in the scene moves the same direction with the camera.

Then, part of the moving objects can be detected in the first image. Unfortunately, this condition always happens for the images captured by the top-view camera of the crane. From one image to another image captured with the top-view camera, foreground objects

(a) The first image

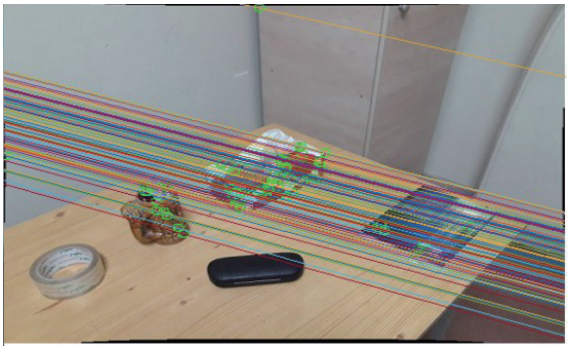
(b) The second image

(c) Optical flow

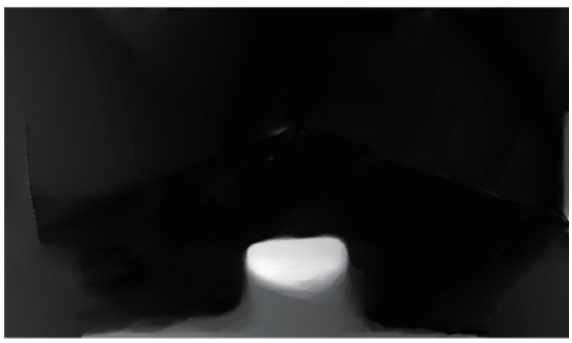


(d) Reconstructed image with optical flow

(e) Image features' motion with optical flow



(f) Features on epipolar lines



(g) Distance from pixels to epipolar lines

(h) Binarized mask of (g) on the image (a)

Fig. 3.3: The experiment result of paper [69]

will always show a complicated movement because of the crane hook's oscillation. Because of this complicated movement, only parts of the foreground object will lie on epipolar lines from the perspective of the second image, so the foreground cannot be detected in full. Figure 3.4 shows the worst case. While the camera moves from camera position C to C', the point p in the world moves to p'. As shown in this figure, the projected point x and x' respective to the point p and p' are both in the epipolar line for the point p. Through the estimation of optical flow, the movement from x to x' can be figured out. But in the distance checking from point x' to the epipolar line, it will fail. The foreground point x' for p will be considered as the background.

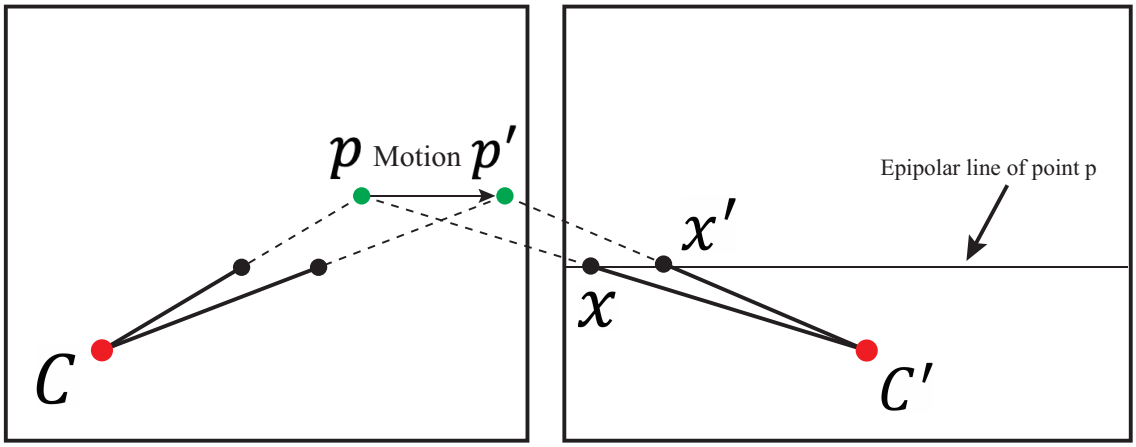


Fig. 3.4: A worst case. As the camera moves from C to C', a world point moves from p to p' by the same time. Through the estimation of optical flow, the pixel moves from x to x'. But the pixel p' is on the epipolar line. The foreground point is considered a background point with this apporoach.

3.3.2 Proposed approach

Our approach is based on two kinds of transformations of pixels from one image to another image, i.e., homography and optical flow. Compared with the approach mentioned in Section 3.3.1, the distance is redefined as the difference of two transformations.

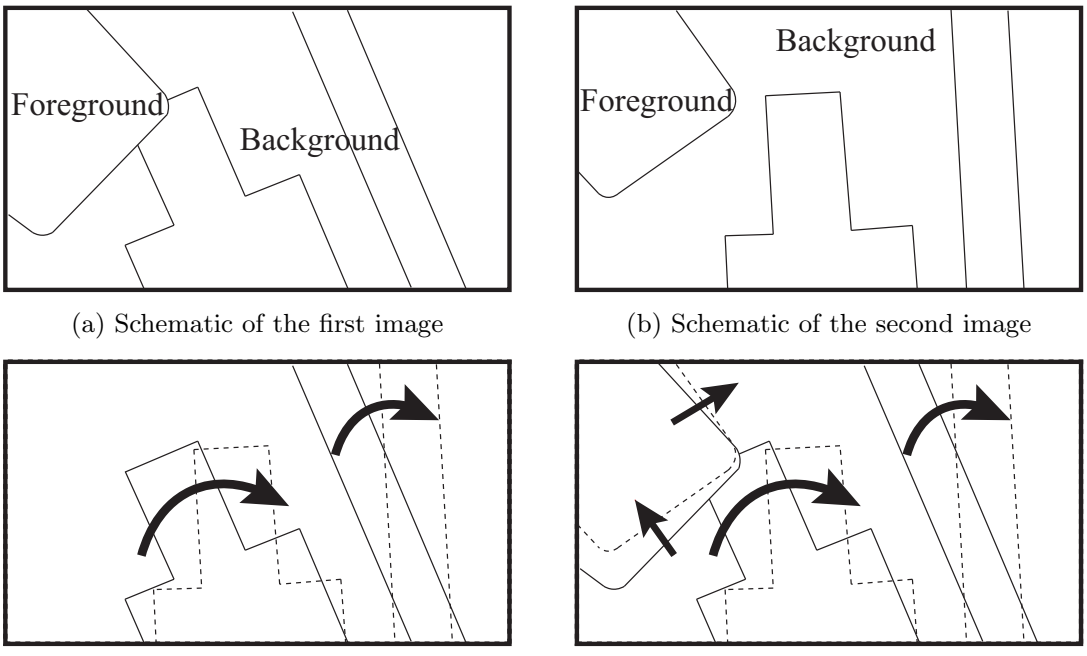
By using the epipolar constraint, it will degenerate in some cases such as shown in Figure 3.4. There are mainly two degenerated cases while using the epipolar constraint in the foreground detection which are:

- Partially detected: If the pixels in the foreground moves arbitrary, some of the pixels after motion are on their correspondent epipolar lines. However, the rest pixels keep a significant distance to their correspondent epipolar lines. It will lead to a detection result which only covers partial area of the foreground. And this is very common by using the epipolar constraint only.
- Fully fail: This is a very extreme case. The example can be imagined that a camera is doing a horizontal moving from left to the right. And the epipolar lines will be horizontal. If the foreground in the image also make a horizontal movement in the image, it can not be detected out as the foreground because the foreground pixels are always on their epipolar lines.

Compared to the approach mentioned in the above section which used the epipolar constraint, the comparison of two motions caused by the optical flow and homography is a more strict constraint. The degenerated cases which always happen in the epipolar constraint will disappear. For the two degenerated cases mentioned above, the pixels' movement can still be computed from by the comparison of the two different pixels' motion. And theoretically, only the foreground pixels' have the same motion patter with background, the foreground can not be detected out. But it is impossible because that if the pixels' have the same motion patter as the background they are the background pixels.

A schematic principle of our approach is demonstrated in Figure 3.5. Figure 3.5a and 3.5b are two schematic drawings for photographs in Figure 3.1a and 3.1b. From Figure 3.5a to 3.5b) the top view camera is moved so that the images are transformed. The

homography represents a linear transformation between two images as shown in Figure 3.5c. In most cases, since the background occupies the large portion of the images, the homography is determined by the background. The homography can be computed by finding correspondence between those two images by matching the features on them. Here as the homography is computed between two very close images taken by the top-view camera at very close positions, it can connect the two images with a relative high precision. The foreground objects follow the motion of the top-view camera and also swing back and forth. But the background objects' motion is only caused by the top-view camera. Optical flow is the pattern of apparent motion of images caused by objects' movement and camera's movement. It represents all the pixels' movement relationship between two images as shown in Figure 3.5d. And the homography can only represent pixels' movement caused by the camera's motion.



(a) Schematic of the first image

(b) Schematic of the second image

(c) Background movement (homography)

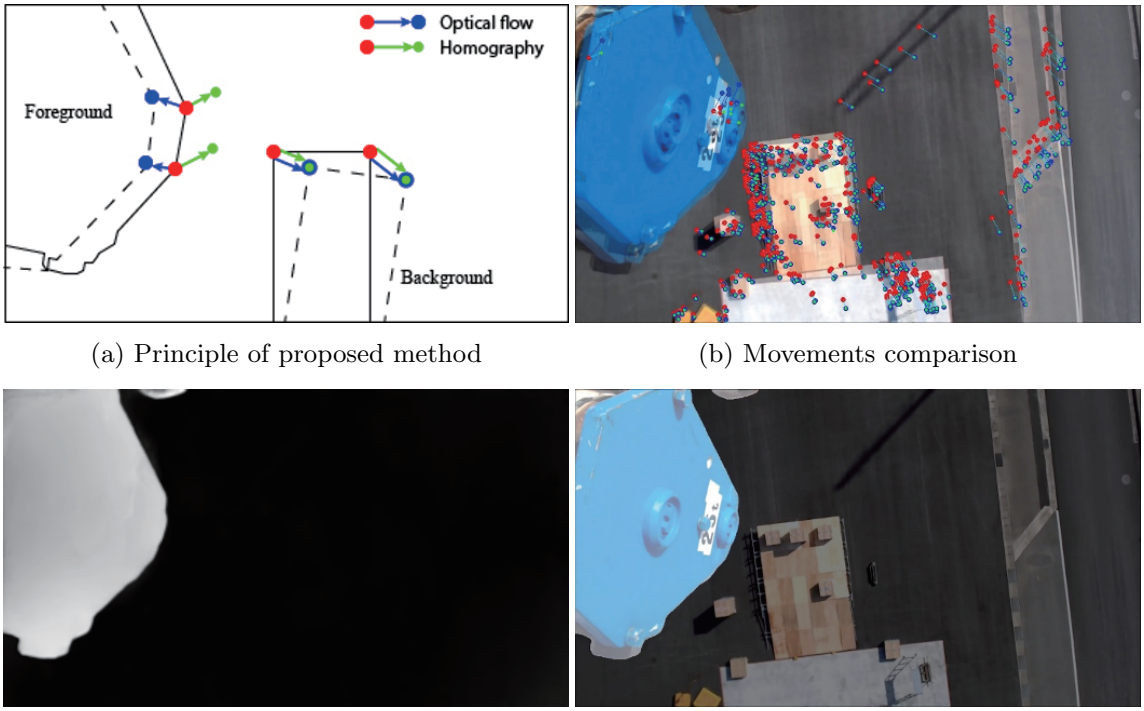
(d) Various movement (optical flow)

Fig. 3.5: Schematic priciple of our approach. (a) and (b) are two schematics corresponding to the images (a) and (b) in Figure 3.1 respectively. (c) is only showing the background, and its movement can be represented with a homography. (d) shows the optical flow of pixels: consistent to the homography on the background and inconsistent on the foreground.

Consequently, we can compute two kinds of vectors representing all the pixels' movement of the first image with its homography and optical flow to the other image. And our basic

idea to distinguish the foreground and the background is based on their relationship. For the background on the image, its two kinds of pixels' movement estimated from the homography and optical flow have a similar orientation. While for the foreground on the image, the two kinds of pixels' movements have very different orientations. To illustrate the difference between two kinds of pixels' movement of foreground and background, a simple diagram of the proposed method appears in Figure 3.6. Both optical flow and homography are representations of pixels' movement from one image to another. In Figure 3.6a, optical flow is represented as the movement from red points to blue points. And the homography is the movement from red points to green points. For background, the movements of red points to blue points and red points to green points are the same, i.e., the pixels' movement estimated from the optical flow and the homography is consistent. However, for the foreground, movements of red points to blue points and red points to green points are not the same, i.e., the pixels' movement estimated from the optical flow and the homography is not consistent. If pixels of the foreground have the same movement with the background, the detection of these pixels will fail. This problem is very significant in structure from motion method as epipolar constraint is not strong enough for the degenerate cases. Our method using homography and optical flow makes a stronger constraint and only keeps low possibility in meeting the degenerate cases.

Figure 3.6b shows a test on images captured with the top-view camera of the proposed method. As can be seen, the foreground is a blue hook. The red points are SIFT features [49]. The homography estimated from the matched features of the two images is represented as the movement from red points to green points. The dense optical flow computed with flownet2 [39] returns the pixel movements from the red points to blue points. Just as mentioned above, in the background, these two movements match. On the other hand, in the foreground, the homography and optical flow of foreground objects are not consistent. By examing the distance for every pixel, we can get the result showing in Figure 3.6c. The brightness of a pixel means the distance of two different motion for the pixel. As the pixel getting brighter, the distance of two different motion get further. Figure 3.6d shows overlaying the binarized mask of Figure 3.6c on the image. It can be seen that the foreground of the image is successfully detected and just covering the big blue hook and a very small portion of the little hook.



(a) Principle of proposed method

(b) Movements comparison

(c) Distance from red points to blue points

(d) Comparison of binarized mask and image

Fig. 3.6: (a) A principle diagram of the proposed method to detect foreground by examine of the distance from the blue point to the green point. (b) A test result of the method on two images captured by the top-view camera showing sparse optical flow and homography. (c) Detection of foreground by comparing the difference between the dense optical flow and homography. (d) Comparison of binarized mask with the image.

3.4 Foreground Detection Algorithm

In this section we describe more details of the above-mentioned approach. We propose this method mainly for our detection of the foreground boom and hook existed in the image captured by the top-view camera. To stitch a panoramic for the crane work site, generally, we will make a pre-scan with the top-view camera and record a video V_{pre}. V_{pre} is a sequence of images from which are captured by rotating the crane around the worksite. Several keyframes for V_{pre} are selected automatically by considering the overlap ratio and used for stitching. The selected keyframes are noted $k_0, k_1, \cdots, k_n$.

The first step is to compute a mask for each key frame. For each of the other key frames k_i, a foreground mask should be detected by computing the homography vectors $v_{homo}(p)$ and optical flow vectors $v_{opt}(p)$ [39] for all the pixels p of k_i. However, the $v_{homo}(p)$ can not represent pixels movement very precise in cases such as the camera is auto-focusing and so on. To make the foreground detection more robust, for a keyframe k_i, a set of support frames was chosen for computing the homography and optical flow. The support frames noted as $s_1, s_2, \cdots, s_M$ are chosen from a set of M frames near k_i in V_{pre} one by one to generate an optimal mask. From k_i to s_j, for each of all the pixels p, the homography vector $v_{homo}(p))$ is computed by the homography H which is a mapping from the key frame k_i to s_j, that is, all the pixels p of k_i are mapped to their corresponding locations $H(p)$ on s_j. Thus $v_{homo}(p)$ can be computed by $v_{homo}(p) = H(p) - p$. For $v_{opt}(p)$, it can be directly estimated with FlowNet2 [39]. Then the distance $D = \parallel v_{homo}(p) - v_{opt}(p) \parallel$ for all the pixels p of k_i can be computed. The foreground pixels p_f and the background pixels p_b have very different values of D. Thus, by filtering with a threshold, the foreground pixels p_f can be picked out. By setting p_f to be white and the other pixels black, we get a binary image m_i as the foreground mask. Therefore, masks $m_1, m_2, \cdots, m_M$ between the support frames $s_0, s_1, \cdots, s_M$ and the keyframe k_i are obtained.

However, some of these masks are not valid due to possible distortion or other reasons for the support frame images. Then a process of selecting a good mask from these obtained mask sets should be done. First, we filter out such masks whose masked area is less than t_min because the foreground should occupy certain areas of the image as the hook is always in the image. Then we find the most appropriate mask from the filtered masks.

As the foreground object such as a hook stays in the same position and thus occupies a similar region of the support images. Therefore, the masks ideally share the same pattern. So we compute the difference of pairs of the masks m_a and m_b and select the pair with the minimum difference. The difference is a simple sum of the difference between the two binary images m_a and m_b. Then we choose the target mask m_i from one of the minimum pairs having the larger masked area.

Figure 3.7 show the pipeline of foreground detection for the ith keyframe. After the selection of the keyframe and its support frames, the first step is to compute the optical flow and homography from the keyframe to the support frames respectively. After that, we will compute the distance between the motion estimated with optical flow and homography. And a lot of masks will be obtained after the second step of distance computation. In the third step of the detection pipeline, pair-wise comparison of the masks for the keyframe and support frames will be conducted. The most-like two masks will be selected out. In the last step, an area size checking will be made, and the target mask finally can be obtained.

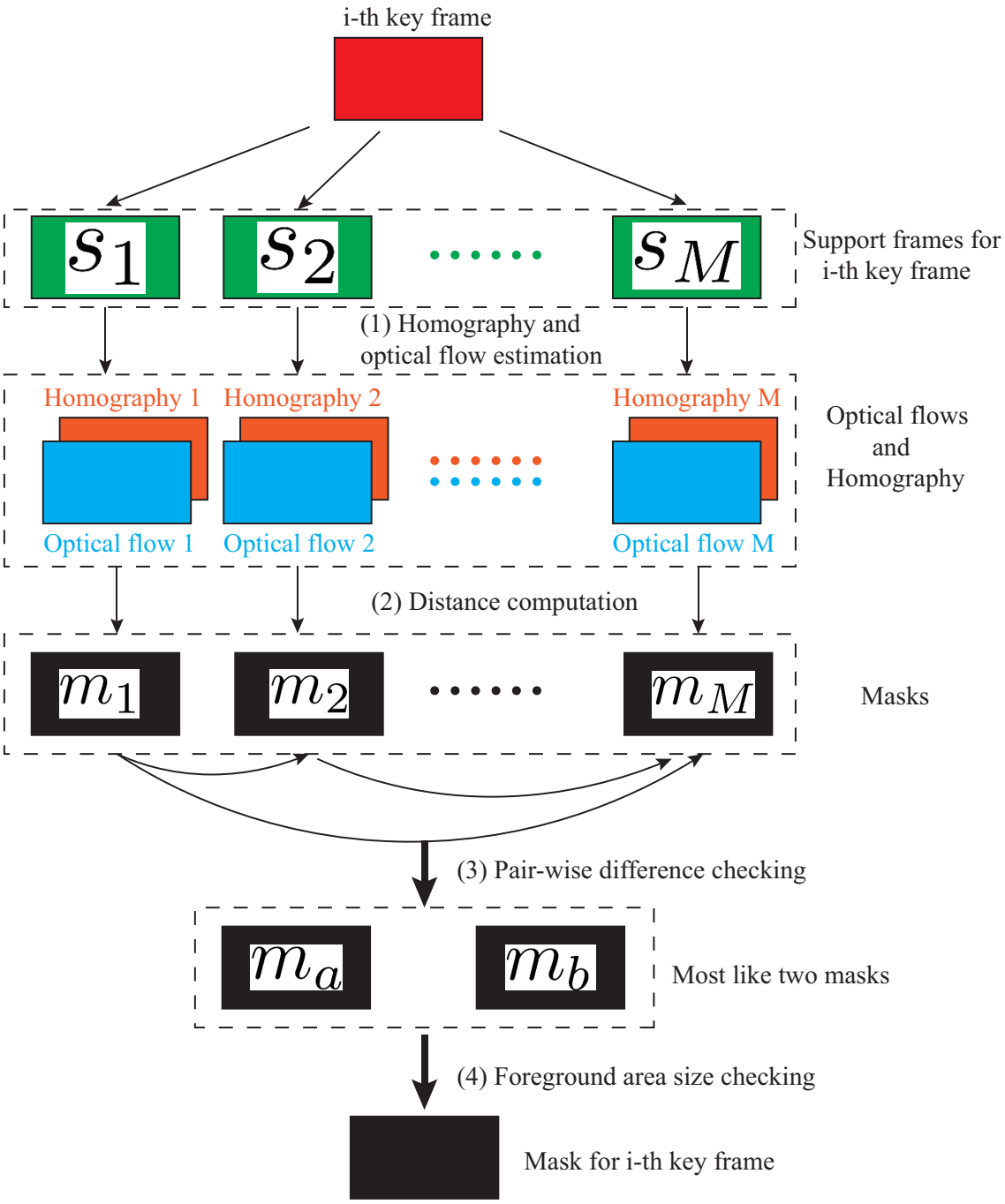


Fig. 3.7: The pipeline of foreground detection for a key frame. After selection of the key frame and its close support frames, with four steps computation, the mask for the key frame can be obtained.

3.5 Experiment and Result

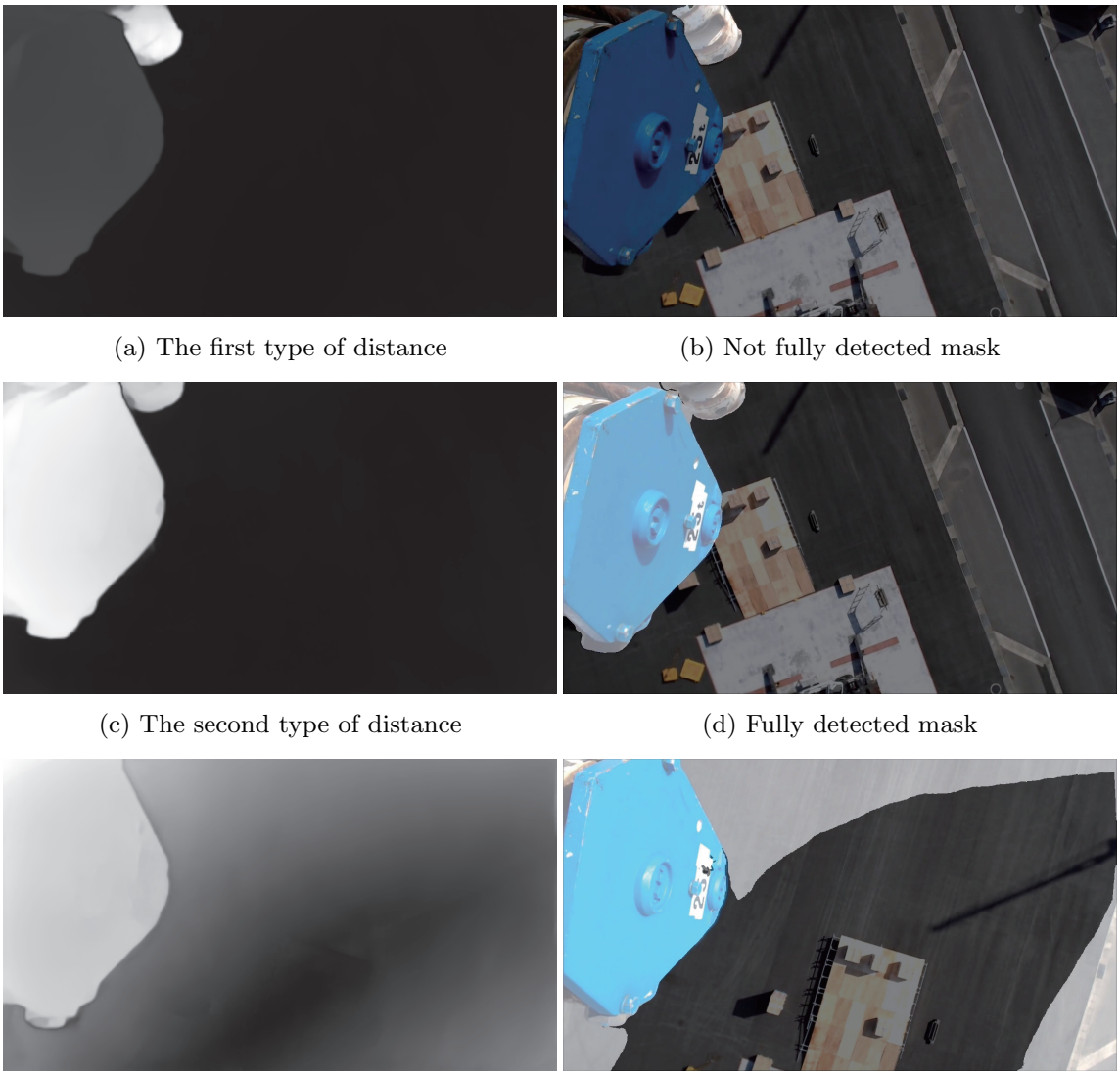
For the proposed approach and our image stitching objectives, two experiments are done with two different pre-shooting videos. The boom length is about 28 meters and the pitching angle is about 50 degrees. In the pre-shooting process, the top-view camera is keeping a height of about 21 meters above the ground. For the first experiment, all the frames have been tested with 30 support frames for the foreground detection. For the second experiment, we have tried to use 40 support frames for each keyframe. Totally 82 keyframes are selected from equidistance of the sequence images of the video.

The original image size of the top-view camera is $1920 \times 1280*$. However, the resizing of the image affected the two different motion the same way. And large image size need a more powerful GPU for the optical flow estimation. The images are all resized to $480*320$. After obtaining the mask, it can be resized up to the original size for the removal of the foreground.

Figure 3.8 shows some results of the first experiment. The left column of Figure 3.8 is the distance of the two different motion estimated with dense optical flow and homography. The right column shows that the binarized mask with otsu thresholding is overlayed on the image. The quality of the detected foreground can be checked. In the first experiment result, there are three types of masks. The first type is shown in Figure 3.8a and 3.8b. For the foreground objects, when they move with different velocity, the otsu thresholding [55] can not find the correct foreground because there will be multiple peaks of the pixel histogram. So only the small hook has been detected out. But as shown in Figure 3.8a, the big blue hook also has obvious motion detected. With a better thresholding method, the foreground can be detected out. The second type of result is shown in Figure 3.8c and 3.8d. It shows the result that the mask is just covering the foreground object. The third type of result is shown in Figure 3.8e and 3.8f. A lot of reasons can affect the result, such as a small motion and inaccurate optical flow.

The result of the second experiment is shown in Figure 3.9. There are 82 masks detected out successfully for 82 keyframes.

A statistics table for all the foreground detection result has been made, as shown in Table 3.1. As there is no ground truth for the foreground detection, the results which are



(a) The first type of distance (b) Not fully detected mask

(c) The second type of distance (d) Fully detected mask

(e) The third type of distance (f) Over detected mask

Fig. 3.8: Three types of result for the first experiment. The first row shows the not fully detected mask. The second row shows the fully detected mask. And the third row shows the over detected mask.

divided into not fully detected, fully detected and over detected are made manually. The three cases of detection are defined as follows:

- Not fully detected: The detected region is only a small portion of the foreground. And there is no obvious background which is detected as the foreground.
- Fully detected: The foreground can closely covering the foreground objects even there are a very small portion covering the background.
- Over detected: A large area of the background is detected as the foreground. No

Fig. 3.9: An example of the second experiment. There are totally 82 mask detections for the key frames.

matter the foreground objects is fully or partially detected as the foreground, if a very significant of background is detected as the foreground, the detection result is an over detection result.

Our approach keeps a very high precision on detecting the foreground.The detection result of manually classification for the three cases of detection results is shown in Table 3.1.

Table 3.1: A statistic analysis of foreground detection experiment

	Not fully detected	Fully detected	Over detected	Totally	Successful ratio
First experiment	8	634	46	688	92.15%
Second experiment	0	82	0	82	100%

The foreground detection of images is for the generation of a clear 2D workspace map in chapter 4. The rate of full detection of foreground determines whether this approach can be applied. For a fully automatic system to generate a clear 2D workspace map, it requires every selected keyframe should have a full detection for the foreground. Assuming there are 20 keyframes selected and need detection for the foreground, the successful detection rate for full detection can have an influence as:

- A low successful rate: if the rate is less than 90 percent, there will be at least 2 images' foreground which is not correctly detected. The influence of the two images will finally be seen in the final panoramic image. And it can be very obvious.
- A high successful rate: under the high success rate, the wrong detection is less than 2 images. The final panoramic is less affected by the foreground objects. Even there are still some ghosts in the final panoramic image. But it still can provide a good vision of the crane's working environment. On the other hand, a replacement selection can be provided to the operator to change the wrong detected ones. There will be a higher second or third chance to replace the wrong ones by their close images as the keyframes. With such a semi-automatic way, the clear 2D workspace map can still be obtained.
- A perfect successful rate: If the success rate can reach 100 percent, the 2D workspace map can be fully automatic and clear.

In the three cases, the highly successful rage and perfect success rate can ensure the approach is applicable. In our detection experiments for the foreground, the successful detection rate (full detection) is about 92 percent in the first experiment. It meets our requirement as at least the high successful detection rate.

Generally for 1 keyframe with 40 support frames, a mask can be generated in several seconds. As the size is resized down to 480×320, the FlowNet2 can estimate 10 to 20 dense optical flows from a keyframe to other support frames. Also for the homography estimation, such a size image won't take time. For one keyframe in our test, it takes about 10 15 seconds to have its mask.

3.6 Conclusion

In this chapter, a new approach for the detection of the foreground is proposed. In order to have a clear workspace with image stitching, the foreground detection and removal for the image is necessary. To achieve the objective, the foreground is detected out on a basic idea of comparing the different motion patterns of the foreground and background of the image. From one image to another image that is very close to each other, pixels are in motion. And the motion for pixels can be estimated through dense optical flow and homography. With FlowNet2 [39] and image view geometry [36], the two different motion of pixels can be estimated. By comparing the difference of two motions, the foreground can be detected out. It is because the motions of foreground pixels are in a big difference.

Considering the error of estimation, a more robust pipeline for selecting the best mask from dozens of masks is proposed. For a keyframe, a lot of support frames will be selected. Hence, a lot of masks will be generated in the beginning. By pair-wise comparison and similarity checking, the best mask for the keyframe can be selected out.

Then two experiments are conducted to test our approach. The statistic Table 3.1 shows that our approach has a good precision on the foreground detection. In the first experiment, about 92% of the keyframes' foreground are detected. For the second experiment, all the 82 keyframes' foregrounds are detected out. The successful detection rate ensures its applicability in the generation of a clear 2D workspace map.

There are limitations for this approach to detect the foreground. For some objects on the ground which moves the same as the background in a short period can not be detected out. Such as for a shadow of the crane boom, even the boom rotates a very obvious angle, the shadow moves little. And the detection to detect the shadow as the foreground is difficult.

Chapter 4

2D Workspace Map Generation and Application

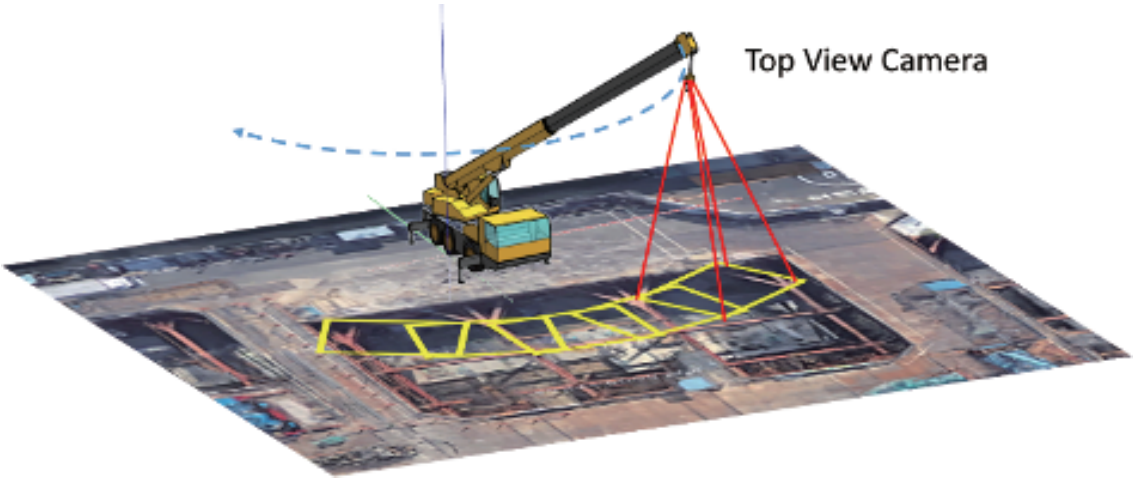
4.1 Introduction

Cranes often work in complicated construction sites. Blind spots, limited information and high mental workload are problems encountered by crane operators. The operators of cranes have to face the constantly changing workspace. The chief dangers to crane operators are a congested working environment, neglect of hidden dangers and a lack of information for making decisions especially for the operators with little working experience and after a long time work. A top-view camera mounted on the boom head offers a valuable perspective on the workspace that can help eliminate blind spots and provide the basis for assisting the operation. But a simply top-view with the-top view camera which is mounted on mobile crane's boom head or tower crane's trolley can not help the operator senses the workspace very well. Thus, providing accurate and overall spatial information about the environment to the crane operators is highly useful and thus required.

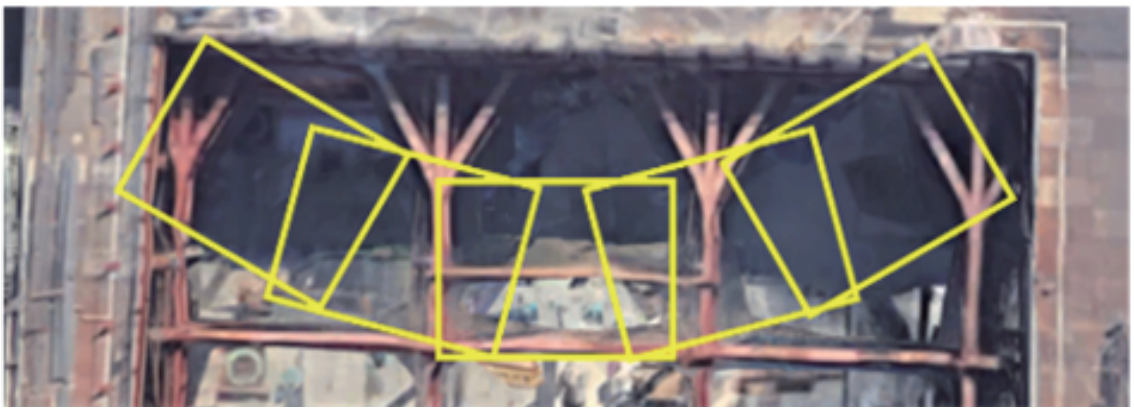
Precise scanning of the environment and data processing tends to be time-consuming and cost a lot. One lab, for example, used data from crane sensors to roughly examine the working environment for path planning [62].

In recent years, many technologies associated with computer vision have been applied in the construction sites to generate environmental information such as Pix4d and SiteTrac.io which are commercial software. These softwares provide solutions for tower cranes and drones but require cloud computation service. Applications involved with work supporting [37] and teleoperation [42,73] with video camera have been proposed and show the obvious result in improving operations.

In this chapter, we rather aim at a light, closed system to be utilized on-site only using top-view camera images. The crane operator's limited visibility and insufficient information about the workspace are the primary concerns. As shown in Figure 4.1a, a top-view camera is mounted on the boom head that moves over the workspace along with the boom. Bird's eye view images can be captured using the top-view camera. With several images captured from the top-view camera, a wide range of the workspace can be represented by stitching and rendering these images, as shown in Figure 4.1b. The stitched top-view camera image can provide a rich range of information to the operator. Herein, the stitched wide-range image is referred to as the workspace map.



(a) A workspace survey with top-view camera



(b) Workspace map generation with automatic image stitching

Fig. 4.1: Workspace map generation. (a) A top-view camera suspended from the boom head continuously capturing images shown in yellow rectangles of the workspace. (b) Automatic image stitching process to stitch these images to produce a wide-range image of the workspace.

To generate the visual map of the crane workspace using the top-view camera, the main technology applied is automatic image stitching. To achieve our objective of having a good and applicable stitched workspace map. Two problems should be figured out first.

The first problem is the projective ambiguity for image stitching. It is almost everywhere for image stitching, but also very easy to neglect. The projective ambiguity for image stitching is caused by different depth and camera translation. As shown in Figure 4.2, the projective ambiguity can be seen obviously. As shown in Figure 4.2, three world points noted as P_1, P_2 and P_3. They are all projected to the same point at the camera position C_1, and finally the point P_1 is represented in the left image as x_1. However, once the camera moves from position C_1 to C_2, the projected result is changed. The three points P_1, P_2 and P_3 are projected to x_2, x_3 and x_4 respectively in the right image. While in the stitching process, only one point of x_2, x_3 and x_4 in the right image can be aligned to x_1 in the left image. By reducing the height variance and camera translation, this phenomenon can be restrained. Also by taking photos at the further positions can also reduce the effect of projective ambiguity while in image stitching. For most cases such as we stitch a panoramic image with our mobile phone, we are always standing far away from the scene been captured and almost only rotate the camera. In such a case, there won't be much projective ambiguity and we can obtain a good panoramic result. For our case, there is a big translation for image stitching with a top-view camera. We choose to lift up the camera at a very high position and only for the application of medium height crane work site.

The second problem is the foreground moving object. In the survey by rotating the top-view camera around the worksite, the image captured always contains a foreground with the boom head and hook. As mentioned in Chapter 3, a directly stitching with the images captured with a top-view camera can only generate unclear workspace maps containing ghosts, as shown in Figure 3.1. With a foreground detection on the images which are used for stitching by the approach proposed in Chapter 3.3, a clear workspace map can be obtained. In addition, the detection of the foreground can also help us eliminate the bad influence and help us find the correct geometry between images, and it is also a useful way to remove the outliers for the commercial software to do the 3D reconstruction through images.

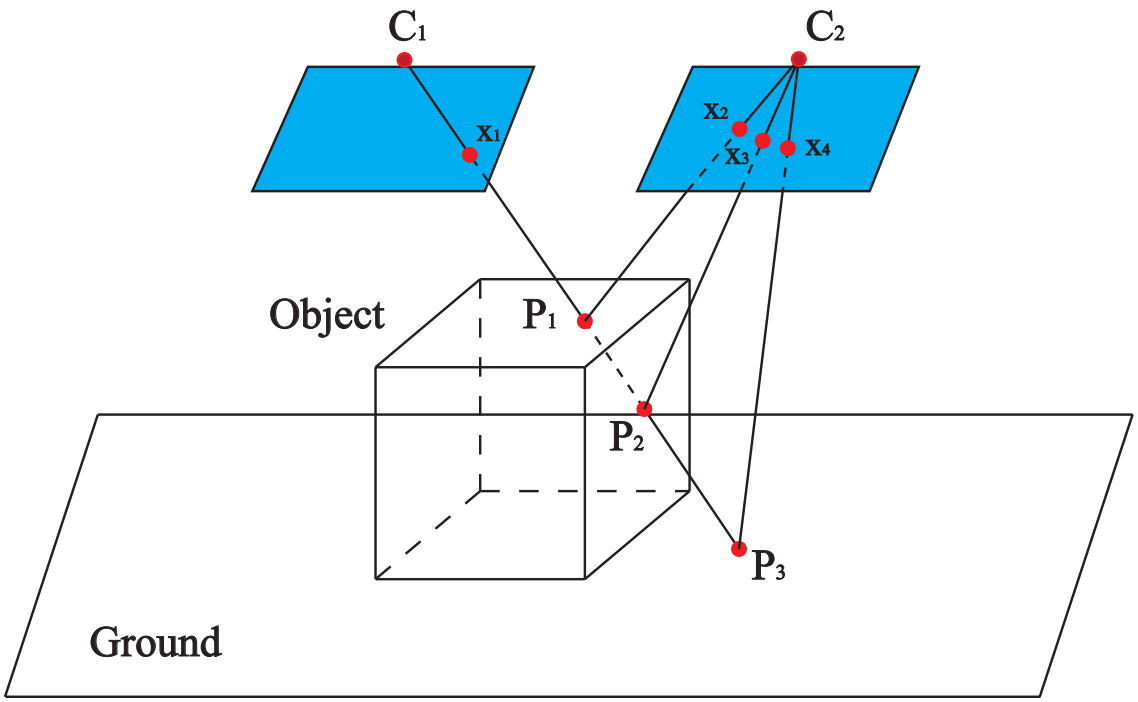


Fig. 4.2: Projective ambiguity

Once an environment map is generated, a variety of assistance applications of the workspace map can be considered. An optimal path to transfer a load can be displayed on the workspace map to aid the operator. Information, such as the position of the boom head and a 2D projection of the lifting path, can also be included in the workspace map, along with other representations that researchers may devise in the future.

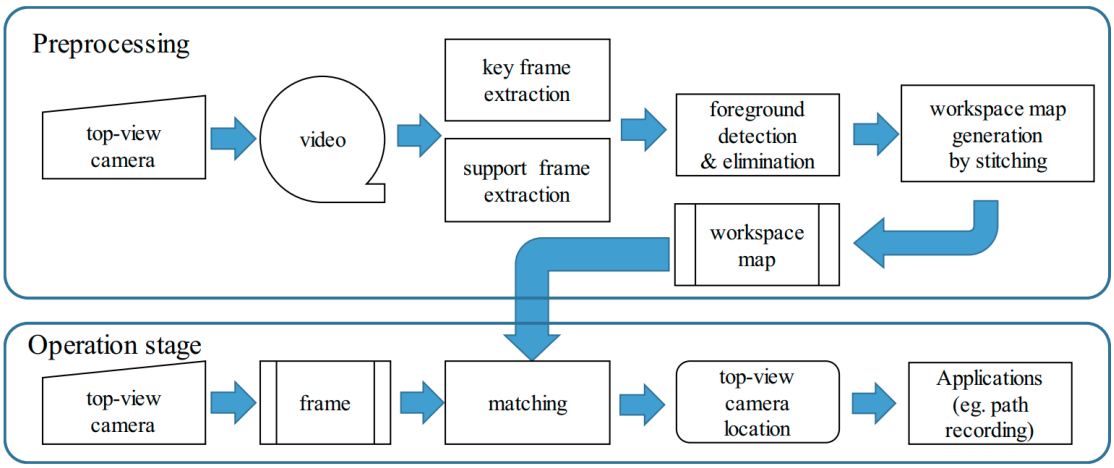


Fig. 4.3: The flow chart of our proposed method. The whole process consists of preprocessing and operation stage. The preprocessing can generate a clear workspace map. In operation stage, with the image frame captured by top-view camera, the workspace map can show and record boom head's position.

4.2 System Overview

The flow chart of our proposed method is shown in Figure 4.3. There are two stages for the system. The first stage is a preprocessing stage aiming at generating a high-quality visual map (workspace map) for later application. The second stage is an operation stage for assisting operation by showing the boom head (hook) location on the workspace map with 2D homography mapping.

To reduce the projective ambiguity and generate the workspace map with high quality, in the process of record a video with a top-view camera, the pre-shooting process requires the following conditions, which are diagrammed in 4.4.

- The top-view camera is located at the top of the boom at a sufficient height to cover a wide area of the workspace.
- The optical axis of the top-view camera should point vertically to the ground.
- While taking the images, the top-view camera rotates with the boom's rotation only. If an extension of the boom is necessary, the height of the top-view camera should be kept constant to make images captured having a close scale.
- Images captured with the top-view camera include background and foreground objects. The background is the ground and objects resting on it. The foreground

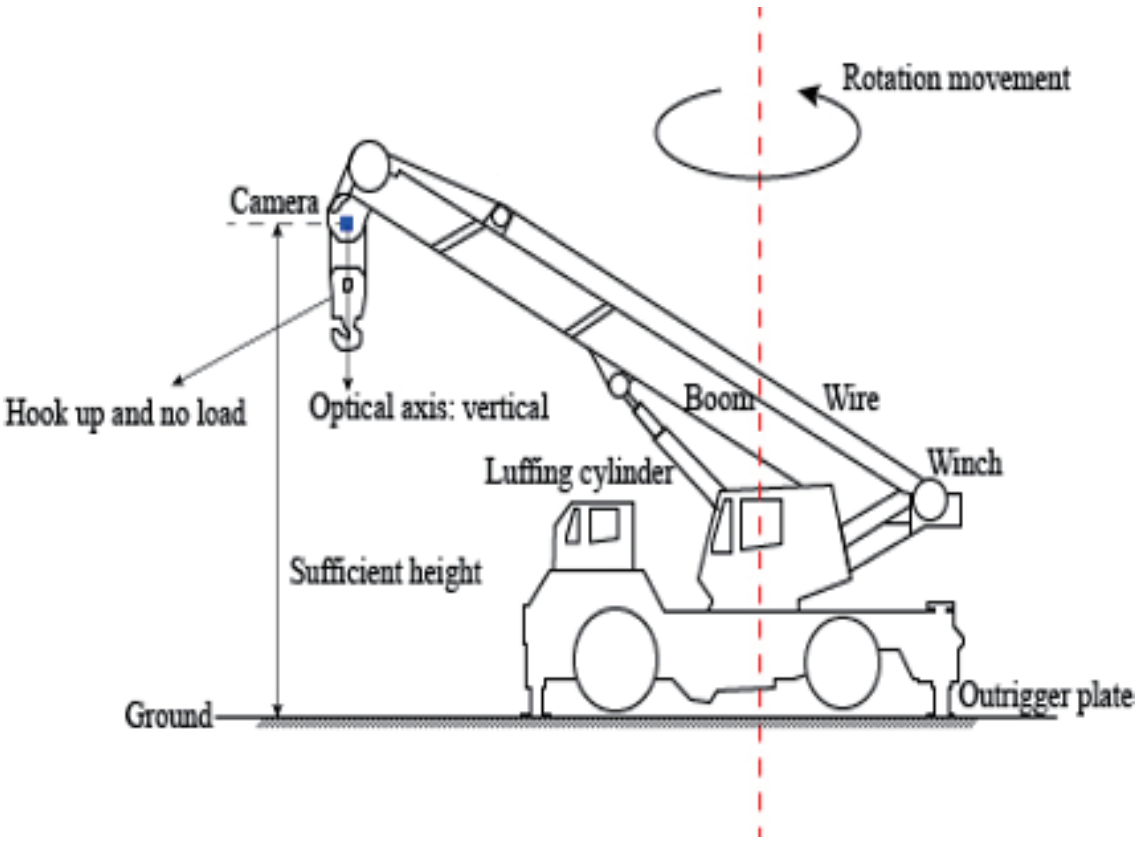


Fig. 4.4: Conditions of pre-shooting of the background images.

includes objects that are hung to the boom and move along with it, such as a hook, a wire, and a swinging load. During the pre-shooting process, the foreground should be removed as much as possible as they must not show up in the workspace map. For this reason, during the pre-shooting, winding up the cable and detaching the load are recommended. But it is impossible to exclude them completely. We need some means to detect the foreground. It is one of the major issues discussed in this paper.

Under these pre-shooting conditions, the images captured with the top-view camera can be composed using automatic image stitching [13]. However, even under the conditions mentioned above, these images still often contain such foreground objects as a boom head. Simply stitching these images will cause ghosts of the foreground objects to appear in the final stitched workspace map. As shown in Figure 3.1, the typical image captured with a top-view camera and a direct image stitching result are presented.

As shown in Figure 4.3, in the preprocessing stage, the first step is the generation

of video through a top-view camera. It should be under the four conditions mentioned above. And after that, many keyframes from the video will be chosen for generating the workspace map. Alongside, near every keyframe, a lot of support frames will be selected out. By applying the approach shown in Figure 3.7, for every keyframe, there will be a mask which is covering the foreground hook. With the removal of the foreground, a clear workspace map can be generated with automatic image stitching [13].

After the preprocessing stage, an operation stage is provided for assisting the operation. In the operation stage, for a new image frame captured with the top-view camera, a matching process will be made to find its location on the generated workspace map. Alongside this, a continuous stream of images can form a moving path on the workspace map.

The main originality of our approach of generating a 2D workspace map for assisting crane operation are:

- A new approach for the detection of foreground by comparing different motions of pixels.
- A 2D workspace map generation pipeline which include keyframes selection, support frames selection, foreground detection and image stitching is proposed for generating a clear and good quality result.
- An application case of locate the boom head (object) in the 2D workspace map for assisting the crane operation.

4.3 Homography and Image Stitching

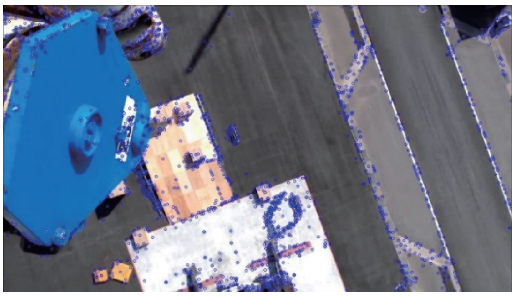
In this section, the process of generating the workspace map with automatic image stitching will be explained in detail. Two core components of generating a clear visual workspace map are the foreground detection and automatic stitching.

The image-stitching problem is well understood. Image alignment and stitching through feature-based matching to estimate a homography are the most important steps. The image stitching can obtain better quality with proper image warping method [18] and blending method [15].

Figure 4.5 shows the image stitching process with four main steps. The geometry relationship of two images for image stitching can be totally described with the 2D homography. How to estimate the homography from one image to the other image robustly and precisely is the key for image stitching.

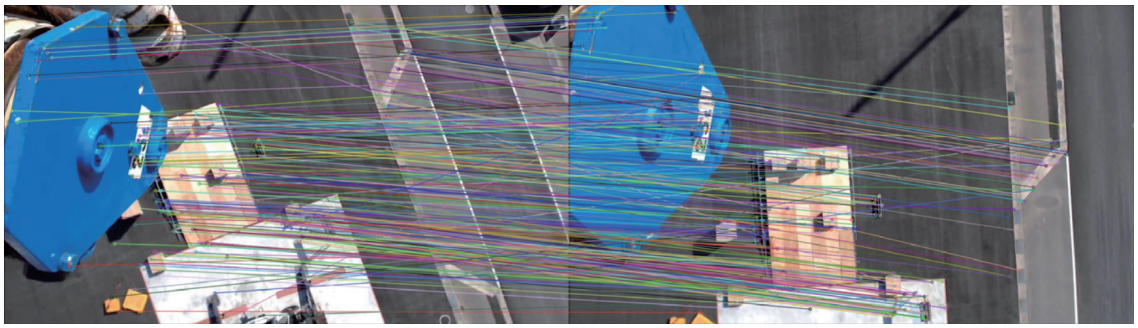
To estimate the homography from one image to another image, the first step is to find robust features. By using robust image features, the corresponding points in the two images can be found out more easily. Although optical flows can also be used for estimation of homography, it is more convenient to use image feature points rather than all pixels. There are many invented features can be chosen, such as SIFT [49], SURF [7], ORB [65] and KAZE features [3]. Here, SIFT features are chosen because of their good scale invariance, rotation invariance and illumination invariance. As shown in Figure 4.5a and 4.5b, SIFT features are extracted from each image.

After the detection of SIFT features from two images, a matching process is conducted to find bunches of corresponding point pairs. For one SIFT feature, it consist of the coordinate location with an orientation (x, y, θ) and a 128 dimensional vector $\boldsymbol{v} = (v_1, v_2, \cdots, v_{128})$. The 128-dimensional vector is called descriptor which describes the relation region of the point (x, y, θ). By comparing the length of the vector of two images, bunches of features can be matched. For example, if n features and m features are found respectively in two images, the i-th feature in the first image with feature location $\boldsymbol{P}_i(x_i, y_i, \theta_i)$ and $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \cdots, v_{i128})$, it will be compared pair-wisely to all the features $\boldsymbol{v}_j = (v_{j1}, v_{j2}, \cdots, v_{j128})$ for $j = 1, 2, \cdots, n$ in the second image by comparing the vector length. It is actually a searching process by comparing the nearest length for

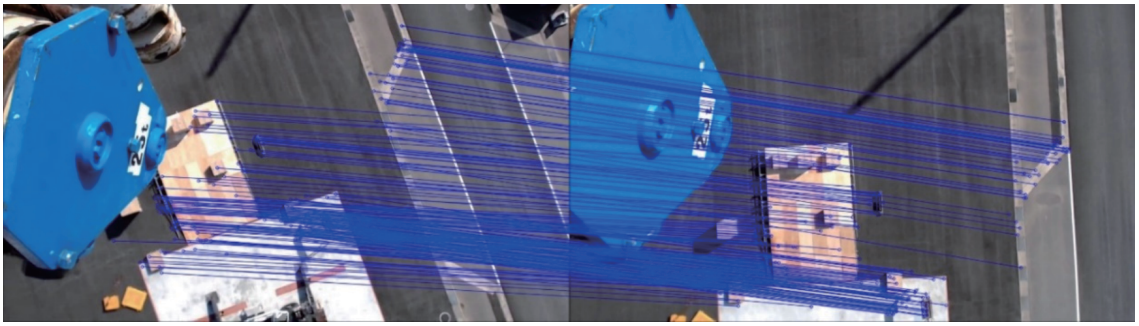


(a) SIFT features in the first image

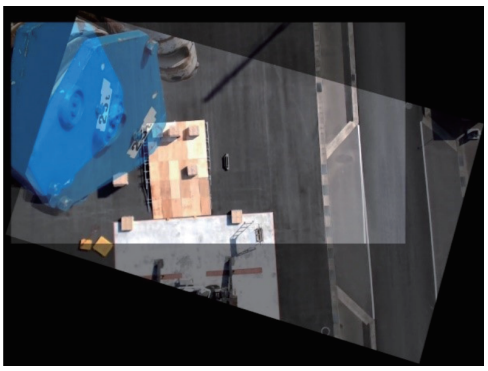
(b) SIFT features in the second image



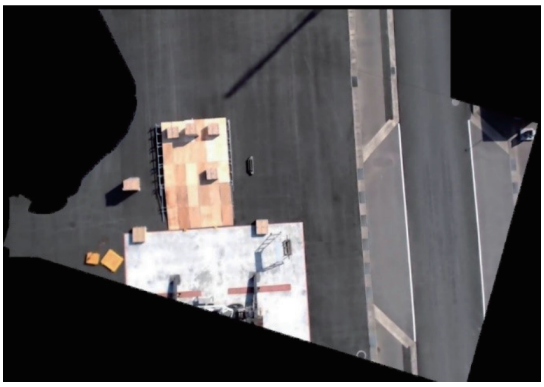
(c) Matched correspondence without RANSAC



(d) Matched features of RANSAC inliers



(e) Aligned result according to homography



(f) Stitching result

Fig. 4.5: Image stitching process. (a)(b) SIFT features are detected from both images.(c) the matched correspondence of SIFT descriptors.(d) RANSAC inliers are extracted from the matched SIFT features. Homography is estimated with these RANSAC inliers. (e)With the applying of the homography estimated in (c), the first image is warped and aligned with the second image. (f) The result of stitching by applying multiband blending on (e).

$\boldsymbol{v}_j$ from all the features in the second image. To improve the result of matching, generally, KNN [27] matching will be used. For the feature point $\boldsymbol{v}_i$ in the first image, several feature points in the second image will be chosen by comparing distance. They are the closest ones for $\boldsymbol{v}_i$. The selected several feature points should meet some condition, then the closest one of the several selected features is considered a correctly matched feature. Generally, the two closest features in the second image will be chosen. For the feature $\boldsymbol{v}_i$ in the first image, $\mathbf{v}_a$ and $\boldsymbol{v}_b$ will be selected from the second image will be chosen. If the distance of $\boldsymbol{v}_a$ is smaller than the 80% of the distance of $\boldsymbol{v}_b$. $\boldsymbol{P}_a$ is the matched feature for $\boldsymbol{P}_i$. As shown in Figure 4.5c, the matched features can be checked. Most of the features are correctly matched. But still, a few features are in the wrong matching relationship.

With most of the correctly matched features and a few wrong matched features, the third step is to estimate the homography robustly. The matched features are noted as $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_i^{'}$. To estimate the homography from these corresponding point pairs robustly and precisely, the RANSAC (random sample consensus) [30] algorithm is implemented. which can robustly identify inliers of the matched features. The homography is represented as a 3 by 3 matrix with 8 degrees of freedom and 1 scale factor. To identify the homography between images, we can select 4 feature correspondences and calculate the homography $\boldsymbol{H}$ between them using the direct linear transformation method [36]. In the matched feature correspondences, the probability for a given matched feature which is correct is p_i. Then after n times of iteration, the probability to find the correct $\boldsymbol{H}$ is $p_c = 1 - (1 - (p_i)^4)^n$. After many times of iteration, for example 500 times with $p_i = 0.5$, p_c is almost for 1. For every H computed, the features in the first image are transformed with computed H. Then a comparison between the transformed features and the features on the second image will be compared. If the transformed feature $\boldsymbol{H}\boldsymbol{x}_i$ is in a tolerance δ pixels away from $\boldsymbol{x}_i^{'}$, this matching result is considered as an inlier of the matching result. The correct H can ensure that there is a maximum overlapping of the transformed features on the features of the second image. Figure 4.5d shows the inliers only of the matched result. Compared with the result shown in Figure 4.5c, the corresponding features are now all matched correctly. It means that the homography matrix $\boldsymbol{H}$ has been successfully estimated.

After the estimation of homography $\boldsymbol{H}$ from the matched features between the two images, the homography $\boldsymbol{H}$ is used to warp the first image with projective geometry.

Coordinates in the image to be warped are represented as $\boldsymbol{P}_i(x_i, y_i, 1)$. The corresponding point in the second image is $\boldsymbol{P}_i^{'}(x_i^{'}, y_i^{'}, 1)$. $\boldsymbol{P}_i^{'}$ can be easily obtained with equation $w\boldsymbol{P}_i^{'} = \boldsymbol{H}\boldsymbol{P}_i^T$, where $w_i$is a scale parameter and $\boldsymbol{H}$ is a 3×3 matrix representing the homography. Figure 4.5e showsthe result after aligning the first image to the second image.

Once the images are aligned, they simply need to be blended together. Multiband blending is used for this process because of its good performance on many examples of image stitching [13]. Figure 4.5f shows the blended result of Figure 3.1c using masks detected with the method proposed in Chapter 3.

4.4 Workspace Map Generation

4.4.1 Workspace Map Generation Process

In order to generate the workspace map for the pre-shooting videos, we stitch the keyframe images. The keyframe images are determined by the overlapping ratio. The first frame in the pre-shooting videos is selected as the first keyframe image k_1. Then we select keyframe k_i by considering its overlapping ratio with the previous keyframe image $k_i - 1$ for $i > 1$ with a pre-warping. If the overlapping ratio is less than our threshold value 0.7, it will be selected as the keyframe image k_i.

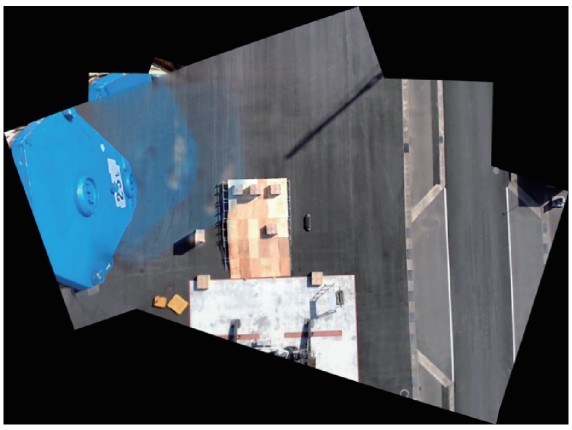
Then a base keyframe k_{base} should be chosen from the selected keyframe images and the rest of the keyframe images should be warped to the base keyframe. However, the workspace may be too large for k_{base} and $k_i$to share a common region. As mentioned in Chapter 4.3, the automatic selection of the keyframes from V_{pre} is constrained in that the near keyframes must have a reasonable overlapping ratio for the homography between two keyframes that can be calculated. If this condition holds, the homography $\boldsymbol{H}_{i,i+1}$ ($\boldsymbol{H}_{i,i-1}$) between two keyframes k_i and k_{i+1} (k_i and k_{i-1}) can be computed successively. Then $\boldsymbol{H}_i$ can be recursively defined as $\boldsymbol{H}_i = \boldsymbol{H}_{i,i+1}\boldsymbol{H}_{i+1}$ for $i < base$ ($\boldsymbol{H}_i = \boldsymbol{H}_{i,i-1}\boldsymbol{H}_{i-1}$ for $i > base$). But this kind of calculation would cause accumulation error which will lead to bad alignment especially when there are many keyframe images. So, with this pipeline mentioned above, we select the base keyframe image from the center of the keyframe images in our first test.

The foreground mask with dilation [23] for each keyframe should be warped together with the keyframe images. Then with the warped masks and the warped key frame images, a multi-band blending process is applied to them to generate the final workspace map [15].

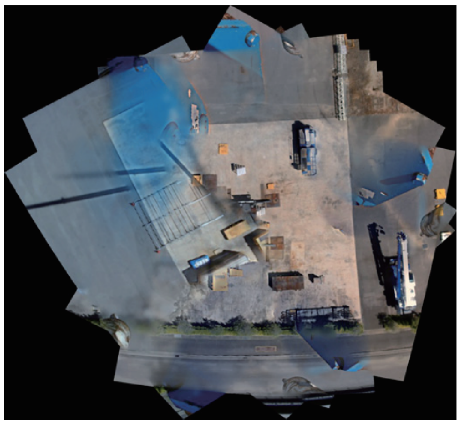
4.4.2 Workspace Map Generation Experiment

Two experiments have been done to verify the preprocessing stage.

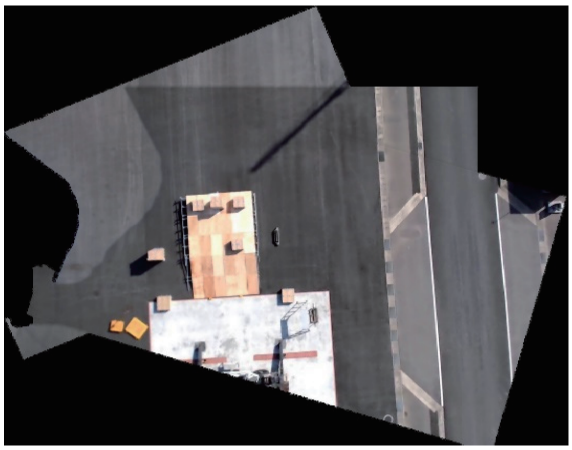
The first experiment is a generation of workspace maps from a short pre-shooting video, and three keyframes images are selected out to make this workspace map. In the second experiment, there are a total of 22 keyframe images picked out automatically from a long pre-shooting video. All the results are shown in Figure 4.6.



(a) Autostich with 3 key frames



(b) Autostich with 22 key frames



(c) Stitching with 3 key frames without foreground

(d) Autostich result with trimed 22 key frames

Fig. 4.6: Image stitching experiment results. (a) Workspace map stitched by auto-stitch with three key frames without foreground detection. (b) Workspace map stitched by auto-stitch with 22 key frames without foreground detection. (c) Workspace map stitched by the simple stitching pipeline with the detection of foreground of three key frames. (d) Workspace map stitched by auto-stitch with the foreground detection of 22 key frames.

Figure 4.6a is a panoramic image stitched with automatic image stitching [13]. Three frames are used to generate it. Its comparison are shown in Figure 4.6c. By stitching the

keyframes with only the background, a clear workspace map is obtained.

When there are a lot of keyframe images such as our second experiment, a total of 22 keyframes are selected out from the pre-shot video. We use auto-stitch software [13, 74] which has integrated the bundle adjustment process to eliminate this accumulation error. Except for providing bundle adjustment to eliminate the accumulation error, the software has integrated more functionalities such as auto-straightening and gain compensation [11]. By the comparison of the stitched result shown in Figure 4.6b and 4.6d, we can find the difference. By a direct image stitching process with images containing foreground, there will be a lot of ghosts which makes the panoramic unclear to people. And the stitching process is unstable by the influence of features on the foreground portion. With the detection of the foreground, a better workspace map can be generated with automatic image stitching.

In the second experiment, there are still many obvious ghosts in the stitched result. In the foreground detection process, the foreground is detected only through dozens of related frames around the keyframe. In such a continuous set of frames, the boom shadow almost keeps static on the ground.

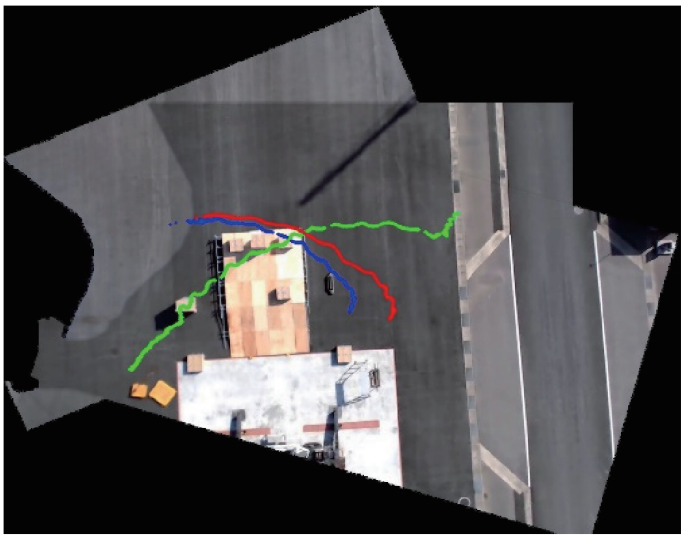
In the two experiments, the input image size from the top-view camera is 1920×1280. However, in the process of the foreground detection, the images are resized just one-forth of the original size. In the first experiment, it takes about 2 minutes to produce the final 2D workspace map with 3 selected keyframes from a short video. And in the second experiment with 22 selected keyframes, it takes about 8 minutes to produce the final 2D workspace map.

4.5 Experiments with Application of Path Location Display

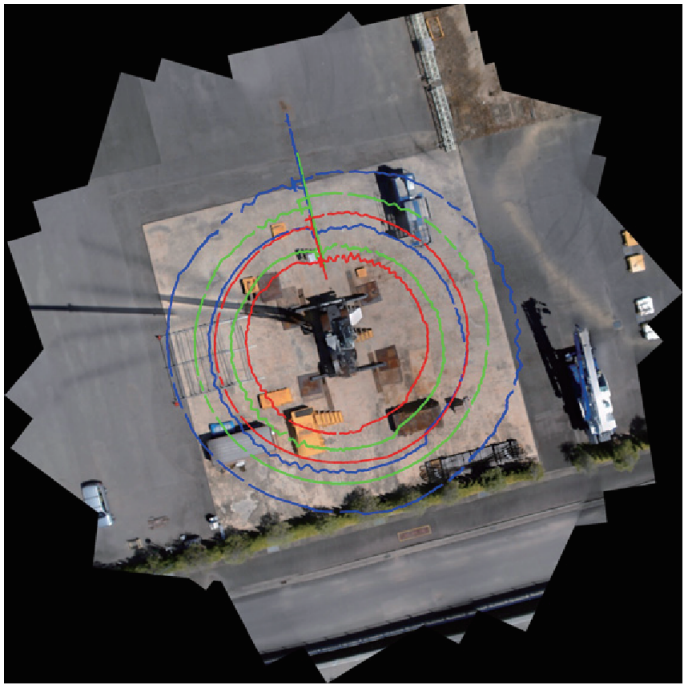
One goal of this study is to utilize the workspace map W for displaying some information to assist the operator. Here an application to overlay the path of the boom on W as a simple and useful piece of information is proposed to assist the operator. Another video V was recorded under the crane's ordinary working conditions to test the process for overlaying the boom head's positions onto the workspace map W. The path of the boom head T can be identified in this input video V and overlaid on the workspace map W. For each frame $f_i \in W$, we computed the homography from fi to W. Assuming the center of the top-view camera is always just below the boom head, the boom head position T_i on W is represented by the center position of f_i. By plotting T_i for all $f_i \in V$, the path of the boom head can be overlaid on W.

Here two experiments are carried out. For each experiment, three videos are used to find locations of boom head on W. The first three video for the first experiment is taken by a top-view camera about 21 meters above the ground. Figure 4.7a shows the results of displaying the boom head's positions on the first stitched workspace map. The first video was recorded under the constrained conditions described in Figure 4.4 with only an obvious hook as the foreground in frames. This video is also used to generate the 2D workspace map. The second and third videos are recordings of crane's ordinary working operations of moving an object with a hook. As shown in Figure 4.7a, the three paths consisting of many locations are clear. Some short gaps appear in the shown paths due to the blurry frames that were removed from the sample videos. Figure 4.7b is the other experiment. These three videos are taken by a top-view camera with different camera height. As can be seen, for each video, the operation is a circle rotation, pitching down and then a circle rotation. The detailed video recording setup are shown in Table 4.1 The difference between the two experiments is only the working environment around the crane.

In application, the location of boom head on the stitched workspace map will help greatly on vision. The operator in the cabin can have an overall understanding of the environment. In operation, the operator knows where the boom head is against the working environment especially there is something to block the vision of the operator. The



(a) Result of the first experiment.



(b) Result of the second experimetnt.

Fig. 4.7: Three clear paths are formed by locating the image's position on the workspace map.

Table 4.1: The second experiment conditions for taking the videos

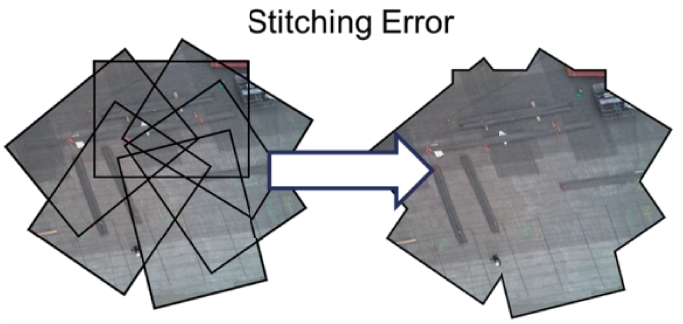
No.	Step	Boom Length (m)	Pitching Angle (deg)	Rotation Angle (deg)	Rotation Radius (m)
1	Rotation	30.5	64.9	0 to 360	11.6
	Pitching		64.9 to 55.1	None	11.6 to 16.3
	Rotation		55.1	360 to 0	16.3
2	Rotation	26.5	65.1	0 to 360	9.6
	Pitching		65.1 to 54.8	None	9.6 to 13.7
	Rotation		54.8	360 to 0	13.7
3	Rotation	23.5	65.0	0 to 360	8.2
	Pitching		65.0 to 54.8	None	8.2 to 11.8
	Rotation		54.8	360 to 0	11.8

recorded path can also help the crane company evaluate the work done by the operator and help to evaluate and train the operator.

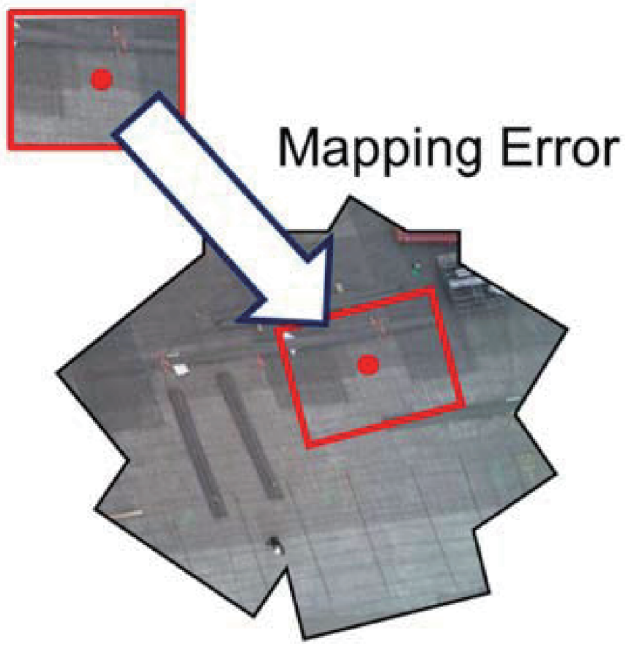
Different image features can be used to estimate the homography and find the location of the boom head in the 2D workspace map. Generally, in our experiment, we have tested SIFT features to estimate the 2D homography. The detection and matching process takes a lot of computation time. For now, it can only proceed with about 15 frames per second. But the crane always moves very slow, a slow frame rate such as 15 fps can be enough. And on the other hand, the feature detection and matching can be accelerated with GPU, which is a possibility to make it be more than 24 fps because the computation for features and matching can be highly parallelized.

4.6 Error Analysis

As shown in Figure 4.7, the position of boom head can be seen in both (a) and (b). One more thing we want to confirm is the precision of boom head's position on the generated workspace map. To verify the precision, we analyze the error of location of the camera's center obtained by the computation pipeline mentioned above. As shown in Figure 4.8, the location is computed by mapping the center point of an image captured with the top-view camera on the workspace map. The location error consists of stitching error and mapping error. The stitching error is in the workspace map caused by the distortion of the image stitching process, and the mapping error exists in the process of estimating homography from the top-view image to the workspace map.



(a) Stitching error.



(b) Mapping error.

Fig. 4.8: Error composition [9]. (a) The stitching error is caused by the image stitching process. Distortions exists while the generation of the workspace map. (b) The mapping error from the homography estimation from one image to the other image.

4.6.1 Stitching Error Analysis

In the image stitching process, many different distortions may exist. To evaluate the workspace map's quality, we need to confirm that it keeps good metric properties [36]. The metric property means that there is a similarity between the real world and the workspace map. So, we can verify the metric property through a comparison of angles and length ratios of some features and objects on the ground. These objects are shown in Figure 4.9.

Fig. 4.9: Features and objects which are used to verify the workspace map's metric property.

Angular Metric

In 4.9, we have labeled the square ground as A and many objects as B to F. There are lines which can be detected from A to F, which provides us a lot of parallel and orthogonal lines to compare. Through the Hough lines detection [5, 40], these lines are detected out successfully. The resolution for our hough space is 0.5 deg. The angles between the lines

we are interested in are picked out for computing their angles. For the angles computed, they differ in the range of 0 deg to 0.5 deg from the parallel and orthogonal relationship as shown in Table 4.2.

Table 4.2: Angle comparison.

	Parallel[deg]	Perpendicular[deg]
A	0.50	89.50-90.00
B	0.0-0.50	89.50
C	0.0-0.50	90.50
D	0.0-0.50	89.50
E	0.0	90.00
F	0.0-0.50	89.50-90.00

Length Ratio

As detected by the parallel lines, we can know the distance between the lines. Even they are not totally parallel to each other, a very close value for distance can be calculated. With the real length ratio measured manually in the real working field, a comparison is shown in Table 4.3.

Table 4.3: Length ratio comparison.

	Actual values			Measured values		
	Length[m]	Width[m]	Ratio	Length[pixel]	Width[pixel]	Ratio
B	5.64	2.35	2.40	314.80	128.00	2.46
C	2.82	1.22	2.30	149.71	65.84	2.27
D	4.20	2.00	2.10	233.00	111.55	2.09
E	4.20	2.00	2.10	232.00	110.00	2.11
F	4.15	1.91	2.17	238.67	108.00	2.21

In the table, the maximum ratio difference is from object B which is 0.06 in difference.

After the process of checking the angles and length ratios, a conclusion can be drawn that the workspace map keeps a good metric property and is similar to the real world. The relative error can be computed for both the angles and the length ratios which are all less than 2 percent. It ensures that the stitched image has a good quality to be displayed to the crane operator. The causes of the stitching are the projective transformation and the distortion of imaging process. The projective transformation will change the metric property which makes the length ratio and angles changed after transformation. And the distortion comes from the camera is because of its lens. The influence of projective

transformation can be reduced by setting the camera strictly point at the ground especially for the first keyframe. It needs a more advanced control system for the camera because that the boom will be bent and shaking while working which make it not point at the ground strictly. And the influence caused by the camera lens can be reduced by calibrating the camera first and then apply the undistortion to the keyframe.

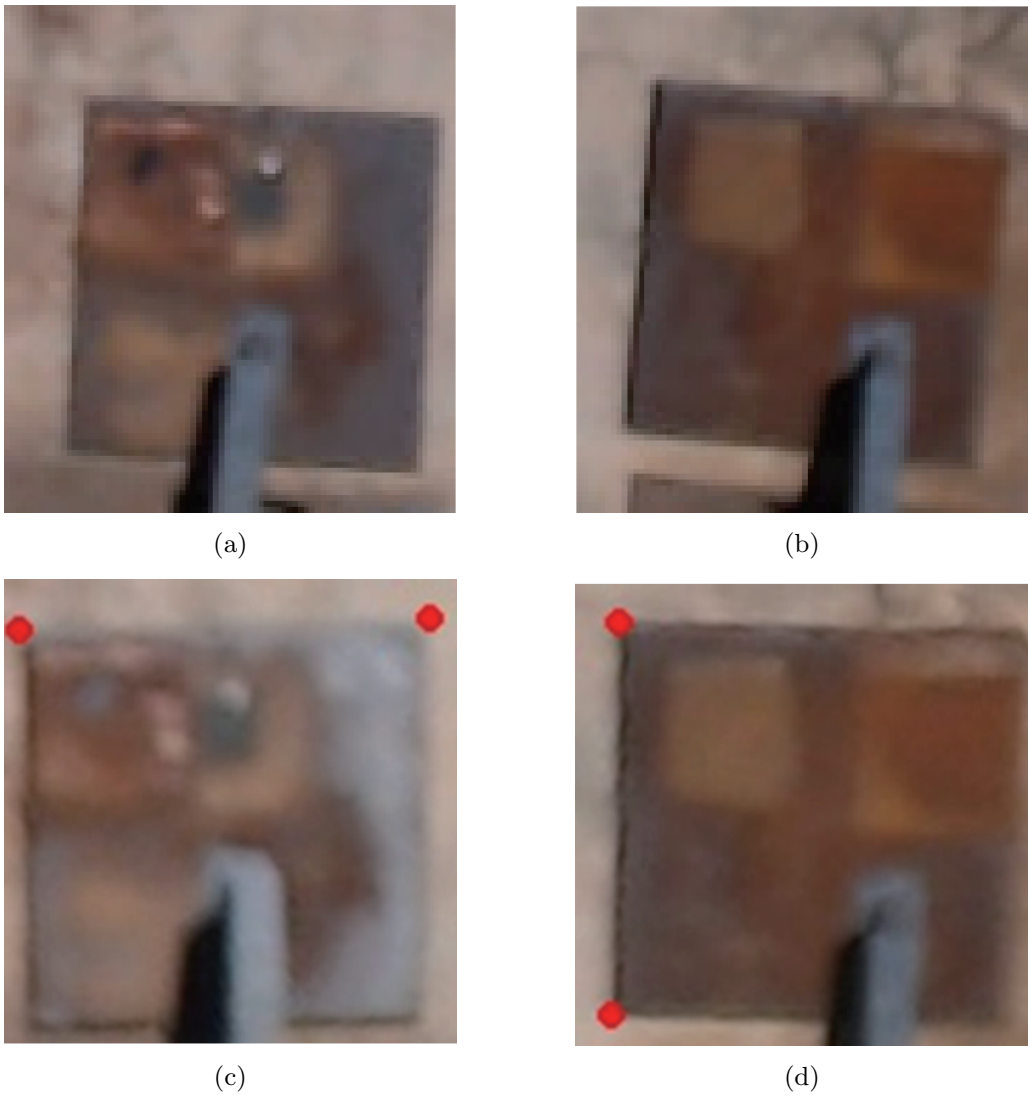
4.6.2 Mapping Error Analysis

The location for an image captured by the top-view camera is computed by a mapping of its center point to the workspace map through the homography. However, this mapping process through the homography involves error. And this error is from the estimation process of homography. To estimate this error, the real position of the image on the workspace map has to be figured out first. In our case, the traditional methods such as GPS and 3D motion capture system are not adoptable as it cannot provide the real position in the stitched workspace map. Through some fixed features on the workspace map and the image captured by the top-view camera, a similarity relationship can be built. And with the similarity, the center of the image's correspondent point on the workspace map can be found. We take this point as the real position noted as p_r.

To build the similarity relationship, we select two objects both existing in the image and workspace map as shown in Figure 4.10a and 4.10b. With canny edge detection [16, 25] and lines detection with RANSAC [30], four points can be detected from the two images. Figure 4.10c and 4.10d show the cross points of lines from Figure 4.10a and 4.10b. With the four correspondent points, a stable similarity relationship can be built. A rigid transform to move the center point of image captured by the top-view camera to the workspace map can be done. As a rigid transformation matrix has a form of $T = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$, T can be known through $Tp_f = p_w$ with the four pair corresponding points where p_f represents the point in the image from top-view camera and p_w represents the point from the workspace map. Then p_r can be found through $p_r = Tp_c$ where the p_c is the center point of the image taken by top-view camera.

In the mapping process, a homography is used to estimate the boom head's location $p_e = Hp_c$. Then the difference between p_r and p_e can be analyzed. A totally 85 tests of a continued frames have been tested and $E = \|p_e - p_r\|$ is calculated. The mean error is $\frac{1}{n}\sum_1^n E_i$ for $n = 85$. Then σ can be got through $\sigma_p = \sqrt{(1/(n-1)))\sum_1^n (E_i - E_m)^2)}$. Final σ is 1.853 pixel. And the maximum error is $L_e = E_m + 3\,_\sigma p$ which is 7.878 pixels

Under different height from the ground, one pixel's correspondent real length is different. And it can be represented with a linear relationship. We have estimated one pixel's real length under different height and made a linear regression to find the possible error for



(a) (b)

(c) (d)

Fig. 4.10: (a) and (b) are the two local regions both existing on the top-view image and workspace map. (c) and (d) shows the detected cross point of three lines. The three lines are detected from the canny edge with RANSAC line fitting.

the ground. The result is shown in Figure 4.11. The line in Fig. 14 has the least square error of E_s^2. Then σ_s can be computed with $\sigma_s = (1/(t-1))E_s^2$. Finally, for a location on the ground, its representation error should be $R_e = 1.9133^{+0.0482}_{-0.0482}$ cm. The final location error should both considering the location error Le and representation error Re with $E_f = L_e \times R_e$. Then we get the final error of $15.073^{0.038}_{-0.038}$ cm.

From the final projection error computation, it can be known that both the representation error for one pixel and the location in pixels contribute to the final result. The representation error is determined by the image stitching process for the 2D workspace

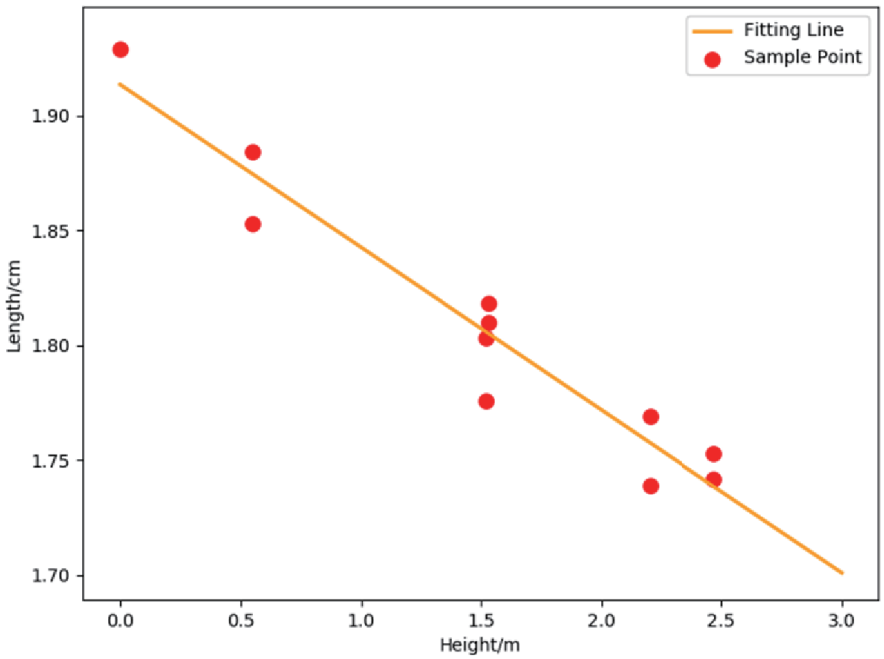


Fig. 4.11: One pixel is representing different real length under different height from the ground.

map. It can be reduced by requiring the camera point more perpendicular to the ground and reduction of initial image distortion which are discussed in the above section. And also for the location error in pixels, one of the causes of the error comes from the distortion and how strict the camera is pointing the ground. Another cause for the location error in pixels is because of the matched features location. A more precise location can be estimated with bundle adjustment by minimizing the sum of all correspondent points' distance.

For the all-terrain crane, it is always handling a very big object. The location error compared is quite small enough for application. Requirements from the crane company mentioned in chapter 1 also think error in 20 cm is enough for application.

4.7 Conclusion

Our program with automatic stitching with the proposed foreground detection process can create a clear workspace map from videos recorded from the boom head of a crane automatically. This application is useful for the median height environment. As shown in the result, few ghosts appear in the maps generated and the locations of objects in the videos are clearly represented in the generated workspace map. All the location points formed a clear moving path in the workspace map. The work is done in this chapter mainly consists of two parts.

- The clear workspace map generation process is consisting of two parts. In Chapter 3, we have proposed a method to detect the foreground by comparing the different motion patterns of the foreground and background of the image. In this chapter, we have integrated this process into automatic image stitching and finally obtained a high-quality workspace map.
- An application to show the location and path of the boom head's position with an impute video is proposed and tested. A preliminary accuracy verification is done to show it is applicable.

The workspace map can be used to help the operator know the location of the boom head's position and its movement path. To conclude, the advantages of this proposed system mainly include three aspects.

(1) The workspace map can remain the most information from the keyframe images. As we only detect the foreground of keyframe image, the information of the background can be kept as much as possible. This is useful when the workspace map is generated with just several keyframe images.

(2) The generated workspace map is clear. As the foreground detection can detect most of the foreground robustly, the workspace map is generated with most of the background information and just few foreground information.

(3) A comprehensive error analysis is made. And the error analysis shows that the generated workspace map is keeping a good metric property which means that it can be used for navigation and recording purpose. It can also present the operator a quite

accurate understanding of the working environment.

Chapter 5

Generation of 3D Spatial Map with Displaying Working Area Limit Line

5.1 Introduction

The construction site is a complicated environment. For a crane working under the complex workspace, the limited information and blind spots are always problems encountered by the crane operators. In Chapter 4, an approach for the crane workspace with only some medium-height objects are proposed to help assist crane operation.

However, while in the condition of existing some very high objects on the ground, to generate the 2D workspace map is impossible as the projection ambiguity which is shown in Figure 4.2 will get very obvious. It is possible to generate a good quality panoramic image as a 2D workspace map with the technology of seam estimation [48]and some better warping methods [32], and the projection ambiguity can be avoided in the 2D workspace map generation process.

But in the location process of finding the image's position on the 2D workspace map, there is still projection ambiguity and the generated 2D workspace map can not be used to help assist the crane operation with the precision requirement. Thus, the location on the 2D workspace map will not be reliable for the environment which contains the high objects. Some locations on the 2D workspace map can even go to the side surface of a high object because of the projection ambiguity in the generation process of the 2D workspace map and the location process for an image.

Besides, simply giving a vertical view of the scene with the top-view camera is ineffective for tall buildings as it lacks the sense of height information of the crane's working environment.

In addition, for the workspace with high objects, providing a 2D workspace map can not assist operation especially with the requirement of height information, such as bringing a load over a high object and down to the ground. Such height information is obtained by the depth information relative to the top-view camera, namely, the distance from the top-view camera. In brief, such height information of the working environment is called as the 3D spatial map.

The 3D spatial map is important and necessary for the safety and efficiency of crane operation, lifting path planning and showing the depth-related information. For example, in the lifting which requires to locate the object with very high precision, it is desirable to navigate the operation with a planned lifting path based on the 3D spatial map. To ensure the safety of the lifting process, the planned lifting path should be collision-free. Without the 3D spatial map, the purpose can be not be achieved. Some important information is also based on the 3D spatial map.

5.2 Research Objectives

In this research, we aim at developing a real-time 3D spatial map generation system based on the crane's top-view camera. The 3D spatial map consists of millions of 3D points and can provide the crane operator with correct environmental 3D information which cannot be provided by the 2D environment map introduced in Chapters 3 and 4. Our target system should satisfy the requirements explained below.

5.2.1 Real-time 3D Reconstruction

For the crane's working environment, it is always in a constantly changing status. Then, the reconstruction speed of the crane's working environment becomes a vital factor. If the reconstruction time is too long, it will influence the work efficiency and is not applicable for assisting construction. So, the real-time reconstruction of a 3D spatial map using the images captured with the top-view camera becomes one of our objectives.

To explain the objective for real-time 3D spatial map reconstruction in more detail, the objective can be clarified as follows:

- Reconstruction speed: As always mentioned, the reconstruction speed should be as fast as possible. It is required in our research to be a real-time reconstruction. For a general video camera, the record frame rate is about 30 frames per second. So, in the reconstruction process for a 3D spatial map, we require the reconstruction can handle at 30 frames per second for reconstruction.
- Reconstruction density: The density for the 3D spatial map is important for the application. For the ground and reconstructed objects, the points should be dense enough to have a more detailed description of the ground and objects. It is a basic requirement for drawing the working area limit line, which discussed in later sections, in the reconstructed 3D spatial map. On the other hand, the noise should be small enough and there should not be too many error points which can also affect the drawing of the working area limit line.
- Reconstruction error: The reconstruction of the 3D spatial map has the error. For different applications, the requirement for error is also different. In this Chapter, the application is to display the working area limit line. For an image captured with the top-view camera which should contain the working area limit line, the working area limit line is usually an arc line across the image. The influence of the final working area limit line in the image caused by the reconstruction error is required in less than a few percents of the size of the displayed image (eg: one percent of width 1080 pixels which is about 10 pixels).

5.2.2 Working Area Limit Line Display

Except for the generation of 3D spatial map generation, we also aim at an application for assisting the crane operators based on the 3D spatial map by displaying the working area limit line on the image captured with the top-view camera. The working area limit line is very important for the safety of crane operation. The lifted object can only be inside the region enclosed by the working area limit line. Many accidents such as boom crack and crane overturning are caused because of lifting the object out of the region enclosed with the working area limit line.

The working area limit is the maximum horizontal working range of crane. The lifted object should not be lifted out of the working area limit. The working area limit is related to the current working status and properties of the crane including current load, counter load and crane's mechanical properties. Just as shown in Figure 5.1, with the different payloads from heavy to light, the working area limit is different. With a heavy lifted payload, the working area limit is close to the crane's rotation center. However, if the lifted payload becomes lighter, the crane can lift the object to a further position. Figure 5.1 only shows the circumstance for one rotation angle. With the rotation of the crane for one circle, the working area limit at all the different rotation angles will finally form a working area limit line. For an assumed or known payload, the crane should always work inside the area enclosed with the working area limit line. For the case that the crane works out of the area enclosed by the working area limit line, the safety accidents can be caused. In most cases, the crane will be turned because the counter load cannot ensure the crane to be balanced. Or otherwise, the boom of the crane will crack as the limitation of the boom's mechanical strength is reached.

The crane operators should pay attention to the working area limit line during operation. But the working area limit line is not shown in the real working environment. The image captured with the top-view camera doesn't contain the working area limit line. And the other displays do not have the working area limit line. Thus, it is strongly requested to show the working area limit line in the image captured with the top-view camera. To draw and overlay the working area limit line to the image captured with the top-view camera, the 3D spatial map is a basic requirement. The working area limit line only considers the

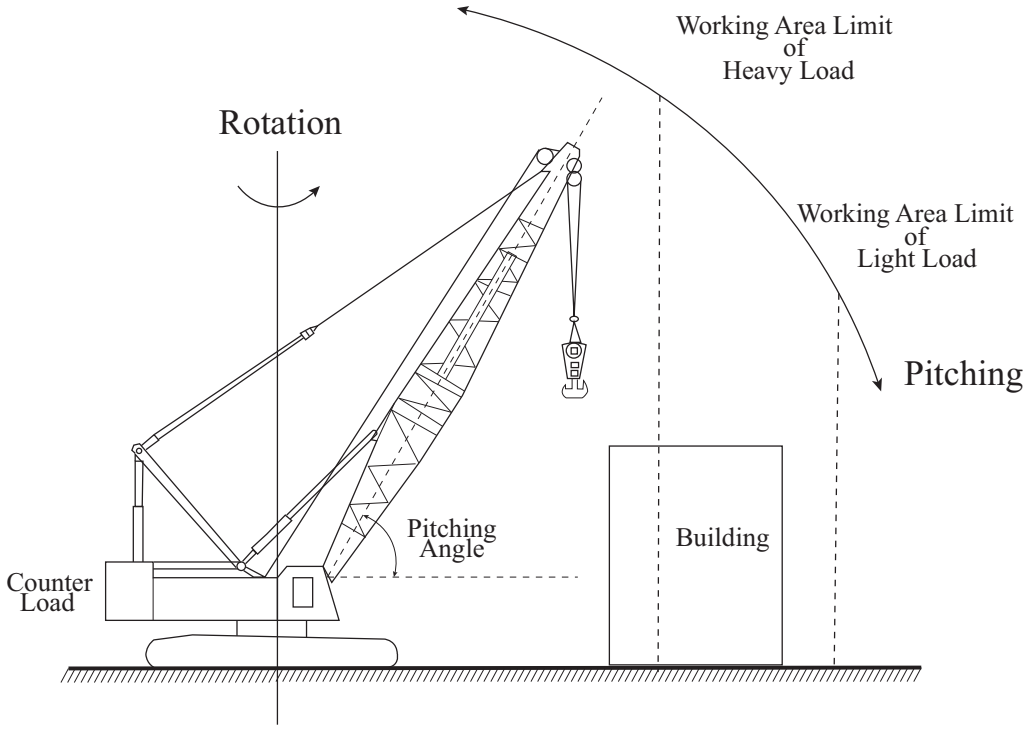


Fig. 5.1: The working area limit of crane. Working area limit is related to many aspects. With different payload on the hook, counter load, and mechanical structure of crane, the working area limit is different.

crane's working environment as flat ground. It is not very precise as it does not consider the height of the objects of the crane's working environment. For the real working area limit line in the crane's working environment, it is affected by the height of the objects when it crosses the objects, which can be seen in Figure 5.1.

The crane's working environment is always very complex. Only some initial construction working environment is almost a flat ground. For most of the situations, there are many tall objects and buildings that existed in the crane's working environment. Hence, the working area limit line showing in the image captured with the top-view camera is not on a pure plane. The working area limit line will be affected by the 3D spatial map of the working environment including the objects and pits. Except correctly drawing the working area limit line in the 3D spatial map, while displaying the working area limit line in the top-view image, the camera pose of the image should also be taken into consideration. The 3D spatial map and the crane's current working conditions determine the working area limit line in the 3D spatial map. The camera pose determines how the working area limit line in the 3D spatial map being projected to the top-view image. Only if the

camera can do an orthogonal projection of the crane's working environment, the working area limit line displayed on the projected image will have no relationship with the height information. But it requires the camera at infinity, which is impossible.

Figure 5.2 shows how the 3D spatial map and camera pose influence the working area limit line of a top-view image. Figure 5.2a and 5.2b shows how the detail of the 3D spatial map influence the working area limit line. The situation of the working area limit line goes through a tall object and a rectangular pit is shown respectively in Figure 5.2a and 5.2b respectively.

Figure 5.2c and 5.2d shows the influence of different camera pose for the working area limit line for the case of going through a tall object. Just as shown, different camera poses will yield different results. In Figure 5.2c, the top-view camera is inside the area enclosed with the working area limit line. The working area limit line on the tall object extends more outside compared with the working area limit line on the ground. However, when the camera moves outside the area enclosed with the working area limit line, we can see that the influence of the tall object for the working area limit line is totally in an inverse case, just as shown in Figure 5.2d. Of course, for the case shown in Figure 5.2d, it is forbidden because it means that the payload is lifted to the danger area.

In a word, to display the working area limit line in the top-view image, we need to know the 3D spatial map and estimate the camera pose against the 3D spatial map. To achieve these objectives, we need to prepare the necessary basics in the preprocessing stage:

- Reconstruct the 3D spatial map M.
- Compute the working area limit line C in the 3D spatial map M.

After the preprocessing, in the operation mode, the working area limit line can be shown in the top-view image with:

- For a new top-view image $\boldsymbol{I}_t$, estimate its camera pose $\boldsymbol{p}_i$ against the map M.
- Project the working area limit line C in 3D with the camera pose $\boldsymbol{p}_i$ and intrinsic $\boldsymbol{K}$ to the 2D working area limit line D_t.
- Overlay the projected 2D working area limit line D_t to the image $\boldsymbol{I}_t$.

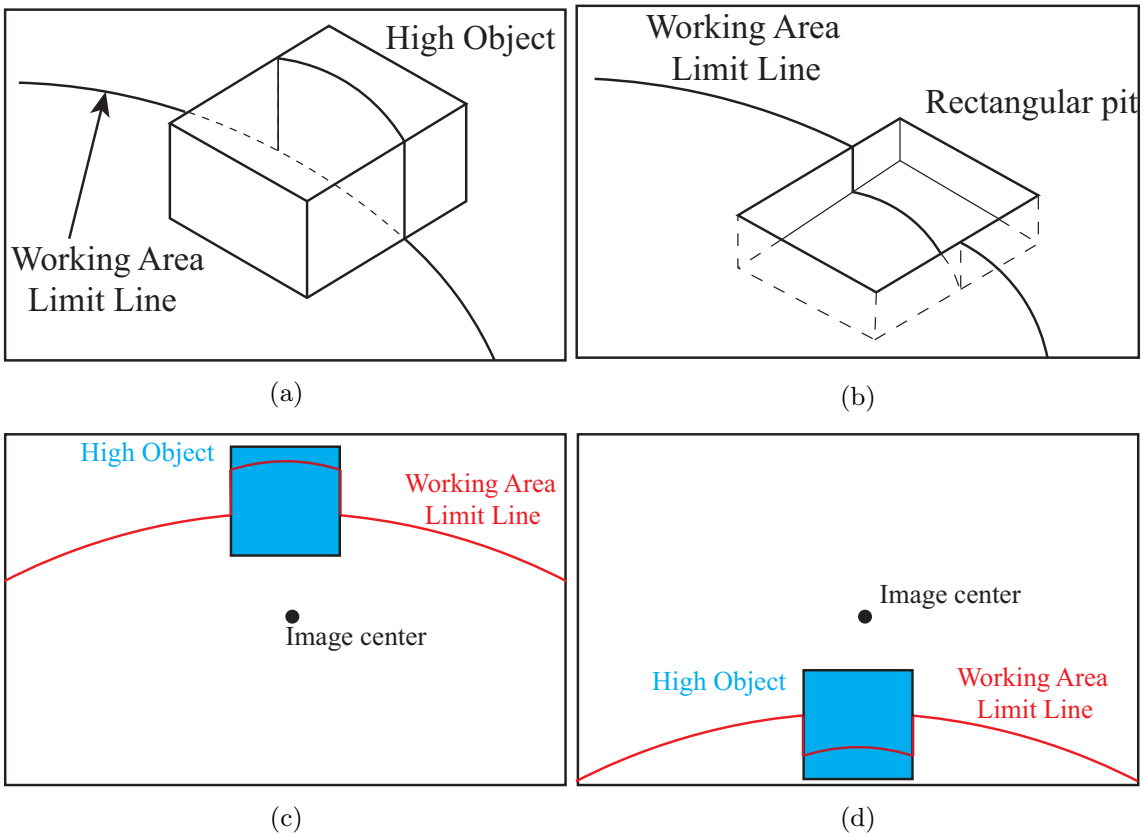


Fig. 5.2: The working area limit line is affected by the height of the environment. (a) shows the influence of a high object. (b) shows the influence of a rectangular pit. (c) The camera pose is inside the working area limit line. (d) The camera pose is outside the working area limit line.

5.3 Approach and Overview

The 3D reconstruction stage is a 3D reconstruction pipeline that can reconstruct the crane's workspace with high efficiency. A pre-shot video taken by the top-view camera will be used to generate the 3D spatial information of the crane's working environment.

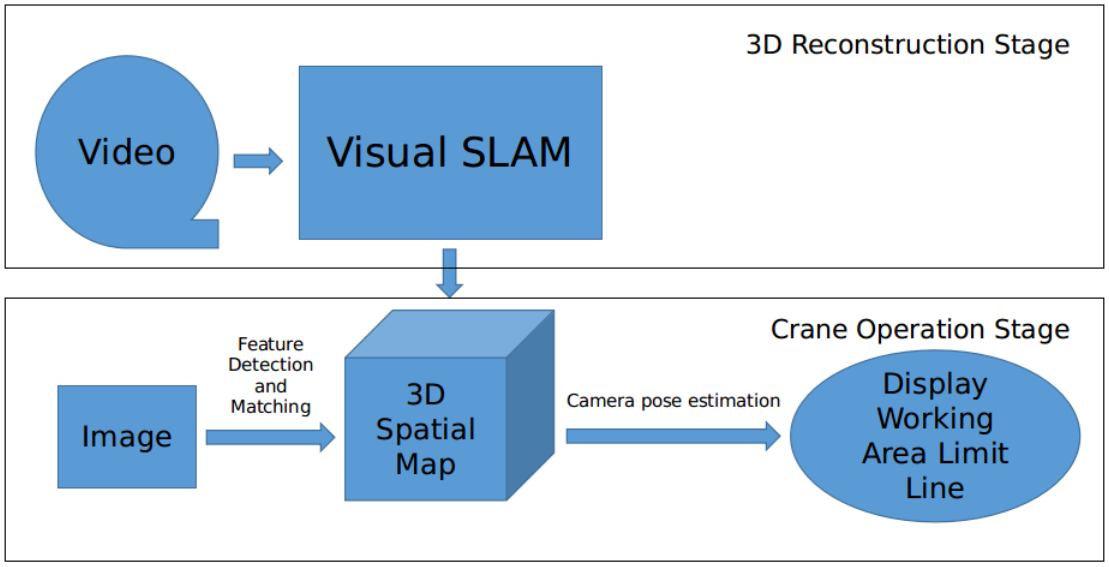


Fig. 5.3: The schematic principle of proposed approach. In the 3D reconstruction stage, the environment will be reconstructed in real time. In the crane operation stage, the working area limit line will be augmented to the image captured with top-view camera.

The systematic overview is shown in Figure 5.3 . There are two stages for the system which are the 3D reconstruction stage and the crane operation stage.

In the 3D reconstruction stage, the objective is to generate a dense 3D spatial map with the working area limit line. A visual SLAM approach has been chosen to achieve the objective. A pre-scan of the crane's working environment will be made by rotating the boom with its length and pitching angle fixed. A video will be generated which contains information about the crane's working environment. Through the visual SLAM, a dense 3D spatial map can be generated. Through trajectory fitting of the top-view camera, the working area limit line can finally be drawn in the 3D spatial map.

After obtaining the 3D spatial map which contains the working area limit line, in the crane operation stage, the objective is to show the working area limit line in the top-view image. For a new image captured by the top-view camera and shown in the display of

the crane's operation cabin, it will be matched with the 3D spatial map and the camera pose of the image will be estimated. Finally, the working area limit line will be projected from the 3D spatial map to the 2D image captured with the top-view camera. The main originality of this research are:

- First trial of using the mixed way SVO and REMODE in the application for the real-time reconstruction of construction site.
- Displaying the working area limit line in the top-view image which considers the influence of the 3D spatial map.

In summary, the major benefits of the proposed system in terms of assisting crane operators are as follows:

- A 3D reconstruction for the generation of the 3D spatial map is selected and applied for the reconstruction of the crane's working environment. The reconstructed 3D spatial map contains various information and is very useful in applications such as displaying useful information. The useful information and applications can be displayed and developed for assisting the crane operators.
- The working area limit line displayed on the top-view image is achieved. The working area limit line is vital in operation for notifying the operators' safety area restriction.
- The working area limit line is drawn in the reconstructed 3D spatial map. By a RANSAC circle fitting of the estimated camera poses with VSLAM, the rotation axis of the crane in the 3D spatial map is computed and the working area limit line is successfully drawn in the 3D spatial map.
- Displaying the working area limit line in the top-view image. By the estimation of camera pose against the 3D spatial map, the working area limit line is projected from 3D in the 3D spatial map to 2D in the top-view image.

5.4 Real Time 3D reconstruction with SVO and REMODE

In the scanning process of the crane's working environment, the camera is doing a 4 DOF motion. The video will be recorded under the camera's 4 DOF motion. In this section, the approach to reconstruct the 3D environment only with a sequence of images in a video is explained. By using a combined VSLAM approach, i.e., SVO and REMODE, the 3D working environment of a working crane can be reconstructed with millions of 3D points.

There are many different kinds of VSLAM approaches that can reconstruct a 3D spatial map. Basically, we select SVO and REMODE as the approach to reconstruct the 3D spatial map of the crane's working environment is on the consideration of our objectives mentioned in the above section.

5.4.1 Selection of SLAM Approach

To obtain the 3D spatial map of the crane's working environment, a survey with depth-sensing capability needs to be conducted first. Various methods can be applied to acquire the 3D spatial map consisting of many 3D points. For the 3D reconstruction with one or several cameras, the method is called visual simultaneously location and mapping (VSLAM).

For the crane, it is convenient to mount just one camera. With only one camera, the monocular VSLAM can be used for reconstruction. For the reconstruction of the crane's working environment, the distance from the top-view camera to the ground is very big. It requires the VSLAM method has a good perception for the long distance to reduce the reconstruction error. The selection is based on the objectives mentioned in section 5.2.1.

In recent decades, as the rapid development of VSLAM approaches, there are many state of the art technologies which are mentioned in Chapter 2 and can be used for 3D reconstruction of the environment. Table 5.1 is showing some of the state of the art VSLAM methods. These approaches have good accuracy in reconstruction and camera pose estimation.

Table 5.1: Comparison of different VSLAM approaches

	Type	Location	Mapping	Density of Map	Speed
PTAM	feature-based	✓	✓	sparse	real-time
ORB-SLAM	feature-based	✓	✓	sparse	real-time
LSD-SLAM	feature-based	✓	✓	semi-dense	real-time
SVO	semi-direct	✓	✓	sparse	real-time
DTAM	direct	✓	✓	dense	real-time
REMODE	direct	x	✓	dense	real-time

The main concerns for the 3D reconstruction of the crane's working environment are clarified in section 5.2.1 which are reconstruction speed, reconstruction density and reconstruction error.

From Table 5.1, the last two approaches DTAM and REMODE can both have a dense reconstruction for the crane's working environment. But considering the scale problem and reconstruction precision, we can only choose REMODE. For DTAM, in the reconstruction process, there is a cost volume which means that it can only reconstruct the environment

within a short range with high precision. The initial scale makes it difficult to initialize automatically also. The speed of the two approaches can both achieve real-time with a good GPU.

But there is a disadvantage for REMODE. It can not initialize and estimate camera pose alone. It is an approach which can only recover the 3D spatial map of crane's working environment. Another approach for camera pose estimation should be given to work with REMODE. The author of REMODE has tested it with SVO for it is the fastest way to estimate camera pose and have good accuracy for camera pose estimation.

Based on the above requirements and considerations, we make our final decision to use SVO and REMODE to reconstruct the 3D spatial map. The indoor experiment has been made in very initial test with REMODE [58]. There are mainly two benefits of using such a combined way for the reconstruction of 3D spatial map which are computation speed and point density of the reconstructed 3D spatial map.

In the following sections 5.4.2 and 5.4.3, the details of the two approaches will be explained.

5.4.2 SVO: Semi Direct Visual Odometry

As shown in Figure 5.4, after the scanning process for the crane's working environment, the reconstruction with SVO and REMODE for the crane's working environment needs to be carried out. As the first component for the 3D reconstruction for the crane's working environment, SVO achieves the estimation for camera pose at a very fast speed for every frame in the recorded video. Even SVO is a fully VSLAM approach which can estimate the camera pose and reconstruct the environment in a sparse point cloud, as the reconstruction result consisting of several thousand 3D points, we only need the fast estimation for camera pose. The reconstruction process for a dense point cloud will be achieved with REMODE.

Figure 5.4 is an overview of SVO. Just like PTAM [45], SVO uses two parallel threads to achieve the purpose of camera pose estimation and environment map reconstruction. The benefit of the two parallel threads structure is the high computation speed.

The first thread is the motion estimation thread. This thread has all the functionalities which are required for REMODE. It can estimate the relative camera pose for two continuous frames. Like the direct VSLAM method, in the motion estimation thread, SVO is trying to estimate the relative camera pose $\boldsymbol{T}_{k,k-1}$ from the previous image $\boldsymbol{I}_{k-1}$ to the present image $\boldsymbol{I}_k$ with the sparse model-based image alignment shown in equation 5.1 [31].

$$\boldsymbol{T}_{k,k-1} = \arg\min_{\boldsymbol{T}} \iint_{\bar{\boldsymbol{R}}} \rho[\delta \boldsymbol{I}(\boldsymbol{T}, \boldsymbol{u})] d\boldsymbol{u} \tag{5.1}$$

The term $\delta \boldsymbol{I}$ is the intensity residual for the two images while they are observing the same environment 3D points. In the image $\boldsymbol{I}_{k-1}$ with the camera pose $\boldsymbol{T}_(k-1, w)$, the pixel $\boldsymbol{u}$ with a depth value d_u can be inversely projected from the 2D image to 3D environment with equation 5.2 [54] at the current camera coordinate frame. The term $\boldsymbol{K}$ is the camera intrinsic and $\dot{\mathbf{u}}$ is a 2D homogenous coordinate for the pixel $\boldsymbol{u}$.

$$\pi^{-1}(\boldsymbol{u}, d_u) = \frac{1}{d} \boldsymbol{K}^{-1} \dot{\boldsymbol{u}} \tag{5.2}$$

For the image $\boldsymbol{I}_k$ with the camera pose $\boldsymbol{T}_{k,w}$, it can project the recovered 3D point in equation 5.2 to the image with equation 5.3. In the last, the intensity residual $\delta \boldsymbol{I}$ can be found with equation 5.4.

$$k(\boldsymbol{I}_k(u), 1)^T = \boldsymbol{K}\, \boldsymbol{T}_{k,w}\, \boldsymbol{T}_{w,k-1} (\pi^{-1}(\boldsymbol{u}, d_u), 1)^T \tag{5.3}$$

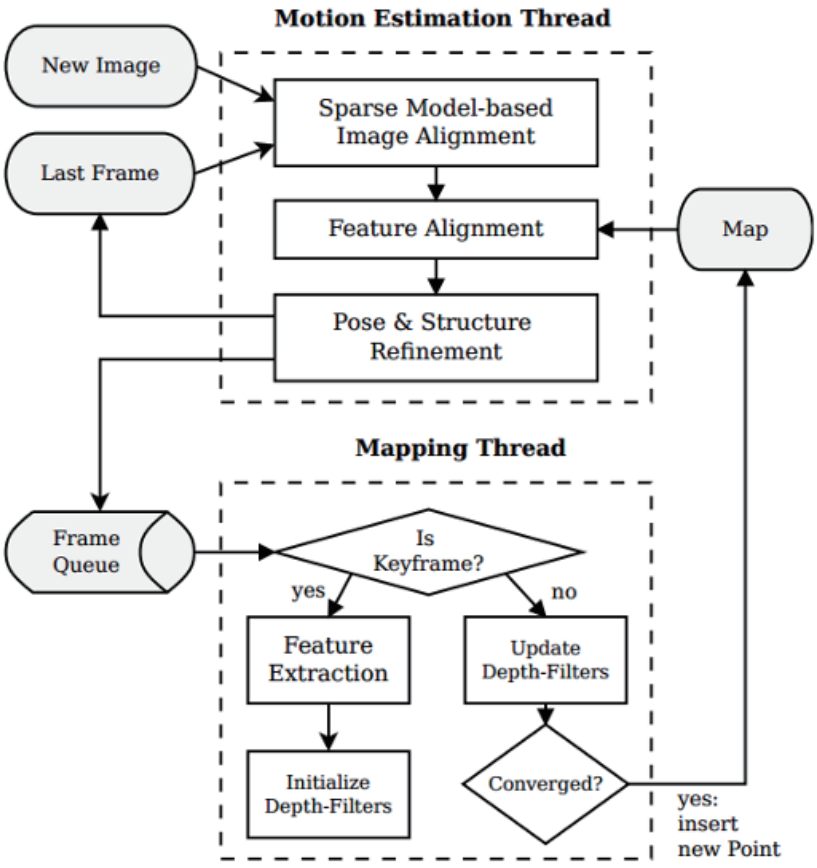


Fig. 5.4: System overview of SVO [31]. There are two parallel threads. The motion estimation thread estimates the camera pose for every image. The mapping thread generate a world map for the fast corners of the key frames.

$$\delta \boldsymbol{I} = \boldsymbol{I}_k(\boldsymbol{u}) - \boldsymbol{I}_{k-1}(\boldsymbol{u}) \tag{5.4}$$

The term $\bar{\boldsymbol{R}}$ denote the image region where in the image $\boldsymbol{I}_{k-1}$ the recovered pixel $\boldsymbol{u}$ can be seen in the image $\boldsymbol{I}_k$. It is defined with equation 5.5.

$$\bar{\boldsymbol{R}} = \{\boldsymbol{u} | \boldsymbol{u} \in \mathbf{R}_{k-1} \; and \; \boldsymbol{I}_k(\boldsymbol{u}) \in \boldsymbol{I}_k\} \tag{5.5}$$

Then the problem becomes to solve the least square of the sum of all the ρ. The $k-th$ term ρ_k can be computed with the equation 5.6.

$$\rho_k = \frac{1}{2} \left\| \delta \boldsymbol{I}_k \right\|^2 \tag{5.6}$$

Different from the other direct VSLAM approach which a large region of pixels depth value is known, in SVO, only the sparse image feature point $\boldsymbol{u}_i$ have a depth value $d_{\boldsymbol{u}_i}$. Then, a small patch of 4×4 pixels around the location $\boldsymbol{u}_i$ is denoted as $\boldsymbol{I}(\boldsymbol{u}_i)$. The

estimation of the relative camera pose $\boldsymbol{T}_{k,k-1}$ from the image $\boldsymbol{I}_{k-1}$ to the image $\boldsymbol{I}_k$ can be achieved by iteratively minimize the equation 5.7 [31].

$$\boldsymbol{T}_{k,k-1} = \arg\min_{\boldsymbol{T}_{k,k-1}} \frac{1}{2} \sum_{i \in \bar{\boldsymbol{R}}} \|\delta \boldsymbol{I}(\boldsymbol{T}_{k,k-1}, \boldsymbol{u}_i)\|^2 \tag{5.7}$$

The equation 5.7 is a nonlinear equation. The Gauss-Newton procedure can be used to solve this equation iteratively by minimizing the least-squares. With the mentioned equations above, the relative camera pose can be estimated. And through the process of estimating the relative camera pose, the feature points on the present image and their back-projection 3D points can be obtained.

But if all the camera poses are estimated iteratively from the present frame to its previous frame, the drifting error will accumulate. To reduce the drift, the camera pose estimation should be made against the map, but not with the previous image. For the image $\boldsymbol{I}_k$, there will be a map consisting of many 3D points. After initially estimation for the relative camera pose from the previous image to the present image, all the 3D points existed in the map are projected to the present image. With the initial relative camera pose estimation, the patch in the previous image is found in the present image. To reduce the drift, now a step by finding the 2D image features' correspondence from the previous reference image to the present image will be conducted. As the image features on the reference image will be used to recover the map. This global to local estimation will reduce the drifting error. This process is achieved by minimizing the photometric error of the patch in the current image with respect to the reference patch in the keyframe, just as shown in equation 5.8 [31]. In the equation, $\boldsymbol{A}_i$ is an affine warping.

$$\boldsymbol{u}_i = \arg\min_{\boldsymbol{u}'_i} \frac{1}{2} \left\| \boldsymbol{I}_k(\boldsymbol{u}'_i) - \boldsymbol{A}_i \cdot \boldsymbol{I}_r(\boldsymbol{u}_i) \right\|^2, \quad \forall i \tag{5.8}$$

Through the equation 5.8, the feature correspondences can be established from the previous key image to the current image. Starting from the initially estimated camera pose, with the established feature correspondences, a final optimization can be made with bundle adjustment (BA) [74] by minimizing the reprojection residuals shown in equation 5.9 [31].

$$\boldsymbol{T}_{k,w} = \arg\min_{\boldsymbol{T}_{k,w}} \frac{1}{2} \sum_{i} \|\boldsymbol{u}_j - \pi(\boldsymbol{T}_{k,w}, w\boldsymbol{P}_i)\|^2 \tag{5.9}$$

Along with the camera estimation thread, the mapping thread will also start to work

with the motion estimation thread. As REMODE only needs the camera pose as input, the mapping thread of SVO will not be explained in this section.

In a word, in the motion estimation thread, the camera pose for the image can be precisely estimated with the equations from 5.1 to 5.9. In a word, there are three steps to estimate the camera pose precisely. Firstly, through the sparse model-based image alignment, the camera pose can be initially estimated from the current image to the previous image. After that, through feature alignment, the features on the keyframe can be figured out in the current image. The image feature correspondences are established with the last step. In the last, with the initially estimated camera pose and established image feature correspondences, the precise camera pose can be estimated with bundle adjustment.

5.4.3 REMODE: REgularized Monocular Depth Estimation

Regularized monocular depth estimation (REMODE) is a probabilistic depth measurement for estimating dense and accurate depth maps [58]. The measurement of depth for a keyframe is a real-time estimation on a per-pixel basis. With the computation of uncertainty, the erroneous estimated pixels which have the wrong depth values will be checked out with the computed uncertainty. The core technologies for REMODE is Bayesian estimation and convex optimization for image processing. REMODE has performed good properties on computation speed and accuracy.

REMODE has three considerations while in the reconstruction from a single moving camera. Firstly, the accuracy of the reconstructed map is taken into consideration. In many dense 3D reconstruction approaches, they pay more attention to the visual result of the reconstructed map. But REMODE aims at providing a more accurate map by measuring the uncertainty for every pixel of the image. It's a solid basis for application based on the reconstructed 3D map. Secondly, REMODE has considered the density of the 3D point cloud for the reconstruction. Generally, the feature-based VSLAM can only generate a sparse point cloud in the last. Many applications such as crane manipulation, obstacle identification and path planning can not be made with the 3D spatial information described by the sparse point cloud. The last consideration of REMODE is reconstruction speed. According to the image and its estimated camera pose, an update for depth estimation need to be made efficiently. And also the estimation of uncertainty can be improved.

Depthmap estimation with multiple images

As mentioned above, firstly, REMODE will estimate depth for the reference image. The depth estimation at the reference image is taken as a Bayesian estimation problem. On the current frame's observation of apixel, its depth value will be updated with a parametric model. And in the last, for the result of the depth map, a smoothness procedure will be made through minimizing a regularized energy functional. With the known camera pose $\boldsymbol{T}_{k,w}$ for the k-th image , a 3D point $\boldsymbol{P}_w$ under the world coordinate frame can be can be transferred to the current coordinate transform with equation 5.10.

$$\boldsymbol{P}_k = \boldsymbol{T}_{k,w} \cdot \boldsymbol{P}_w \tag{5.10}$$

An observation of a camera will form an image. Along with its camera pose, the observation can be noted as $(\boldsymbol{I}_k, \boldsymbol{T}_{k,w})$. A video recorded consisting of m images with a moving camera can be denoted with equation 5.11. The reconstruction for a reference image will be made from a continuous sequence of images in the video. For the reference image $\mathbf{textif}\mathbf{I}_r$, its following n images will be used. In this subset, from the views r to k, a depth hypothesis d_k can be generated with triangulating the pixel $\boldsymbol{u}$ of the reference image $\boldsymbol{I}_r$.

$$V = \{(\boldsymbol{I}_k, \boldsymbol{T}_{k,w}) | k = 1, 2, \cdots, m\} \tag{5.11}$$

Then, from the reference image $\boldsymbol{I}_r$ to the other images of the sequence, a set of noisy measurement is estimated as d_k for $k = 1, 2, \cdots, n$. The depth information is sensed through a distribution which mixes a good measurement and an outlier measurement. The good measurement is assuming that the real depth $\hat{d}$ is the center of a normal distribution. And the outlier measurement is uniformly distributed in an interval $[d_{min}, d_{max}]$ which is known to contain the depth for the structure of interest [58]. The sensing process can be described with equation 5.12, where ρ and τ_k^2 are the probability and the variance of good measurement.

$$p(d_k|\hat{d}, \rho) = \rho\mathcal{N}(d_k|\hat{d}, \tau_k^2) + (1 - \rho)\mathcal{U}(d_k|d_{min}, d_{max}) \tag{5.12}$$

The observations from r to $r + n$ are assumed to be independent. Hence for $\hat{d}$, with all the measurements from d_{r+1} to d_{r+n}, its Bayesian estimation can be obtained through the posterior from equation 5.13 with $p(\hat{d}, \rho)$ being a prior on the true depth and the ratio of good measurements which is supporting it.

$$p(\hat{d}, \rho|d_{r+1}, \cdots, d_k) \propto p(\hat{d}, \rho) \prod_k p(d_k|\hat{d}, \rho) \tag{5.13}$$

At the time step $k - 1$, the estimation is applied to make a sequential update. This update is prior to combining with the observation at time step k. For the inlier ratio, it can be estimated with the product of a Gaussian distribution for the depth and a Beta distribution [76] as shown in equation 5.14. The terms α_k and b_k controls the Beta distribution.

$$q(\hat{d}, \rho|\alpha_k, b_k, \mu_k, \sigma_k^2) = Beta(\rho|\alpha_k, b_k)\mathcal{N}(\hat{d}|\mu_k, \sigma_k^2) \tag{5.14}$$

With the k-th image, equation 5.15 will make the update. By matching the first and second-order moments for $\hat{d}$ and ρ for a *Beta timesGaussian* distribution, the true poste-

rior can be estimated approximately [68]. The parameters for a_k, b_k, μ_k and σ_k^2 can then be known.

$$q(\hat{d}, \rho | d_{r+1}, \cdots, d_k) \approx q(\hat{d}, \rho | a_{k-1}, bk-1, \mu_{k-1}, \sigma_{k-1}^2) \tag{5.15}$$

With the Bayesian estimation process, a depth map $D(\boldsymbol{u})$ can be obtained. And a regularized posterior for the depth map should be made to denoise the depth map. For every pixel in the reference image $\boldsymbol{I}_r$, its depth estimation ν_k and confidence σ_k^2 are given upon the k-th image. The problem for denoising the depth map $D(\boldsymbol{u})$ can be noted as $F(\boldsymbol{u})$ which is a minimization of equation 5.16. The regularizer and the data term is balanced and controlled by the free parameter λ.

$$\min_F \int_\Omega \{G(\boldsymbol{u}) \left\|\nabla F(\boldsymbol{u})\right\|_\epsilon + \lambda \left\|\nabla F(\boldsymbol{u}) - D(\boldsymbol{u})\right\|_1\} d\boldsymbol{u} \tag{5.16}$$

Paper [10] has explained the " G-Weighted Total Variation " $G(\textbf{textitu})$. $G(\textbf{textitu})$ is a weighting function. Based on the Huber norm and gradient, the non-smooth surfaces can be penalized with a regularization term shown in equation 5.17. The Huber norm is always a good choice because it allows smooth reconstruction while preserving discontinuities at strong depth gradient locations [54].

$$\left\|\nabla F(\boldsymbol{u})\right\|_\epsilon = \begin{cases} \frac{\|\nabla F(\boldsymbol{u})\|_2^2}{2} & if\ \|\nabla F(\boldsymbol{u})\|_2 \leqslant \epsilon \\ \|\nabla F(\boldsymbol{u})\|_1 - \frac{\epsilon}{2} & otherwise. \end{cases} \tag{5.17}$$

In the computation process, the regularization is affected by the weighting function $G(\boldsymbol{u})$. Hence the the strength of regularization is estimated on the basis of the measure confidence for $\boldsymbol{u}$ with equation 5.18. The expected value of the inlier ration and the variance which is denoted as $\mathbb{E}[q]$ and σ^2 can account for the specific pixel $\boldsymbol{u}$. The regularization term is affected by the weighting function 5.18. For the inlier ration ρ, it is controlled by the measurement variance σ^2 for measurements with a high expected value. The regularization will have just little influence on reliable measurements which is characterized by a small variance. However, ff the result is a measurement showing an expected value which is quite small or high variance, the measurement will be considered an unreliable measurement and the regularization term will have a strong influence on the measurements:

$$G(\boldsymbol{u}) = \mathbb{E}[q](\boldsymbol{u}) \frac{\sigma^2(\boldsymbol{u})}{\sigma_{max}^2} + \{1 - \mathbb{E}[q](\boldsymbol{u})\} \tag{5.18}$$

The minimization for equation 5.16 can be computed iteratively with the work of paper [17]. The primal-dual formulation is exploited for the algorithm shown in equation 5.19. With alternating the gradient descent and ascent procedures, the function can be solved in the primal F and dual variables F^*:

$$\min_{F} \max_{F^*} \langle \mathbf{diag}(G)\nabla F, F^* \rangle + \lambda \left\| C - D \right\|_1 - \delta_{F^*}(F^*) - \frac{\epsilon}{2} \left\| F^* \right\|_2^2 \tag{5.19}$$

The term $\delta_{F^*}(F^*)$ is a indicator function defined in equation defined with equation 5.20.

$$\delta_{F^*}(F^*) = \begin{cases} 1 & if \ \ \left\| F^* \right\|_1 = 0 \\ \\ \infty & otherwise. \end{cases} \tag{5.20}$$

For the primal and dual variables, it is assumed that t and t^* to be the time steps receptively. For the weighted Huber denoising function 5.16, the update steps will take places with the equation 5.21. The resolvent operators are shown in equation 5.22 and 5.25. For a specific pixel $\boldsymbol{u}$ of a reference image, the value d is the noisy depth value of the pixel.

$$\begin{cases} F_{n+1}^* = \text{prox}(\frac{F_n^* + t^*(\text{diag}(G)\nabla \quad \bar{F})}{1 + t^*\epsilon}) \\ \\ F_{n+1} = \text{shrink}(F_n - t(\nabla^T \text{diag}(G) F_{n+1}^*) \\ \\ \bar{F_{n+1}} = 2F_{n+1} - F_n \end{cases} \tag{5.21}$$

$$\text{prox}(\tilde{f}^*) = \frac{\tilde{f}^*}{\max(1, \left| \tilde{f}^* \right|)} \tag{5.22}$$

$$\text{shrink}(\tilde{f}) \begin{cases} \tilde{f} - t\lambda & if \ \ \tilde{f} - d > t\lambda \\ \\ \tilde{f} + t\lambda & if \ \ \tilde{f} + d > -t\lambda \\ \\ d & if \ \ \left| \tilde{f} - d \right| \leqslant t\lambda \end{cases} \tag{5.23}$$

Measurement uncertainty

In the process of depth map estimation mentioned in the above section, the measurement uncertainty τ_k^2 is used. In this section, the definition and computation of the measurement uncertainty will be explained.

As shown in Figure 5.5, the definition of measurement uncertainty is straightly defined as the distance from ${}_r\boldsymbol{P}$ to ${}_r\boldsymbol{P}^+$ which can be noted as equation 5.24.

$$\tau_k^2 = (\|{}_r\boldsymbol{P}^+\| - \|{}_r\boldsymbol{P}\|)^2 \tag{5.24}$$

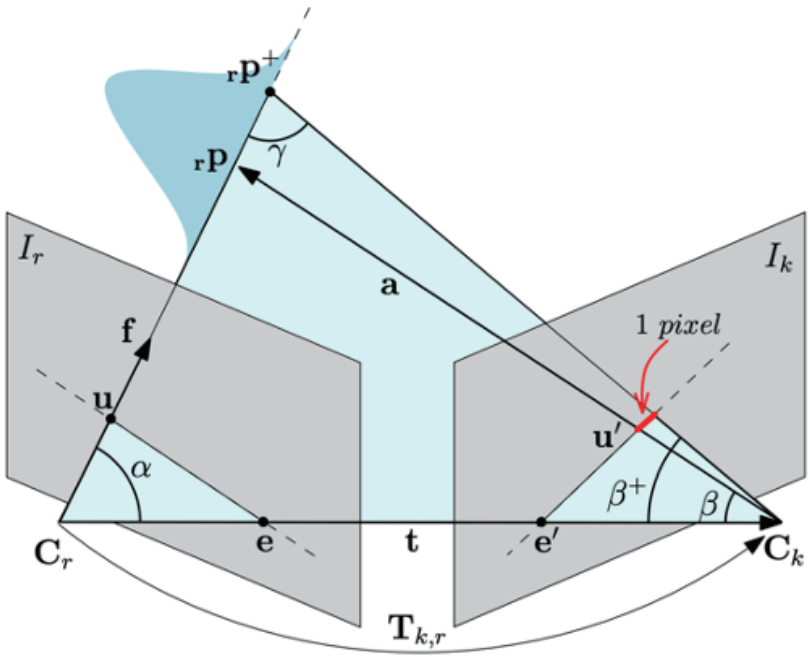


Fig. 5.5: The measurement uncertainty of REMODE [58]. Two images I_r and I_k are taken with different camera poses. The relative pose is $\boldsymbol{T}_{k,r}$. The measurement uncertainty is defined as the square of distance from ${}_r\boldsymbol{P}$ to ${}_r\boldsymbol{P}^+$. Through geometry, the measurement uncertainty can be computed.

As shown in Figure 5.5, the pixel $\boldsymbol{u}$ is a projection from the 3D point ${}_r\boldsymbol{P}$ in the reference image $\boldsymbol{I}_r$. The transformation $\boldsymbol{T}_{k,r}$ contains a rotation $\boldsymbol{R}$ and a translation $\boldsymbol{t}$. The $\boldsymbol{f}$ is a unit vector along the pixel $\boldsymbol{u}$'s projection direction defined as $\boldsymbol{f} =_r \boldsymbol{P}/(\|{}_r\boldsymbol{P}\|)$. Let f be the focal length of the camera.

With the following geometry equations [58] shown in Figure 5.5 and equation 5.24, the

measurement uncertainty can be computed.

$$\begin{cases} \alpha = {}_r\boldsymbol{P} - \boldsymbol{t} \\ \alpha = \arccos(\frac{\boldsymbol{f}\cdot\boldsymbol{t}}{\|\boldsymbol{t}\|}) \\ \beta = \arccos(-\frac{\boldsymbol{a}\cdot\boldsymbol{t}}{\|\boldsymbol{a}\|\cdot\|\boldsymbol{t}\|}) \\ \beta^+ = \beta + 2\tan^{-1}(\frac{1}{2f}) \\ \gamma = \pi - \alpha - \beta^+ \\ \left\|{}_r\boldsymbol{P}^+\right\| = \|\boldsymbol{t}\|\,\frac{sin\beta^+}{sin\gamma} \end{cases} \tag{5.25}$$

5.4.4 3D Reconstruction System

The pipeline which integrated SVO and REMODE is shown in Figure 5.6. The pipeline works in the 3D reconstruction stage which is required to generate a dense 3D point cloud for the crane operation stage.

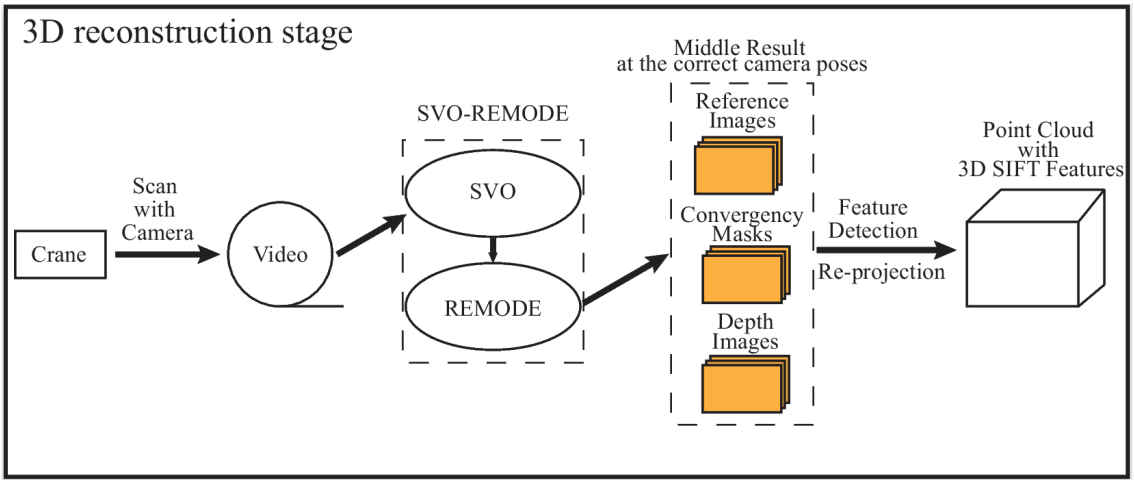


Fig. 5.6: The reconstruction pipeline

The reconstruction starts from the pre-shot of the crane's working environment. The crane boom mounted with a top-view camera on its head will rotate around the working environment and take a video. The video will be sent to SVO for camera pose estimation. Every frame in the video will have a stamped camera pose. Along with the stamped camera pose, each frame will be sent to REMODE for depth estimation. SVO-REMODE will finally give out some middle results. The middle results consist of reference images, convergency masks and depth images which are:

- Reference image: In REMODE, not every image will be estimated for depth. Only the reference image will be estimated for depth.
- Convergency image: For a reference image that is estimated for depth, not all the pixels' depth value are reliable. Only the convergent pixels' depth value is correct and can be used for reconstructing the 3D spatial map.
- Depth image: It contains the depth value for all the pixels of the reference image.

For the reference images, a process of detecting SIFT feature point will be made. If a

SIFT feature is in the convergency mask, it will be inversely projected from 2D to the spatial map. So the final result of the 3D reconstruction stage is a dense point cloud containing many 3D SIFT feature points.

5.4.5 Experiment

In order to evaluate SVO-REMODE pipeline for the 3D reconstruction of crane's working environment, evaluation is done for both the simulation videos and the field experiment video.

Simulation Experiment

Before the field experiment, it is better to perform the simulation experiment to evaluate the capabilities of the approach first. Therefore, the simulation experiment is conducted based on Gazebo [64]. Gazebo is a simulation software for robotics with many models and sensors which can be directly picked out to construct our virtual crane's working environment.

We have used a tower crane model. A textured simulation 3D working environment for a tower crane can be changed by replacing objects in the scene. A top-view camera is fixed beside the hook. The camera can take a video with an image size of 640×480 at the frame rate of 60 frames per second. In the simulation process, the crane will rotate around the environment to generate a simulated video. To make the captured image more like the real captured image, the foreground is in the range of the camera's projection area just as shown in Figure 5.7.

Two videos are recorded based on two different virtual construction environment of a tower crane. The top-view camera will rotate above the crane's working environment for 1.25 circle to ensure that the construction is a complete construction for each direction. The field of view angle is about 45 degrees. For all the two experiment, the camera is at a height of 28 meters and 21 meters far away from the crane's rotation center.

To test SVO-REMODE for different objects distributed in the crane's working environment, we have set up two virtual environments to record the videos and do the test. The environment setup is shown in Figure 5.8. The first virtual environment is shown in Figure 5.8a. Around the tower crane, there are four box objects with different lengths, width and height. And the illumination is set to a low level to test the reconstruction process's sensibility to illumination. The second virtual environment is shown in Figure 5.8b. There are three buildings in the environment with different textures and shapes. The environment illumination is set up to a higher level compared with the previous example.

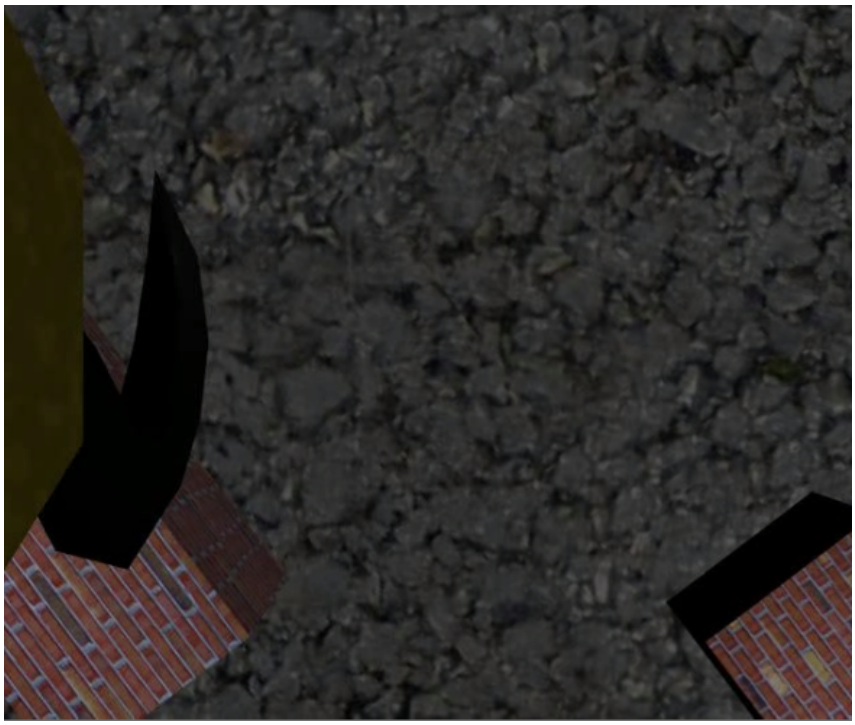


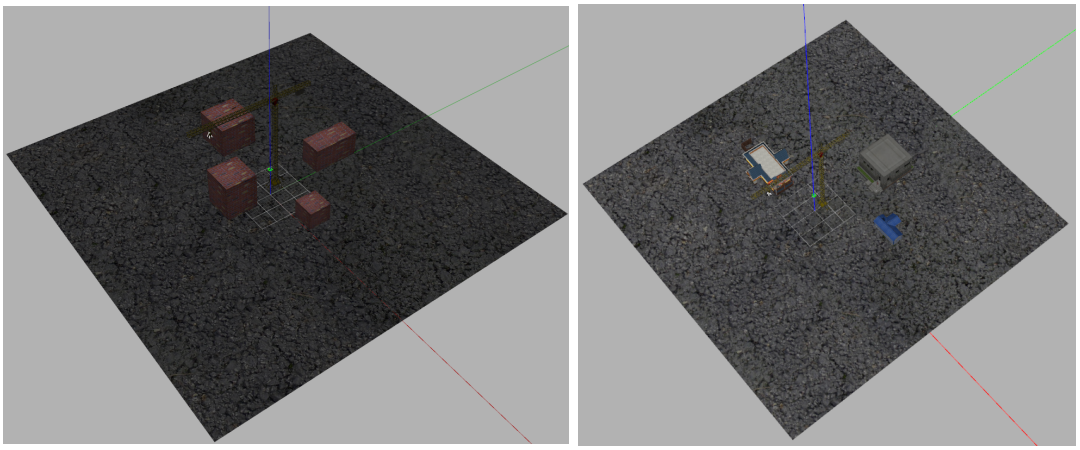
Fig. 5.7: A simulated sample image of a recorded video captured with the top-view camera

After the generation of simulated videos from our created virtual working environment for the tower crane, the reconstruction for the crane's working environment is tested. The reconstruction results are shown in Figure 5.9. The first result is shown in Figure 5.9a. There are 38 reference images which are selected from the simulated video. The 38 reference images along with their correspondence convergence images and depth images have recovered the scene shown in this figure. In the first simulation result, there are about 3.5 million 3D points used to describe the 3D spatial information under the tower crane. The second experiment is shown in Figure 5.9b. Totally 33 reference image has generated about 5.5 million 3D points to describe the working environment of the tower crane.

Field Experiment

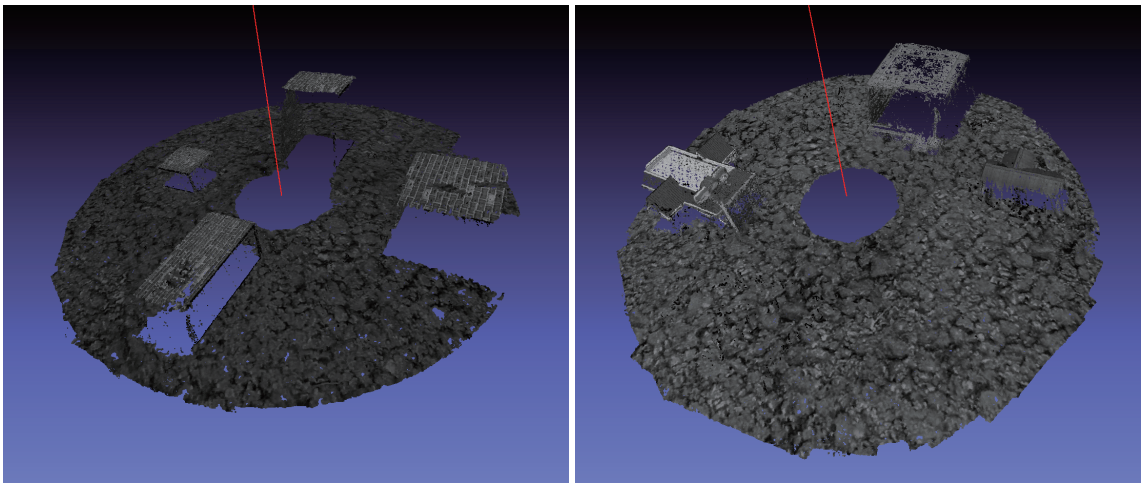
After the simulation experiment, we have a preliminary understanding of the 3D reconstruction for the crane's working environment with SVO-REMODE. Hence we have made the field experiment.

The videos are generated by a top-view camera mounted on the boom head of a mobile crane. The boom of the crane will rotate a circle to record the crane's working envi-



(a) The first simulation environment (b) The second simulation environment

Fig. 5.8: Two simulation environment are setup with Gazebo



(a) The first simulation (b) The second simulation

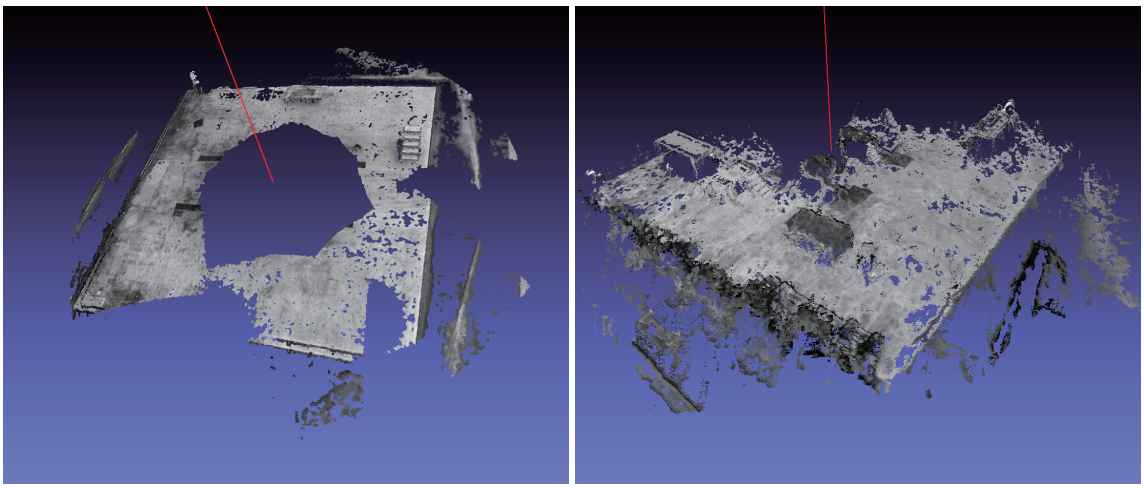
Fig. 5.9: 3D reconstruction with SVO-REMODE approach for simulation test

ronment. The first video is generated by a top-view camera which 15 meters above the ground. It moves a circle around the rotation axis above the ground. The second video is the third experiment video shown in Table 4.1.

The first field experiment is recorded by a PoE camera FLIR with a focal length of 24 mm. The resolution of the image is 1280×1024. The second experiment is using a camera named logicool c920R. The resolution of the camera is 1920×1280.

Even REMODE can be a real-time reconstruction for 3D spatial map. It can not handle large size image for real-time computation. For the second experiment which has a resolution of 1920×1280, it will be resized to one half of its original size which is 960×640.

The reconstruction result is shown in Figure 5.10. As shown in Figure 5.10a, the environment of the first scene is only a very flat ground with some very low objects. It is successfully reconstructed with 1.4 million 3D points to describe the crane's working environment. The second field experiment contains many higher objects in the crane's working environment. In this experiment, the crane's boom shadow has appeared in some regions. To eliminate its influence, the experiment is only conducted for about 75 percent of the crane's working environment. The result is shown in Figure 5.10b consisting of about 0.8 million 3D points.



(a) The first field experiment (b) The second field experiment

Fig. 5.10: 3D reconstruction with SVO-REMODE approach for field experiment

5.4.6 Summary

In this section, we have introduced the selection and details of the 3D spatial map reconstruction approach SVO and REMODE. With SVO and REMODE, a real-time reconstruction for the crane's working environment can be made in real-time. The working processes and principles of SVO and REMODE have been introduced and explained in a very detailed way.

In the last, the experiments have been done to test the capabilities of SVO and REMODE to reconstruct the crane's working environment. From the experiment results, the reconstruction efficiency for computation can be real-time. For the reconstructed 3D spatial maps, they contain millions of 3D points which can be considered as a dense reconstruction for the crane's working environment. The dense reconstruction is considered as a benefit for applications. For the objectives of reconstruction speed and reconstruction density, the experiments perform:

- Reconstruction speed: both for the two different image sizes which are 640×480 and 960×640, the reconstruction pipeline can achieve the objective of real-time 3D reconstruction.
- Reconstruction density: the reconstructed 3D spatial map contains millions of 3D points which make the 3D spatial map applicable for the objective of further applications.

In later sections of this Chapter, we will show the working area limit line using the dense reconstruction result. Furthermore, an error analysis will be made in later sections to verify the reconstruction error of SVO and REMODE.

5.5 Working Area Limit Line Display

To augment the information of the working area limit line on the top-view image shown in the cabin's display, two steps need to be conducted. Firstly, the working area limit line should be reflected in the densely reconstructed 3D environment generated with SVO-REMODE. To achieve the purpose, the rotation center of the crane in the reconstructed 3D environment should be computed first. And then by checking every 3D point's distance to the rotation axis, the working area limit line can be drawn in the reconstructed 3D environment. Secondly, after the working area limit line being successfully drawn in the reconstructed 3D environment, it should be accurately projected to the top-view image shown in the cabin's display. This process needs an accurate estimation of the camera pose.

As mentioned in the objectives for the final error for displaying the working area limit line in the top-view image, for about 1000 pixels, it is better to require the error to be within 10 pixels which is about one percent error in displaying the working area limit line.

5.5.1 Working Area Limit Line for 3D Spatial Map

After the reconstruction of the crane's working environment, a dense 3D point cloud will be obtained. To show the working area limit line in the image captured with the top-view camera, firstly, it should be drawn to the point cloud. The working area limit line can be different such as a circle, an ellipse, or some other more complicated shapes. In most cases, the working area limit line will be a circle, especially in the case of the tower crane, as the working area limit in any direction for a tower crane should be the same. To draw the circle in the reconstructed 3D point cloud, we just need to put the center of the circle to the rotation center of the crane. So first, the rotation center in the 3D point cloud should be computed.

After finding the rotation center, the working area limit line can be represented in the point cloud. To draw the working area limit line in the image of the cabin display of a top-view camera, a projection from the 3D point cloud to the image is required. To make this projection accurately, the camera pose for the image in the 3D point cloud should be estimated with enough precision.

Rotation Axis Estimation

As mentioned above, in order to draw the working area limit line in the point cloud, the rotation center should be figured out first. It can be achieved by a circle fitting with the camera positions estimated from SVO. In the reconstruction process of SVO, the camera poses for all the images can be estimated with high precision. To make the estimation more robust, the circle fitting is made with the RANSAC [30] algorithm.

As in our application, the camera is a downward-looking camera. It means that only with a 2D circle fitting with x and y coordinate of camera position, the rotation axis can be estimated. With the computed camera positions noted as $C_i(x_i, y_i)$ for $i = 1, 2, \cdots, n$, a circle fitting with RANSAC will be made to figure out the rotation center in the 3D point cloud. RANSAC is a procedure in the estimation of parameters robustly in many applications. In the data samples which contain some wrong data, RANSAC can always separate the inliers and outliers. The outliers are the wrong data. For a robust estimation, RANSAC always goes before the least square estimation. The removal of wrong data can significantly improve the precision of the least square estimation.

To define a circle, three points are required. With RANSAC, we only need to repeat the following process t times.

- Pick out three points from the $C_i(x_i, y_i)$ for $i = 1, 2, \cdots, n$ randomly.
- Compute the circle by the picked 3 points.
- Check the inliers and record the inliers' count.

The circle model computed with the three points which have the most inliers is the correct circle estimated. And then the probability of finding the correct circle after t times is

$$p(Circle\ is\ correct) = 1 - (1 - (p_i)^r)^t \tag{5.26}$$

where p_i is the ratio of inliers and $r = 3$ for our circle fitting.

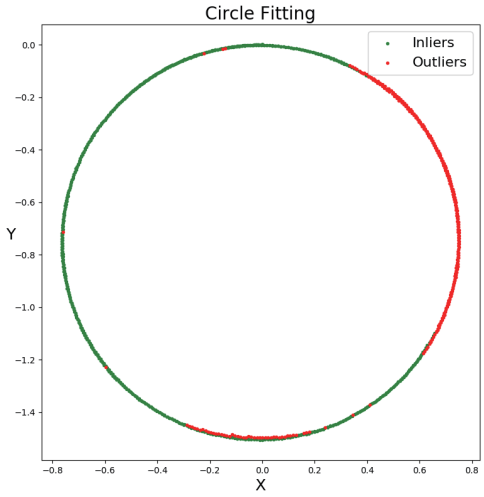
Figure 5.11 shows the circle fitting results of the simulated video and field experiment video. The red points are the outliers and the green points are the inliers. Figure 5.11a and 5.11c shows the path of the simulated videos. The circle is almost perfectly matched with the estimated circle model as shown. Figure 5.11c shows the result of the first field test.

After the boom rotates two circles around the working environment, the camera keeps a drifting error. This is because that SVO lacks a global optimization on the camera pose. The later estimated pose will have a drifting error. The drifting error after two circles becomes a little bit obvious, but the estimation for the circle is still precise enough. It is precise because the initial part of the path which keeps only very little drifting error is considered the inlier. The last one shown in Figure 5.11d is not a complete two circle rotation. It is a more complicated movement with boom rotation and pitching. There is about one-fourth missing as there is boom shadow in the region which will influence the result a lot.

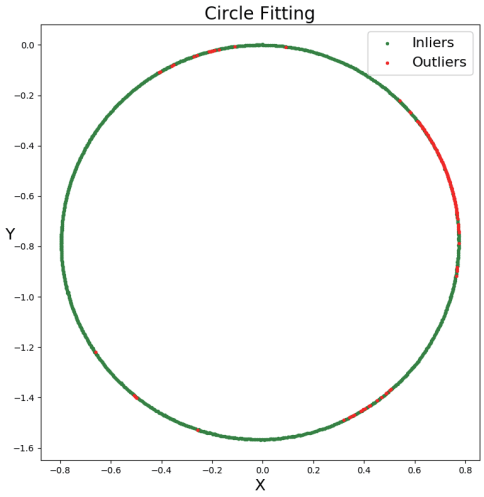
Working Area Limit Line Drawing

After the crane rotation axis has been successfully estimated, the working area limit line can be drawn in the reconstructed point cloud. This objective can be easily achieved by checking every point's distance to the rotation axis. The working area limit at rotation angle θ is d_l. The 3D point in the direction θ is noted as $P(x, y, z)$ with the rotation axis estimated as $l_r(x_0, y_0, m)$. The distance from P to l_0 can be computed with $d = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. To show the working area limit line clear, a threshold d_t is set. If $|d - d_l| < d_t$, the point will be marked as a point on the working area limit line.

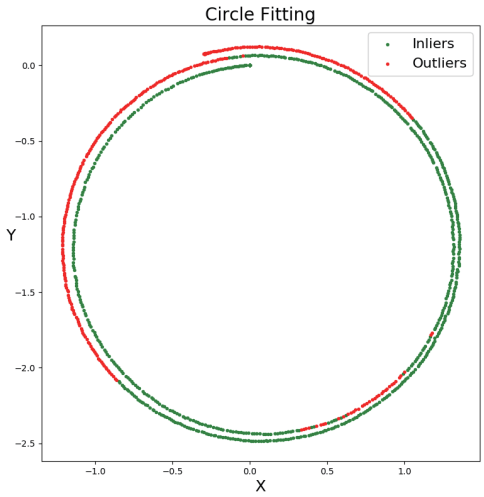
The circle is a commonly used working area limit line for tower crane and the rough crane with four outriggers fully extended. The experiment results are shown in Figure 5.12. The four results are correspondent to the results shown in Figure 5.11 respectively.



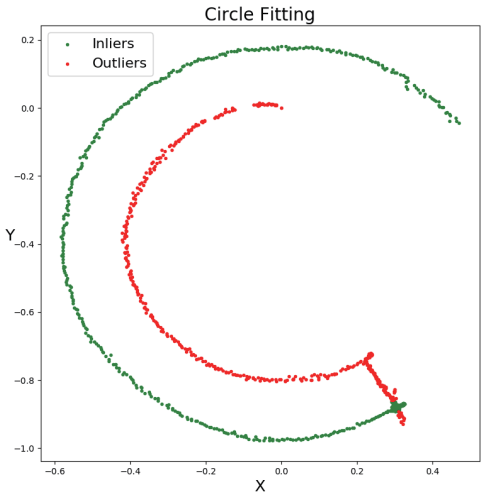
(a) The first simulation



(b) The second simulation

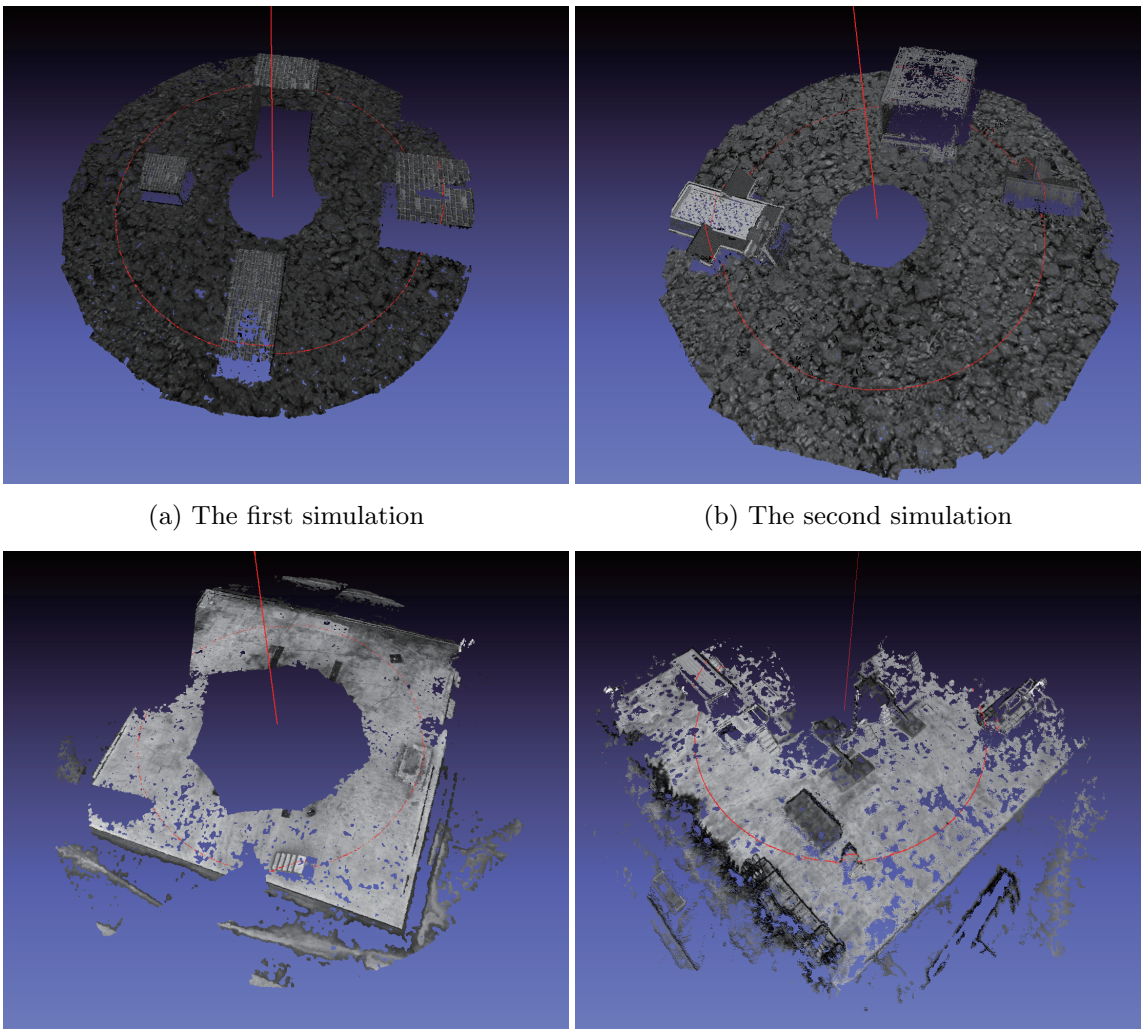


(c) The first field test



(d) The second field test

Fig. 5.11: Rotation center estimation result throught circle fitting



(a) The first simulation (b) The second simulation

(c) The first field test (d) The second field test

Fig. 5.12: Working area limit line drawing experiment.

5.5.2 Camera Pose Estimation and Projection

Even SVO can estimate the camera poses. There is a disadvantage in using it. The error drifting will be very obvious after a long-time estimation with SVO. It can be seen from Figure 5.11c.

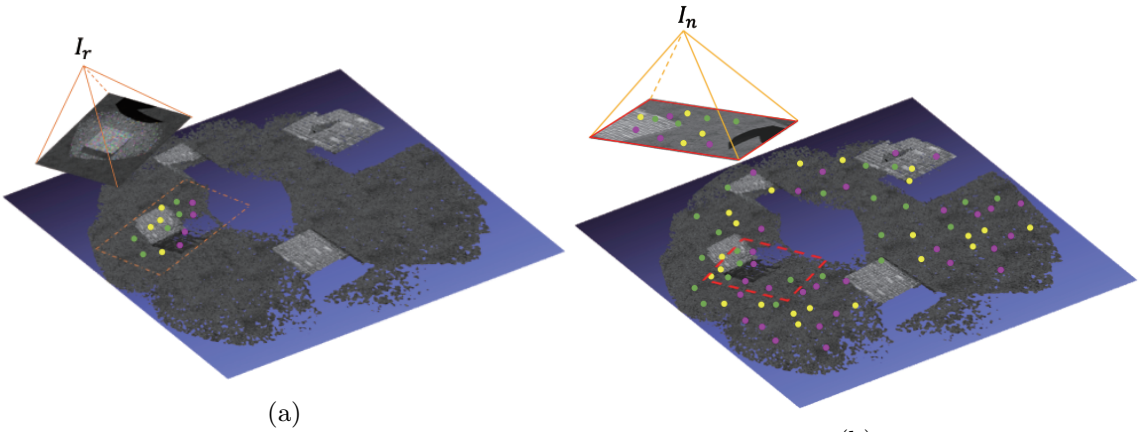
In reconstruction, this effect is not obvious and good reconstruction results can be generated by REMODE with the poses provided by SVO. To make the location more reliable, PNP (Perspective N Points) with RANSAC shall be applied to achieve the objective.

From SVO-REMODE pipeline, the middle results are reference frames, depth images, convergence images and camera poses. The final 3D point cloud with intensity is generated with these middle results.

For a new image captured with the top-view camera, it is necessary to find its camera pose in the reconstructed 3D point cloud. Then the working area limit can be augmented to the image. PNP needs to know at least 3 matches between 3D points and 2D points. To find out the matches required, in the reconstruction process from the middle result to the final result, we will use 3D SIFT features. For traditional SIFT, there is a feature point consist of the feature's pixel location and a 128-dimensional vector descriptor. The descriptor is used for matching, and the location is used for computing the camera pose in PNP.

In the reconstruction process, the reference frames with their related depth image and convergency image are used. Figure 5.13a is showing how a reference frame is used for reconstruction. As shown in the reference image I_r, the SIFT feature point which keeps a converged depth value will be re-projected to the reconstructed 3D environment. The reference image is a projection on the rectangle area in the reconstructed result. And in the reconstruction process of the reference image, the SIFT features shall be detected along with the reconstruction. If a SIFT feature is in the converged area of the reference image, it will be used to generate a 3D SIFT feature in the reconstructed result. Just as shown in the rectangle area, there are many points with color. They are the 3D SIFT features in the reconstructed 3D environment.

By reconstruction with all the reference images, the 3D environment with 3D SIFT features can be generated. Figure 5.13b shows how a new image's camera pose is estimated.



(a) (b)

Fig. 5.13: The reconstruction with 3D SIFT features and camera pose estimation with RANSAC PNP. (a) A reference image with 2D SIFT features is re-projected to the environment map. (b) By matching image feature points with all the feature points in the reconstructed environment, with PNP and RANSAC, the camera pose can be estimated.

When a new image I_n captured with the top view camera comes, the SIFT features on it will be detected. The rectangle area in the reconstructed 3D environment should be the area that a camera looking at. The colored points on the image I_n are a schematic representation of the 2D SIFT features. By matching the 2D SIFT features on the image I_n and the 3D SIFT features in the reconstructed 3D environment, there will be many points correspondence from 3D to 2D, here noted as $X_i \leftrightarrow x_i$ for $i = 1, 2, \cdots, n$, where X_i is the 3D points and x_i is the image points. As shown in Figure 5.13b, the colored points are a schematic representative of 3D SIFT features in the reconstructed environment. The math definition of PNP is a projection from world points to image points with the equation below.

$$sx_i = K[R|t]X_i \tag{5.27}$$

where:

- s is a scale factor for homogrphy.
- x_i is the 2D homography representation of pixel coordinate.
- K is 3 by 3 matrix representing the camera intrinsic parameters.
- R is a 3 by 3 rotation matrix.
- t is a 3 by 1 translation vector.

- X_i is the 3D homography representation of world points.

With enough matched correspondent pairs $X_i \leftrightarrow x_i$, by using DLT (Direct Linear Transformation) [36] method with RANSAC [30], the robust estimation on R and t can be made.

After the camera pose is successfully estimated, the working area limit line in the 3D spatial map can be projected to the image.

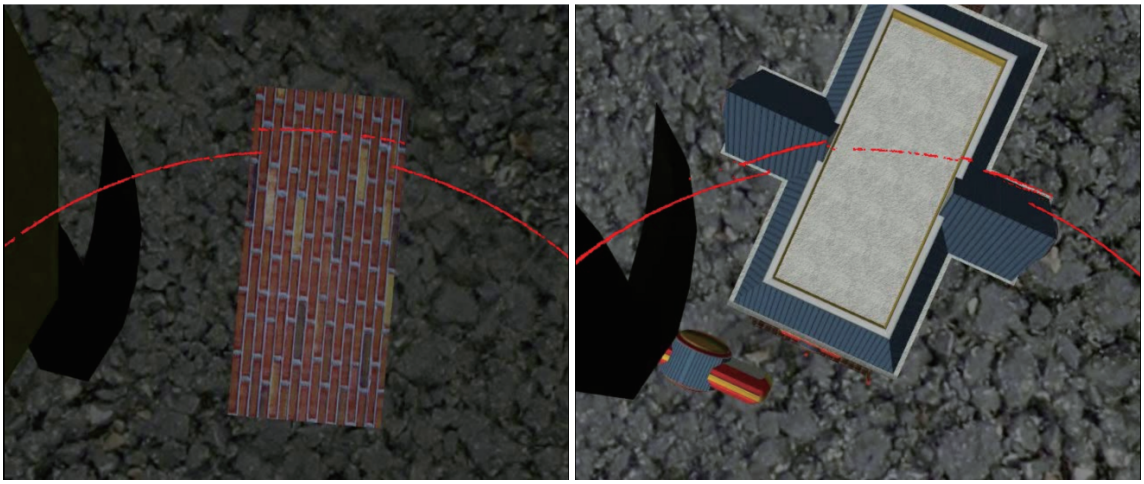
5.5.3 Experiment

After clarifying all the requirements and solutions to draw the working area limit line in the image captured with a top-view camera, three experiments have been conducted. To show the obvious effect of the high object, the field experiment with a flat ground is not tested in showing the working area limit line in the image.

Figure 5.14 is showing the three results. Figure 5.14a and 5.14b is for a simulated image. As shown in the simulation experiment result, the camera is inside the safe area. Under such a camera pose, when the working area limit line going across a higher object, the working area limit line will not be continuous. The safe area is affected by the height information on the ground.

The experiment of a field recorded video has the same result as the simulated result. As the object's height in the field experiment is not as obvious as in the virtual environment, this effect in the field experiment result can just be seen not so obviously.

From the experiment results, it can be clearly seen that the working area limit line is affected by the object on the ground. For the experiment of simulation, when the working area limit going across the box object, the circle on the ground is not continuous with the working area limit arc line on the box object. Especially in the second experiment of the simulation shown in Figure 5.14b, the building's roof is not a flat roof. When the working area limit line gets across the roof from right to left. The height above the ground for the building becomes higher. Then the working area limit line extends more and more from left to right. Through the three experiments, the influence can be obviously identified. The working range of the crane can be recognized according to the different height information and camera position.



(a) (b)

(c)

Fig. 5.14: Working area limit line display. (a) and (b) are the simulated experiment. (c) is a field experiment result.

5.6 Evaluation

After the experiment of 3D spatial map reconstruction with SVO-REMODE and working area limit line displaying, an error analysis is necessary. In this section, the reconstruction error and the projection error of the working area limit line are analyzed.

5.6.1 Reconstruction Error Analysis

In section 5.4, the reconstruction of the 3D spatial map with SVO-REMODE is explained in detail. The experiments have also been conducted for both the simulated video and field test video. To use the 3D spatial map reconstructed with SVO-REMODE, the reconstruction error should be evaluated first.

The evaluation of reconstruction error is conducted by checking the reconstruction quality of the ground plane. In the experiment, the ground plane contains most of the 3D points. To achieve the purpose, we need to do:

- Segment the ground plane by RANSAC plane fitting with a wide threshold. The reconstruction contains many side surfaces of tall objects. With a wide threshold, both the partial side surfaces and ground plane's reconstruction noises will be included in the data used for estimating the reconstruction error. The side surfaces will make the error bigger than it is supposed to be.
- Estimate the ground plane by the least square. The estimated ground plane with the least-squares will be used for figuring the reconstruction error.
- Find the maximum error with pauta criterion.

RANSAC Plane Fitting

Through RANSAC plane fitting, the ground plane with noises and partial side surfaces of the objects will be segmented out from the reconstructed 3D spatial map. The point cloud totally containing n 3D points can be noted as a collection shown in equation 5.28.

$$\mathbf{P}_c = \{\boldsymbol{P}_i | i = 1, 2, \cdots, n\} \tag{5.28}$$

A plane can be confirmed with three points which are not lying on the same line. From $\mathbf{P}_c$, generally through a random selection of three points $\boldsymbol{p}_o$, $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$, a plane can be figured out with equation 5.29, where $\boldsymbol{p}_x$ is the point in the plane spanned by $\overrightarrow{\boldsymbol{p}_o\boldsymbol{p}_a}$ and $\overrightarrow{\boldsymbol{p}_o\boldsymbol{p}_b}$.

$$(\overrightarrow{\boldsymbol{p}_o\boldsymbol{p}_a} \times \overrightarrow{\boldsymbol{p}_o\boldsymbol{p}_b}) \times \overrightarrow{\boldsymbol{p}_o\boldsymbol{p}_x} = 0 \tag{5.29}$$

Equation 5.29 can be rewritten in coefficient form as:

$$c_1 x + c_2 y + c_3 z + c_4 = 0 \tag{5.30}$$

Then, for the point $\boldsymbol{p}_i(x_i, y_i, z_i) \in \mathbf{P}_c$, its distance can be estimated with equation 5.31.

$$d_i = \frac{|c_1 x_i + c_2 y_i + c_3 y_3 + c_4|}{\sqrt{c_1^2 + c_2^2 + c_3^2}} \tag{5.31}$$

The estimated distance d_i will be compared with a threshold d_s with equation 5.32 to judge whether the current point is a inlier point or outlier point.

$$\begin{cases} inlier & if \quad d_i < d_s \\ \\ outlier & if \quad d_i > d_s \end{cases} \tag{5.32}$$

The random point selection of $\boldsymbol{p}_o$, $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ will be made many times. The inlier point count will be recorded every time. Finally the plane equation which can have the maximum inlier point count will be the plane required. And all the inlier point will be used for error analysis. The inlier point can be noted with equation 5.33.

$$\mathbf{P}_{plane} = \{\boldsymbol{p}_i | i = 1, 2, \cdots, m\} \tag{5.33}$$

Least Square Plane Fitting

With a rough plane fitting with RANSAC, the inlier points can be successfully selected out. A precise plane can be figured out with the least square estimation. The coefficients which keep the least squares The plane with the least squares can be assumed as:

$$b_1 x + b_2 y + z + b_3 = 0 \tag{5.34}$$

Then the least squares can be expressed as:

$$S = \sum_{i=1}^{m} (b_1 x_i + b_2 y_i + z_i + b_3)^2 \tag{5.35}$$

To solve this least square problem, we only need to solve the linear equations 5.36.

$$\begin{cases} \frac{\partial S}{\partial b_1} = 0 \\ \\ \frac{\partial S}{\partial b_2} = 0 \\ \\ \frac{\partial S}{\partial b_3} = 0 \end{cases} \Leftrightarrow \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & \sum m \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix} \tag{5.36}$$

Maximum Reconstruction Error

After obtaining the least square plane, the error analysis for maximum reconstruction error can be achieved. For all the points $\mathbf{P}_{plane}$, their distance to the least square plane can be computed as a collection shown in equation 5.37.

$$\mathbf{D} = \{d_i = \frac{|b_1 x_i + b_2 y_i + z_i + b_3|}{\sqrt{b_1^2 + b_2^2 + 1}} \mid \boldsymbol{p}_i(x_i, y_i, z_i) \in \mathbf{P}_{plane}\} \tag{5.37}$$

The mean error $\bar{d}$ and then the error σ can be estimated with equation 5.38. Finally, the maximum reconstruction error will be 3σ.

$$\begin{cases} \bar{d} = \frac{1}{m} \sum d_i \\ \sigma = \sqrt{\frac{1}{m-1} \sum (d_i - \bar{d})^2} \end{cases} \tag{5.38}$$

The four reconstructed 3D spatial maps are used for the evaluation of reconstruction error. Finally, the two 3D spatial maps reconstructed with two simulated videos have a maximum error of 0.3386 and 0.3767 meters. While for the two field experiments, the maximum reconstruction error is 0.1503 meters for the experiment with flat ground and 0.3433 meters for the experiment with some tall objects on the ground. The reconstruction error is less than our objective 0.5 meters. The error comes mainly from the method itself. As known, the VSLAM approaches will keep a significant reconstruction error than the depth-sensor based approaches. Every step in a VSLAM approach will yield errors. One example is using matched features to solve the relative pose of two image. It is a solution of the least squares for an over determined linear system. In SVO and REMODE pipeline, the reconstruction of the final 3D spatial map is using the camera pose from SVO and the depth image from REMODE. And in estimation of the depth image, REMODE also takes the camera poses from SVO. So the error of camera pose from SVO affect the result more than REMODE. To have a more precise estimation of the camera pose, it is a promising way to have a bundle adjustment locally and globally. A local estimation will ensure that REMODE takes a more precise camera pose locally. As REMODE estimates the depth image locally, it will make the depth estimation more precise. A global optimization with bundle adjustment can make the drifting error smaller which will make the final alignment more precise. And then it can reduce the reconstruction error.

5.6.2 Camera Position Estimation Error Analysis

For the camera pose estimation to project the working area limit line from 3D to 2D, there will be an error. The estimation error for camera pose and reconstruction error will influence the projection result, namely, the location of the working area limit line in the image of the cabin display. The error analysis will only consider the accuracy of the camera position which is a 3 degrees of freedom translations.

In the estimation for the camera pose with PNP-RANSAC, the position of the camera can be estimated as:

$$\mathbf{C} = \{\boldsymbol{c}_i(x_i, y_i, z_i) \mid i = 1, 2, \cdots, n\} \tag{5.39}$$

There is a ground truth for the trajectory of camera pose which is a circle. The estimated camera position should be close to the circle except for some wrong estimations. The center of the circle is estimated in section 5.5.1 which can be expressed with the rotation axis $\boldsymbol{c}(c_x, c_y, z)$ and the radius r. If the camera position is correctly estimated, it should be in the range from $r - d_t$ to $r + d_t$ as shown in equation 5.40.

$$\|\boldsymbol{c}_i - \boldsymbol{c}\| < d_t \tag{5.40}$$

For the camera positions in $\boldsymbol{C}$ meeting the condition shown in equation 5.40, they will be considered as the inlier camera positions which can be used for the estimation of the error:

$$\mathbf{C}_{inlier} = \{\boldsymbol{c}_i(x_i, y_i, z_i) \mid i = 1, 2, \cdots, m\} \tag{5.41}$$

The camera position has three components.Obviously, with only the trajectory in 2D, the error can only be roughly estimated for x direction and y direction of the camera position. Assuming that the movement all happens in y direction (radial direction), we can roughly estimate the error with $\sqrt{3}$ of the maximum error happening in the radial direction. Still by the evaluation of the mean value $\bar{r}$ and the error σ for the error in radial direction, the process is shown in equation 5.42.

$$\begin{cases} \boldsymbol{R} = \{r_i = \|\boldsymbol{c}_i - \boldsymbol{c}\| \mid \boldsymbol{c}_i \in \boldsymbol{C}_{inlier}\} \\ \bar{r} = \frac{1}{m}\sum r_i, \quad r_i \in \boldsymbol{R} \\ \sigma = \sqrt{\frac{1}{m-1}\sum (r_i - \bar{r})^2}, \quad r_i \in \boldsymbol{R} \end{cases} \tag{5.42}$$

The final maximum error in one direction is considered as 3σ. For the two simulation tests, the maximum error for camera position estimation is 2.9 mm and 1.2mm. The two field tests have a maximum error of 24.6mm and 25.0mm. The maximum error for the field experiment is far more obvious than the simulation experiment. The error analysis is only made by estimating the camera position. To improve the precision for camera position, the bundle adjustment can be applied by minimizing the reprojection error from 3D to 2D.

5.6.3 Projection Error

The error in the projection process of the 3D working area limit line to the image plane of the cabin display should both consider the reconstruction error and camera position error. The maximum reconstruction error and position error have already been estimated as e_r and e_p. The projection error can be defined as equation 5.43 at the camera coordinate frame.

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & e_p \\ 0 & 1 & 0 & e_p \\ 0 & 0 & 1 & e_p \end{bmatrix} \begin{bmatrix} x \\ y \\ z + e_r \\ 1 \end{bmatrix} \tag{5.43}$$

From the equation, the range for the projected coordinate (u, v) can be estimated. And for the ground point, the value z is the height of the camera above the ground. The influence can be expressed with:

$$e_{proj} = \frac{fd}{z + e_r + e_p} + \frac{fe_p}{z + e_r + e_p} - \frac{fd}{z} \tag{5.44}$$

where d is the distance of the 3D point to the optical axis of the camera. The maximum error will be reached which the e_r is negative and e_p is positive. For the simulation experiment, the projection error is 2.91 and 3.88 pixels. And for the field experiment, the projection error is 3.00 and 3.99 pixels respectively. The final term of projection error is a result affected by both the reconstruction error and the camera position error. However, the final projection error should consider the camera pose error, not only the position of the camera. In the future, the error analysis needs to also consider the 3 degrees of freedom for rotations.

5.7 Summary

In this chapter, an approach for displaying the working area limit line based on a real-time 3D spatial map reconstruction is developed. The approach has two stages, i.e., the 3D spatial map reconstruction stage and the working area limit line displaying stage.

In the 3D spatial map reconstruction stage, through a pre-scanning with a top-view camera around the crane's working environment, SVO-REMODE can achieve a real-time 3D reconstruction for the crane's working environment. Both the simulation experiment and field experiments have been conducted to test the efficiency and reconstruction quality. The reconstructed 3D spatial map contains millions of 3D points which can have a very detailed description of the crane's working environment. An error analysis in the last is made to verify the reconstruction quality. By analyzing the reconstruction error for the ground plane, the reconstruction error is finally figured out. The maximum reconstruction error is about 0.38 meters.

In the working area limit line displaying stage, the working area limit line is successfully projected from the 3D spatial map to the image captured with a top-view camera. After the reconstruction of the 3D spatial map with SVO-REMODE, by circle fitting, the rotation axis of the crane in the 3D spatial map is figured out. And following that, the working area limit line can be presented in the 3D spatial map. For a new image captured with the top-view camera, its camera pose against the 3D spatial map is estimated. After that, the working area limit line in the 3D spatial map is projected to the image with the estimated camera pose. Finally, error analysis for estimating the camera position error and projection error is made. From the error analysis, we can know that in the last the projected point on the working area limit line can reach a maximum of 3.99 pixels which is accurate enough for application.

Chapter 6

Conclusion and Future Work

6.1 Thesis Summary

Cranes are widely used and expected to play one of the vital roles in all kinds of construction projects. For the crane operated by a human sitting in the small cabin, the limited visual information is a problem encountered by the crane operator. Desirable visual assistance which contains intuitive visual information is necessary for helping the crane operator.

Chapter 1 has clarified background and challenges first. For a crane operator sitting in a small cabin, the limited visual information increases the difficulties for the crane operation. Following that, the objectives and approaches are discussed in detail. Two image generation systems are developed to assist crane operation for initial construction stage and middle/last construction stage respectively. For different construction stages, the crane operator needs different visual image assistances:

- In the initial construction stage, the crane's working environment is with little height variance caused by some low objects. The lifting under this stage is easier than the later stage. Compared to a local view provided by the top-view camera which can only provide the crane operator a limited vision, the overall 2D workspace map for the crane operator is more promising. By providing the 2D workspace map and finding the current boom head location on the 2D workspace map, the crane operator can have good guidance with the location on the 2D workspace map.
- In the middle/last construction stage, as the height variance gets very obvious, some height-related information is important to the crane operator. To help display such information, the system to provide the height-related information in the top-view

image is developed. In this thesis, one of the most important information working area limit line is shown in the top-view image.

In Chapter 2, the fundamentals including many concepts and technologies for understanding this thesis is explained.

Chapter 3 and Chapter 4 describe the 2D workspace map generation system which can provide a clear 2D workspace map and location for the initial construction stage. As the image captured with a top-view camera always contains the foreground which mainly consists of a hook and a boom head, the automatic image stitching to generate a panoramic can only make a panoramic with many ghosts. So first in Chapter 3, we proposed a new approach to detect the foreground. Many experiments are made to show its effectiveness in the detection and removal of the foreground. The foreground detection is a vital component for generating a clear 2D workspace map. Chapter 4 has introduced and explained the 2D workspace map generation pipeline and its application. It contains two stages in the system. In the preprocessing stage, the keyframes and support frames in a video will be selected out. With the foreground detection and image stitching, a clear 2D workspace map can be obtained. In the crane operation stage, the location and path of boom head can be estimated and shown on the 2D workspace map which can provide the crane operator visual information. Two experiments have been done and the results show that the system can successfully give a clear 2D workspace map and find the location of boom head on the 2D workspace map. And in the last, the error has been estimated to ensure the system's applicability.

Chapter 5 is the system for the middle and last construction stage. It can provide one of the most important height-related information working area limit lines in the captured top-view image. The system is also a two stages system consisting of a preprocessing stage and a crane operation stage:

- Preprocessing stage: In this stage, the 3D spatial map is reconstructed with VSLAM. We have chosen SVO and REMODE to reconstruct a dense and precise 3D point cloud as the 3D spatial map. In the reconstruction process, the features have been detected and also recovered to the 3D point cloud. The reconstructed 3D spatial map is a dense 3D point cloud containing partial 2D descriptors which will

be used in the next stage.

- Crane operation stage: In the crane operation stage, first the working area limit line will be drawn in the 3D spatial map by fitting the trajectory of the camera and distance testing. Then for a new image captured by the top-view camera, it will be matched to the dense 3D point cloud and the camera pose for the image will be estimated. In the last, the working area limit line in the 3D spatial map can be projected and displayed on the top-view image.

Experiments have been conducted to test the reconstruction for the 3D spatial map and displaying the working area limit line in the top-view image. Finally, the error analysis is made to evaluate the reconstruction error and final projection error to ensure its precision and applicability for assisting the crane operation.

6.2 Future Work

The foreground detection to generate a clear workspace map is a little bit time-consuming. There is a balance in time consuming and accuracy. The masks are searched by comparison a keyframe to many support frames. As shown in our test in Figure 4.6d, the shadow on the ground is not detected out. It is because the speed of the shadow is very slow from the keyframe to its support frames. Even the motion of optical flow and 2D homography is completely opposite, the detection result is not good for the shadow. The crane has to work under the sun and the shadow caused by the crane itself can not be avoided. The method should be improved to have detection on such shadows to give a better 2D workspace map.

In the 3D reconstruction with SVO-REMODE pipeline, SVO yields accumulation error which is called drifting error. Every later pose of the camera is estimated with the front keyframe in SVO. The REMODE pipeline can estimate the depth for pixels with the poses provided by SVO. As the locally relative poses of SVO is accurate, the point cloud generated by REMODE is accurate. But in the final registration, the drifting error caused by SVO will be made the result not so good as we want. To have a better registration of the final point cloud, a global optimization on the camera poses for reconstruction is required. It can be achieved with the bundle adjustment. As shown in Figure 5.12c, the drifting error of camera pose makes the registration worse than the other result shown in Figure 5.12.

In chapter 5, we have only introduced and implemented one of the most important height-related information working area limit line. Actually, there are many other important height-related information and assisting approaches can be developed with the reconstructed 3D spatial map and rotation axis.

Acknowledgement

I have been here in Japan for almost 4 years. I want to thank many people for their kindly help with my study and life here.

First and foremost, I would like to thank my supervisor **Prof.SUZUKI Hiromasa**. Department of Precision Engineering, The University of Tokyo. I can receive his sincere advice and study guidance almost at every weekly meeting. His extremely hardworking is not only on doing research, but also supervise his students in doing research. I will never forget the opportunity and suggestions he gave me which will be invaluable in my future life. I would extend my utmost gratitude to him.

I would also like to express my sincere gratitude to **Prof.OHTAKE Yutaka**. He has given much valuable advice and help in my study and life.

Moreover, I would like to extend my gratitude to **Dr.Noguchi Shinji**, **Mr. UNTEN Hiroki**, **Mr.KOSAKA Takayuki**, **Dr.YONEDA Mizuki** from TADANO LTD.. They have given us a lot of advice from the application's view and provide us the support for the field experiments. All the experiments are set up and conducted under the discussion with them.

Thanks for **Prof.OTA Jun**, **Prof.NAGATANI Keiji**, **Prof.YAMASHITA Atsushi** and **Prof.MAEDA Yusuke**. Thank you for reviewing my thesis manuscript and valuable advice.

Thanks to **Mrs.TSUJIGUCHI** who makes me feel the Lab like home sometimes. As a Japanese beginner, many documents are under the help of her.

My sincere thanks also go to **Prof.NAGAI**, **Prof.YATAGAWA**, **Mr.KATAYAMA**, **Dr.Kemal**, **Dr.Li**, **Mr.TAKAHASHI**, **Mr.WATANABE**, **Mr.LIU**, **Mr.TANG** and **all the other lab members**.

I would also like to my motherland China for the CSC scholarship and care under the COVID-19.

Finally, I would like to express my thanks to my parents. I can feel their understanding of the phone call every time. Their love has filled my life.

Bibliography

[1] Automatic moment limiter (aml-c).

[2] Motilal Agrawal, Kurt Konolige, and Luca Iocchi. Real-time detection of independent motion using stereo. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, volume 2, pages 207–214. IEEE, 2005.

[3] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *European Conference on Computer Vision*, pages 214–227. Springer, 2012.

[4] Hirokazu Araya, Makoto Kakuzen, Hideki Kinugawa, and Tatsuo Arai. Level luffing control system for crawler cranes. *Automation in construction*, 13(5):689–697, 2004.

[5] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. In *Readings in computer vision*, pages 714–725. Elsevier, 1987.

[6] E Bargazov, T Uzunov, O Alipiev, S Antonov, and D Bortyakov. New structure of the gantry cranes level luffing jib system. *Machines. Technologies. Materials.*, 12(1):3–7, 2018.

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[8] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[9] Alexander Braun, Sebastian Tuttas, André Borrmann, and Uwe Stilla. Automated progress monitoring based on photogrammetric point clouds and precedence relationship graphs. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 32, page 1. IAARC Publications, 2015.

[10] Xavier Bresson, Selim Esedolu, Pierre Vandergheynst, Jean-Philippe Thiran, and Stanley Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and vision*, 28(2):151–167, 2007.

[11] M Brown and DG Lowe. Recognising panoramas. computer vision, 2003. In *Proceed-*

ings. Ninth IEEE International Conference on, volume 2, page 1, 2003.

[12] Matthew Brown and David G Lowe. Invariant features from interest point groups. In *BMVC*, volume 4, 2002.

[13] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[14] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International journal of computer vision*, 61(3):211–231, 2005.

[15] Peter J Burt and Edward H Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, 2(4):217–236, 1983.

[16] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[17] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[18] Che-Han Chang, Yoichi Sato, and Yung-Yu Chuang. Shape-preserving half-projective warps for image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3254–3261, 2014.

[19] Michael M Chang, A Murat Tekalp, and M Ibrahim Sezan. Simultaneous motion estimation and segmentation. *IEEE transactions on image processing*, 6(9):1326–1333, 1997.

[20] Jingdao Chen, Yihai Fang, and Yong K Cho. Real-time 3d crane workspace update using a hybrid visualization approach. *Journal of Computing in Civil Engineering*, 31(5):04017049, 2017.

[21] Zhichao Chen and Stanley T Birchfield. Person following with a mobile robot using binocular feature-based tracking. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 815–820. IEEE, 2007.

[22] Yong Cho, Chao Wang, Mengmeng Gai, and Jee Woong Park. Rapid dynamic target surface modeling for crane operation using hybrid ladar system. In *Construction Research Congress 2014: Construction in a Global Network*, pages 1053–1062, 2014.

[23] Michel Coster and Jean-Louis Chermant. Image analysis and mathematical morphol-

ogy for civil engineering materials. *Cement and Concrete Composites*, 23(2-3):133–151, 2001.

[24] Arturo De la Escalera and Jose María Armingol. Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. *Sensors*, 10(3):2027–2044, 2010.

[25] Rachid Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International journal of computer vision*, 1(2):167–187, 1987.

[26] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[27] Sahibsingh A Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4):325–327, 1976.

[28] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Improved multi-person tracking with active occlusion handling. In *ICRA Workshop on People Detection and Tracking*, volume 2. Citeseer, 2009.

[29] John G Everett and Alexander H Slocum. Cranium: device for improving crane productivity and safety. *Journal of construction engineering and management*, 119(1):23–39, 1993.

[30] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[31] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.

[32] Junhong Gao, Seon Joo Kim, and Michael S Brown. Constructing image panoramas using dual-homography warping. In *CVPR 2011*, pages 49–56. IEEE, 2011.

[33] Marc Gelgon and Patrick Bouthemy. A region-level motion-based graph representation and labeling for tracking a spatial image partition. *Pattern Recognition*, 33(4):725–740, 2000.

[34] Ilias Grinias and Georgios Tziritas. Motion segmentation and tracking using a seeded

region growing method. In *9th European Signal Processing Conference (EUSIPCO 1998)*, pages 1–4. IEEE, 1998.

[35] US GSA. Gsa bim guide for 3d imaging. *Washington, DC: US General Services Administration. http://www. gsa. gov/graphics/pbs/GSA_BIM_Guide_Series_03. pdf*, 2009.

[36] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[37] Taketsugu Hirabayashi, Kazuki Abukawa, Tomoo Sato, Sayuri Matsumoto, and Muneo Yoshie. First trial of underwater excavator work supported by acoustic video camera. *Journal of Robotics and Mechatronics*, 28(2):138–148, 2016.

[38] Berthold KP Horn and Brian G Schunck. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, pages 319–331. International Society for Optics and Photonics, 1981.

[39] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[40] John Illingworth and Josef Kittler. A survey of the hough transform. *Computer vision, graphics, and image processing*, 44(1):87–116, 1988.

[41] Michal Irani and P Anandan. A unified approach to moving object detection in 2d and 3d scenes. *IEEE transactions on pattern analysis and machine intelligence*, 20(6):577–589, 1998.

[42] Masaru Ito, Yusuke Funahara, Seiji Saiki, Yoichiro Yamazaki, and Yuichi Kurita. Development of a cross-platform cockpit for simulated and tele-operated excavators. *Journal of Robotics and Mechatronics*, 31(2):231–239, 2019.

[43] Donghyun Kim and Richard B Langley. On ultrahigh-precision gps positioning and navigation. *Navigation*, 50(2):103–116, 2003.

[44] Donghyun Kim, Richard B Langley, JH Kim, and SN Kim. A gantry crane autosteering system based on gps rtk technology. *The European GNSS*, 2003.

[45] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and*

augmented reality, pages 225–234. IEEE, 2007.

[46] Ghang Lee, Joonbeom Cho, Sungil Ham, Taekwan Lee, Gaang Lee, Seok-Heon Yun, and Hyung-Jun Yang. A bim-and sensor-based tower crane navigation system for blind lifts. *Automation in construction*, 26:1–10, 2012.

[47] Ung-Kyun Lee, Kyung-In Kang, Gwang-Hee Kim, and Hun-Hee Cho. Improving tower crane productivity using wireless technology. *Computer-Aided Civil and Infrastructure Engineering*, 21(8):594–604, 2006.

[48] Tianli Liao, Jing Chen, and Yifang Xu. Coarse-to-fine seam estimation for image stitching. *arXiv preprint arXiv:1805.09578*, 2018.

[49] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[50] Panadda Marayong, Hen-Geul Henry Yeh, Edgar Coronado, Vinay Ganji, and Avadhut Chaudhari. Computer-aided container handling assistance for ergonomic crane operation. *METRANS Project*, 2012.

[51] Nkhesani Mkansi, Noelle Ramsamy, and Yahya Laher. Drones in construction: a tool to measure progress.

[52] Rahul Kumar Namdev, Abhijit Kundu, K Madhava Krishna, and CV Jawahar. Motion segmentation of multiple objects from a freely moving monocular camera. In *2012 IEEE International Conference on Robotics and Automation*, pages 4092–4099. IEEE, 2012.

[53] Richard L Neitzel, Noah S Seixas, and Kyle K Ren. A review of crane safety in the construction industry. *Applied occupational and environmental hygiene*, 16(12):1106–1117, 2001.

[54] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

[55] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[56] Dong ping Tian et al. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 8(4):385–396, 2013.

[57] SA Pix4D. Pix4dmapper, 2014.

[58] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2609–2616. IEEE, 2014.

[59] Shrinivas J Pundlik and Stanley T Birchfield. Real-time motion segmentation of sparse feature points at any speed. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(3):731–742, 2008.

[60] Liyana Ramli, Z Mohamed, Auwalu M Abdullahi, HI Jaafar, and Izzuddin M Lazim. Control strategies for crane systems: A comprehensive review. *Mechanical Systems and Signal Processing*, 95:1–23, 2017.

[61] Autodesk ReCap, Autodesk ReCap360, and Autodesk ReCap360 Ultimate. Autodesk recap. *Accessed March*, 30, 2014.

[62] Weijun Ren, Zifeng Wu, and Lei Zhang. Real-time planning of a lifting scheme in mobile crane mounted controllers. *Canadian Journal of Civil Engineering*, 43(6):542–552, 2016.

[63] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[64] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):105–119, 2008.

[65] E Rublee, V Rabaud, K Konolige, et al. Orb: an efficient alternative to sift or surf. iccv. In *2011 IEEE International Conference on*, 2011.

[66] Konstantin Rumyantsev and Dmitry Petrov. An analysis of feature detection efficiency on images with a priori unknown and changing viewing conditions. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 570–573. IEEE, 2016.

[67] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Learning 3-d scene structure from a single still image. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[68] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In

2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), volume 1, pages 519–528. IEEE, 2006.

[69] Reza Serajeh, Amir Mousavinia, and Farzad Safaei. Motion segmentation with hand held cameras using structure from motion. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 1569–1573. IEEE, 2017.

[70] Naai-Jung Shih. The application of 3d scanner in the representation of building construction site. *NIST SPECIAL PUBLICATION SP*, pages 337–342, 2003.

[71] Ashit Talukder and Larry Matthies. Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 4, pages 3718–3725. IEEE, 2004.

[72] Pingbo Tang, Daniel Huber, Burcu Akinci, Robert Lipman, and Alan Lytle. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in construction*, 19(7):829–843, 2010.

[73] Takanobu Tanimoto, Ryo Fukano, Kei Shinohara, Keita Kurashiki, Daisuke Kondo, and Hiroshi Yoshinada. Research on superimposed terrain model for teleoperation work efficiency. *Journal of Robotics and Mechatronics*, 28(2):173–184, 2016.

[74] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[75] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5038–5047, 2017.

[76] George Vogiatzis and Carlos Hernández. Video-based, real-time multi-view stereo. *Image and Vision Computing*, 29(7):434–441, 2011.

[77] John YA Wang and Edward H Adelson. Layered representation for motion analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–366. IEEE, 1993.

[78] Hao Wu, Jing Tao, Xinping Li, Xiuwen Chi, Hua Li, Xianghong Hua, Ronghua Yang,

Sheng Wang, and Nan Chen. A location based service approach for collision warning systems in concrete dam construction. *Safety science*, 51(1):338–346, 2013.

[79] Jianjie Wu and Qifu Wang. Feature point detection from point cloud based on repeatability rate and local entropy. In *MIPPR 2007: Automatic Target Recognition and Image Analysis; and Multispectral Image Acquisition*, volume 6786, page 67865H. International Society for Optics and Photonics, 2007.

[80] Ye Xuan. Design of embedded infrared detection-based anti-collision system for bridge cranes. *Hoisting and Conveying Machinery*, 2, 2010.

[81] Chang Yuan, Gerard Medioni, Jinman Kang, and Isaac Cohen. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *IEEE transactions on pattern analysis and machine intelligence*, 29(9):1627–1641, 2007.

[82] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European conference on computer vision (ECCV)*, pages 822–838, 2018.