A divide-and-conquer approach for Visual Odometry with minimally-overlapped multi-camera setup

by

Jaeheon Jeong

B.A., Mechanical Engineering, Yeungnam University, 2000

M.S., Mechanical Engineering, Yeungnam University, 2002

M.S., Computer Science, University of Colorado, 2010

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2014

This thesis entitled:
A divide-and-conquer approach for Visual Odometry
with minimally-overlapped multi-camera setup
written by Jaeheon Jeong
has been approved for the Department of Computer Science

Nikolaus Correll

Prof. Clayton Lewis

Prof. James Martin

Prof. Tom Yeh

Prof. Min Choi

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Jeong, Jaeheon (Ph.D., Computer Science)

A divide-and-conquer approach for Visual Odometry
with minimally-overlapped multi-camera setup

Thesis directed by Prof. Nikolaus Correll

Pose estimation of a moving camera rig from the images alone has been investigated by the computer vision community for decades, because the location and direction information of the cameras are the basis for more advanced applications, such as 3D reconstruction and Simultaneous Localization and Mapping (SLAM). Visual Odometry (VO) is the accumulation of the relative pose estimation while the camera rig moves. There are some visual odometry methods for mono view, stereo, omnidirectional and multi-cameras that require additional sensor input(odometry, compass, e.g,) and/or synchronized cameras. However, in our Virtual Exercise Environment (VEE) system, and other low-cost multi-camera setups, none of the above methods can be applied. The Bundle Adjustment(BA) is the general approach for a non-regular case like the VEE system. BA puts all the known variables in one huge matrix and solves the unknowns at once with Levenberg-Marquardt iterations. Thus, the BA is computationally expensive in nature with the complexity of $O(n^3)$, and sometimes infeasible. In this thesis, I propose a 'divide and conquer' approach that generates additional observations from consecutive images in neighboring camera pairs. I show this approach to solve the critical condition and drastically speed up pose estimation when compared to BA. The performance with different conditions and sub-algorithms are also tested and discussed.

Dedication

To my love.

Acknowledgements

I would like to express my special appreciation and thanks to my advisor Prof. Nikolaus Correll. I would like to thank my committee members, professor Prof. Clayton Lewis, Prof. James Martin, Prof. Tom Yeh and Prof. Min Choi for serving as my committee members. Without a question, many thanks are also owed to Dr. Jane Milligan for the years of support.

In the course of my studies here, I have met many wonderful people who have enriched my life professionally and personally. In particular, Dr. Wei Xu, Dr. Yuli Liang, Aymman Hammuda, and Timsy Apel for the days of discussing numerous research topics. Thanks to Dr. Michael Otte, Dave Coleman, Nicholas Farrow, Timothy Caldwell, Anshul Kanakia, John Klingner, Erik Komendera, Andy McEvoy, Halley Profita, Rowan Wing, Heather Hava, Dana Hughes and Scott Mishra for the encouragement and friendships.

In addition, I would like to thank to Prof. Song, Dong Joo, Prof. Sah, Jong Youb, and Prof. Kim, Young-Tak for encouraging me to study abroad.

Finally, very special thank to my family.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

Since the industrial revolution, machines have replaced manual labour. Mass production has been supported by machines and their operators. Although machines contributed to the development of civilization, the industrial age still needs help from human beings to complete the production of products. Robot arms increased their share in the industry assembly line with better performance than people conducting jobs. People want to train robots for complex works. An autonomous robot should do the job just with a command from a user. An autonomous robot need to understand what task need to be done, and some basic information. To understand the task and the environment, a robot may use many sensors and have advantages over human since they can use some sensors such as ultrasound, thermal image, and LIDAR, which human beings do not have. For an autonomous mobile robot, the location and the pose of the robot are important information, and many mechanisms are developed on that purpose.

There are many methods to find out the current location of robots. Some of them are listed on Table 1. Line tracers read the signs on the floor, like drivers read signs on the road. Infrared markers for indoor robots work as Polaris and constellations for navigators in the age of discovery. Beacons are good landmarks of specific location such as lighthouses. Global Positioning System(GPS) is using the signals from the artificial satellites on the stationary orbits. These methods are able to be used only with the infrastructure already made. Dead Reckoning(DR) just uses the agent's own information like the previous location and speed to estimate the current position. Dead Reckoning(DR) is used in the same way that the navigators use a compass and the

Method	Data for computing
Global Positioning System(GPS)	Signals from satellites
Line Tracer	Line Maps on the ground
Infrared Marker	Patterns on the ceiling
Beacons	Radio, Acoustic, or RADAR signals from the beacons
Inertia Measure Unit(IMU)	Signals from gyro and accelerometer
Visual Odometry	Images from cameras

Table 1.1: Odometry Methods

speed of the ship in the age of discovery. Visual Odometry(VO) is the technology for estimating the changes of position and orientation from the images of a camera on a moving agent over time.

Usually, odometry can be calculated with the composition of various sensors such as a speedometer, an accelerometer, a gyroscope, or a Global Positioning System(GPS). Each sensor has its advantages and drawbacks in their ability to find specific locations. Speedometers measure the signals of rotations from the wheels. However, slipping wheels make errors, and a legged robot does not have wheels. Accelerometers and gyroscopes, typically called an Inertia Measure Unit(IMU), have been used in airplanes for years, but there are accumulated error problems. GPS mechanisms provide current position through satellite signals, but cannot reach underground structures. Visual Odometry(VO) is based on images from one or more cameras, and a camera is one of the most popular forms of equipment on robots. One of the advantages of visual odometry is that the camera is so common on contemporary robots. Images provided by the robots help the supervisor of the robot help to understand the surroundings, and the environment information can be extracted while computing the current location through the Visual Odometry(VO). The disadvantages rest with the accumulated errors, and drifts in Visual Odometry(VO). The same disadvantages of accumulated problems are formed on Visual Odometry(VO) and Inertia Measure Unit(IMU). In Inertia Measure Unit(IMU), there is no way to take care of the problem itself. However, in Visual Odometry(VO) the environment can be used as a landmark to fix the drift. To recognize the landmark, the agent

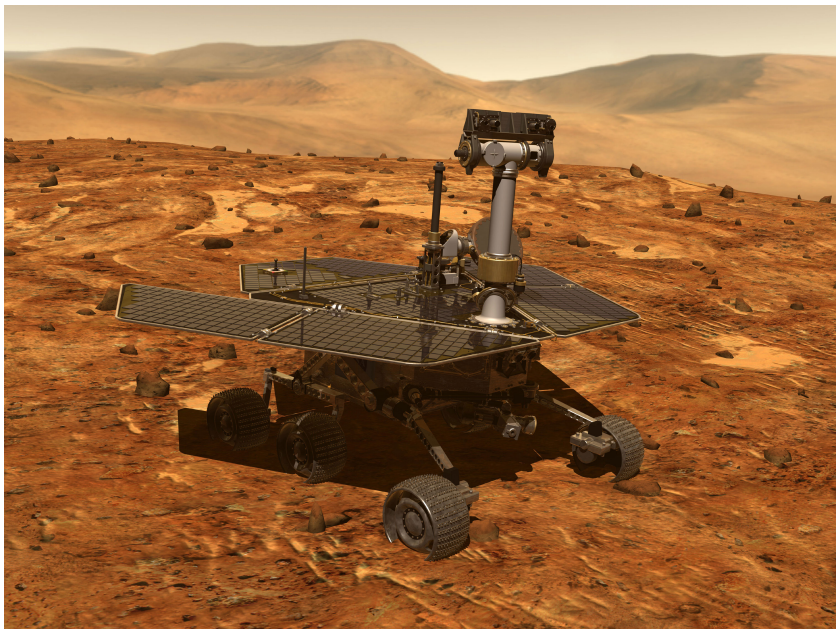
should visit the same place at least twice.

Since it is simple to reconstruct depth images, typical Visual Odometry(VO) comes with a set of stereo camera. With depth images, each tracking point can be registered to a map, and the clouds of matched points between the movement give the relative camera position through the Iterative Closest Point(ICP)[111]. The Iterative Closest Point(ICP) algorithm can be used as the tracking and mapping method for the robots[63]. Figure 1.1 shows the Spirit Mars Rover and its Visual Odometry(VO) track on planet mars[17]. The Mars Rover is designed to have several cameras which have their own examination purpose. For the VO, the stereo navigation cameras are installed. Using only one camera, it is impossible to get a real world distant measurement, which is called the scale factor. Thus, additional sensors or some initial information is necessary for the scale factor in one-camera systems. In multi-camera systems, which do not have enough overlapped portions for a depth image by stereo, there is a different problem. The motions of a bunch of cameras fixed on one camera rig can be regarded as just one motion of the camera rig by the combination of each movement of cameras. However, under degeneracy conditions, such as zero rotation or pure rotation, the system of equations goes trivial and fails. The Bundle Adjustment(BA)[103], which computes the camera positions and 3D locations of feature points together iteratively, seems to be the only option for these degeneracy cases. However, BA is computationally expensive due to the Levenberg-Marquardt iterations in it.

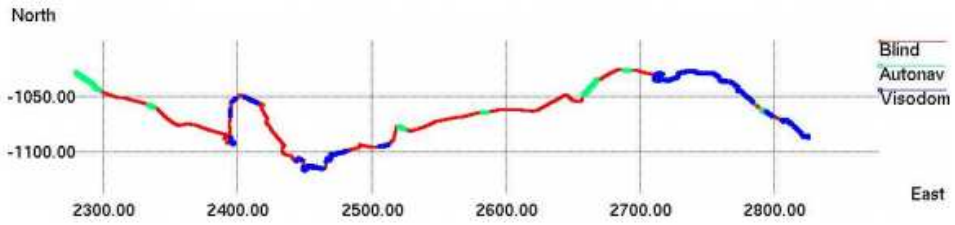
For the purpose of visual odometry, multi-camera without stereo is not a good setup. However, the setup appears in our daily life for information of the surroundings, for example, an around view of an automobile as seen in Figure 1.2. The recording setup for the Virtual Exercise Environment(VEE) project also has the purpose of larger viewing angle and thus does not have enough overlapped area for the stereo visual odometry. Figure 1.3 shows a user experiment of VEE. Note that there is three screens playing three videos, those are recorded from three camcorders. In the VEE project, tagging distance for each frame of the stitched image has been an issue. If the distance tags of the images are not smooth, the playing speed of image for the user changes in rough. An improvised speed meter was used at first. Moving on the Global Positioning System(GPS) was

not successful. The GPS did not get enough resolution in time and the tolerance in space was too large. For example, on a corner of a trail, the distance tags interpolated with GPS data have less distance values than the ground truth, and they lose smoothness on showing the corner. For distance tags problems, Visual Odometry(VO) is a good alternative to distance sensors and VO can take an advantage from the many images from the multi-cameras which are already set up for the good user experience in VEE project.

In next chapter, I start with the VEE project. Then, the basic information about the geometry with two cameras is explained in the next chapter. After that, I propose a novel solution for the visual odometry for a multi-camera system which does not have enough overlapped area for the stereo visual odometry. A 'divide and conquer' approach that generates additional observations from consecutive images in neighboring camera pairs is presented and its geometrical binding is suggested. I show this approach to solve the critical condition and drastically speed up pose estimation when compared to BA. The performance with different conditions and sub-algorithms are also tested and discussed.

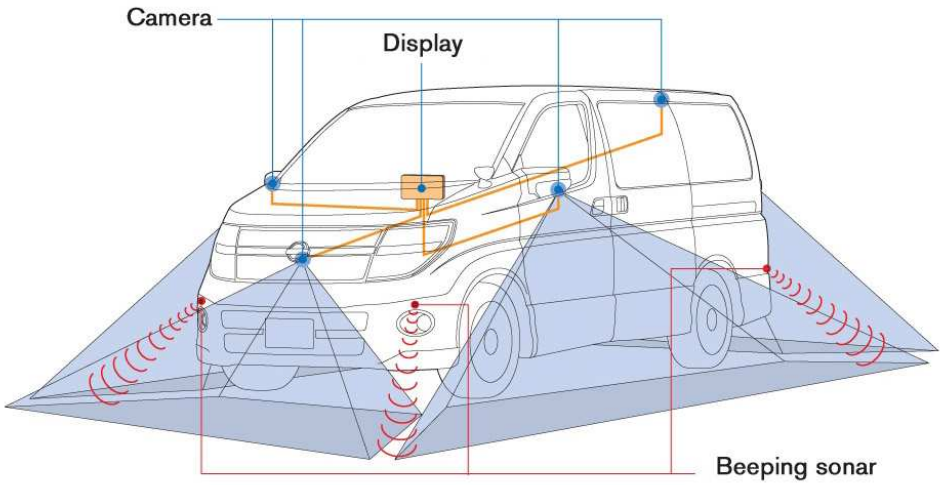


(a) Mars Rover, Spirit, Courtesy of NASA



(b) Plot of Spirit's traverse history using Visual Odometry

Figure 1.1: Example of Visual Odometry, Spirit traverse on Mars [17]



(a) Cameras on Automobile for Around View

(b) Aroundview Display on a Screen

Figure 1.2: Around View of an Automobile, as an example of multi-camera system for wide angle of view, Courtesy of Nissan[98]

Figure 1.3: A Multi-screen Virtual Exercise Environment in Use

Chapter 2

Virtual Exercise Environment(VEE)

The Virtual Exercise Environment(VEE) project was sponsored by the National Institute on Disability and Rehabilitation Research (NIDRR) through the Rehabilitation Engineering Research Center on Recreational Technologies(RERC Rec-Tech). People who do not have enough cardiac exercise have high risk of cardiac diseases, especially who can not go outside easily. People in wheelchairs have a higher possibility on cardiovascular disease than walking people since they do not use muscles in their legs. The only cardiac exercise equipment they can use is the ergometer while people who are able-bodied may choose one among a treadmill, a stationary bike, an elliptic, a stair climber, and so on. Thus, cardiac exercise only with an ergometer is easily boring. Adhering to an exercise program using stationary exercise machine at home or gym is the main task for the VEE project.

In this chapter, we introduce VEE systems, the first and second generations with the characteristic points of each generation. Each generation consists of a recording system to capture video, distance and incline data about real trails, and a playback system which displays both video and terrain data in the form of video speed and resistance. The first generation has 360 degrees of view while the second generation has full HD quality images of the scene and supports the communication with other users. Trails are played back according to the speed, which the user generates on the stationary exercise equipment. The system uses commodity capture and display devices and supports standard interfaces for existing exercise equipment. Finally, we discuss the challenges on developing the VEE systems.

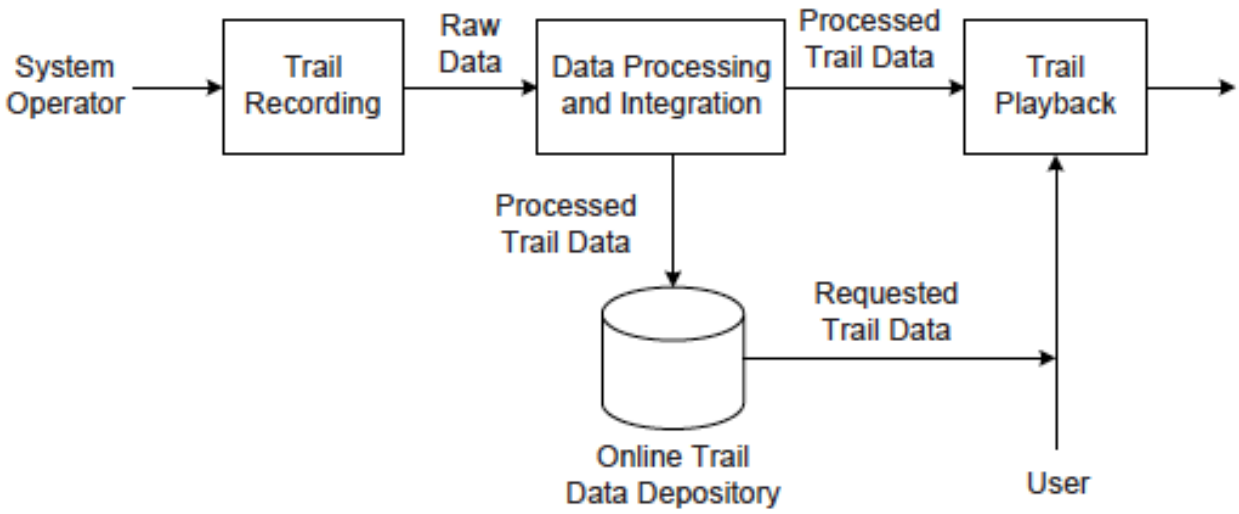


Figure 2.1: VEE System Overview

2.1 VEE, First Generation

In this section, we explain how 360 degree panoramic video is made and the way how interact with users to make them feel like they are on a real exercising trail. The VEE consists of a recording system to capture video, distance and incline data about real trails, and a playback system. The playback system displays both video and terrain data in the form of video speed and resistance, according to the speed, which the user generates on the stationary exercise equipment. The system also supports standard interfaces for existing exercise equipment.

2.1.1 Cameras and Sensors

To get a 360 degrees of panoramic view, five cameras are placed on the camera header as Figure 2.2. The Fire-i digital camera from the Unibrain[40] has a lens for 107 degrees of horizontal viewing angle and 400Mbps IEEE 1394a port for data transfer through CMU 1394 digital camera driver[23]. Those cameras are powered by two 7.2V, 2000mA batteries for outdoor environment. The maximum frame rate on recording is 30fps, but it was dropped down to 10 fps when five cameras are all connected to one firewire port on a laptop. The fps is dropped down to around 7 fps while recording since the voltage of the laptop's battery dropped down and the hard disk drive slows down with the accumulation of images data. The distance meter is devised with a hall sensor and three magnets on a rear wheel of a trike. An incline meter is also equipped to take a note of the grade in slopes. The data from the incline meter and hall sensor is sent to the serial port of the laptop through an AVR board. All the equipments, the camera head, sensors with a communication board, batteries, and a laptop, are set on a trike as seen in Figure 2.3.

2.1.2 Panorama Stitching

The homography between images are computed with SIFT[58] points through the first several frames. Then, the five images for one frame are stitched to one panorama image. On each frame, the overlapped area of the images are blended by the scheme of scaled images[11]. The four pyramids of

Name	Category	Manufacture	Property	Interface
FlexDeck	Treadmill	LifeFitness	slope	CSAFE
LifeCycle	Exercise Bike	LifeFitness	Resistance Level	CSAFE
SciFit Pro I	Arm Ergometer	SciFit	Resistance Level	CSAFE
PCGamerbike	Arm Ergometer	SciFit	Virtual Gear	FitFx
Crankcycle	Ergometer	Matrix	Virtual Gear	Arduino

Table 2.1: List of Compatible Exercise Machines

the images are enough for the blending. In addition, the distance from the edge of each overlapped area is used as weight for the blending. Before the blending, the correction of the brightness should be done with the relative brightness on the overlapping areas, since the brightness of each cameras are different due the different direction of the sunshine on each camera. The brightness correction with the gain compensation method[9] does not remove the seam clearly, but it is enough for the VEE since only part of the panorama image can be seen on the HMD. After five images in 320 by 240 pixel resolution are stitched, the resolution of the panoramic image is 1431 by 261.

2.1.3 Video Playback

The Z800 3DVisor manufactured by eMagin[26] is introduced to the system as a Head Mount Display(HMD) as seen in Figure 2.6. The HMD has an accelerometer, a magnetometer, and a rate gyroscope for the head movement. Yaw and pitch values are used in our playback system to determine to show which part of the panoramic image appears on the display.

The HMD is good to show the part of panoramic image with the head movement, but has some problems. First, it can not be used with a treadmill due to a safety issue. A user with HMD on the treadmill is actually blind, because he/she can not see where his/her feet step. Thus, the user may step on the sides of the treadmill, and fall down. Second, the HMD costs more than one thousand USD and is not affordable for every user. Third, only one part of the panoramic images are able to be seen and is crude pixel resolution. Even though the HMD supports 800 by 600 pixels of resolution, the panoramic images are not big enough to cover up the full resolution of the HMD.

Last, the yaw sensor of the HMD has an accumulation of errors. Though the HMD has a compass in it, the compass was used only once at the start and ignored while it is on the move and the yaw value has drifted with noises. In this case, even if the user want to see the forward direction, but need to keep the head direction on a little left and more.

2.1.4 Communication with Stationary Exercise Machine

The protocol, Communications Specification for Fitness Equipment(CSAFE)[30], is used in the VEE project. The protocol is agreed by eight major exercise equipment manufacturers in the market. Treadmills, stationary bikes from Lifefitness, and ergometer from SciFit are tested and used for the VEE project through the CSAFE protocol. The incline data from the terrain recorded are applied to each machine, for example, inclining of the treadmill and the resistance change for the stationary bike and the ergometer.

When we exhibit our system on the Rehabilitation Engineering and Assistive Technology Society of North America(RESNA) conference, people ask that more equipment can be worked with the VEE system. Thus, an affordable PCGamingBike has been included. The PCGamerBike is connected with USB 2.0 cable. It just provides the speed of the equipment, and it does not allow the playback program to change the resistance level. A concept of changing gears while going up the hill and the incline data are compensated into the distance in the software.

2.2 VEE, Second Generation

The first generation of VEE is successful, but still has some problems. The stitched images take too much disk space. The image resolution on the HMD is not good enough. The HMD is somewhat expensive and not affordable for some users. It is dangerous to use the HMD on the treadmill and needs a separate version of playback system for non-HMD. The recording equipment is not simple to recreate another to record more exercising trails, but is necessary to record more trails somewhere else. Thus, we move onto the second generation of VEE system. For the disk space, camcorders are introduced to use the mpeg compression. Instead of HMD, three big screen

TVs are introduced for the gym version, and also one screen version for the home users. In addition, an Arduino board with hall sensor with magnets are devised for regular exercise equipment with out CSAFE protocol. The Matrix Bike without any electric equipment, was tested with the novel arduino rpm meter on the second generation.

2.2.1 Recording Equipment

Three camcorders are equipped on a stroller with a GPS tracker. The camcorder from JVC has 1920 by 1080 pixels of resolution. We tested with various formations of a triangle for better panoramic images, and concluded that the forward looking camera should be 15 centimeters in front of the side camcorders, and the side cameras should be 17 centimeters away from the center axis and about 45 degrees turned to outsides. Thus, the distance between center and a side camera is about 20.8 centimeters.

A GPS tracker for hikers records for the trail as seen in Figure 2.8, but it records only one position for a minute. It means just one position tag for 1800 frames. Figure 2.9 is an iPhone app from MotionX[32], which can notate a GPS data. However, the recorded marks are only two times for a minute. In addition, there is no time synchronization method among a GPS recorder and recording cameras at all. Therefore, a Visual Odometry(VO) algorithm is necessary for better resolution in position on the second generation VEE system.

2.2.2 Panorama Stitching

Since the off-the-shelf camcorders do not have any equipment for the shutter synchronization, the panoramic stitching is quite a challenging problem. The brightness problem still exists due to the sunshine. The HD quality videos take lots of time to stitch. Panoramic stitching of there three videos are part of Wei's research[106] and is not in the range of this thesis. After stitching, the images are divided and enlarged to a full HD resolution and encoded to three full HD videos for playing.

2.2.3 Video Playback

Playing three videos on three screens with synced frame is quite challenging since the multi-threaded players do not decode each videos at the same time. One thread decodes faster and the others are slow. Thus, to meet the speed, the faster thread was forced to slow down. For the productivity in implementation, Microsoft's Multimedia Foundation API's [56] are used as mpeg decoder and it just supports four times of regular speed in theory. It means that it plays only four times faster than the recording speed, even if the user runs faster than four times of recording speed. The 4x speed limitation may from the the decoding speed of the MF APIs. In addition, there was a bottleneck with the video cards in PC. Two screens are connected to one video card, and the other screen is connected to another video card. Without the forcing the speed that the videos play down to the slowest speed, the screen which occupies the whole video card goes faster. Thus, the playing speed was not fast, but is enough to attract the users.

2.2.4 Multi-user Communication

The exercising buddy is a good motivator to keep the users exercising. A communication through the internet for two users is introduced. The connection is through the TCP/IP protocol. Two users can share and collaborate the same exercise trails whatever the exercise setup is. The speed of each user on the equipment is shared through the network and the playing speed of the video is set to the average of speeds for the two users to see same scene of the trail at the same time. Thus, the compensation for the different speed due to the other equipments matters and has been dealt with. The Voice over IP(VoIP) could be included in the playback system, but was too heavy to handle in the program. Thus, commercial VoIP services such as Skype[61] are recommended for the voice communication between the exercising buddies.

2.2.5 More Exercising Machines

There are many exercise machines which do not support the CSAFE protocol. A Crank Cycle manufactured by Matrix does not have any electronic parts and thus does not support the CSAFE

protocol. For the VEE system, at least one information, the speed of the exercising equipment is necessary. A speed meter is devised with a hall sensor, three magnets and an Arduino board[2]. The Arduino board has an USB port and is connected to the computer. The communication is set up like an additional serial port. Like the CSAFE machines, the serial port is fast enough in 9600 bps. Figure2.14 shows a Matrix's crank cycle with an Arduino speedometer in exhibition at the State of Science Conference. Some equipment those only provide the speed of the equipment and does not have any method to change the level of resistance for the exercise need another method to apply the terrestrial condition of the trails. Thus, a virtual gear system is introduced. The speed of exercising equipment is adjusted in the VEE playback system along the slope of the trails.

2.2.6 Focus Group

The VEE systems have been tested by a group of people, called focus group. The second generation of VEE system, which has three full HD screens has good response from the group. One person by one, they had workouts with the exercise machine they chose, and easily concentrated in exercising as they had been in the trail. To feel like in the trail, most member of the focus group wanted turn the light off. They also enjoyed the breeze of winds by a fan in the room. Maybe, the status of wind when recording can be helpful data if we can add fan control method in VEE systems. However, they pointed that the playing speed of the video was somewhat slow, which can be fast as the VEE program decodes the videos quickly.

2.3 Summary

Since the project started, the VEE system has been upgraded yearly for the better user experience. The three stages, which are recording the trails, processing the data, and playing the video with user's response, have been set up in clear. In near future, most of the exercise machine may support new protocol such as Message Queuing Telemetry Transport(MQTT) protocol for the machine-to-machine connectivity, and the VEE system may provide better environment. In addition, adding new feature such as the wind by a electric fan, will help more realistic user

experiences, and make people exercise more and healthy, which is the goal of the VEE project. An online demo of the VEE system can be found at Youtube: http://www.youtube.com/watch?v=4gEAfuAbntk. In addition, the VEE project motivated us to develop a novel method for multi-camera visual odometry when stereo VO can not be applied due to the lack of overlapped areas.

In conclusion, in the middle of the data process, the tagging distance for the each image frame is very important for the reality, and a VO algorithm is required.

Figure 2.2: Camera Header with Five Unibrain Cameras

Figure 2.3: Equipments on the Recording Trike

Figure 2.4: Trike on Recording

Figure 2.5: A Stitched Panorama Image from Five Cameras

Figure 2.6: eMagin Z800 Head Mount Display(HMD)

(a) Recorded surround video and terrain features are played back on synchronized HMD and exercise bike.

(b) Playback using synchronized computer monitor and treadmill.

Figure 2.7: VEE system, First Generation

Figure 2.8: A Tested GPS Tracker, Garmin eTrex Vista HCx for Hikers

Figure 2.9: A Tested iPhone App, MotionX GPS

Figure 2.10: Recording Stroller for the Second Generation VEE

Figure 2.11: A Panorama Stitched Image for the Second Generation VEE

Figure 2.12: One Screen Setup of the Second Generation VEE Playback System with a Desktop Ergometer

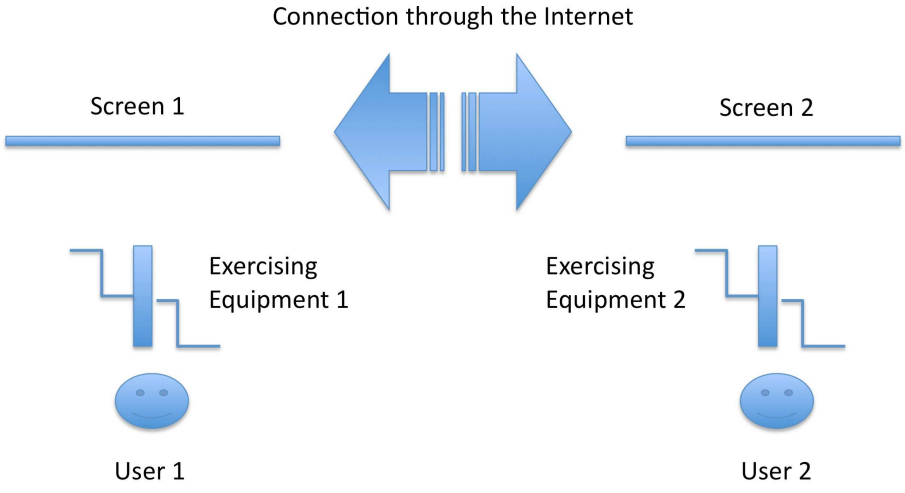


Figure 2.13: Overview of Multi-user Communication

Figure 2.14: Three Screen Gymnasium Setup of Second Generation VEE System with Matrix Crankcycle at SOS Conference

Figure 2.15: Three Screen VEE playback System with an Ergometer at the Focus Group

Chapter 3

Camera Model and Epipolar Geometry

A camera model is necessary to write a real world phenomena of a camera into equations. Starting with the structure of an eyeball, we are going to set up the geometric camera model. There are two types of the camera parameters which explain the status of a camera. The intrinsic camera parameters express the internal information of a camera. The extrinsic camera parameters are about the pose of a camera. To explain the relative poses of two cameras, the epipolar geometry is going to be discussed. Then, the methods to get the relative pose from the images, such as the eight point algorithm for the fundamental matrix and RANSAC, will be discussed in this chapter.

3.1 Camera Model

An object is projected to retina in an eye through the cornea, pupil, lens, and vitreous as seen in Figure3.1. A simple pin hole camera has the similar structure with the eyeball. Figure3.2 shows the structure of pin hole camera. The pin hole camera has a pin hole instead of the lens in the eyeball, and the size of pinhole is the aperture that the pupil changes in the eye. The object is projected onto an image plane in a pin hole camera, as projected onto the retina in the eye. The difference between the retina and the image plane is that the retina is curved but the image plane on the camera is a flat plane, where the radial distortions arise. As seen in Figure3.3, the geometric camera model is similar with the pin hole camera, but it is different that the image plane is placed between the object and the center of the camera in the geometric camera model. By locating the center of projection, the image plane, and the object in order, it is more intuitive to understand

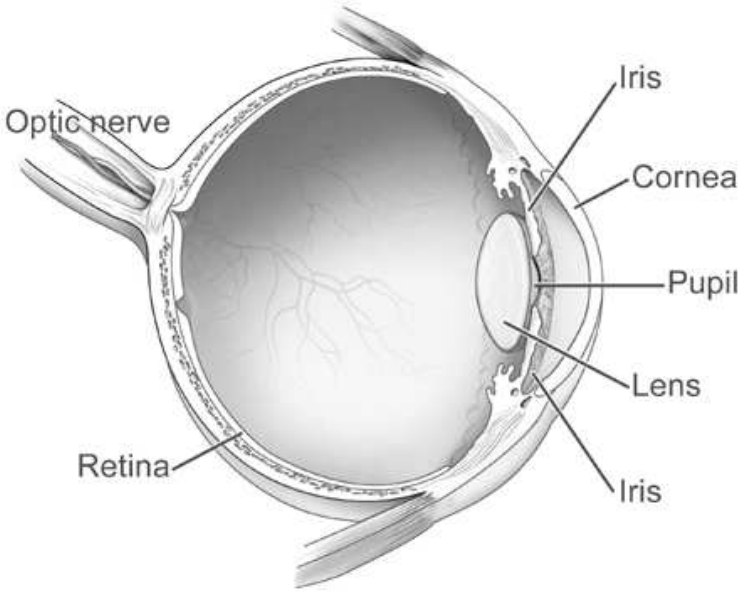


Figure 3.1: Structure of an Eyeball, courtesy of National Eye Institute

the image coordinate since the direction of U and V axes has the same direction with the object, while the image of the object is reflective in the pin hole camera.

In general, the origin of an image is on the left-top corner of the image and U axis increases to right direction and V axis increases to down. We are going to use the same direction of the axis in the image to the camera. Thus, the X and Y axes of the camera point the same directions of U and V, respectively. In addition, the direction of the Z axis always points to the object by the right-hand law. The pixel on the image plane is expressed by two integers. Thus, there exists a transform matrix between two coordinates.

In the camera, there is an image sensor, which converts an optical image to electrical signals, and the signals are turned to digital data by the image processor in the camera. There are two types of the image sensors, which are most common in the market. They are Charge-Coupled Device (CCD) and Complementary Metal-Oxide-Semiconductor(CMOS) active pixel sensors. Both sensors have their own advantages and disadvantages. CMOS has less static power consumption, but high noises in a dark surroundings. CCD has better sensitivity in the dark than CMOS but has vertical smears. The sensor parameter we are interested in is the size of the sensor and pixel

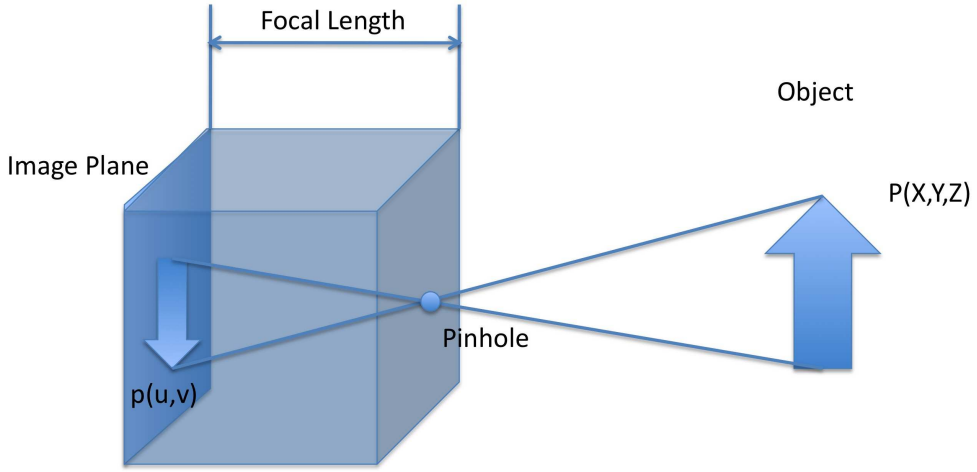


Figure 3.2: Pinhole Camera

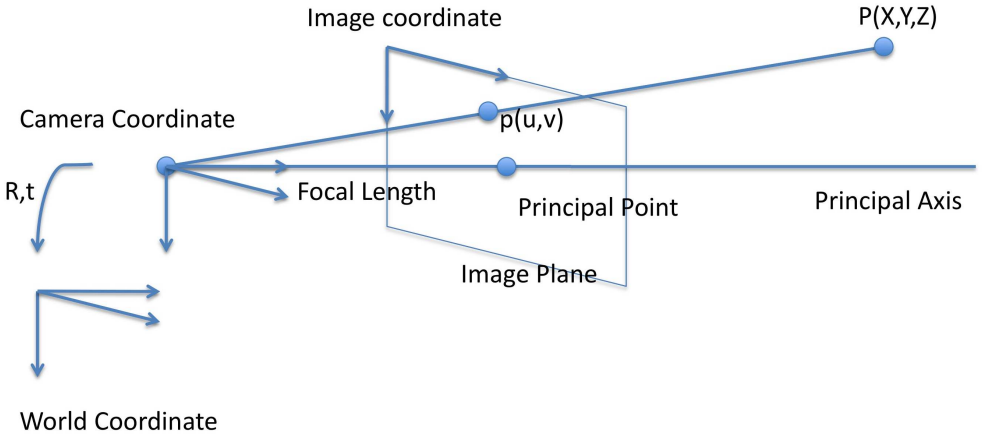


Figure 3.3: Geometric Camera Model

resolutions. In these days, one cell for a pixel is just a few micrometers on the image sensor. Thus, the scaling parameter and the parameter for the principal point has larger values in the intrinsic camera parameter matrix, K.

$$K = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

where,

$$\alpha_x = f \cdot m_x \alpha_y = f \cdot m_y \tag{3.2}$$

where, γ is skew coefficient, f is the focal length of the camera, and m_x and m_y are scale factors relating the pixels to distance on the image sensor. u_0 and v_0 represents the principal point, which is in the center of the image, but not exactly on the center of image sensor in practically.

The intrinsic camera parameters in K are usually set in the factory of the camera manufactures and the focal length only can be changed by users sometimes. By the process, called as camera calibration, the intrinsic camera parameters can be computed.

The extrinsic camera parameters are defined with the respect to the reference of the world reference coordinates, by the camera location and direction where the camera sees. Euler angles with roll, pitch, and yaw are usually used to express the angle of rotation. However, the order of the rotation is confusing in certain conditions. The alternatives of Euler angle are the rotation matrix and the quaternions. A rotation matrix is orthogonal, and its determinant is *one*. We are going to use the rotation matrix, R, as the way to express a rotation in this thesis. The rotation matrix, R, and the translation vector t are unique to express where the camera is located and which direction the camera sees. In addition, it is easy to handle the points in the reference coordinate along the change of the camera poses.

When the pose of the camera is determined, the points in the world also can be transferred to the camera coordinate to be projected into the image plane. In homogeneous coordinate, the transform matrix, T, can be easily composed with camera poses, R and t. For the transfer, the inverse of the rotation matrix, R^{T}, and the $-R^{\mathrm{T}}t$ is used for the points transfer, and the transferred

point, P_c is defined as

$$P_c = \begin{bmatrix} R^{\mathrm{T}} & -R^{\mathrm{T}}t \\ 0_3 & 1 \end{bmatrix} P = TP, \tag{3.3}$$

where 0_3 is $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$. Since the rotation from the world coordinate to the camera coordinate is R and the translation is t, the point, P, in world coordinate need to be transferred to camera coordinate through the transfer matrix, T.

3.2 Epipolar geometry

Epipolar geometry is known as the geometry of two scenes of one object. An epipole is the projection of the other camera center. The camera coordinate locates on O_1 and O_2. An epipolar plane is defined by two camera centers and one point, P, on an object. As seen in Figure 3.4, two cameras see a point P on an object in the three dimensional world. The point, P, and two camera centers, O_1 and O_2, make the epipolar plane, π. Two epipoles, e_1 and e_2, are on the epipolar plane. Two cameras have their own image planes, π_1 and π_2. The point, P, is projected onto the image planes, π_1 and π_2. Let p_1 and p_2 be the projected images of P on the image planes, π_1 and π_2, respectively. While P moves on the line $\overline{O_1P}$ and gets new position P', the projected image, p_1 on the image plane, π_1, does not move, but the p_2 moves along the line of intersection of two planes, π and π_2, and get new projection at ${p_2}'$. The line on the image plane, during the movement of P along the line $\overline{O_1P}$, is called an epipolar line, l_2. In the same way, l_1 is an epipolar line from the movement of P along the line $\overline{O_2P}$. Thus, there is a relationship between a projected point, p_1 and an epipolar line, l_2. The transformation from the point, p_1, to the epipolar line, l_2 is the essential matrix, E, and we have

$$l_2 = Ep_1, \tag{3.4}$$

where, $p_1 = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T$ and $l_2 = \begin{bmatrix} a_2 & b_2 & c_2 \end{bmatrix}^T$ and it satisfies

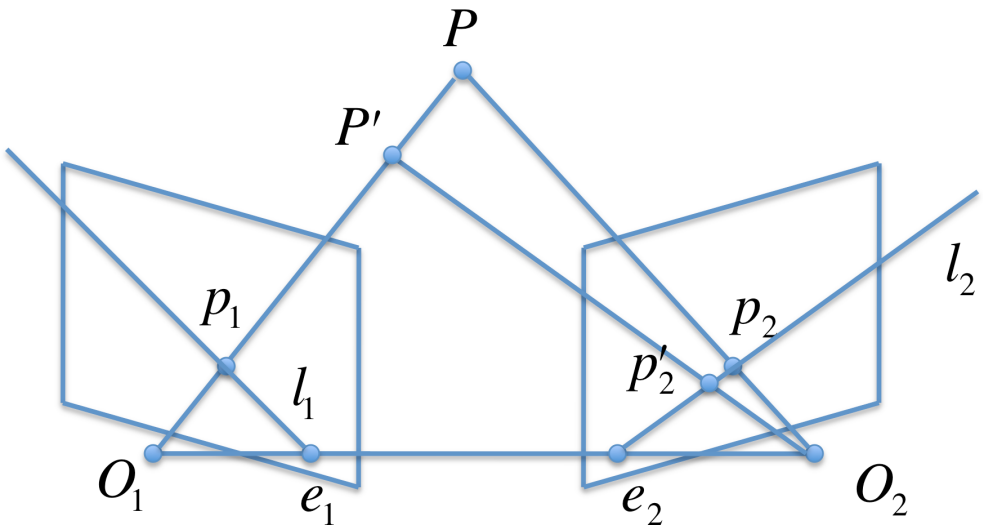


Figure 3.4: Epipolar geometry, two cameras locate at O_1 and O_2 and the point P is projected onto the image planes. A projected point of the other camera on the image plane is an epipole.

$$a_2x + b_2y + c_2 = 0 \tag{3.5}$$

on the image plane, π_2. There is also the other relationship between p_2 and l_1 and it is E^{-1}. The line, which connects two camera centers, O_1 and O_2, is called the baseline. When we set the reference coordinate on the camera coordinate on O_1, the pose of camera 2 can be explained with two parameters, the translation vector, $\vec{t}$, and the rotation matrix, R. We now construct vector equations from the relationship of points, images of points, and image planes with the translation vector and rotation matrix as follows,

$$\vec{t} = \overrightarrow{O_1O_2}, \tag{3.6}$$

$$\overrightarrow{O_2P} = \overrightarrow{O_1P} - \vec{t}, \tag{3.7}$$

$$\overrightarrow{O_2P} = sR\overrightarrow{O_1P}, \tag{3.8}$$

$$\overrightarrow{O_1P} = \frac{1}{s}R^{\mathrm{T}}\overrightarrow{O_2P}, \tag{3.9}$$

where s is a scalar scale factor, $\frac{|\overrightarrow{O_2P}|}{|\overrightarrow{O_1P}|}$. The scalar factor s indicates that the ratio of the distances from the cameras to the point, P.

Because the cross product of the vector $\vec{t}$ and $\overrightarrow{O_1P}$ become the surface normal of the epipolar plane, π, and the three vertices, O_1, O_2, and P are on a plane, the dot product between the surface normal and any vector on the plane should be zero. Thus we obtain

$$\hat{n} = \vec{t} \times \overrightarrow{O_1P}, \tag{3.10}$$

$$\overrightarrow{O_1P} \cdot \hat{n} = 0. \tag{3.11}$$

From equation (3.11), by substituting equation (3.9) into vector $\overrightarrow{O_1P}$ and equation (3.11) into $\hat{n}$ in equation (3.11), we have equation (3.12) below,

$$\frac{1}{s} R^{\mathrm{T}} \overrightarrow{O_2P} \cdot \vec{t} \times \overrightarrow{O_1P} = 0. \tag{3.12}$$

Because the cross product $\vec{t}\times$ can be expressed with a rank deficient skew-symmetric matrix, T_{x},

$$T_{\mathrm{x}} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}, \tag{3.13}$$

where $\vec{t} = (t_x, t_y, t_z)^{\mathrm{T}}$, we now obtain equation (3.14) from the matrix T_{x} and equation (3.12).

$$R^{\mathrm{T}} \overrightarrow{O_2P} \cdot T_{\mathrm{x}} \overrightarrow{O_1P} = 0. \tag{3.14}$$

The scale factor, s, can be dropped in equation (3.14) since the right hand side is zero. In addition, it is known that the dot product between two vectors can be expressed with a row vector and a column vector in matrix form, and thus we have equation (3.15) and (3.16),

$$(R^{\mathrm{T}}\overrightarrow{O_2P})^{\mathrm{T}}T_{\mathrm{x}}\overrightarrow{O_1P} = 0, \tag{3.15}$$

$$\overrightarrow{O_2P}^{\mathrm{T}}RT_{\mathrm{x}}\overrightarrow{O_1P} = 0, \tag{3.16}$$

from equation (3.14).

Here the matrix RT_{x} shows us the information of the rotation and translation, in other words, the relationship between the two vectors, $\overrightarrow{O_1P}$ and $\overrightarrow{O_2P}$. We call this matrix, RT_{x}, as an essential matrix, E,

$$E = RT_{\mathrm{x}}, \tag{3.17}$$

which is defined with the rotation matrix, R, and the translation vector's cross product form, T_{x}. Thus, the essential matrix, E has rank 2, because of the rotation matrix, R, has full rank and T_{x} has rank 2. In addition, the rotation matrix, R, is orthogonal, and T_{x} is skew-symmetric. Thus, the essential matrix, E, has two same non-zero singular values.

Thus, equation (3.16) can be rewritten as

$$\overrightarrow{O_2P}^{\mathrm{T}}E\overrightarrow{O_1P} = 0, \tag{3.18}$$

and it means that the the projection of P can transferred onto π_2 and forms an epipolar line, l_2. Since the projection of P onto π_2 is on the epipolar line, l_2, the dot product of the projected point and the epipolar line is zero.

3.3 Relative pose estimation

Pose estimation of the second camera relative to first camera is based on the epipolar geometry. The essential matrix has only extrinsic parameters of the relative pose. However, the points we can get from the camera are projection of the three dimension points onto the image plane through

the camera matrices which also include the intrinsic parameters. As same as equation 3.18, we define the fundamental matrix, F, which satisfies the relationship,

$$p_2^{\mathrm{T}} F p_1 = 0, \tag{3.19}$$

on the image coordinates, where $p_1 = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T$ and $p_2 = \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix}^T$. The fundamental matrix is a 3×3 matrix as same size as the easential matrix.

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}, \tag{3.20}$$

and equation (3.19) is

$$x_2x_1f_{11} + x_2y_1f_{12} + x_2f_{13} + y_2x_1f_{21} + y_2y_1f_{22} + y_2f_{23} + x_1f_{31} + y_1f_{32} + f_{33} = 0, \tag{3.21}$$

and equation (3.21) can be rewritten as

$$\begin{bmatrix} x_2x_1 & x_2y_1 & x_2 & y_2x_1 & y_2y_1 & y_2 & x_1 & y_1 & 1 \end{bmatrix} \vec{f} = 0, \tag{3.22}$$

where $\vec{f} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{21} & f_{22} & f_{23} & f_{31} & f_{32} & f_{33} \end{bmatrix}^{\mathrm{T}}$.

For n 3D points, the ith 3D point, P^i, has the projections, ${p_1}^i$ and ${p_2}^i$, respectively on image planes, then

$$\begin{bmatrix} {x_2}^1{x_1}^1 & {x_2}^1{y_1}^1 & {x_2}^1 & {y_2}^1{x_1}^1 & {y_2}^1{y_1}^1 & {y_2}^1 & {x_1}^1 & {y_1}^1 & 1 \\ {x_2}^2{x_1}^2 & {x_2}^2{y_1}^2 & {x_2}^2 & {y_2}^2{x_1}^2 & {y_2}^2{y_1}^2 & {y_2}^2 & {x_1}^2 & {y_1}^2 & 1 \\ \quad ... & & & & & & & & \\ {x_2}^n{x_1}^n & {x_2}^n{y_1}^n & {x_2}^n & {y_2}^n{x_1}^n & {y_2}^n{y_1}^n & {y_2}^n & {x_1}^n & {y_1}^n & 1 \end{bmatrix} \vec{f} = 0. \tag{3.23}$$

Setting up the n by 9 matrix to a matrix A, we obtain

$$A\vec{f} = 0. \tag{3.24}$$

The information we usually get is that the matched points from two images. What we want to know is the fundamental matrix, F, which has 9 unknowns, but the rank of F is two, and thus the matrix A should have at least *eight* 3D points, in other word, *eight* couples of the projections. The matrix A can be decomposed with Singular Value Decomposition(SVD). More about the epipolar geometry is explained in [104], [112], and [35].

Using SVD, the matrix, A, is decomposed as

$$A = UDV^T, \tag{3.25}$$

where U and V are unitary matrices. The matrix D is a diagonal matrix and the diagonal elements are the singular values. The SVD is a method to measure the distance from the matrix A to projection plane as we can see in A. Thus, each singular value is the distance measured and corresponds the column vector of the matrix V as the basis of projection plane. Since what we need on equation (3.24) is f, the null space of matrix A and the singular value we want should be *zero*. Practically, the smallest singular value is not zero, and we choose the smallest one among the singular values in the diagonal element of the D matrix, and also choose the corresponded column vector of V as f. Ideally, Af should be *zero*, and it means the smallest singular value is also *zero*. However, it does not happen practically, and thus our solution for the best is minimizing $\parallel Af \parallel$ subject to the condition $\parallel f \parallel = 1$.

Even the f is calculated, it may not satisfy the condition that the fundamental matrix, F, is a rank 2 matrix. In other words, determinant of the F should be *zero*. To modify the fundamental matrix to have rank 2 without losing principal information that we get from the matrix A, we use the SVD again for F, and we have

$$F = U_F D_F V_F^T, \tag{3.26}$$

where the fundamental matrix, F, is decomposed to three matrices, U_F, D_F, and, V_F. The fundamental matrix, F, is a 3 by 3 matrix and have three singular values, and thus one singular value

among three should be zero. The singular value matrix D_F is a diagonal matrix. Replacing the smallest singular value in the D_F by zero gives modified diagonal matrix, D_{FM}, and the modified fundamental matrix, F_M, can be composed again,

$$F_M = U_F D_{FM} V_F^T. \tag{3.27}$$

The relationship between the fundamental matrix and the essential matrix is that the fundamental matrix has the additional intrinsic parameters, which we already know from the camera calibration. We now have

$$F = K_2^{-T} E K_1^{-1}, \tag{3.28}$$

where K_1 and K_2 is the intrinsic camera matrix of first camera and second camera.

Thus, the essential matrix, E, can be computed by multiplying the camera intrinsic parameter matrix, K,

$$E = K_2^T F K_1. \tag{3.29}$$

The essential matrix, E, can be decomposed to the rotation matrix, R and the unit translation vector, t as seen in equation (3.17)[69].

However, we still have the cheirality problems after we get the rotation matrix, R and the unit translation vector, t, [35]. Cheirality problem appears because projected points in the image plane does not tell that the corresponded three dimension points were in front of the camera or behind of the camera. Thus, we have four possibility of choosing the correct case. The cheirality problem can be selected by the vote of triangulated point locates in front of or behind the camera, since all the three dimensional points should be in front of the camera in our camera model[109]. We have four possible cases, which are (R, t), $(R, -t)$, $\left(R^T, t\right)$, and $\left(R^T, -t\right)$. For a couple of projection, the reconstructed three dimensional point, $P_r e$ should be in front of both cameras. Thus, each reconstructed point by the triangulation has a vote for one of four possible cases, and the most voted couple of R and t is chosen in the end.

After computing the fundamental matrix from the randomly selected 8 points, it is necessary

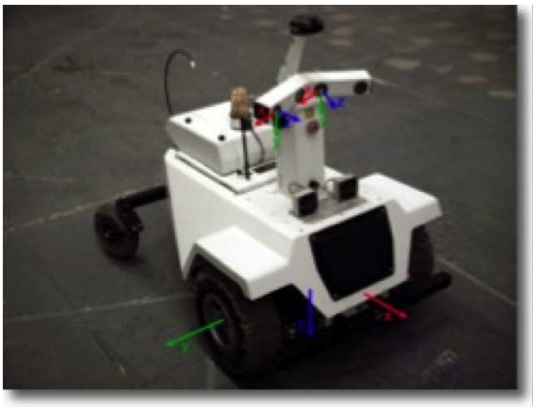
a. extract SIFT features from frame n
b. extract SIFT features from frame n+1
c. find matching points
d. normalize the points to reduce errors
e. randomly choose 8 point
f. calculate fundamental matrix and essential matrix
g. Chirality vote for R and t
h. compute re-projection errors and count inliers
i. repeat e, f, g, h, till the number of inliers is larger than the RANSAC threshold

Table 3.1: Relative pose estimation algorithm

for all match points to measure the re-projection errors. The sum of the distance between the reprojected point on image plane and the location of feature point extracted from SIFT can be regarded as an error that should be minimized. Thus, we count the inliers which has less distance than a threshold that we set before the subroutine. By repetition of choosing 8 points randomly, calculating fundamental matrix, voting for the chirality, and counting inliers with the reprojection errors, the most acceptable essential matrix can be given. The whole relative pose estimation algorithm for R and t is shown in Table 3.3.

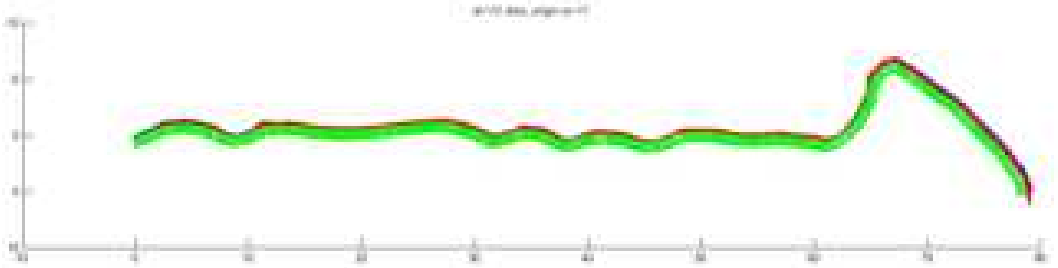
3.4 Test case

The relative pose estimation algorithm is described as the Table 3.3. The accumulated relative pose estimation with one camera is regarded as one view visual odometry. One camera visual odometry is impossible without the scale factor since the translation vector, t, is a unit vector. Thus, additional information for the scale factor is necessary for the one view visual odometry. The one camera visual odometry algorithm through the epipolar geometry can be tested with the LAGR[100] robot data. The LAGR robot has two sets of stereo camera, GPS, and IMU. Computing the poses of robot using just one view of the stereo camera with the additional information of a scale factor by the IMU is tested.



(a) LAGR robot

(b) sample image of the LAGR robot



(c) The track of LAGR robot by one view visual odometry with additional scale factor by IMU

Figure 3.5: One view visual odometry test by epipolar geometry

3.5 Summary

A projective camera model is set up and the relative pose estimation by the epipolar geometry is discussed. Through the camera model and the epipolar geometry, one view visual odometry method are implemented and tested. For more accurate calculation, some subroutines in the relative pose estimation (see Table 3.3) are tested. Other method for computing the fundamental matrix is also tried[96], but it is too slow and is not so much difference on the test case. The resolution of images and the focal length are important property for the accuracy, and the algorithm is impracticable without normalizing method[79] when the diagonal elements of the intrinsic camera matrix, K, have large values. The number of RANSAC[29] trial is also sensitive parameter in the algorithm, and practically it depends on the number of good matches in the all the matched point. The SIFT[58] features gives good results, but its computational burden is somewhat heavy to compare many points to find matched points. The harris corner detector[34] gives to much mismatch points and requires more trials in the RANSAC process than the SIFT features. How to set a threshold is still sensitive for case by case. The threshold should be less than one pixel for each re-projection in usual, and a smaller threshold requires more RANSAC iteration but does not guarantee better accuracy than the 1 pixel threshold.

Chapter 4

Multi-Camera Visual Odometry

Multi-camera system is useful for wide angle of view. An around view of an automobile is a good example. Four cameras are generally used on an around view to show front view, rear view and side views as seen in Figure 1.2. Another example is the Google Street View[97]. A bunch of cameras are set in a rig for panoramic images as seen in Figure 4.1. Our application, the VEE system, also uses three cameras for wide angle of views as seen in Figure 4.2. Visual odometry with a multi-camera system, which is designed for wide viewing angles, is not as popular as the stereo visual odometry. In this chapter, starting with explaining our VEE recording equipment, other researchers' works related on our problem are going to be discussed. Then, a set of equation with the geometric constraints is derived. Finally, the reason why the non-stereo multi-camera visual odometry is avoided will be discussed.

4.1 System Overview

In this section, the camera rig is explained for the purpose to introduce the geometric constraints of the system. Three cameras are mounted rigidly with divergent views as seen in Figure 4.2 These settings are good for wide field of views, which is one of the important attraction for the VEE users. However, there are not enough overlapped area for stereo comparisons for the stereo visual odometry. Mono view visual odometry of each camera gives the rotation matrix and the direction of motion, but for the metric scale factors. Those three independent motions of cameras might be fused to get the movement of the camera rigs. However, the critical conditions, which

(a) Google Street View Car

(b) Closeup of the Camera on the Google Street View Car

Figure 4.1: Google Street View System, courtesy of Google

will be discussed in section 4.4, make it difficult to get the information in stable.

We propose to divide the problem of metric motion estimation for a diverging trinocular rig into five single-view relative pose estimation problems and then calculate the scale factors of each single-view motion estimate. While the trinocular setup lends itself to three single-view problems, we construct two additional views by tracking feature points in consecutive image frames between the center and peripheral cameras.

The key idea here is that feature points on the frontal view will eventually appear in one of the side cameras after some frames as the camera rig moves forward as seen in Figure 4.3 In other words, with the front camera, the single view relative pose estimation which we discussed in section 3.3 should be done first. Then, the pose estimation with a side camera alone on time lapse we have to calculate, and the pose estimation with the other side alone should be compute. Fusing these three relative pose estimations does not give us reliable results near the critical condition. Thus, the relative pose estimation between the front camera of the previous time frame and one of side cameras on current time frame should be computed. Then, the same one but with the other side camera must be added. Finally, these five independent relative pose estimation are fused to estimate the metric scale factors. By adding two more relative pose estimations between forward camera and side cameras with time lapse, the critical condition near zero rotation is solved.

The five mono view relative pose estimation can be constructed simultaneously and the scale factors are calculated based on our new system of constraints. As discussed on section 3.3, the single-view visual odometry algorithm is based on the approach of Nister et al. [70], and uses RANSAC[29] and SIFT features[58].

4.2 Related Works

Pose estimation of camera rig from the images are investigated by computer vision society for decades, because the location and direction information of the cameras are the basis for more advanced applications, such as 3D reconstruction, SLAM and so on. As the Visual Odometry(VO) is coined in 2004[68], VO algorithms with single camera[68, 12], stereo camera[68], and omnidirectional

camera[10, 85, 54] were developed. Stereo schemes take advantage of depth information[63] and performed well in applications[60, 38]. Monocular visual odometry cannot resolve the scale factor on the translation distance. Omnidirectional camera also needs a special algorithm for VO to take care of its panoramic images.

Typically, multi-camera visual odometry uses metric reconstruction from stereo pairs, which are strongly calibrated, and highly overlapped, to resolve the scale factors[73]. If there is a lack of overlapped area, stereo based visual odometry cannot be used. The Bundle Adjustment[103] is the general approach for any kind of 3D reconstruction problems, but too slow as the complexity is $O(n^2 log(n))$, where n is unknown parameters, even well implemented with the sparse scheme[49]. With the number of features in the order of tens of thousand, the BA is infeasible without a powerful machine.

Once the general framework for multi-camera systems was proposed in [75, 31, 90], several authors have addressed the question of estimating camera motion or pose for multi-camera systems with limited or no overlap. The key problem as articulated by [65] was that the parameters for certain critical motions cannot be determined.

In particular one such motion is straight line motion with zero rotation angle, which is common in vehicles and mobile robots. [44] proposed a method to solve for camera motion by estimating and averaging the frame to frame rotations extracted from the essential matrices from individual VO calculations, but was not successful on the critical condition with zero-rotation. Thus, they just avoided the critical conditions in their test application by rotating their ladybug camera by 10 degrees for each frame[45, 46].

[77] also addressed a similar structure from motion application and needed to solve for relative position from a movement of the camera rig, however, they incorporate data from a Global Positioning System and an Inertial Navigation System.

[33] also proposed a solution for computing the visual odometry scaling factor for non-overlapped views. However, the difference is that we use not only the matched features for different frames on same camera, but also the matched features between different frames on neighboring

cameras.

[43] presented a framework for 6D absolute scale motion, they used the generalized camera model. They stacked up N-frames for the accuracy of scale factors. However, their Least Squares equations also have degeneracies when there is no rotation.

[80] also introduced the multiple non-overlapping camera model. They used the Extended Kalman Filter and median arbiter for four camera's rotation. However, their model equation is trivial when the rotation matrix is a identity matrix. In addition, their frameworks need the depth initialization for each camera at the first frame.

[21] also proposed a factorized framework for a generalized camera model. They divided the problem into rotation, translation and structure estimation, and averaged the rotation. They also admitted the possible degeneracies, but no further study is done yet and the framework is with the synthetic data simulation only.

The applicability of a spherical approximation for multiple camera was studied in [47]. Their research show better results than the 17-point algorithm for a generalized camera model. However, there is a strict sufficient condition with the distance between cameras and the distance to the feature point. To apply on our VEE system cases, the distance between cameras are too small to meet.

[42] proposed to use a multi-camera rig for SLAM. Other researchers also studied SLAM with a multi-camera rig[15]. The SLAM and autonomous navigation were integrated in their research, but they also used the wheel odometry, and the visual information for pose estimation was limited on their visual SLAM.

Our system uses three off-the-shelf camcorders which do not have any triggering or synchronization method, giving rise to temporal synchronization problems. [16] and [105] proposed methods for aligning video sequences, but the methods do not fit in our case because the different time stamp of the frames is continued with specific time lapse.

Calibration of our rig is also a tough problem. The system has slightly overlapped regions among images for stitching panoramic views. Stereo calibration for the two overlapping pairs was

performed using the Matlab Calibration Toolbox [7], but the result was not sufficiently accurate. There is some previous work addressing calibration for non-overlapping cameras [28, 59, 83], however, they all assumed synchronized shutters.

In our VEE system, none of the above methods can be applied and the usual option to solve was Bundle Adjustment(BA)[103], which puts all the known variables in one huge matrix and solves at once with Levenberg-Marquardt iterations. Thus, the BA is computationally expensive in nature with the complexity of $O(n^3)$. As seen in Figure 4.2, the system has three off-the-shelf full HD camcorders, which have diverted angles on the camera rig for wider field of view since the camera rig was designed for panoramic stitching for the purpose of wider field of view in VEE systems[107], and thus there is not enough overlapped area for stereo algorithms.

4.3 Geometry and Equations

We have three cameras on our camera rig. One is looking forward, another is looking 45 degrees left, and the other is 45 degrees looking right. Since these three cameras are for playing videos on screens as seen in Figure 2.15 , we call these cameras as left, center, and right camera. In addition, we numbered the center camera as 1, the left camera as 2, and the right camera as 3. We assume that we already know the intrinsic parameters and the transformation matrix between the center camera and each side cameras. On Figure 4.5, these two transformation matrices are shown as T_{CL} and T_{CR}. We assume that T_{CL} and T_{CR} are the previously calibrated camera transform matrices from the center to the left and right cameras respectively. The motion of camera rig causes ${}_6C_2$ camera transformations including the T_{CL} and T_{CR}. Here we focus on just five of them. Three transformations are for each camera with time difference, and two transformations are between center and side cameras with time differences. We also number the five relative pose estimation. The pose estimation of center camera alone in time lapse is numbered as 1, the one of left as 2, and the right one as 3. The additional comparison between the center camera in previous time frame and the left one in current time frame is numbered as 4, and the one between the center in previous time frame and the right one in current time frame is numbered as 5.

Let the transformation matrices T_1, T_2, and T_3 for each cameras with time differences and T_4 and T_5 for center camera with left and right cameras with time differences. These constraints on Figure 4.5 can be translated into the matrix equations,

$$T_2 T_{CL} = T_{CL} T_1, \tag{4.1}$$

$$T_3 T_{CR} = T_{CR} T_1, \tag{4.2}$$

$$T_2 T_{CL} = T_4, \tag{4.3}$$

$$T_3 T_{CR} = T_5, \tag{4.4}$$

$$T_{CL} T_1 = T_4, \tag{4.5}$$

$$T_{CR} T_1 = T_5, \tag{4.6}$$

$$T'_{CL} T_4 = T'_{CR} T_5, \tag{4.7}$$

$$T'_{CL} T_2 T_{CL} = T'_{CR} T_3 T_{CR}, \tag{4.8}$$

$$T'_{CL} T_4 = T'_{CR} T_3 T_{CR}, \tag{4.9}$$

$$T'_{CL} T_2 T_{CL} = T'_{CR} T_5, \tag{4.10}$$

where T'_{CL} and T'_{CR} are inverses of the T_{CL} and T_{CR}, respectively. These constraints are good as far as we have the translation vectors. However, the independent five monocular visual odometries give the rotation matrices and the unit translation vectors only, not with the length of the vector.

Let $p_i(k)$ be a matrix of image points from camera i at time k. The 5 standard monocular VO calculations are performed one each for cameras from $p_1(k)$ to $p_1(k + \Delta k)$, from $p_2(k)$ to $p_2(k + \Delta k)$ and from $p_3(k)$ to $p_3(k + \Delta k)$ in the usual way, and two more from $p_1(k)$ to $p_2(k + \Delta k)$ and to $p_3(k + \Delta k)$, respectively using [70] (see Figure 4.5). These yield five rotation matrices, $(R_1, R_2, R_3, R_4, R_5)$, and five unit translation vectors, (v_1, v_2, v_3, v_4, v_5), for the rig motion with respect to each temporal image pair. Here, Δk is chosen such that objects are likely to shift from the center camera to one of the outside camera views, which is a function of the speed of the camera setup and the frame rate. As $\Delta k >> 1$ in practice, the lack of synchronization between individual

views becomes neglible. As the transformation matrices are composed of the rotation matrices and the translation vectors, each transformation matrices can be written as

$$T_i = \begin{pmatrix} R_i & t_i \\ 0 & 1 \end{pmatrix}, \tag{4.11}$$

where i is from 1 to 5, and the scaled translation vectors, t_i, can be written as

$$t_i = a_i \cdot v_i, \tag{4.12}$$

where a_i is a scholar scale factor for each transformations. Also, the rotation matrices, R_{CL} and R_{CR}, and the translation vectors t_{CL} and t_{CR} are also known. Now we move on to the vector equations for the scale factors a_1, a_2, a_3, a_4, and a_5 for each of the 5 sequences. The scale factor a_1 is for the forward looking camera, a_2 for the left one, a_3 for the right one, a_4 for center at time k to left at time $k+\Delta k$, and a_5 for center at time k to right at time $k+\Delta k$. Using the center camera as the reference t'_{CL} and t'_{CR} are relative translation vectors transformed by estimated camera rotation R_1:

$$t'_{CL} = R_1 t_{CL}, \tag{4.13}$$

$$t'_{CR} = R_1 t_{CR}. \tag{4.14}$$

Figure 4.5 illustrates the basic geometry arising from the rig motion (R_1, t_1) from time k to $k+\Delta k$. Note that the origin of camera rig coordinate locates at the center camera. Cameras move from positions C_1, C_2 and C_3 to C'_1, C'_2 and C'_3 respectively. Given known t_{cl} and t_{cr}, estimated t'_{cl} and t'_{cr}, and unit translations v_i we can construct a series of constraints relating the old and new camera positions.

For example the location of C'_2 relative to C_1 can be described by the sum of the vector from C_1 to C'_1, which is $t_1 = a_1 v_1$ plus the vector from C'_1 to C'_2, which is the estimated t'_{cl}. We can describe the same location as the sum of the known vector t_{cl} from C_1 to C_2 plus the vector from C_2 to C'_2, which is $t_2 = a_2 v_2$. This vector algebra yields an equation constraining the unknown scale factor a_i (see equation (4.15)).

We construct 10 such equations exploiting these redundant relationships. The key to the robustness of our approach to degenerate motions is that we include the cross sequences from C_1 to C_2' and C_3', which inherently include the extrinsic rotations R_{cl} and R_{cr}, and thus will not have zero rotation when $R_1 = I$. For example, equation (4.21) describes the motion from C_1 to C_1' through C_2' $(a_4 v_4 - t_{cl}')$ and through C_3' $(a_5 v_5 - t_{cr}')$. The full set of equations is:

$$a_1 v_1 + t_{cl}' = t_{cl} + a_2 v_2, \tag{4.15}$$

$$a_1 v_1 + t_{cr}' = t_{cr} + a_3 v_3, \tag{4.16}$$

$$a_4 v_4 = t_{cl} + a_2 v_2, \tag{4.17}$$

$$a_5 v_5 = t_{cr} + a_3 v_3, \tag{4.18}$$

$$a_1 v_1 + t_{cl}' = a_4 v_4, \tag{4.19}$$

$$a_1 v_1 + t_{cr}' = a_5 v_5, \tag{4.20}$$

$$a_4 v_4 - t_{cl}' = a_5 v_5 - t_{cr}', \tag{4.21}$$

$$t_{cl} + a_2 v_2 - t_{cl}' = t_{cr} + a_3 v_3 - t_{cr}', \tag{4.22}$$

$$a_4 v_4 - t_{cl}' = t_{cr} + a_3 v_3 - t_{cr}', \tag{4.23}$$

$$t_{cl} + a_2 v_2 - t_{cl}' = a_5 v_5 - t_{cr}'. \tag{4.24}$$

From the 10 vector equations above, we rearrange the terms to form a matrix equation for solving the five scale factors, and thus we obtain

$$\begin{bmatrix} v_1 & -v_2 & 0 & 0 & 0 \\ v_1 & 0 & -v_3 & 0 & 0 \\ 0 & -v_2 & 0 & v_4 & 0 \\ 0 & 0 & -v_3 & 0 & v_5 \\ v_1 & 0 & 0 & -v_4 & 0 \\ v_1 & 0 & 0 & 0 & -v_5 \\ 0 & 0 & 0 & v_4 & -v_5 \\ 0 & v_2 & -v_3 & 0 & 0 \\ 0 & 0 & -v_3 & v_4 & 0 \\ 0 & v_2 & 0 & 0 & -v_5 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} t_{cl} - t'_{cl} \\ t_{cr} - t'_{cr} \\ -t_{cl} \\ -t_{cr} \\ -t'_{cl} \\ -t'_{cr} \\ t'_{cl} - t'_{cr} \\ -t_{cl} + t'_{cl} + t_{cr} - t'_{cr} \\ t'_{cl} + t_{cr} - t'_{cr} \\ -t_{cl} + t'_{cl} - t'_{cr} \end{bmatrix}. \tag{4.25}$$

We have five unknown scale factors and 10 vector equations, and the system 4.25 is an overdetermined system. An approximation for a_1, a_2, a_3, a_4, and a_5 is computed by solving the system. Each vector equation has three dimensions, and thus three scala equations. Matlab provided a function for the least square problems. However, the Singular Value Decomposition(SVD) can used to decompose the system for Least Squares(LS) techniques, if necessary.

In addition to the matrix system above, the weighted least squares scheme was also applied. For small rotations, the translation vectors of three same camera VO solutions, v_1, v_2, and v_3, are almost parallel, and they do not have a significant role in the system. Thus, the weighted least squares scheme is introduced and we apply larger weights to the equations with v_4 and v_5 than the other equations. Equations (4.19)-(4.21) are weighted twice than other equations. The vector v_2 and v_3 are quite sensitive to the errors and result bad approximations. Thus, the equations without v_2 and v_3, but with v_4, and v_5 are more weighted and gave better result empirically. The sensitiveness of the equation will be shown with an example in next chapter.

A large degree rotation of the camera rig by a sharp turn of the vehicle is also an important

case. In this case there are not enough matches for v_4 or v_5, because either the right or left camera will not see the same features as the center view at a prior time step. When we observe an inconsistent result from one of the mono view relative pose estimation solutions, a scheme to reject one of the five vectors, v_1, v_2, v_3, v_4, or v_5, is applied. The vector with the largest error after the least squares approximation is rejected and the least squares approximation is repeated to improve the result. Omitting a translation vector from the system not only reduces the length of the unknown vector in the system, but also reduces the 10 constraints to 6 constraints. This rejection method is helpful in the unstable cases due to inaccuracy in the relative pose calibration for the three cameras, and when temporal synchronization inaccuracies are exacerbated by jitter due to uneven road surfaces.

4.4 The Critical Condition

The critical condition is called when some degenerated situations happens on the motion of camera rig. One example is pure translation. Another degenerate situation is that pure rotation, which usually happens with a camera rig, which can pan and/or tilt itself. In VEE recording cart, the most common critical condition we suffer is the pure translation, in practical, near zero rotation, because the cart moves forward slowly for smooth videos. Without the proposition that the features seen on the frontal camera should appear on one of the side camera, we can not make sure the vector v_4 and v_5 exist in Figure 4.5.

Let us see what will be happen on our vector equation system without the vector v_4 and v_5 as seen in Figure 4.6, and thus the system of 10 vector equations in equation (4.25) is reduced to a system of three vector equations,

$$\begin{bmatrix} v_1 & -v_2 & 0 \\ v_1 & 0 & -v_3 \\ 0 & v_2 & -v_3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} t_{cl} - t'_{cl} \\ t_{cr} - t'_{cr} \\ -t_{cl} + t'_{cl} + t_{cr} - t'_{cr} \end{bmatrix}. \tag{4.26}$$

Whereas the set of equations (4.26) is sufficient to solve for the unknown scaling factors a_1, a_2, and a_3 under normal conditions, some camera motions can lead to singular conditions. For example, when moving in a perfectly straight line, all rotation matrices become the identity matrix and the system cannot be solved as seen in [65]. Of course, in pure translation, right hand sides of equation (4.26) is all zero since zero rotation means that the rotation matrix, R is an identity matrix. It transforms the vector t_{cl} to vector t'_{cl}, but is same as t_{cl}. In the same way, t_{cr} is equal to t_{cr}. Thus, the set of equations (4.26) goes trivial and can not solve the case. This problem can be tackled by adding the vectors v_4 and v_5. In our three camera visual odometry, the camera rig moves forward at a walking speed of about 2 $miles/h$. The purpose of our videos in [107] requires a smooth transition even if the user on the treadmill walks in the minimal speed, 0.5 $miles/h$. It cannot be faster than the walking speed for the video quality. There are typically only gradual turns, so it is assumed that all camera movement between frames is near the critical condition. The added vectors, v_4 and v_5, have the important role in the system because we have more forward motion than rotation, therefore the weights on equations with v_4 and v_5 are greater than the other weights for the weighted least squares method. Due to the sensitivity of the least squares method, if one motion estimate is an outlier, excluding that the vector from the solution helps stabilize the system. In that case, the matrix system shrinks to 6 vector equations and 4 unknowns. This rejection system is also effective for the case of sharp turns, where not enough matches are found for v_4 or v_5.

4.5 Summary

The visual odometry for a multi-camera system, which stereo algorithm can not be applicable with, is studied. Existing methods have restrictions to apply on our VEE application and a new approach is necessary to solve the problem. By the observation that feature points, seen in the forward looking camera, eventually appear on side cameras, the multi-camera visual odometry with additional time lapse relative pose estimation is proposed. The method proposed in this chapter is stable even in the critical condition.

Figure 4.2: Three camcorders on the camera rig

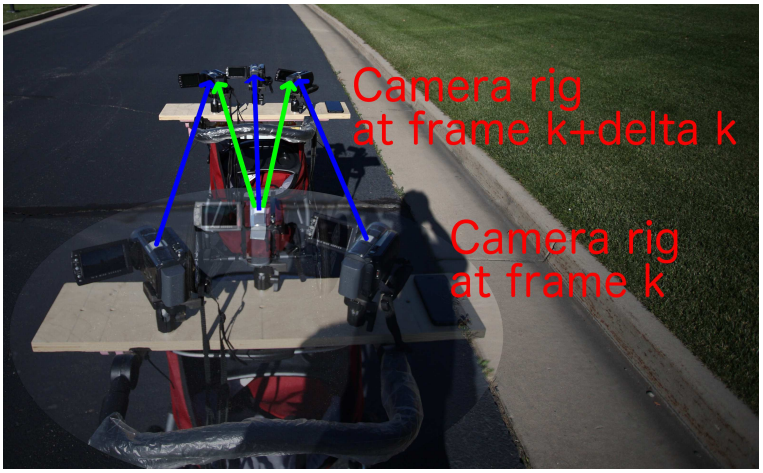


Figure 4.3: The basic idea is that the features on front camera on time time k should appear on one of the side camera at the time $k + \Delta k$.

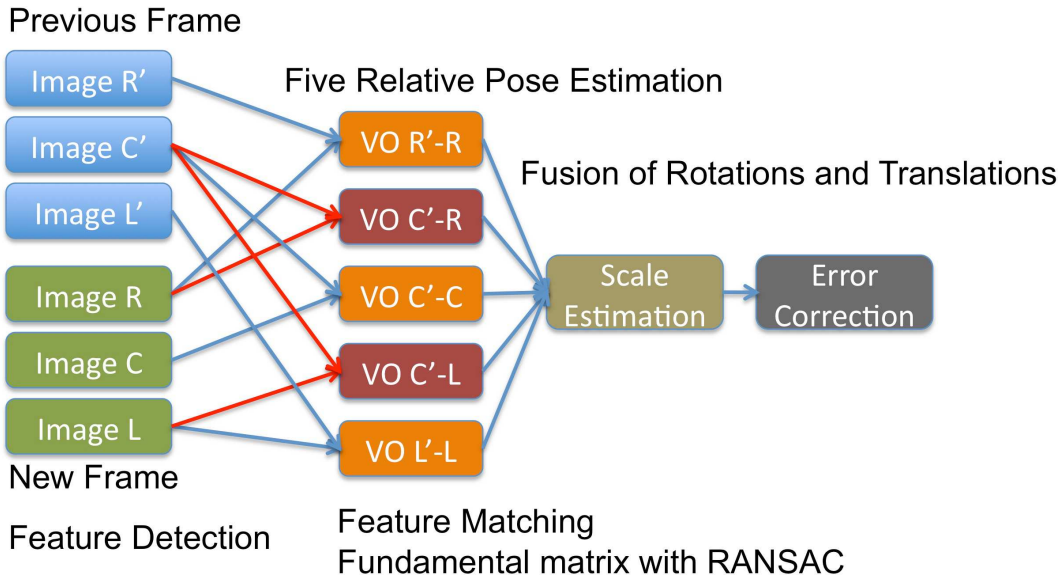


Figure 4.4: System Overview: Images from three cameras for five mono view relative pose estimations, with which scale estimation for metric distance, and the correction of the scale factors when too much errors are calculated for one frame step.

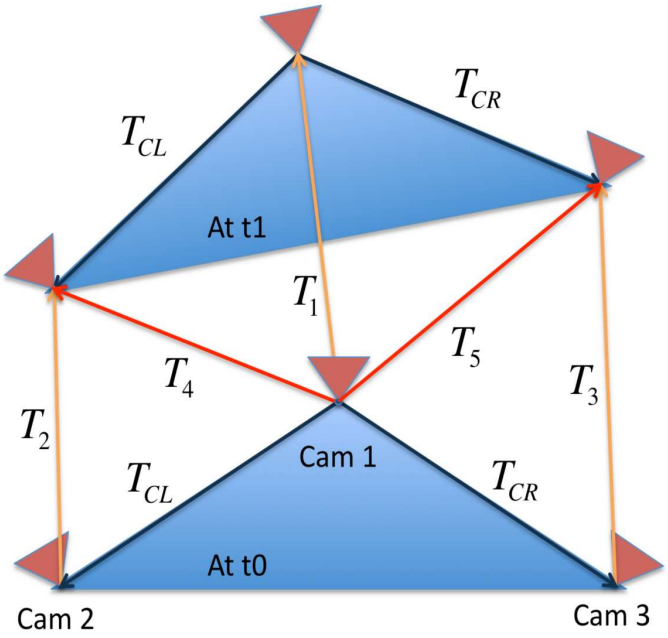


Figure 4.5: Basic Geometry, where T is a 4 by 4 transformation matrix, which has the information of rotation and translation.

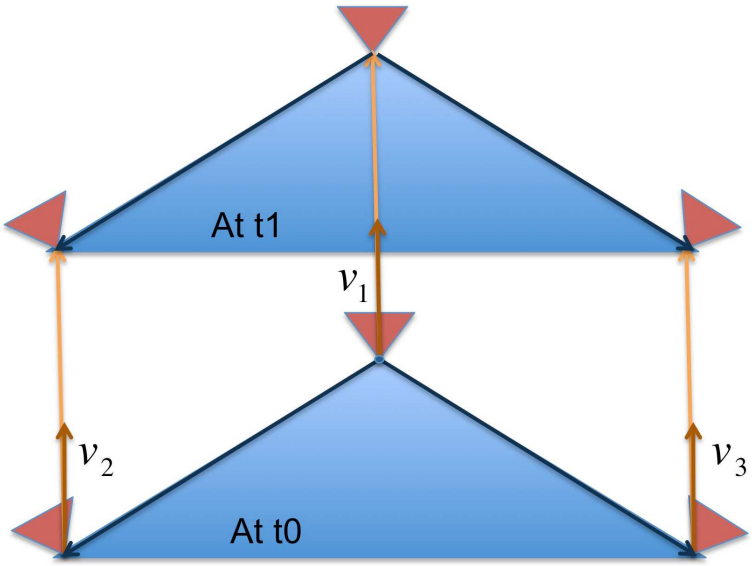


Figure 4.6: Basic Geometry without the v_4 and v_5 on a degenerated case of pure translation.

Chapter 5

Experimental Results

As a physical system introduces a series of complications, such as errors from the temporal difference with unsynchronized shutters, bad camera calibration, or pixel errors on video recorded with off-the-shelf camcorders, we conduct a first series of experiments using synthetic data to test the proposed method is absence of these problems. This also allows us to test exact straight-line movement, which is difficult to obtain experimentally. We then apply the proposed method to real-world data sets captured using the system shown in Figure 2.10. First data set is a short course near the engineering building. Through this data set, we investigate how one bad estimation of five relative pose estimation effects on the whole result. Then, we will see how the problem solved. Next, the proposed algorithm is applied to a longer 20 minutes track for the VEE program. Due to the thousands of images, it takes a lot of time. Thus, some speed up techniques are tried and discussed.

5.1 Synthetic Data Experiment and Gaussian Pixel Errors

Figure 5.1 shows the synthetic data used to test the diverging-view VO system. The camera test path is generated by **a)** a sine function and **b)** a straight line. Both paths pass 1000 random 3D data points, which are used to generate uniquely identifiable image features for the environment. The camera rig moves from $(0, 0, 1)$ to $(10, 0, 1)$. 1000 data points are set in the space which spans $[0, 30]$ on x axis, $[-30, 30]$ on y axis, and $[0, 3]$ on z axis. The arrows in Figure 5.1 indicate the axis

of camera rig. The intrinsic camera matrix, K, used here, is

$$K = \begin{bmatrix} 1000 & 0 & 500 \\ 0 & 1000 & 500 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.1}$$

Figure 5.2 and 5.3 show the synthetic camera track plotted with blue o's and the calculated results with red stars. It shows a perfect match, and the machine errors only exist.

We now test the proposed method with Gaussian distributed pixel errors on matching features (from 0.2 to 1 pixel errors). The pixel error applies on the image plane for each projection, to which each 3D synthetic point is projected through a camera matrix. For the same situation we already did with the synthetic data, the pixel errors are added, and the results are plotted on Figures. Figure 5.4 and Figure 5.5 show the absolute and relative errors on the scale factor with respect to Gaussian correspondence error with σ ranging from 0.2 to 1 pixels. Figure 5.6 shows the rotation error in degrees. Figure 5.7 shows the five different paths. The path with an error rate of 0.2 pixel is quite good and the one with 0.4 pixel error is still acceptable except one big jump during the calculation. Cases with 0.8 and more than 0.8 pixel errors illustrate how bad the results become due to accumulated error. For the error rate of more than 0.8 pixels, the algorithm needs another scheme that can filter unexpected jumps and the wrong cheirality. By the way, 100 times of RANSAC trial was good enough to find the best model for all matching points. For the strait motion with the pixel error, there is much less errors than the sinusoidal motion, and Figure 5.7 shows very little differences compared to the non-error model.

These results show that our algorithm is quite sensitive on pixel errors but also show acceptable results within 0.6 pixel errors. Thus, for real scenes, a feature extraction scheme with high accuracy is required. In practice, SIFT[58] gives us good matches in resolution. In addition, a high standard on the threshold to distinguish the inliers is necessary.

5.2 Real Data Experiment and the Erroneous Estimation Rejection Scheme

We performed a real data experiment on a set of short videos captured near the engineering building. Our algorithm is tested with a 3100 frame sequence recorded at $30fps$, making VO estimates at every $\Delta k = 15$ frame (≈ 200 pose estimates over 130 meters). The speed of cart with three camcorders on recording is regular walking speed, 2 $miles/h$. The distance between center cam and left cam is about 18 centimeters. For stable scale factor, at least one for 30 frames should be computed, its the maximum frame step, in practice. For sharp turns, in which the features are not enough on the comparison, the frame step cannot be too large. The outer side seeing camera may have a lack of features at sharp turn, and that situation is dealt with the rejection scheme. However, if the turn is so sharp and thus there are not enough matches on the front looking camera, it fails. Another significant error we should consider is shutter synchronization. Without any shutter synchronization method, each frame has the maximal temporal differences of 16.7ms between camcorders. With the frame step, $\Delta k = 15$ and $30fps$, the time stamps between two consecutive frames are $500 \pm 16.7ms$, making the error from the lack of synchronization negligible. Thus, we start with 15 for Δk, and will investigate with variable frame steps in a longer track. The RANSAC trial for this video is 1000 trial or 70% of matches are good for the case was used, and the threshold of pixel error for inliers is 1.0 pixels as sum of mutual-reprojections.

In Figure 5.8, the superimposed green dots, blue dots, and red dots illustrate the matched features between a temporal frame k and the next processed frame $k+15$, on left, center, and right sequence images respectively. The cyan dots show the matched features between a center view for frame k (Figure 5.8(e)) and the left view for frame $k+15$ (Figure 5.8(a)), and the magenta dots do the same between the center view for frame k and the right view for frame $k+15$ (Figure 5.8(c)).

A typical result for a scale factor calculation is shown in Figure 5.9. A cyan line shows the camera rig pose at frame k, and the blue line shows the camera rig pose at frame $k+15$. The green lines show the estimated translation vectors for the left, center and right camera. The red and magenta lines show the vectors v_4 and v_5. The right vector shows some obvious errors.

This erroneous situation happens when a wheel of the stroller rolls over a small rock. Even if we increased the frame step, Δk, there is still time difference is exist and the mono view relative pose estimation of the right camcorder makes errors, in this example.

In Figure 5.10, the right vector, v_3 is removed and the scale vector of the motion is recalculated. This means that the 2nd, 4th, 8th and 9th rows of the matrix equation (4.25) are removed in the error correction. Note that the scale factor of the front camcorder, where locates the origin of camera rig coordinate. Before the erroneous v_3 removed, the scale factor is less than 0.5. After the v_3 is removed, the scale factor is bigger than 0.6. All the metric scale used in chapter is *meter*, if not mentioned any other metrics.

Figure 5.11 plots the estimated camera path for the engineering building image sequence. While going forward around 130 meters there are two smooth turns. Then, at around the 10 meter mark, the motion changes to the $-x$ direction. In addition, since the camera rig was not perfectly level at the start position, there is 6 meters of height change. The GPS data are recorded simultaneously by an iPhone app[32] and appear with O markers in Figure 5.11. Note that the scattered position of GPS points shows the sampling frequency. The accuracy of the GPS data is known to be less than one meter. This example demonstrates the robustness of our approach to stabilize the critical straight-line motions as well as upward motion.

5.3 The East Campus Videos and Speed Up Techiniques

The East Campus full HD videos, resolution of 1920 by 1080 pixels, recorded at the east campus are around 12 minutes long and have 24000+ frames for each camera. The closed-loop trail is seen on Figure 5.12 and consists of a tree-lined recreational trail around a large office building. Figure 5.12 also shows 1000+ pose estimations on East Campus loop, with 15 frame step on Full HD scale images. Note that the GPS ground truth shows roughness at some corners. The estimated track form the smooth shape of the track. However, there is 30 meters difference from the start position to the end position on estimation and 30 meters difference on elevations. A drift comes from the accumulation of errors on each pose estimation. If the camera rig is not level at the start

position, it gets higher elevations at the end position, even though the end position is same as the start position. There are some researchers studying to fix the drift problems [92], and it is postponed to future works, in this thesis.

Table 5.1: Average numbers of features extracted and matched with time consumed, number of inliers after mono view relative pose estimatoin with time spent, and seconds for our algorithm and Sparse Bundle Adjustment algorithm. The East Campus videos are tested for this scaled down cases.

	Full HD	Half scale	Quarter scale	1/6 scale	1/8 scale
Features per Image	11388	3659	994	403	272
Time for Feature Extraction	11.65	3.22	0.84	0.39	0.28
Matches on Center	959	756	315	134	112
Matches on Sides	547	459	215	99	83
Matches on Center-Sides	327	277	166	46	35
Time for Matching	54.79	5.70	0.31	0.045	0.032
Inliers through Mono View VOs	449	395	178	76	62
Time for Mono View VOs	43.38	3.47	0.526	0.465	0.330
Time for Tricam Algorithm	0.01288	0.0012	0.0018	0.00147	0.00204
Time for SBA	10.27	7.12	2.88	1.84	1.178

As shown in Table 5.1, the number of SIFT features extracted from an image is usually more than ten thousand, and the matched features between consecutive images for forward motion is around one thousand, and around 500 for sideways motion, and the features moving form the center to the sides are around 300 when the frame step is 15. As seen in Table 5.1 , the most time consuming process is to find the matches between images for the full HD images. Finding matches takes around two minutes for a frame of the camera rig movement. The full-HD images from the cameras have the resolution of 1920 by 1080 pixels. We post-processed the existing video by down-sampling it from full HD to half (960 by 540), quarter (480 by 270), 1/6 (320 by 180), and 1/8 (240 by 135) scales, and measured the time it takes to gather a position estimate from frame to frame as well as the resulting accuracy. The number of feature points and matching features between images are gradually decreasing as the scale of the input images is reduced. Since the 8-point algorithm is used for the mono view visual odometry for each camera, the tricam algorithm theoretically also only needs 8 correct matches in theory. However, passing cars and other disturbances in the videos

requires additional features to achieve stable numerical solutions. To be done at least one frame in a second, the images should be scaled down to 1/6 scale. All the time measures in Table 5.1 is seconds, and measured in Matlab environment on Windows 7 platform with an Intel Core2Quad 9450 processor. The Sparse Bundle Adjust algorithm[57] is also tested and measured in Matlab with mex compiler, just for the comparison of time consumption against our method. Figure 5.13 shows the tracks on the different scales when the frame step, Δk, is fixed on 15. It shows that the 1/6 scale is not acceptable and the quarter scale is the best for the time efficiency. Figure 5.13 shows that the time spent for a frame along with the scales in log scale. The 1/6 scale is not so much faster than the quarter scale, and it would be the best if quarter scale can be done a little fast.

We also dropped more frames to increase speed. In order to be called '$real - time$', we need to be able to finish all calculations within 0.5 seconds when frame step is 15, while it can take one second when frame step is 30. Unfortunately, the frame step cannot be too large as this decreases the number of matched features between center and side cameras, which is critical for our algorithm. We tested from 15 to 29 frame steps at quarter scale, increasing the accumulated error over 1km distance by an order of magnitude. The errors are mostly made in two regions where a car passed by nearly and it also means that dropping frames affects more when matched points are unstable. The result in Figure 5.15 shows that the increased frame step around 23 was quite acceptable, and if bigger than that, the moving objects in the scene disturbed more.

To speed up, the Matlab code is parallelized for the quad-core processor. The algorithm has several steps and some of them is independent and can be parallelized easily. Searching features on three new images can be parallelized to three tasks. Matching features for five different relative pose estimation are divided to five tasks. The Core2Quad has only 4 cores, but the additional vectors, v_4 and v_5 does not take much times since there are not so much matches than the front view. For a better efficiency, the RANSAC process also is able to be parallelized.

Since the most time consuming process in the algorithm is feature extracting and matching, the SURF [6], instead of the SIFT features, are tried. Figure 5.3 shows that the comparison of SIFT

Table 5.2: Average numbers of matched points and Inliers after RANSAC, and average time consumed for each process. Since the RANSAC threshold is set to 0.75, SIFT with $\Delta k = 15$ aborts RANSAC loop most quickly among the cases.

	SIFT,$\Delta k = 15$	SIFT,$\Delta k = 21$	SURF,$\Delta k = 15$	SURF,$\Delta k = 21$
Feature Extraction Time	3.22	3.14	0.312	0.316
Matches on Center	756	658	287	254
Matches on Sides	459	358	201	165
Matches on Center-Sides	277	255	103	94
Time for Matching	5.70	5.82	0.178	0.128
Inliers on Center	683	658	202	172
Inliers on Sides	404	243	106	82
Inliers on Center-Sides	242	161	57	51
Time for RANSAC	3.47	128.8	47.7	40.76
Matches-Inliers ratio	0.886	0.705	0.595	0.564

and SURF features on feature matching. SURF has less features and thus less matches. Figure 5.17 shows the comparison of the tracks of SIFT and SURF features with the frame step of 15 and 21 on half scale images. On quarter scale, SURF does not make enough match points, regularly. The GPS ground truth is a black line, and the SURF features can be a good alternative for SIFT on the half scale videos. Table 5.2 shows that the time spent on each steps. Note that the SURF consumes significantly small time on feature extraction, and less than half features compared on SIFT. Thus, SURF spent less time on matching process. The RANSAC trial time tells something more. Since the threshold for breaking from the RANSAC cycle is three quarter of matching point, the SIFT with frame step of 15 breaks out the loop quickly, but others takes all the maximum number of RANSAC trials and the time spent depends on the number of matched features. The FREAK[72] feature is also tested, but FREAK does not make enough features even on half scale videos. The BRISK[53] feature is also considered, but has the binary descriptor as FREAK and not tested.

5.4 A Limitation of the Algorithm

The proposed algorithm, adding time lapse pose estimations between the front camera and a side camera, gives us reliable result on our VEE camera rig. We tried to expand our algorithm to other multi-camera panoramic system, the Point Grey's ladybug camera[76]. Figure 5.18 shows the

ladybug camera, which has six CCD sensors. We do not have the ladybug camera, we downloaded the New College Data set[87]. It includes ladybug images, ground truth, and other data with the other equipments. Each image has 384 by 512 pixel resolution, and the recording frequency is 3Hz. Our system of equation needs just three of them and only the frontal, front-left, front-right images are in computation.

The result is not good enough. The ladybug camera gets less than two frames per a second, and it means that all the frames should be computed. The vehicle which equipped all the sensors moves to fast and it makes hard to find matched features. Most important, the distance between cameras is too small. Our VEE systems has $20cm$ between the front camera and a side camera, and it gives us the accuracy in one meter of movement. When the VEE cart move forward 1 meter, the angle made at the previous frontal camera for current frontal camera and the current side camera is 11.3 degrees and it is large enough for calculation. However, the ladybug camera has less than $20mm$ for the distance between adjacent cameras, and the movement of the vehicle was so fast that the distance between frames are larger than 1 meter. In that range, the vector equation system, our algorithm based on, loses its accuracy. Even though we could not get a meaningful result, the limits of our algorithm is revealed.

5.5 Disscussion

We have proposed a 'divide and conquer' approach for visual odometry on divergent camera views, which provides robust estimates of translation and rotational motion even when camera overlap is small and camera motion is straight. The small overlap prevented us from using more common approaches, such as to rely on internally calibrated cameras and the 5-point algorithm[67]. The 5-point algorithm does not provide us with the scale factor therefore could not be used in our application, however.

Similarly, we chose to combine the constraints from three cameras by introducing additional correspondences between consecutive images in the outer and central camera instead of treating the whole system as one non-central camera as suggested by [36] and using generalized P3P [71, 75]

as the small amount of overlap in our data is not sufficient for robust stereo matching.

Another common challenge with visual odometry is to distinguish moving from static objects. If there is a moving object like a pedestrian or a car on the videos, some SIFT features are also on the moving objects and confuse the estimate. In these cases, RANSAC rejects the SIFT features on the moving objects. In addition, the proposed scheme to reject those vector with obvious errors helps to stabilize the results.

The most time consuming processes in our algorithm are comparison of features between scenes (whose complexity is quadratic to the number of features) and the RANSAC step. Due to the HD resolution of the video, usually more than ten thousand of SIFT features are detected and the comparison of features takes a lot of time (in the order of minutes on a Core2 processor). The RANSAC algorithm terminates in a few seconds, but it could be speed up by tuning the proportion of inliers and how many features should be included.

We devised scaling down and dropping frames to make the proposed algorithm faster. Figure 5.14 shows the time spent to process one frame, including feature extraction, feature matching, five mono view visual odometries and the tricam algorithm for the scale factors. Since the time consumed depends on the number of features, the calculation time decreases exponentially with scale. At least one pose estimation per a second is necessary for real time calculation, which we achieve only at 1/6 scale, although with a poor accuracy. We also observe that dropping frames has strong impact on accuracy in comparison to reducing scale. This can be explained by the observation that resolutions down to a fourth of Full HD still provide sufficient number of features, but dropping frames loses important information in particular when perturbed with passing-by objects.

5.6 Summary

The algorithm tests with synthetic data and real video are performed. With the synthetic data, the proposed algorithm is proved with the assumption that the feature point on frontal camera should appear on side cameras with time lapse when the camera rig moves forward. In addition, the stability of the algorithm is found and the limit to use the proposed algorithm, the pixel error

on the image plane must less than 0.5 pixels to get a reliable estimation.

With the short videos around the engineering building, the proposed algorithm is necessary to deal with the errors from the shutter synchronization problem and from inaccurate camera calibrations. The most errored estimation of the five relative pose estimation is ignored and the result get better than the result with all five estimations.

For the east campus videos, which is long enough for the VEE program, the computing time is concerned, and some speed-up techniques are tested. Various frame step test finds the most time effective frame steps. Tests with scaled down images shows that the quarter scale is good enough for the proposed algorithm. Introducing other features shows the advantage of SURF features on time consumption and its limits on the number of features and the accuracy. FREAK feature algorithm is faster than any other algorithm but has not enough features. It seems that the features with short descriptor is not good on nature scenes since there are so many of similar features such as trees and grasses, which is hard to distinguish.

Expansion to the ladybug panorama camera reveals the limit of the proposed algorithm. The distance between adjacent camera is directly related to the accuracy of the algorithm. If the larger the distance between cameras, the better the results is. However, the distance can not be too large since the step on time frame is also related.

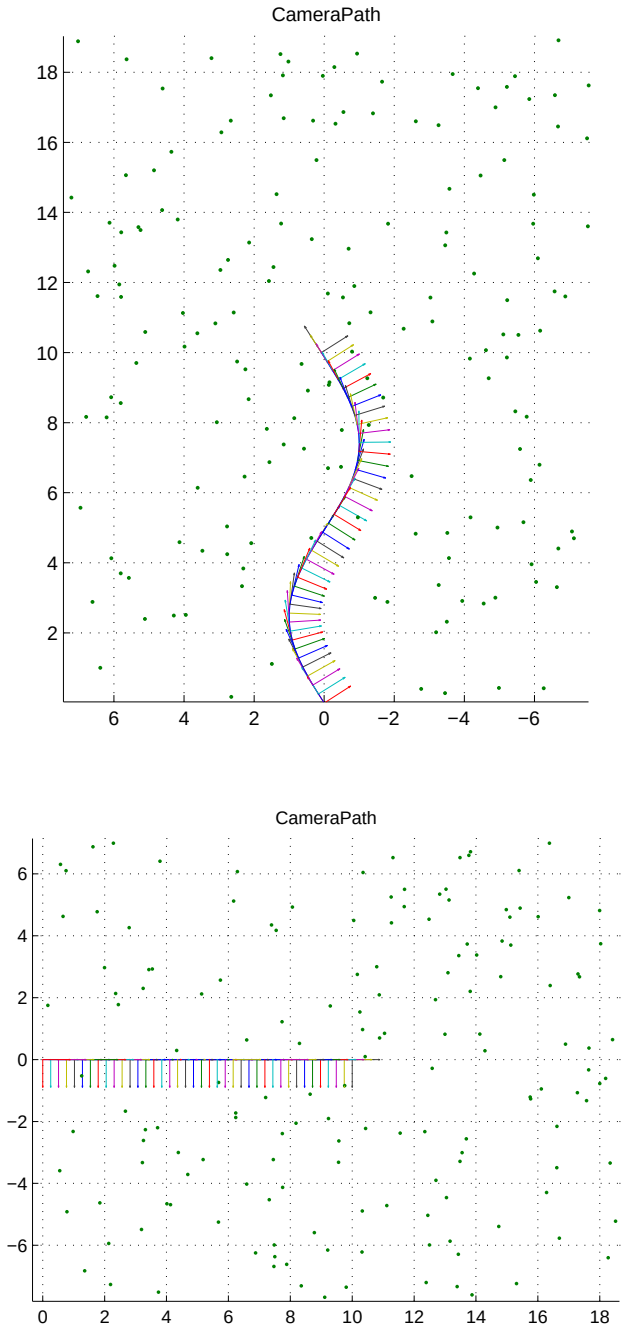


Figure 5.1: Synthetic camera path and Data points for a sinusoidal (top) and straight path (bottom).

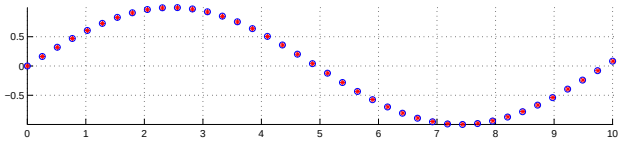


Figure 5.2: Synthetic camera path (blue o) and estimated path (red *)

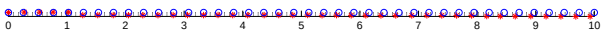


Figure 5.3: Synthetic camera path (blue o) and estimated path (red *)

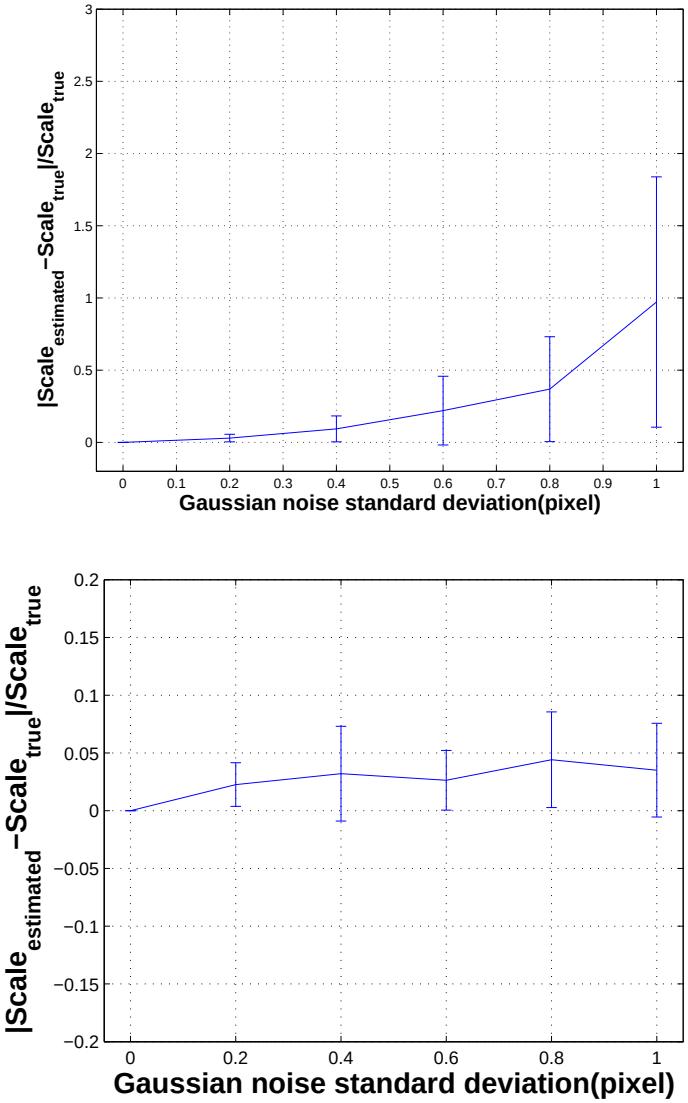


Figure 5.4: Scale Errors for pixel errors for sinusoidal (top) and straight motion (bottom).

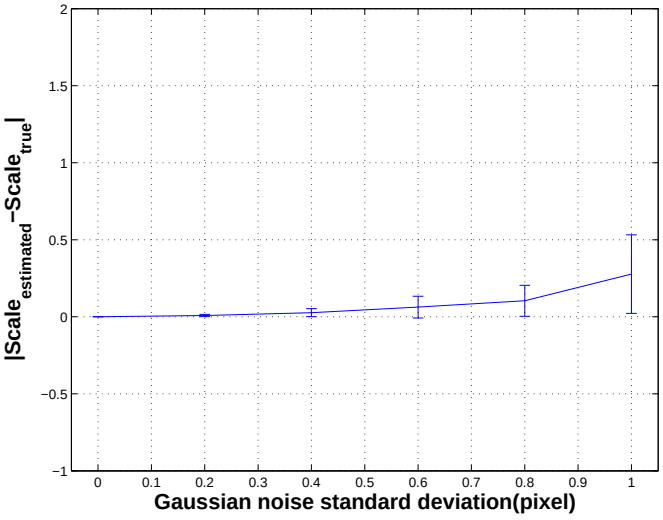


Figure 5.5: Relative Scale Errors for pixel errors during sinusoidal motion.

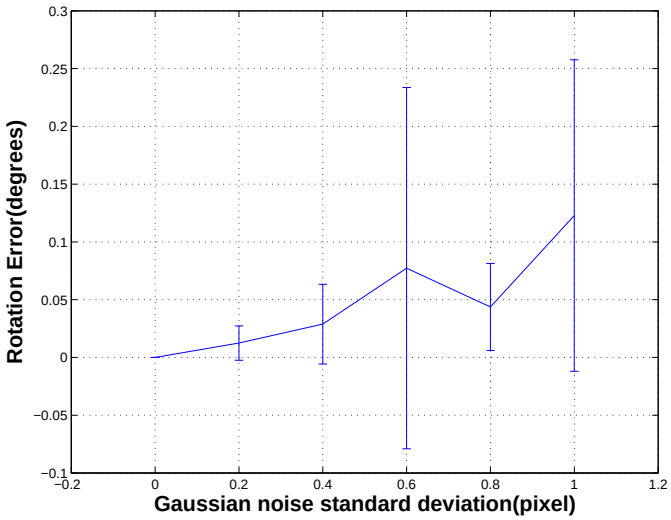


Figure 5.6: Rotation Errors for pixel errors during sinusoidal motion.

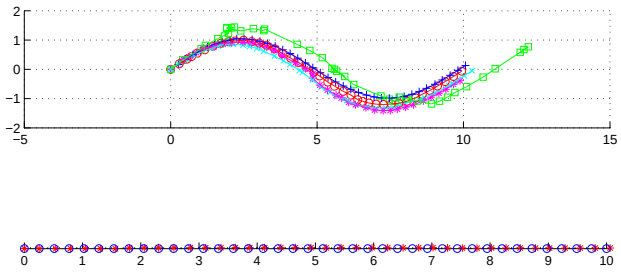
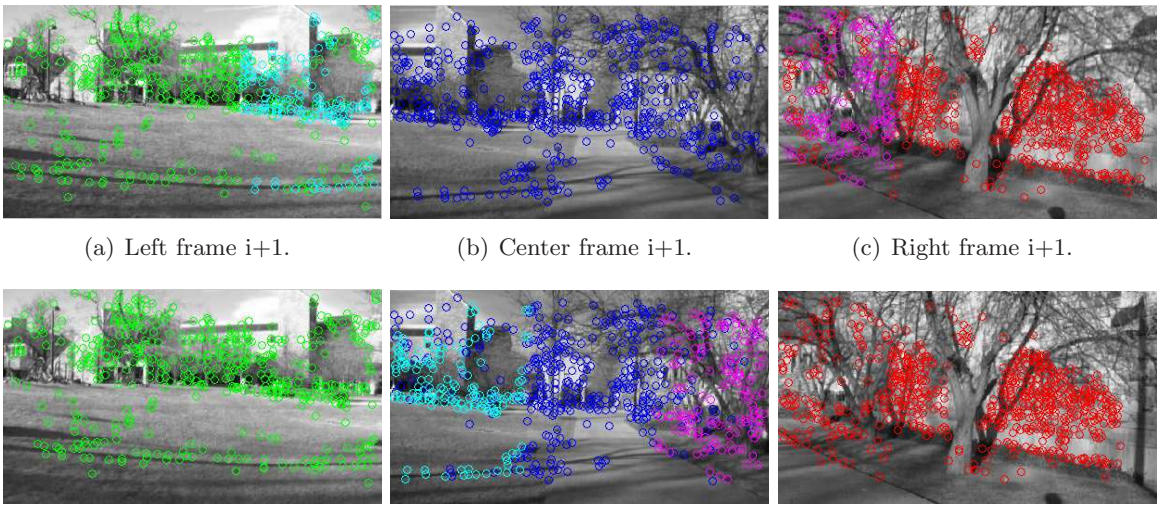


Figure 5.7: Camera paths with pixel errors The marker +, o, star, cross and square shows for the path with 0.2, 0.4, 0.6, 0.8 and 1 pixel errors, respectively, for sinusoidal (top) and straight motion (bottom).



(a) Left frame i+1. (b) Center frame i+1. (c) Right frame i+1.

(d) Left frame i. (e) Center frame i. (f) Right frame i.

Figure 5.8: Spatial and temporal correspondence patterns for consecutive frames.

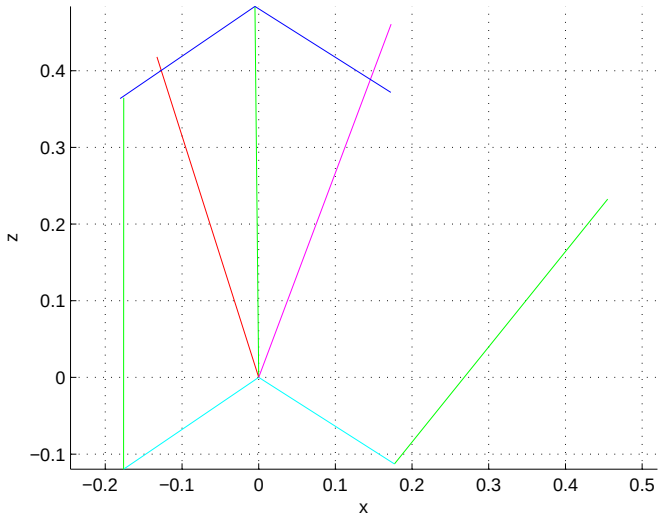


Figure 5.9: Estimated camera motion between frames shown in Figure 5.8 showing an errorneous estimation of the translation vector for the right camera.

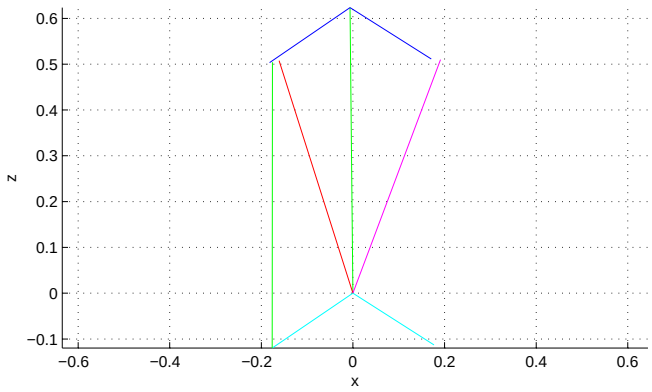


Figure 5.10: The camera motion between frames in Figure 5.8 after v_3 is removed.

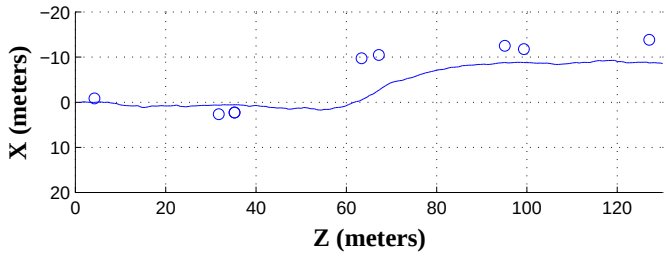


Figure 5.11: The camera path through frame 501 to frame 3636

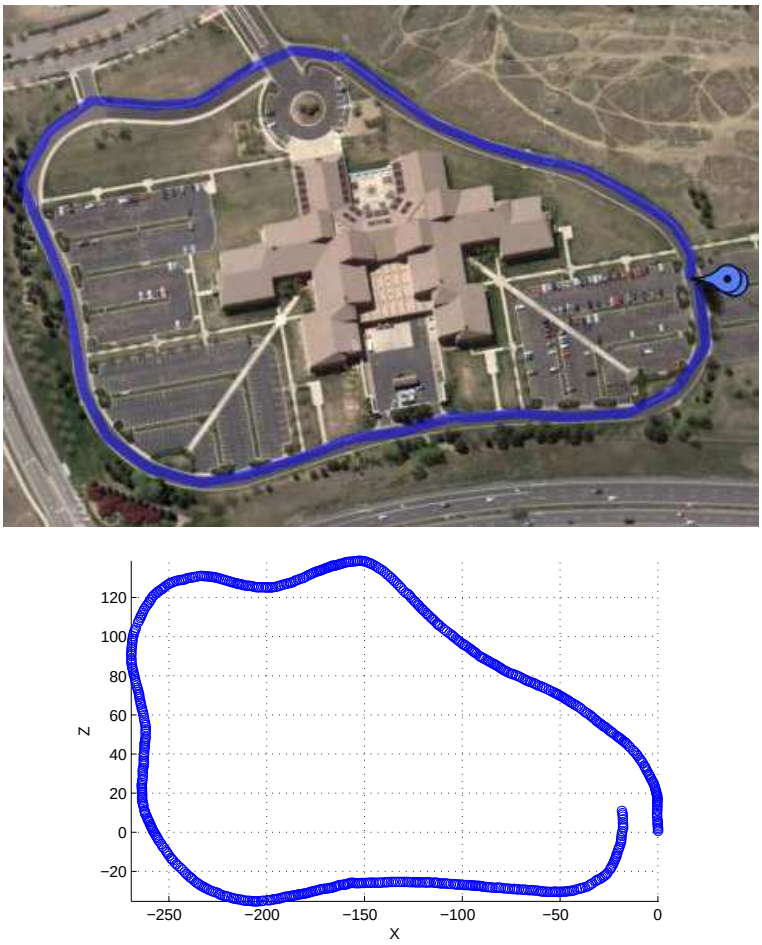


Figure 5.12: The camera path on East Campus. Ground truth data (top) and open-loop visual odometry showing drift after return to the initial position.

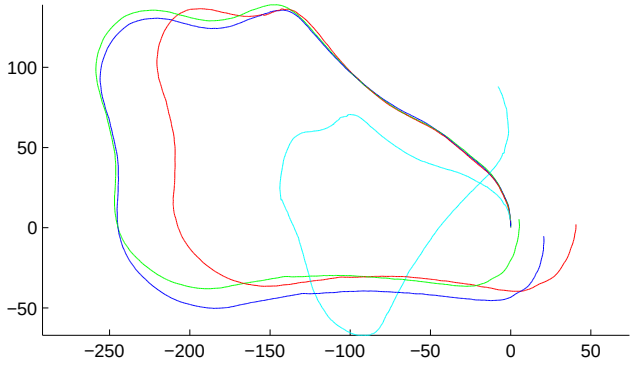


Figure 5.13: Estimated tracks of Full HD, Half, Quarter, and 1/6 scale, which corresponds blue, green, red, and cyan, respectively.

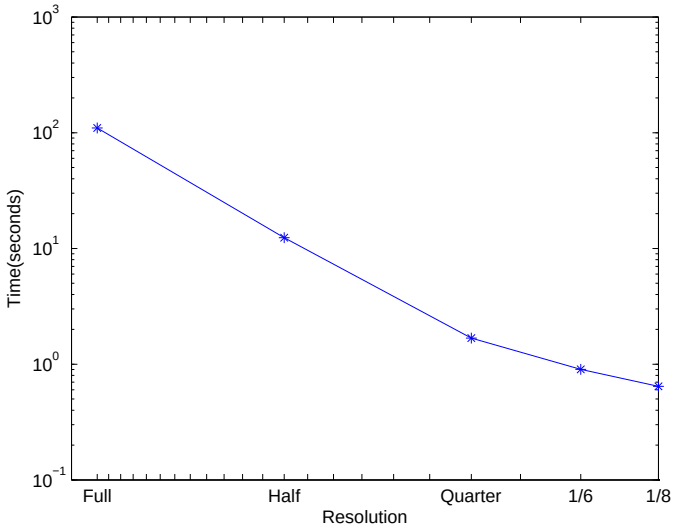


Figure 5.14: Scale down vs Time consumption graph in log scale

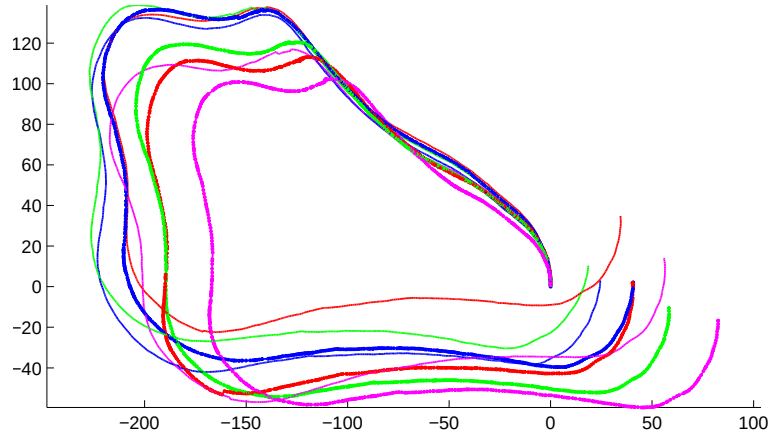
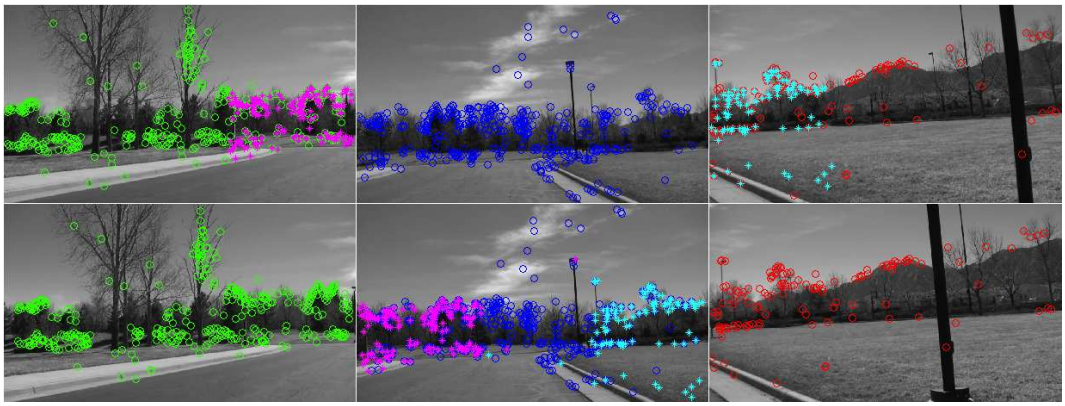
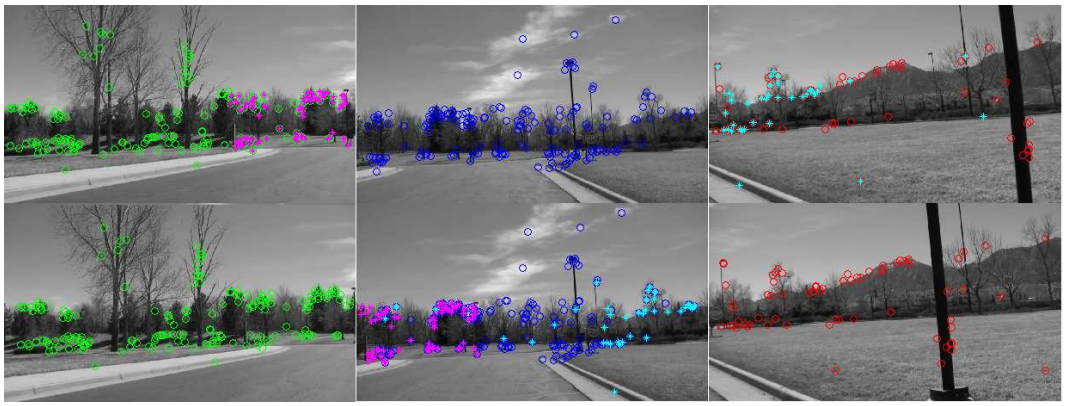


Figure 5.15: Estimated tracks with Frame step of 15, 17, 19, 21, 23, 25, 27, and 29, in blue, red, green, magenta, thick blue, thick red, thick green, and thick magenta



(a) The matched SIFT features on frame 10036(up) and 10021(down)



(b) The matched SURF features on frame 10036(up) and 10021(down)

Figure 5.16: Comparison of SIFT and SURF features on Half scale East Campus videos

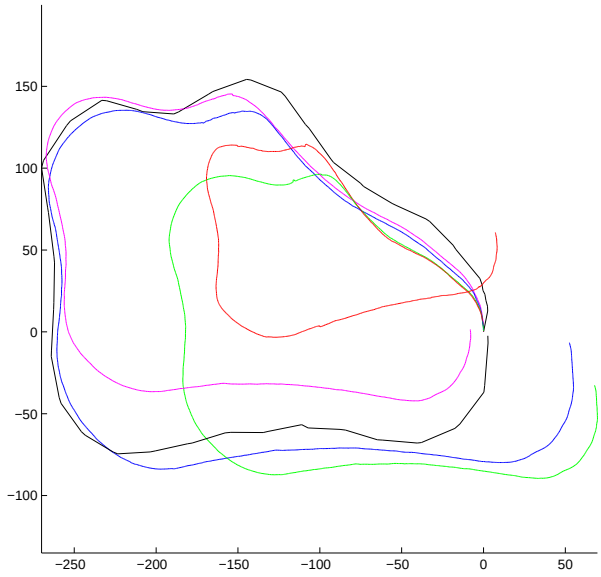


Figure 5.17: The black line is the tracked GPS data. The blue and green lines are the track with SURF features with frame step of 15 and 21, respectively. The red and magenta lines are the track with SIFT features with frame step of 15 and 21, respectively.

Figure 5.18: Ladybug panoramic camera with 6 small image sensors, courtesy of PointGrey.

Chapter 6

Conclusion and Future work

I have proposed a novel way to solve a visual odometry problem for divergent camera using a 'divide and conquer' approach and exploiting correspondences from consecutive image frames. This allows us to reduce the trinocular visual odometry problem to five mono view visual odometry problems, which in turns allows for robust odometry estimates even in underdefined situations such as straight line motion, and drastically reduces computational time when compared to the bundle adjustment method. This approach is applicable whenever the camera rig is constantly moving forward, which is the case in mapping and robotic application, where the status of the robot (moving or not moving) is known.

In this work, we assume the camera rig's speed to be nearly constant, which allows us to determine the distance between consecutive frames that lead to strong correspondences empirically. The step on time frame can be various in calculation with adaptive schemes.

An additional challenge for our method is that it is sensitive to the camera calibration which is difficult to achieve for camera rigs with limited view overlap. Auto calibration can help to resolve this problem with better accuracy. Also, a more robust visual odometry algorithm and weighting equations with machine learning may give better results.

There is always the drift problems. Since the error of the pose estimation for each frame is accumulated and the initial pose can not be level and see north exactly in nature, there exist the drift. To fix these drift problems, some extra correction process where the loop closes is necessary. A GPS signals or IMU data of a cellular phone also might be fused for a better result.

In further work, we wish to find ways to estimate the pose in '*realtime*' with faster environment than Matlab. Feature extracting and comparing algorithm is still studied for quicker and better methods. Maybe a better feature algorithm is developed in near future and it can be tested with the proposed algorithm.

For computation time, scaled down images and frame step is tested. Adaptive method on scaled down images also possible. A trade off of time and accuracy is always a deal to consider. If too much time taken, a calculation with scaled down images are better choice, but if too less matched points, back on the larger images are better. Of course, if more cores to access, these computations can be parallelized and done together in future.

Bibliography

[1] Gaurav Aggarwal, Soma Biswas, Patrick J. Flynn, and Kevin W. Bowyer. A sparse representation approach to face matching across plastic surgery. In WACV, pages 113–119. IEEE, 2012.

[2] Arduino. http://www.arduino.cc/.

[3] Sheldon Jay Axler. Linear Algebra Done Right. Springer, New York.

[4] Tim Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. Robotics Automation Magazine, IEEE, 13(3):108–117, Sept 2006.

[5] H. Harlyn Baker and Robert C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. In In IJCV, 1989.

[6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). Comput. Vis. Image Underst., 110(3):346–359, June 2008.

[7] J. Y. Bouguet. Camera calibration toolbox for Matlab, 2008.

[8] Matthew Brown and David G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In 3DIM, pages 56–63. IEEE Computer Society, 2005.

[9] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. Int. J. Comput. Vision, 74(1):59–73, August 2007.

[10] R. Bunschoten and B. Krose. Visual odometry from an omnidirectional vision system. In Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, volume 1, pages 577–583 vol.1, 2003.

[11] Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. ACM Trans. Graph., 2(4):217–236, October 1983.

[12] Jason Campbell, Rahul Sukthankar, Illah Nourbakhsh, and Aroon Pahwa. A robust visual odometry and precipice detection system using consumergrade monocular vision. In in Proceedings of the 2005 IEEE International Conference on Robotics and Automation ICRA 2005, pages 3421–3427, 2005.

[13] Rodrigo Carceroni, Ankita Kumar, and Kostas Daniilidis. Structure from motion with known camera positions. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1, CVPR '06, pages 477–484, Washington, DC, USA, 2006. IEEE Computer Society.

[14] G. Carrera, A. Angeli, and A.J. Davison. Slam-based automatic extrinsic calibration of a multi-camera rig. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 2652–2659, May 2011.

[15] Gerardo Carrera, Adrien Angeli, and Andrew J. Davison. Lightweight slam and navigation with a multi-camera rig. In Achim J. Lilienthal, editor, ECMR, pages 77–82. Learning Systems Lab, AASS, rebro University, 2011.

[16] Y. Caspi and M. Irani. Alignment of non-overlapping sequences. In Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, volume 2, pages 76–83 vol.2, 2001.

[17] Yang Cheng, Mark W. Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging. IEEE Robot. Automat. Mag., 13(2):54–62, 2006.

[18] Howie Choset and Keiji Nagatani. Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. IEEE T. Robotics and Automation, 17(2):125–137, 2001.

[19] Javier Civera, Oscar G. Grasa, Andrew J. Davison, and J. M. M. Montiel. 1pointt ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry. J. Field Robot., 27(5):609–631, September 2010.

[20] Brian Clipp, Jae-Hak Kim, Jan-Michael Frahm, Marc Pollefeys, and Richard I. Hartley. Robust 6dof motion estimation for non-overlapping, multi-camera systems. In WACV, pages 1–8. IEEE Computer Society, 2008.

[21] Yuchao Dai, Mingyi He, Hongdong Li, and R. Hartley. Factorization-based structure-and-motion computation for generalized camera model. In Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on, pages 1–6, 2011.

[22] A.J. Davison, Y. González Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, July 2004.

[23] CMU 1394 Digital Camera Driver. http://www.cs.cmu.edu/ iwan/1394/.

[24] H. Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. Robotics Automation Magazine, IEEE, 13(2):99–110, June 2006.

[25] Damien Dusha and Luis Mejas. Error analysis and attitude observability of a monocular gps/visual odometry integrated navigation filter. I. J. Robotic Res., 31(6):714–737, 2012.

[26] eMagin Corporation. http://www.emagin.com/.

[27] Sandro Esquivel, Felix Woelk, and Reinhard Koch. Calibration of a multi-camera rig from non-overlapping views. In FredA. Hamprecht, Christoph Schnrr, and Bernd Jhne, editors, Pattern Recognition, volume 4713 of Lecture Notes in Computer Science, pages 82–91. Springer Berlin Heidelberg, 2007.

[28] Sandro Esquivel, Felix Woelk, and Reinhard Koch. Calibration of a multicamera rig from non-overlapping views. In IN LECTURE NOTES IN COMPUTER SCIENCE 4713 (DAGM, 2007.

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

[30] Communications Specification for Fitness Equipment. http://www.fitlinxx.com/csafe/.

[31] Jan-Michael Frahm, Kevin Kser, and Reinhard Koch. Pose estimation for multi-camera systems. In CarlEdward Rasmussen, HeinrichH. Blthoff, Bernhard Schlkopf, and MartinA. Giese, editors, Pattern Recognition, volume 3175 of Lecture Notes in Computer Science, pages 286–293. Springer Berlin Heidelberg, 2004.

[32] Inc. Fullpower Technologies. http://gps.motionx.com.

[33] Jae hak Kim, Jan michael Frahm, Marc Pollefeys, and Richard Hartley. Robust 6dof motion estimation for non-overlapping, multi-camera systems. In in IEEE Workshop on Applications of Computer Vision, pages 1–8, 2008.

[34] Chris Harris and Mike Stephens. A combined corner and edge detector. In In Proc. of Fourth Alvey Vision Conference, pages 147–151, 1988.

[35] Richard Hartley and Andrew Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[36] Michal Havlena, Toms Pajdla, and Kurt Cornelis. Structure from omnidirectional stereo rig motion for city modeling. In Alpesh Ranchordas and Helder Arajo, editors, VISAPP (2), pages 407–414. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2008.

[37] K. Hoffman and R.A. Kunze. Linear algebra. Prentice-Hall mathematics series. Prentice-Hall, 1971.

[38] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pages 3946–3952, 2008.

[39] Gibson Hu, Shoudong Huang, and Gamini Dissanayake. Evaluation of pose only slam. In IROS, pages 3732–3737. IEEE, 2010.

[40] Unibrain Inc. http://www.unibrain.com/.

[41] Jaeheon Jeong, Jane Mulligan, and Nikolaus Correll. Trinocular visual odometry for divergent views with minimal overlap. Robot Vision (WORV), 2013 IEEE Workshop on, 2013.

[42] M. Kaess and F. Dellaert. Probabilistic structure matching for visual SLAM with a multi-camera rig. Computer Vision and Image Understanding, CVIU, 114:286–296, feb 2010.

[43] T. Kazik, L. Kneip, J. Nikolic, M. Pollefeys, and R. Siegwart. Real-time 6d stereo visual odometry with non-overlapping fields of view. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 1529–1536, 2012.

[44] Jae-Hak Kim, Richard Hartley, Jan-Michael Frahm, and Marc Pollefeys. Visual odometry for non-overlapping views using second-order cone programming. In Yasushi Yagi, SingBing Kang, InSo Kweon, and Hongbin Zha, editors, Computer Vision ACCV 2007, volume 4844 of Lecture Notes in Computer Science, pages 353–362. Springer Berlin Heidelberg, 2007.

[45] Jae-Hak Kim, Hongdong Li, and R. Hartley. Motion estimation for multi-camera systems using global optimization. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, 2008.

[46] Jae-Hak Kim, Hongdong Li, and Richard I. Hartley. Motion estimation for nonoverlapping multicamera rigs: Linear algebraic and l_∞ geometric solutions. IEEE Trans. Pattern Anal. Mach. Intell., 32(6):1044–1059, 2010.

[47] Jun-Sik Kim, Myung Hwangbo, and Takeo Kanade. Spherical approximation for multiple cameras in motion estimation: Its applicability and advantages. Computer Vision and Image Understanding, 114(10):1068–1083, 2010.

[48] Jun-Sik Kim and Takeo Kanade. Degeneracy of the linear seventeen-point algorithm for generalized essential matrix. Journal of Mathematical Imaging and Vision, 37(1):40–48, 2010.

[49] Kurt Konolige. Sparse sparse bundle adjustment. In Proc. BMVC, pages 102.1–11, 2010. doi:10.5244/C.24.102.

[50] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss, and Alexander Kleiner. On measuring the accuracy of slam algorithms. Auton. Robots, 27(4):387–407, November 2009.

[51] Abhijit Kundu, K. Madhava Krishna, and C. V. Jawahar. Realtime multibody visual slam with a smoothly moving monocular camera. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, ICCV, pages 2080–2087. IEEE, 2011.

[52] Pierre Lébraly, Eric Royer, Omar Ait-Aider, and Michel Dhome. Calibration of non-overlapping cameras - application to vision-based robotics. In Proceedings of the British Machine Vision Conference, pages 10.1–10.12. BMVA Press, 2010. doi:10.5244/C.24.10.

[53] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, pages 2548–2555, Washington, DC, USA, 2011. IEEE Computer Society.

[54] Anat Levin and Richard Szeliski. Visual odometry and map correlation. In CVPR (1), pages 611–618, 2004.

[55] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In Proceedings of the 18th International Conference on Pattern Recognition - Volume 01, ICPR '06, pages 630–633, Washington, DC, USA, 2006. IEEE Computer Society.

[56] Microsoft Media Foundation Library. http://msdn.microsoft.com/en-us/library/windows/desktop/ms694197(v=vs.85).aspx.

[57] Manolis I. A. Lourakis and Antonis A. Argyros. Sba: a software package for generic sparse bundle adjustment. ACM Transactions on Mathematical Software, pages 1–30, 2009.

[58] David G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60:91–110, 2004.

[59] Pierre Lbraly, Eric Royer, Omar Ait-Aider, and Michel Dhome. Calibration of non-overlapping cameras—application to vision-based robotics. In Frdric Labrosse, Reyer Zwiggelaar, Yonghuai Liu, and Bernie Tiddeman, editors, BMVC, pages 1–12. British Machine Vision Association, 2010.

[60] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. Journal of Field Robotics, Special Issue on Space Robotics, 24:2007, 2007.

[61] Microsoft. http://www.skype.com/.

[62] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis & Machine Intelligence, 27(10):1615–1630, 2005.

[63] Annalisa Milella and Roland Siegwart. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In in IEEE International Conference on Computer Vision Systems, page 21, 2006.

[64] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 1, pages 363–370, 2006.

[65] Daniel Muhle, Steffen Abraham, Christian Heipke, and Manfred Wiggenhagen. Estimating the mutual orientation in a multi-camera system with a non overlapping field of view. In Uwe Stilla, Franz Rottensteiner, Helmut Mayer, Boris Jutzi, and Matthias Butenuth, editors, Photogrammetric Image Analysis, volume 6952 of Lecture Notes in Computer Science, pages 13–24. Springer Berlin Heidelberg, 2011.

[66] Richard A. Newcombe, Steven Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, ICCV, pages 2320–2327. IEEE, 2011.

[67] D. Nister. An efficient solution to the five-point relative pose problem. In Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, volume 2, pages II–195–202 vol.2, 2003.

[68] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 1, pages I–652–I–659 Vol.1, 2004.

[69] David Nistér. An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell., 26(6):756–777, June 2004.

[70] David Nistr, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. Journal of Field Robotics, 23:2006, 2006.

[71] David Nistr and Henrik Stewnius. A minimal solution to the generalised 3-point pose problem. Journal of Mathematical Imaging and Vision, 27(1):67–79, 2007.

[72] Raphael Ortiz. Freak: Fast retina keypoint. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12, pages 510–517, Washington, DC, USA, 2012. IEEE Computer Society.

[73] Taragay Oskiper, Zhiwei Zhu, Supun Samarasekera, and Rakesh Kumar. Visual odometry system using multiple stereo cameras and inertial measurement unit. In Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, pages 1–8, 2007.

[74] Frank Pagel. Calibration of non-overlapping cameras in vehicles. In Intelligent Vehicles Symposium, pages 1178–1183. IEEE, 2010.

[75] R. Pless. Using many cameras as one. In Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, volume 2, pages II–587–93 vol.2, 2003.

[76] PointGrey. http://www.ptgrey.com/products/ladybug/ladybug.pdf.

[77] M. Pollefeys, D. Nistr, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewnius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. International Journal of Computer Vision, 78(2-3):143–167, 2008.

[78] Marc Pollefeys, Reinhard Koch, and Luc J. Van Gool. Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters. In ICCV, pages 90–95, 1998.

[79] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. Int. J. Comput. Vision, 59(3):207–232, September 2004.

[80] M. E. Ragab and K. H. Wong. Multiple nonoverlapping camera pose estimation. In ICIP, pages 3253–3256. IEEE, 2010.

[81] Abhiram G. Ranade. Some uses of spectral methods in computer science.

[82] Cyril Roussillon, Aurelien Gonzalez, Joan Sol, Jean-Marie Codol, Nicolas Mansard, Simon Lacroix, and Michel Devy. Rt-slam: A generic and real-time visual slam implementation, 2012. cite arxiv:1201.5450Comment: 10 pages.

[83] T. Ruland, H. Loose, T. Pajdla, and L. Kruger. Hand-eye autocalibration of camera positions on vehicles. In Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pages 367–372, 2010.

[84] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. Robotics Automation Magazine, IEEE, 18(4):80–92, Dec 2011.

[85] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart. Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In Computer Vision, 2009 IEEE 12th International Conference on, pages 1413–1419, 2009.

[86] Stephen Se and Piotr Jasiobedzki. Instant scene modeler for crime scene reconstruction. cvprw, 0:123, 2005.

[87] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. The International Journal of Robotics Research, 28(5):595–599, May 2009.

[88] J. Sola. Multi-camera VSLAM: from former information losses to self-calibration. In International Conference on Intelligent RObots and Systems 2007, 2007.

[89] Joan Solà, Teresa Vidal-Calleja, Javier Civera, and José María Montiel. Impact of landmark parametrization on monocular ekf-slam with points and lines. Int. J. Comput. Vision, 97(3):339–368, May 2012.

[90] Henrik Stewenius and Kalle strm. Structure and motion problems for multiple rigidly moving cameras. In Tom Pajdla and Ji Matas, editors, Computer Vision - ECCV 2004, volume 3023 of Lecture Notes in Computer Science, pages 252–263. Springer Berlin Heidelberg, 2004.

[91] G. Strang. Linear Algebra and its Applications. Harcourt Brace Jovanovich, 3rd edition, 1988.

[92] H. Strasdat, J. M. M. Montiel, and A. Davison. Scale drift-aware large scale monocular slam. In Proceedings of Robotics: Science and Systems, Zaragoza, Spain, June 2010.

[93] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Real-time monocular slam: Why filter? In ICRA, pages 2657–2664. IEEE, 2010.

[94] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Editors choice article: Visual slam: Why filter? Image Vision Comput., 30(2):65–77, February 2012.

[95] Yasuyuki Sugaya and Ken ichi Kanatani. Highest accuracy fundamental matrix computation. In Yasushi Yagi, Sing Bing Kang, In-So Kweon, and Hongbin Zha, editors, ACCV (2), volume 4844 of Lecture Notes in Computer Science, pages 311–321. Springer, 2007.

[96] Yasuyuki Sugaya and Kenichi Kanatani. High accuracy computation of rank-constrained fundamental matrix. In BMVC. British Machine Vision Association, 2007.

[97] Google Street View System. http://www.google.com/maps/about/behind-the-scenes/streetview/.

[98] Nissan Around View System. http://www.nissan-global.com/en/technology/overview/avm.html.

[99] Thorsten Thormählen, Hellward Broszio, and Patrick Mikulastik. Robust linear auto-calibration of a moving camera from image sequences. In Proceedings of the 7th Asian Conference on Computer Vision - Volume Part II, ACCV'06, pages 71–80, Berlin, Heidelberg, 2006. Springer-Verlag.

[100] Learning Applied to Ground Robots(LAGR). http://www.rec.ri.cmu.edu/projects/lagr/.

[101] Lloyd N. Trefethen and David Bau. Numerical Linear Algebra. SIAM, 1997.

[102] B. Triggs. Autocalibration and the absolute quadric. In Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), CVPR '97, pages 609–, Washington, DC, USA, 1997. IEEE Computer Society.

[103] Bill Triggs, Philip Mclauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment a modern synthesis. In Vision Algorithms: Theory and Practice, LNCS, pages 298–375. Springer Verlag, 2000.

[104] Emanuele Trucco and Alessandro Verri. Introductory Techniques for 3-D Computer Vision. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[105] A. Whitehead, R. Laganiere, and P. Bose. Temporal synchronization of video sequences in theory and in practice. In Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on, volume 2, pages 132–137, 2005.

[106] Wei Xu. Panoramic Video Stitching. PhD thesis, University of Colorado, Boulder, 2012.

[107] Wei Xu, Jaeheon Jeong, and Jane Mulligan. Augmenting exercise systems with virtual exercise environment. In 5th International Symposium on Visual Computing (ISVC09), 2009.

[108] Wei Xu, Jaeheon Jeong, and Jane Mulligan. Augmenting exercise systems with virtual exercise environment. In Advances in Visual Computing, volume 5875 of Lecture Notes in Computer Science, pages 490–499. Springer Berlin Heidelberg, 2009.

[109] Wei Xu and Jane Mulligan. Robust relative pose estimation with integrated cheirality constraint. In ICPR, pages 1–4. IEEE, 2008.

[110] Wei Zhang and J. Kosecka. Ensemble method for robust motion estimation. In Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on, pages 100–100, June 2006.

[111] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision, pages 119–152, 1994.

[112] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. Int. J. Comput. Vision, 27(2):161–195, April 1998.

Appendix A

Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) is known as most useful matrix decomposition in computer vision. The SVD is the final and the best factorization of matrix for now. It is also known as the SVD gives an optimal solution to any problem in least squares method. In this Appendix, we will briefly introduce the SVD not only in aspect of mathematician's points with backgrounds, but also in aspect of computer scientist's points with various applications of SVD. We also introduce some important Theorems and topics related with the SVD.

There are couple of different ways to explain SVD and approaches of the book [3] and [101] are quite interesting to introduce with different views, namely algebraic view and geometric view. In this Appendix, we explain the SVD in a form suitable for our purpose including basics of linear algebra. The introductory concepts, Theorems, Corollaries, Propositions are taken from classical books, [3], [91], [101], and [37].

A.1 Introduction to linear algebra

A.1.1 Vector spaces and linear maps

A vector space is a set V with operators $+$ and $\cdot$ on V such that commutativity, associativity, additive identity, additive inverse, multiplicative identity, and distributive properties hold, where scalars are in either $\mathbf{R}$ or $\mathbf{C}$. A vector space over field $\mathbf{F}$, $\mathbf{R}$ or $\mathbf{C}$, is called a vector space V over $\mathbf{F}$. Examples are a real vector space and a complex vector space, which are a vector space over $\mathbf{R}$ and a vector space over $\mathbf{C}$. We now introduce another example of vector space, which contains

polynomials. A polynomial with coefficients in $\mathbf{F}$ is a function $p : \mathbf{F} \to \mathbf{F}$,

$$\text{if there exist } c_0, ..., c_n \in \mathbf{F} \text{ such that } p(x) = c_0 + c_1 x + ... + c_n x^n \tag{A.1}$$

for all $x \in \mathbf{F}$. Define $\mathscr{P}(\mathbf{F})$ be the set of all polynomials with coefficients in $\mathbf{F}$, then it is also a vector space.

Points or vectors are elements of vector space. Let $v_1, ..., v_n$ be vectors in V. Given $a_1, ..., a_n \in \mathbf{R}$, $a_1 v_1 + \cdots + a_n v_n$ is a linear combination of the vectors. There are finite dimensional vector space and infinite dimensional vector spaces. It depends on finite or infinite vector list spans the vector space. The function space and $\mathscr{P}(\mathbf{F})$ are typical examples of the infinite dimensional vector space and $\mathbf{R}^n$ is the finite dimensional vector space.

A list of vectors are linearly independent *if* $a_1 v_1 + \cdots + a_n v_n = 0$ *iff* $a_1 = \cdots = a_n = 0$. If a list of vectors is linearly independent and spans V, then the list of vectors $(v_1, ..., v_n)$ is called a basis of V. For example, a standard basis of $\mathbf{R}^3$ is $((1, 0, 0), (0, 1, 0), (0, 0, 1))$.

Corollary A.1.1. ***([3], p.29)*** *Every finite-dimensional vector space has a basis.*

Theorem A.1.2. ***([3], p.31)*** *Any two different bases of a finite-dimensional vector space have the same length.*

A function $L : U \to V$ is a linear map from U to V with additivity and homogeneity properties.

$$\begin{aligned} &\textit{Additivity}: && L(u + v) = L(u) + L(v) \\ &\textit{Homogeneity}: && L(\alpha u) = \alpha L(u) \end{aligned}$$

We denote the set of all linear maps from U to V as $\mathscr{L}(U, V)$, and the set of all linear maps from V to V as $\mathscr{L}(V)$. Some examples of linear maps are zero, identity map, differentiation, and integration. The differentiation and integration are defined by $L(p) = p'$ and $L(p) = \int p(x)dx$, where $L \in \mathscr{L}(\mathscr{P}(\mathbf{R}), \mathscr{P}(\mathbf{R}))$.

We now introduce an m-by-n matrix in connection with basis of the vector spaces. Consider we have the m-by-n matrix

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}. \tag{A.2}$$

Let $L \in \mathscr{L}(U, V)$. Suppose there are bases $(u_1, \cdots, u_n)$ of U and $(v_1, \cdots, v_m)$ of V. Because of the linear combination and basis $v's$, we write $L(u_j)$ uniquely as

$$L(u_j) = a_{1,j}v_1 + \cdots + a_{m,j}v_n = \Sigma_{i=1}^{m} a_{i,j}v_j, \tag{A.3}$$

where $a_{i,j} \in \mathbf{F}$ for each $i = 1, ..., m$ and $j = 1, ..., n$. Then the matrix of linear map L with respect to each bases of U and V is (A.2). We now rewrite (A.2) as

$$\begin{array}{c} \\ v_1 \\ \vdots \\ v_m \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & \cdots & u_n \end{array} \\ \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix} \end{array}.$$

We denote the matrix representation of the linear map by $\mathscr{M}(L, (u_1, \cdots, u_n), (v_1, \cdots, v_n))$ or simply by $\mathscr{M}(L)$. Thus for each linear map from U to V we have a matrix with respect to its bases. At this point we introduce when we have a diagonal matrix in association with some basis.

Proposition A.1.3. ***([3], p.89)*** *Suppose $\lambda_1, ..., \lambda n$ are distinct eigenvalues of L, and $L \in \mathscr{L}(V)$. Then L has a diagonal matrix with respect to some basis of V iff there is a basis of V consisting of eigenvectors of L.*

This proposition will be connected with theorem in next section, and also used to prove SVD.

A.1.2 Inner product spaces and linear functional

In order to discuss orthonormal basis, introducing an inner product space and norm of vectors are inevitable. The inner product on W is a function and it maps an order of pair, $(v, w) \in W$, to a scalar, $\langle v, w\rangle \in \mathbf{F}$ with positivity, definiteness, additivity in first slot, homogeneity in first slot, and conjugate symmetry. Here the definiteness is $\langle w, w\rangle = 0$ *iff* $w = 0$. The norm of the vector is a distance from the origin and it is the length of the vector. The norm of vector $\mathbf{w} = (w_1, \cdots, w_n) \in \mathbf{R}^n$ is defined by $\|\mathbf{w}\| = \sqrt{w_1^2 + \cdots + w_n^2}$. We call a vector space with an inner product on W is an inner product space.

The vectors v and w are orthogonal if $\langle v, w\rangle = 0$. Geometrically it is the same as perpendicular. If the list of vectors in $\mathbf{w}$, $(w_1, \cdots, w_n)$, of W are pairwise orthogonal and the norm of $\mathbf{w}$ is 1, then we call orthonormal. The orthonormal list of vectors is linearly independent and thus orthonormal basis of W is a list of orthonormal vectors.

Theorem A.1.4. ***([3], p.107)*** *Suppose $(v_1, ..., v_n)$ is an orthonormal basis of V. Then*

$$w =< w, v_1 > v_1 + \cdots + < w, v_n > v_n \tag{A.4}$$

for every $w \in V$.

This theorem A.1.4 states the most importance of orthonormal bases.

Corollary A.1.5. ***([3], p.109)*** *Every finite-dimensional inner-product space has an orthonormal basis.*

A linear functional on W is a linear map from W to scalar, $\mathbf{R}$, which also satisfies additivity and homogeneity.

Theorem A.1.6. ***([3], p.117)*** *Suppose is f a linear functional on W. Then there is a unique vector $w \in W$ such that*

$$f(v) =< v, w > \tag{A.5}$$

for every $v \in W$.

Suppose V and W are a finite-dimensional, nonzero, inner product space over $\mathbf{F}$. An adjoint of $L \in \mathscr{L}(V,W)$ is denoted by L^* and it is a linear, $L^* \in \mathscr{L}(W,V)$ if $L \in \mathscr{L}(V,W)$. For better understanding, consider a linear transformation, $L : \mathbf{R}^3 \to \mathbf{R}$, $L(x,y,z) = 2x - y + 4z$. This linear transformation is a linear functional on $\mathbf{R}^3$, and we can rewrite the map as $L(x,y,z) =< (x,y,z),(2,-1,4) >$. The inner product is in fact a bilinear. Let a linear functional map from v to $< Lv, w >$, for a fixed $w \in W$. Then L^*w is the unique vector in V such that

$$< Lv, w >=< v, L^*w > \tag{A.6}$$

for all $v \in V$.

A.2 Singular Value Decomposition

A.2.1 Operators on inner product spaces

Suppose we still have the finite-dimensional, nonzero, inner vector product space V over field $\mathbf{F}$, and let $L \in \mathscr{L}(V)$. If $L = L^*$, then we call L is self-adjoint. It is also known as Hermitian. For example, let L is the operator on $\mathbf{R}^2$ and the matrix of this operator with respect to standard basis is

$$\begin{pmatrix} a & c \\ c & b \end{pmatrix}.$$

Then the matrix A is self-adjoint and it is symmetric. The self-adjoint with $< Lu, u >\geq 0$, for all $u \in V$ is called as a positive operator L.

Proposition A.2.1. ***([3], p.128)*** *Every eigenvalue of a self-adjoint operator is real.*

If the operator can commute with its adjoint, then we call normal. It means $LL^* = L^*L$, for $L \in \mathscr{L}(V)$. Every self-adjoint is normal, but we have normal operator that is not self-adjoint. Note that any skew-symmetric matrix is also normal, but not self-adjoint. If v is an eigenvector of L, which is normal, then v is also an eigenvector of L^*.

Corollary A.2.2. ***([3], p.132)*** *If $L \in \mathscr{L}(V)$ is normal, then eigenvectors of L corresponding to distinct eigenvalues are orthogonal.*

A.2.2 Singular Value Decomposition : Algebraic point of view

We now introduce an important theorem of self-adjoint operator and for instance, it implies that a symmetric matrix can be factored into $A = U\Lambda U^T$, where U is an orthogonal matrix and Λ is a diagonal matrix. The orthogonal matrix U contains orthonormal eigenvectors and the diagonal matrix Λ contains eigenvalues[91]. The book [3], states the theorem over complex inner product and real inner product spaces. Both theorems state that every self-adjoint operator has a diagonal matrix with respect to some normal basis by Proposition A.1.3.

Theorem A.2.3. ***([3], p.133, Complex Spectral Theorem)*** *Suppose that V is a complex inner-product space and $L \in \mathscr{L}(V)$. Then V has an orthonormal basis consisting of eigenvectors of L iff is normal.*

Notice that the above theorem considers the case that a complex skew-symmetric matrix.

Theorem A.2.4. ***([3], p.136, Real Spectral Theorem)*** *Suppose that V is a real inner-product space and $L \in \mathscr{L}(V)$. Then V has an orthonormal basis consisting of eigenvectors of L iff L is self-adjoint.*

Suppose $L \in \mathscr{L}(V)$ and let $\sigma_1, ..., \sigma_n$ are eigenvalues of $\sqrt{LL^*}$. Then $\sigma_1, ..., \sigma_n$ are singular values of L. The theorem below shows that every operator on V can be factored into in terms of its singular values and orthonormal bases of V. If an operator preserves norms, for instance a distance in a metric space, then we call it is an isometry : If $\|Su\| = \|u\|$ for all $u \in V$, then an operator $S \in \mathscr{L}(V)$ is an isometry. The theorem below states the operator is written as the isometry times a positive operator.

Theorem A.2.5. ***([3], p153, Polar Decomposition)*** *If $L \in \mathscr{L}(V)$, then there exists an isometry $S \in \mathscr{L}(V)$ such that $L = S\sqrt{L^*L}$.*

The idea of Polar Decomposition comes from the analogy between $\mathbf{C}$ and $\mathscr{L}(V)$. We consider $\bar{z}z$ as L^*L, and suppose the complex number z is in the unit circle. It means that $\bar{z}z = 1$, similarly

$L^*L = I$. Equivalently, we write $\bar{z}z = 1$ as $(\frac{z}{|z|})\sqrt{\bar{z}z}$. Then $(\frac{z}{|z|})$ is in the unit circle, and we can guess the existence of isometry in Polar Decomposition.

Theorem A.2.6. ***([3], p.156, Singluar-Value Decomposition)*** *Suppose $L \in \mathscr{L}(V)$ has singular values $\sigma_1, ..., \sigma_n$. Then there exist orthonormal bases $(u_1, ..., u_n)$ and $(v_1, ..., v_n)$ of V such that*

$$Lw = \sigma_1 < w, u_1 > v_1 + \ldots + \sigma_n < w, u_n > v_n \tag{A.7}$$

for every $w \in V$.

In order to prove this theorem, we first apply the Spectral Theorem to $\sqrt{L^*L}$. Then there is an orthonormal basis $(u_1, ..., u_n)$ of V such that $\sqrt{L^*L}u_j = \sigma_j u_j$ for $j = 1, ..., n$. We now use the Polar Decomposition to $\sqrt{L^*L}w$, and thus apply the isometry $S \in \mathscr{L}(V)$ such that $L = S\sqrt{L^*L}$ to both side of $\sqrt{L^*L}u_j = \sigma_j u_j$. Because of Theorem A.1.4, we have $w =< w, u_1 > u_1 + \cdots + < w, u_n > u_n$, and then let $v_j = Su_j$ for each j. Notice that the above theorem is $L \in \mathscr{L}(v)$. When we consider a matrix of this linear map, it is a square matrix, but it is not necessary normal or self-adjoint. If we only consider the real symmetric matrix of operator L, then the matrix is a square matrix, and Spectral Theorem and SVD state the same results. We also note that the SVD gives us to use two different bases.

We now introduce the SVD for any matrix M. When we discuss the SVD with the matrix instead of the operator on inner produce spaces, we easily notice that SVD is applied for any m by n matrix from literatures. However implementations or proofs are closely related with the proof of Theorem A.2.6.

Theorem A.2.7. ***([91], p.443, Singular Value Decomposition)*** *Any m by n matrix M can be factored into*

$$M = U\Lambda V^*, \tag{A.8}$$

where the columns of m by m matrix U are eigenvectors of MM^* and the columns of n by n matrix V are eigenvectors of M^*M. The matrix Λ is a m by n diagonal matrix with singular values, and the singular values of the square roots of the nonzero eigenvalues of both MM^* and M^*M.

Notice that the matrix U and V in the Theorem A.2.7 are unitary, i.e., $UU^* = U^*U = I$ and $VV^* = V^*V = I$. In other words, $U^* = U^{-1}$ and $V^* = V^{-1}$. The columns and rows of unitary matrix are orthonormal vectors. This version of SVD is mostly introduced from the literature for the application problems, but we note that mathematician's point, operators on inner product space, allows us to discuss SVD as a bounded operator in a classical function space, which is infinite dimensional inner product space, such as Hilbert space.

A.2.3 Singular Value Decomposition : Geometric point of view

In this subsection, we introduce the SVD from a geometric point of view. The geometric interpretation of the SVD in 2D is known as it maps the unit circle to an ellipse. From equation (A.8), we have an equivalent equation $MV = U\Lambda$. For each column of V, we also rewrite $MV = U\Lambda$ as

$$Mv_j = \sigma_j u_j, \tag{A.9}$$

where σ_j is singular values and $j = 1, ..., n$. From equation (A.9), we now obviously see that M maps the orthonormal vectors to scaled orthonormal vectors, $\sigma_j u_j$. More specifically, we also can state Theorem A.2.7 as followings(see Figure(A.1)) :

- Rotate the unit circle with the orthonormal vectors of V.
- Scale by Λ
- Rotate an ellipse with the orthonormal vectors of U.

From the literature, we also see that $\{u_j\}$ is called left singular vectors and $\{v_j\}$ is called right singular vectors. In most applications, a reduced SVD is used instead of a full SVD under assumption that the matrix M has full rank n, $m \geq n$. The reduced SVD consists of m by n matrix $\tilde{U}$ and n by n matrix $\tilde{V}$ with orthogonal columns, and n by n diagonal matrix $\tilde{\Lambda}$ with positive real numbers. Theorem A.2.7 is the full SVD, and the assumption of full rank n is not necessary. For the rank deficient matrix, the full SVD is appropriate, but the reduced SVD can be also used by

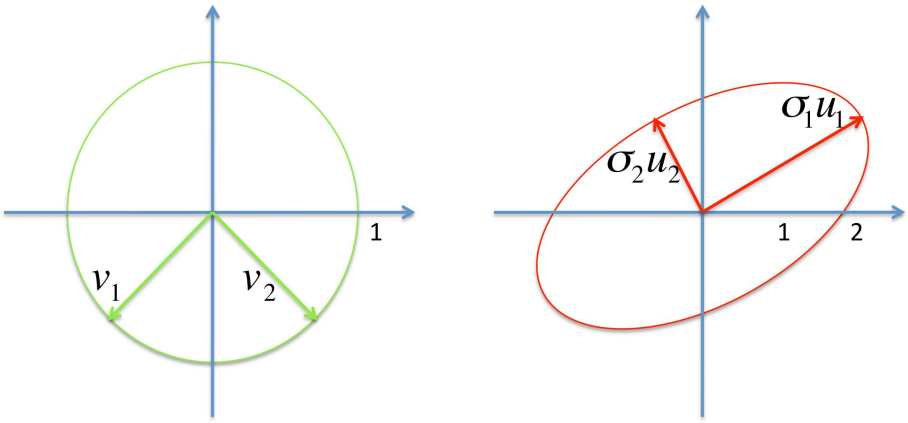


Figure A.1: Geometric view of Singular Value Decomposition at a 2×2 matrix

modifying it(See [101]). Note that the range of linear map L is defined as $range\ L = \{Lv : v \in V\}$ for some $v \in V$. The rank of matrix representation of L is $dimension\ of\ V - dimension\ of\ null\ L$, where $null\ L = \{v \in V : Lv = 0\}$. The rank deficient matrix implies that the rank of matrix is less than $min\{m, n\}$ because rank of matrix is $min\{m, n\}$.

Theorem A.2.8. ***(p.35, [101])*** *For any ν, $0 \leq \nu \leq r$, we define*

$$M_\nu = \Sigma_{j=1}^{\nu} \sigma_j u_j v_j^*. \tag{A.10}$$

If $\nu = \{p\} = min\{m, n\}$, and define $\sigma_{\nu+1} = 0$, then

$$\|M - M_\nu\|_2 = \inf_{A \in C^{m \times n}, rank(A) \leq \nu} \|M - A\|_2 = \sigma_{\nu+1}. \tag{A.11}$$

The above theorem is known as low rank approximation and it has a good geometric interpretation. Theorem A.2.8 tells us the partial sum of rank one matrix $M_\nu = \Sigma_{j=1}^{\nu} \sigma_j u_j v_j^*$ is best approximation of M in 2-norm. Here 2-norm of matrix M is denoted by $\|M\|_2$ and defined as $\max_{\|v\|_2=1} \|Mx\|_2$. Thus the 2-norm of matrix M, $\|M\|_2$, is equal to the largest singular value of the matrix M.

A.3 Applications of Singular Value Decomposition

Singular Value Decomposition is used in the proposed method in this thesis three times. The first one is for the fundamental matrix, F. From the randomly chosen 8 points, the matrix, size of

8 by 9, is set up. Then, the basis, corresponded on the smallest singular value is the element of the fundamental matrix, F. Where, we used the SVD to get the basis of the smallest projection errors of 8 points. It means that there are 8 projections on 9 dimensional space and the singular values is the distance of 8 projections from the basis. Thus, the basis corresponds to the smallest singular value is the basis with the smallest errors, in other words, the optimum basis.

Second, SVD is used to fix the fundamental matrix. Since the result is computational and the application has many kind of errors, the rank of the fundamental matrix, F, is not 2, in practical. The result of SVD on the fundamental matrix, F, shows three singular values. Since its rank should be 2, the smallest singular value of the fundamental matrix, F, should be zero. By replacing the smallest singular value of the fundamental matrix, F, and composing $\tilde{F}$, the errors on reprojection is reduced.

Third, SVD can be used for weighted least squares method on solving equation (4.25). For the system matrix, A, and the right hand side vector, b, we are going to get the vector, x, which minimize $(b - Ax)\prime W (b - Ax)$, where W is a diagonal matrix and the diagonal values are the weights of each equation. For now, a least square function provided by matlab is used, but it is possible to rewrite in SVD version when ported to other program language for the purpose of speed-up.