

# PANORAMA GENERATION FOR STEREOSCOPIC VISUALIZATION OF LARGE-SCALE SCENES

EDGARDO MOLINA

A DISSERTATION SUBMITTED TO  
THE GRADUATE FACULTY IN COMPUTER SCIENCE IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY,  
THE CITY UNIVERSITY OF NEW YORK

## COMMITTEE MEMBERS:

DR. ZHIGANG ZHU (ADVISOR)

DR. YINGLI TIAN

DR. IOANNIS STAMOS

DR. QUYNH DINH

MAY 15, 2015

© Copyright by Edgardo Molina, 2015.

All rights reserved.

# Abstract

In this thesis, we address the problem of modeling and stereoscopically visualizing large-scale scenes captured with a single moving camera. In many applications that image large-scale scenes the critical information desired is the 3D spatial information of stationary objects and movers within the scene. Stereo panoramas, like regular panoramas, provide a wide field-of-view that can represent the entire scene, with the stereo panoramas additionally representing the motion parallax and allowing for 3D visualization and reconstruction of the scene. The primary issue with stereo panorama construction methods is that they are constrained for a particular camera motion model; typically the camera is constrained to move along a linear or circular path. Here we present a method for constructing stereo panoramas for general camera motion, and we develop a (1) *Unified Stereo Mosaic Framework* that handles general camera motion models. To construct stereo panoramas for general motion we created a new (2) *Stereo Mosaic Layering* algorithm that speeds up panorama construction enabling real-time applications.

In large-scale scene applications it is often the case that the scene will be imaged persistently by passing over the same path multiple times or two or more sensors of different modalities will pass over the the same scene. To address these issues we developed methods for (3) *Multi-Run and Multi-Modal Mosaic Alignment*. Finally, we developed an (4) *Intelligent Stereo Visualization* that allows a viewer to interact and stereoscopically view the stereo panoramas developed from general motion.

## Acknowledgements

This work has been supported by AFRL Award #FA8650-05-1-1853, AFOSR Award #FA9550-08-1-0199, the 2011 Air Force Summer Faculty Fellow Program (SFFP), and NSF I-Corps Award #1243737. The work is also partially supported by NSF under Award #EFRI-1137172, Award #CNS-0551598, and ARO Award #W911NF-08-1-0531.

The first person I must thank is my advisor, Prof. Zhigang Zhu, for his tireless and superb mentorship. Beyond a research advisor, he is dedicated to service and has been a role model for what it means to be a collaborator, a mentor, and especially an educator.

Thanks also go to my committee members, Dr. Yingli Tian, Dr. Ioannis Stamos, and Dr. Quynh Dinh, for their insightful comments and feedback over the course of the development of my dissertation.

I would also like to thank Mrs. Olga Mendoza-Schrock, Dr. Todd Rovito, Dr. Clark N. Taylor and Dr. Kevin L. Priddy, for their inspiring discussions and advising during my summer research at the Air Force Research Laboratory, WPAFB. While there I met many great people who were visiting scholars, contractors, and Air Force employees. I am proud to call many of them my friends, especially Olga and her husband, Chris Schrock, for welcoming me so warmly.

Finally, I must thank my parents, my brother, and my sister for their motivation and support throughout my academic career. They instilled in me the value of knowledge and of giving it back.

To my parents, René and Teresa.

# Contents

Abstract . . . . .	iii
Acknowledgements . . . . .	iv
List of Figures . . . . .	ix
<b>1 Introduction</b>	<b>1</b>
1.1 Panorama Applications . . . . .	2
1.2 Problem Statements . . . . .	3
1.3 Goals . . . . .	5
1.4 Summary of Contributions . . . . .	8
1.5 Organization . . . . .	10
<b>2 Related Work</b>	<b>12</b>
2.1 Image Registration . . . . .	12
2.2 Panorama Mosaicking . . . . .	13
2.2.1 Circular Motion . . . . .	13
2.2.2 Linear Motion . . . . .	14
2.2.3 Large-Scale Scenes . . . . .	14
2.2.4 Panorama Seams . . . . .	15
2.3 Stereoscopic Visualization . . . . .	15
<b>3 Panorama Generation Under Linear Motion</b>	<b>18</b>
3.1 View Perpendicular to Camera Motion . . . . .	19

3.1.1	Parallel-perspective geometry and panoramas . . . . .	20
3.1.2	Stereo and multi-view panoramas . . . . .	21
3.1.3	Results . . . . .	23
3.2	View Parallel to Camera Motion . . . . .	25
3.3	Unified Stereo Mosaic Framework . . . . .	26
<b>4</b>	<b>Panorama Generation Under Circular Motion</b>	<b>28</b>
4.1	Viewing In . . . . .	28
4.2	Viewing Out . . . . .	31
4.3	Viewing Perpendicular . . . . .	36
4.4	Unified Stereo Mosaic Framework . . . . .	40
<b>5</b>	<b>Panorama Generation Under General Motion</b>	<b>44</b>
5.1	Direct Layering Approach . . . . .	45
5.1.1	Multiple Runs and Results . . . . .	48
5.2	Unified Stereo Mosaic Framework . . . . .	49
<b>6</b>	<b>Stereoscopic Visualization using Multi-view Panoramas</b>	<b>52</b>
6.1	Stereo Viewing and Motion Path Alignment . . . . .	52
6.2	Multi-modal / Multi-pass Alignment . . . . .	54
6.3	Interactive Viewing Choices . . . . .	55
6.3.1	Adaptive Baseline/Disparity . . . . .	57
6.3.2	Interactive View Angle . . . . .	57
6.3.3	Automatic Viewer Tracking . . . . .	58
6.4	Stereoscopic Display Types . . . . .	60
6.5	Memory Management . . . . .	61
<b>7</b>	<b>Conclusion and Discussion</b>	<b>62</b>
7.1	Unified Stereo Mosaic Framework . . . . .	62

7.2	Large Scale Scene Issues . . . . .	63
7.3	Discussion . . . . .	64
7.4	Future Work . . . . .	67
<b>A</b>	<b>Data Sources</b>	<b>69</b>
A.1	CLIF 2006 . . . . .	69
A.1.1	Data details . . . . .	69
A.1.2	Issues . . . . .	70
A.2	CLIF 2007 . . . . .	70
A.2.1	Data details . . . . .	70
A.2.2	Issues . . . . .	71
A.3	CSUAV . . . . .	71
A.3.1	Data details . . . . .	71
<b>B</b>	<b>Publications</b>	<b>73</b>

# List of Figures

1.1 (a) Camera viewing perpendicular to motion direction (b) Camera viewing along motion direction. . . . .	6
1.2 Camera moving along a circular trajectory around an axis and viewing in different directions. (a) Camera viewing outside of the circle. (b) Camera viewing inside the circle. (c) Camera viewing perpendicular to the plane of the circle. . . . .	6
1.3 Camera moving along an arbitrary motion curve with any camera view. . . . .	7
3.1 (top) The optical flow is parallel to the motion direction when the camera is viewing perpendicular to the motion direction. (bottom) The rotation (roll) of the camera about its optical axis does not affect the optical flow from remaining parallel to the motion. . . . .	19
3.2 A slit taken from a frame. . . . .	20
3.3 A parallel-perspective panorama constructed from a fixed slit. . . . .	21
3.4 Interpolating parallel rays for parallel-perspective panoramas. . . . .	22
3.5 Stereo parallel-perspective projection geometry. A point imaged by a forward angle slit will be viewed at a later time in the frame sequence from a backward angle slit. . . . .	23
3.6 Multiple slits can be used to create panoramas. This enables the selections of adaptive baselines for improved 3D viewing and 3D reconstruction. . . . .	23
3.7 Two of 8 IR panoramas generated of a scene captured with linear motion. . . . .	24

3.8	Two of 8 EO panoramas generated of a scene captured with linear motion.	24
3.9	The optical flow radiating from the FOE when camera optical axis is parallel to motion direction.	26
3.10	Illustrating how an FOE slit can be transformed into a linear slit with a polar transformation. The camera is moving towards the FOE, which is at the line $r = 0$ in the polar image. Slits are perpendicular to the direction of camera motion in this polar image.	27
4.1	Top view of cyclograph geometry for a camera viewing in towards the center of the path.	29
4.2	Image coordinates for circular projection panoramas	30
4.3	Computing depth $D$ from cyclograph geometry.	30
4.4	A camera flying on a circular path around the earth	31
4.5	Top view of concentric geometry for a camera viewing outwards from the center of the rotation path.	32
4.6	Computing depth of a scene point $P$ using concentric stereo panoramas.	32
4.7	Multimodal alignment of two offset cameras on a rotating platform.	33
4.8	Multimodal alignment results for a platform with an EO and IR camera superimposed.	36
4.9	Circular flight geometry.	37
4.10	Stereo geometry for circular flight.	37
4.11	PRISM geometry for circular flights.	39
4.12	A 360 degree panorama for a circular simulation (in polar coordinates).	40
4.13	Complete circular panorama (in cartesian coordinates).	40
4.14	A circular panorama for a real scene capture (in polar coordinates).	40
4.15	Cyclograph stereo geometry for very large scale scenes. A camera's optical center path moves around the earth's terrain.	41

5.1	A general motion path can be segmented into three constrained paths. . . . .	44
5.2	GENERATEPANORAMAS procedure. . . . .	48
5.3	Illustration of the GENERATEPANORAMAS procedure. Under ideal conditions, each multi-viewpoint panorama <i>pano</i> [ <i>i</i> ] is constructed from a similar view angle on the scene. . . . .	48
5.4	The CLIF 2007 sequence makes 4 approximately circular passes of an area. The units in both horizontal and vertical axes are in pixels. The dashed black box represents the last frame. Red boxes represent every 50th frame. (a) Image alignment without error correction. (b) Image alignment after error correction. . . . .	50
5.5	Panoramas for (a) base cycle and (b) cycle 2. . . . .	50
6.1	The visualization window follows along the motion path, keeping the local windows epipolar lines approximately horizontal. . . . .	53
6.2	(a) MVSV displaying EO in stereo. (b) MVSV displaying IR in stereo. (c) MVSV displaying IR overlay on EO (not in stereo). . . . .	54
6.3	A viewer perceiving depth beyond a stereo display. . . . .	56
6.4	A viewer perceiving depth in front of a stereo display. . . . .	56
6.5	MVSV allowing the view angle to be changed in stereo. (a) MVSV displaying the left most view angle. (b) Viewing through the center angle. (c) MVSV displaying the right most view angle. . . . .	58
6.6	Interactive view angles are changed when viewer's detected face enters a new numbered view zone. . . . .	59
6.7	As the viewer moves parallel to the screen, the view angle is changed to provide corresponding views. . . . .	60

# Chapter 1

## Introduction

Capturing imagery and video of large scale scenes has never been as accessible as it is today. The availability of affordable small cameras with high resolutions, different modalities, and unmanned aerial and ground systems has made it possible to capture large amounts of imagery over long periods of time. This has renewed interest in applications that can make use of such data to continuously monitor and model areas of interest. The large amounts of data that can be automatically captured has created a problem for operators and users of the image data who do not have enough time to view all of the video data being captured. Typically, the critical information contained within the image data is the 3D spatial information of static objects and movers. In this dissertation we model large scale scenes captured with a single camera using stereo panoramas to provide a snapshot of the entire scene, along with a stereoscopic visualization for an operator to interact with the data in a spatial context. In addition we develop a method for aligning persistently captured data over an area whether it is captured with the same or different camera/sensor modality. Our aim is to show a Unified Stereo Mosaic Framework for generating stereo panoramas from imagery captured under different motion models, that can be used for stereoscopic visualization.

## 1.1 Panorama Applications

*Panoramas* are images that provide a very wide *field-of-view* (FOV) of a scene. Such images are beneficial in mapping, tracking, surveillance, and scene visualization applications among many others. Panoramas can be captured either by a wide angle lens or a camera in motion. Commonly available wide angle lenses for cameras approach 100 degrees of angular FOV, with some specialty fish eye lenses approaching 180 degrees. This is typically not wide enough for many of the applications where the area of interest may be an entire town or many miles of road. Instead, panoramas can be constructed from a camera with a regular lens and in motion. The images can then be overlapped and stitched into a very large panorama. One of the most popular examples for capturing a panorama would be that of a user holding a camera in hand with their arm stretched out, and rotating completely in place to capture images 360 degrees around their body, these images are then used to construct a single panoramic image. But in fact there are many large scale scene applications for panoramas, and some not so obvious ones.

Perhaps at the largest scale of applications we have satellite imaging, which has an imager (sometimes a pushbroom sensor) capturing many images and combining them into one large mapping or panorama that allows users to view an entire area the size of a city at once. Aerial imaging is an area with renewed interest due to the recent consumer availability of Unmanned Aerial Vehicles (UAV) technology. Planes and helicopters have long been outfitted to capture imagery and video from all altitudes. In many mapping and surveillance applications this has meant capturing the imagery along a path and creating a panorama of towns, cities, roads, etc. With the availability of UAV's, even consumers are looking to image large scale scenes of events and capture them as panoramas. Very large scale scenes are also being captured by ground vehicles such as bikes, cars, and trains. These vehicles are being outfitted with cameras to capture imagery of long stretches of miles. Perhaps the biggest examples of such applications are Google [16] and Bing [26] maps and their development of street view and route panoramas that aim to cover every city in the world. Another large scale use of

panoramas that is not obvious is in medical imaging. When we think of endoscopies, and pill cameras that can be swallowed, it is ideal for the doctors to have a large scale mapping or panorama of the gastrointestinal tract that was imaged. This would allow them to quickly view any area of the tract within its spatial context without having to take time to rewind or forward a video, frame-by-frame, until they discover the area of interest.

## 1.2 Problem Statements

Capturing panoramas and the 3D information of a very large scale scene requires the use of a camera in motion versus a still camera with a wide angle lens. Here we look at the problems and benefits with both methods.

Commonly available wide FOV lenses are not able to capture a wide enough FOV for many large scale scene applications. Instead, specialty lenses such as Fish-Eye lenses are commonly available to capture up to a 180x180 degree FOV of scenes. Other lenses are designed to capture 360 degrees around an axis, so called omnidirectional lenses. A primary benefit of such lenses is that they capture all view directions at the same time, therefore each image represents a snapshot of a single point in time. But, in large scale applications, problems arise with the use of these specialty lenses: (1) First, since cameras have a fixed sensor resolution, the angular resolution of a scene decreases as the lenses get wider. (2) Second, these lenses create a lot of undesirable distortion to be useful for many applications. (3) Third, a lens cannot solve occlusion issues that arise from imaging a large scale scene such as imaging a winding road.

In these large scale scenarios, panoramas are best captured with a camera in motion over or around the scene as it *captures many images that are aligned and then mosaicked into a single panoramic image*. This addresses various problems faced with the use of specialty lenses: *resolution is improved* by mosaicking various full resolution images from a given sensor, *distortion is improved* because regular lenses can be used in motion, and *occlusions*

*are overcome* by moving the camera around those occluding objects. The one tradeoff from using a specialty lens is that the panorama no longer represents an instantaneous snapshot of time, but for many of these very large scale applications there is no viable alternative.

While panoramas provide benefits in large scale applications, generating them requires that we address a set of problems with (1) parallax, (2) independent moving objects, (3) image alignment, and (4) seamless stitching/mosaicking. Before a panorama is constructed, all of the images captured in motion must be aligned with a registration procedure. Motion parallax and moving objects are both problems that cause error in the alignment procedure. Once the images are aligned, then they have to be combined into the final panorama, which should appear as a continuous image, parallax and alignment errors all produce erroneous seams in the final panorama. Special consideration has to be taken during these procedures to produce a correct result.

Since many of the large scale scene applications are primarily interested in the 3D spatial information of object and movers in the scene we use *stereo panoramas* to model the scenes, which encode the 3D information. Stereo panoramas are two or more panoramas that are in alignment and provide a stereoscopic view of the scene. These stereo panoramas can be used for both stereoscopic visualization as well as for the 3D reconstruction of the scene.

Building stereo panoramas requires that we handle the problems listed above, as well as deal with new problems. First, we create two or more panoramas from the same sequence of images, that is a second camera is not used in achieving the stereo pairs. Second, the motion parallax that is problematic to image alignment, contains the depth cues that stereo panoramas use to offset points and provide 3D information. Therefore, the mosaicking method employed to create stereo panoramas must take the geometry into account and not simply perform a blending of two images.

One of the biggest problems has been that methods developed for generating stereo panoramas have been constrained to a specific camera motion model, due to the varying

motion parallax exhibited in different types of camera motion. The camera may have a dominant direction of linear motion, it may be moving in a circle, or more commonly it can have general unconstrained motion. In the past, we treated the linear and circular motion cases with separate motion and mosaicking models, but in this dissertation we developed a unified mosaicking framework for general motion and show how it relates to the constrained linear and circular cases. Doing this provides many benefits to large-scale applications that will be outlined below.

### 1.3 Goals

In this dissertation we were particularly interested in constructing a *Unified Stereo Mosaic Framework* for imagery captured under general unconstrained motion. Stereo panoramas are two or more panoramas that are generated of a scene that also provide parallax for the purposes of stereoscopic viewing and for 3D reconstruction. Generating stereo panoramas has two major requirements:

First, we need to understand the motion a camera undertook while capturing the images to construct the panorama, such that the disparity between the two stereo panoramas corresponds to the human visual system. This is done by ensuring that the baseline of the two panoramas is kept horizontal and approximates the distance between human eyes. In addition, the epipolar curves should be horizontal lines so that the human visual system can comfortably converge on image points at various depths. It turns out that these epipolar geometry properties also simplify the problem of feature correspondence when performing 3D reconstruction. On the other hand, for 3D reconstruction it is beneficial to have large baselines to improve the depth error, so a solution with an adaptive baseline is also desired.

Second, we must perform the alignment of many images under motion parallax. Methods for generating panoramas are complicated by the various types of motions a camera can take. Our approach is first to look into each of the methods for constrained motion, find

their differences and their connections, and then build a unified framework such that one single mosaicking method can deal with all the motion types.

The following are the different camera motion models typically encountered in applications:

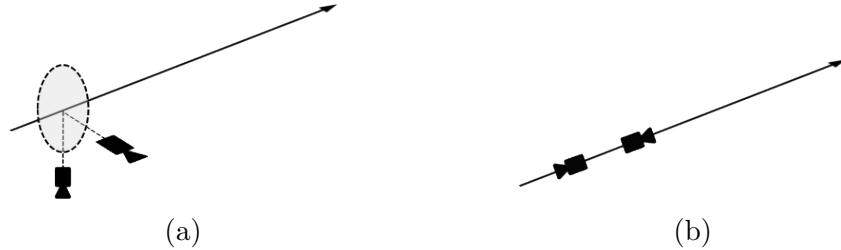


Figure 1.1: (a) Camera viewing perpendicular to motion direction (b) Camera viewing along motion direction.

**Linear Motion:** a camera that moves along a dominant direction is said to have linear motion. This case can be further broken down into two special cases:

- L(a) the camera is viewing in a direction perpendicular to the motion direction, similar to the pushbroom model,
- L(b) the camera is viewing along the direction of motion, either forward or backwards.

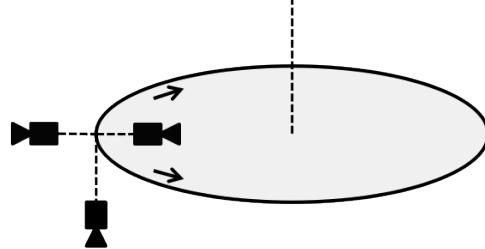


Figure 1.2: Camera moving along a circular trajectory around an axis and viewing in different directions. (a) Camera viewing outside of the circle. (b) Camera viewing inside the circle. (c) Camera viewing perpendicular to the plane of the circle.

**Circular Motion:** a camera that moves along a circle path can be further divided into 3 categories:

- C(a) the camera is viewing outside of the circle, which creates concentric panoramas,

- C(b) the camera is viewing inside of the circle, which is a cyclograph,  
 C(c) the camera is viewing perpendicular to the plane formed by the circle.

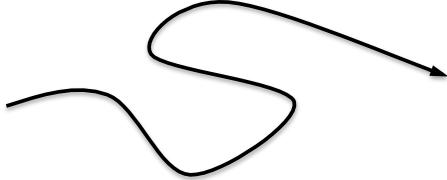


Figure 1.3: Camera moving along an arbitrary motion curve with any camera view.

**General Motion:** a camera that moves without constraint in all directions. This motion can be handled in two ways:

- G(a) the motion is broken down into piecewise segments of linear and circular cases,  
 G(b) the motion is handled directly with a unified motion model.

To achieve the goal of developing a Unified Stereo Mosaic Framework for general motion that covers all of the motion models listed above, we addressed the following problems:

P.1. *How do we generate panoramas under the various motion models? And how are they similar?*

- We considered the methods for generating panoramas under a specific constrained motion, such as linear or circular motion. We then related these to each other and to our unified general motion model to demonstrate that each constrained model can be treated as a special case of other motion models through a conversion. We did this to show that we can use a unified mosaicking algorithm, such as PRISM (Parallel Ray Interpolation for Stereo Mosaics) [57], instead of having to develop separate methods for each of the specific motion types.

P.2. *How do we generate stereoscopic views under different conditions? Such as motion models, frame rates, and different modalities?*

- In real world scene captures we run into many variable conditions, such as fast or low frame rates, varying speeds as the camera moves over a large scale scene, and cameras with different modalities. Another condition that arises is continuous coverage of the same area using a single modality or multiple modalities of sensors (such as both EO and IR). In these condition it is desired for multiple runs over a scene to be aligned for exploitation with a single sensor modality, and for different spectral bands to be aligned with multimodal sensors.

### P.3. *How do we visualize these stereo mosaics for large scale scenes?*

- When we generate panoramas of large scale scenes, such as along a highway, or of a town scale, we begin to encounter memory issues in creating such large panoramas. We will discuss methods for managing memory in these cases. For general motion panoramas, the epipolar curve will follow the camera motion; in these cases we also need to handle the problem of properly visualizing the stereoscopic mosaics in a way that maintains the epipolar curves horizontal for the viewer.

## 1.4 Summary of Contributions

In our development of methods to generate stereo panoramas for large scale scenes under general motion and address the problems outlined above, we made the following contributions:

### C.1. A Unified Stereo Mosaic Framework:

- We show that a unified mosaicking approach can be used to handle general camera motion, by demonstrating that methods for constrained motion types are similar. We developed models for creating mosaics for motions L(a), C(a), C(c), G(a), and G(b). We have also shown that for really large circular paths, such as around the earth  $L(a) \approx C(b)$ . We have shown that a polar transformation can make C(c) a

special case of L(a), and L(b) can be transformed to L(a) via a polar conversion, and we show that the general motion model covers all of these motions. By showing that all of the motion models can be handled by the general motion model, we can use either the PRISM or Fast Layering (covered in chapter 5) algorithms for panorama generation. These methods have been shown to produce the best output in terms of depth error, visualization, and/or computation

#### C.2. Multi-Run and/or Multi-Modal Mosaic alignment:

- We have also developed work that is able to align multiple run panoramas. This has two important uses: (1) first, in persistent surveillance scenarios, the camera may continuously (or over some days) image the same scene. In these cases it is beneficial to be able to align the panoramas for multiple runs for visualization, detecting, and tracking tasks. (2) It is also beneficial to be able to align and visualize panoramas of the same scene that comes from different modalities (such as EO imagery with IR imagery).

#### C.3. Stereo Mosaic Layering:

- We developed a faster alternative to the PRISM algorithm for mosaicking. This has the benefit of enabling realtime applications while also handling arbitrary camera motion and without greatly sacrificing the quality of the panoramas.

#### C.4. Intelligent Stereo Visualization:

- We developed a smart interface that can correctly visualize general motion panoramas by aligning the imageries epipolar curves horizontally for display, picking the correct disparity, and allowing for multi angle viewing. The visualization application also takes care of loading in large-scale multi-view panoramas efficiently by using a scaling and tiling approach.

## 1.5 Organization

In the following chapters we present our methods, in addition to the state of the art, for generating stereo panoramas for constrained motion as well as for general motion. We separate the methods based on the type of camera motion model (linear, circular, or general), we will present the type of epipolar geometry produced under each type of motion, and methods developed for the mosaicking of images into panoramas. Under all of these camera motion models, we are optimizing the panoramas for stereo visualization; in many cases this will also provide benefits in 3D reconstruction but this will not be the focus. At the end of each motion models chapter, we look at the mosaicking method similarities and through some conversions, we show they can be treated with the same mosaic methods. This is particularly useful to show that any arbitrary unconstrained motion can be effectively used with our Unified Stereo Mosaic Framework while preserving the epipolar and baseline properties desired for stereo panoramic visualization. The rest of this dissertation is organized as follows:

Chapter 2 presents a brief literature review for the state of the art methods in image registration, stereo panoramas, mosaicking, and stereoscopic visualization.

For Chapters 3, 4, and 5 which outline the three types of camera motion models, we first describe the state of the art methods, followed by a discussion of our contributions under each model and how they relate to each other, building up to the Unified Stereo Mosaic Framework, which is summarized in the conclusion.

Chapter 3 presents Panorama Generation under Linear Motion. We will show how two types of linear motion (camera motion with parallel motion parallax and motion with an FOE parallax) can be connected by using a polar transformation and transformed into a linear pushbroom model, therefore the PRISM or Fast Layering algorithm can be used in both cases.

Chapter 4 presents Panorama Generation under Circular Motion. Three types of circular motion are discussed and connected: cyclograph, concentric and circular flight. We also

show that they can all be converted into linear motion and therefore the PRISM or the Fast Layering algorithm can also be utilized. In this chapter we also present our work on multimodal stereo panorama generation with a color camera and an IR camera on a rotating platform.

Chapter 5 presents Panorama Generation under General Motion. We will first briefly describe a divide-and-conquer approach that can utilize the PRISM algorithm, then focus on a faster mosaicking algorithm for high resolution video images captured with low frame rate and a rather un-smooth motion. In this chapter, a spatial-temporal alignment algorithm is also going to be presented for panorama generation and alignment with multiple runs over a scene.

Chapter 6 presents our work in stereo visualization with stereo panoramas. We show how to deal with memory limitations in creating very large scale panoramas. And we also demonstrate our approach for presenting stereoscopic views of general motion panoramas to a user interactively.

We conclude with chapter 7, which will summarize the Unified Stereo Mosaic Framework, which shows how each of the constrained motion models are related, and to show that the general motion method developed in chapter 5 can be used to handle the other motion models. We also present some future work left in stereo panorama generation and visualization.

# Chapter 2

## Related Work

Creating and visualizing stereo panoramas for large scale scenes consists of multiple methods. For large scale scenes, the camera is typically in motion to capture a large area, therefore the first step is to perform image registration. The next step is to mosaic the registered imagery into panoramas, and further, stereo panoramas. Lastly, there are various display methods available for the visualization of stereo panoramas. The following sections are a brief literature review of all of these areas. The focus of the work developed in this dissertation is in mosaicking techniques for imagery captured under various motion models and how to stereoscopically visualize that imagery.

### 2.1 Image Registration

Before a panorama is constructed, the imagery must be registered and aligned. This is one of the most critical steps in constructing a seamless and correct stereo panorama pair. With large scale scenes, small interframe errors rapidly accumulate to large global alignment errors, also known as drift errors. The basic registration methods are based on feature point matching or correlation, and recently the most popular methods are bundle adjustment [20], SLAM [29], and Visual Odometry [41]. While these methods work well in most cases, they also have limitations since ideally they work best when features are maintained in view.

With large scale scenes it may take multiple passes before that happens. Some methods for improving drift is the use of key frames to restart the algorithms, but a lot of research has been dedicated to reducing drift and augmenting the basic registration methods for large scale scenes. Caspi and Irani [9] have studied alignment of non-overlapping sequences from two cameras but with a common viewpoint.

A common approach to registration is to augment the imagery with position data from GPS and INS/IMU sensors. These approaches have been studied by Heiner and Taylor [17], Oskiper et al. [34], and Zhu et al. [61]. The alignment obtained with these methods are geo-referenced, but the sensors results are not accurate for seamless panoramas, especially for large scale scenes, where the scene is captured at a great distance. These methods also require additional hardware. Another approach is to use existing reference imagery that covers the entire scenes, such as satellite imagery or DEM's (Digital Elevation Maps), and register all of the images to that by matching known landmarks and features. Kumar et al. [23], Lin et al. [24], and Zhu et al. [62] and Oskiper et al. [33], have studied these methods, which also produce geo-registered imagery. Olson et al. [32] describe techniques for registering images from sequences of aerial images captured of the same terrain on different days. While these methods work well, they don't account for the initial acquisition, and for major changes in the terrain.

## 2.2 Panorama Mosaicking

### 2.2.1 Circular Motion

The construction of stereo panoramas from two rotating cameras was proposed by Huang & Hung [18]. The use of mirrors to create two omni-directional cameras aligned for stereo panoramas was proposed in Gluckman et al. [15]. Methods for constructing stereo panoramas from a single off-center camera were proposed by Peleg et al. [35], and Shum & Szeliski [44]. More recently we have studied the development of two rotating multi-modal off-center cam-

eras in [37], and we have also studied stereo panoramas for a single camera flying above a scene, with a nadir view and following a circular path [28]. Smith et al. [45] has studied the use of onmi-stereo panoramas created from a rotating camera for the immersive visualization of cultural heritage landmarks and imagery. Zhang et al. [55] describes the setup of a rotating stereo panorama rig.

### 2.2.2 Linear Motion

Building stereo panoramas from a linearly translating camera has been studied by Zhu et al. [60], and Chai & Shum [10]. Stereo panoramas from translating cameras mounted to ground vehicles have been studied in [56] and for aerial surveys has been studied in [57] and [58]. Schechner and Nayar [42] developed a generalized mosaicing camera rig using a filter in front of the camera. This allows for computing High Dynamic Range in panoramas, which was also studied in [3]. Rousso et al. [40], and Wang et al. [49, 43] have studied creating panoramas from a camera translating parallel to its optical axis, as an application the paper [49] looks at cameras mounted to the front of a train; stereo panoramas are created for track condition monitoring. Tang and Zhu [46] has studied the use of linear stereo panoramas for a content-based 3D reconstruction of scenes.

### 2.2.3 Large-Scale Scenes

Constructing panoramas for long sequences has been studied by Peleg and Herman [36], in Zheng’s work on Route Panoramas [56], the parallel-perspective stereo mosaics of Zhu et al. [57], and the multi-viewpoint panorama work of Agarwala et al. [1]. Stereo panorama generation for general motion of large scale scenes has more recently been studied by us in [27]. Recently, high-resolution linear and circular stereo panoramas and their visualization has been studied by Richardt et al. [39], and Yan et al. [54] have studied the interesting problem of creating ”infinite” panoramas by stitching multiple stereo images of similar content but from physically different locations.

#### 2.2.4 Panorama Seams

Once the imagery captured is aligned, it has to be combined into a panorama, which ideally does not present seams along the boundary of different images. In this way, it will appear as if the panorama was captured from a single camera. This is important in stereo visualization since seams can cause incorrect depth to be perceived by the viewer. Amongst the methods developed to prevent seams are: Peleg and Herman, in [36], where they described panoramas that perform a blending of panorama pixels where multiple aligned images overlap. The works of Agarwala et al. [2], Burt and Adelson [6], and Jia and Tang [21] have all described general purpose methods for smooth seams in combined imagery. Recently, Gao et al. [14], have proposed a method that uses a dual-homography warping to combine images with two dominant planes with smooth seams. Panorama generation has become of interest on mobile devices recently, Cho et al. [12] present a stereo panorama stitching framework for handheld devices.

### 2.3 Stereoscopic Visualization

One of the major benefits of modeling a large scale scene with stereo panoramas is the ability for users to immediately use the stereo pair and visualize the depth even before a 3D reconstruction method is applied. There are many display types that can be used for stereoscopic visualization. There are displays that require a viewer to wear special glasses and displays that do not. All of these displays aim to present only one of the stereo pair images to each eye of the viewer so that they perceive depth.

Color-multiplexed displays [47, 53, 51, 52] combine stereo pairs by mixing complementary colors from each image into a final image. The viewer then wears glasses with complementary color filters. The most popular of these methods is the red-cyan anaglyph, in which the red channels from the left image are combined with the blue-green channels of the right image. The glasses contain a red and cyan filter over the same eyes.

Time-multiplexed displays [25, 31, 47] very quickly switch from displaying the left image, and then the right image in alternating fashion, at a speed greater than 120Hz. The viewer then wears active glasses that synchronize with the display and also alternately cover each eye so that the viewer only sees the left image with their left eye, and the right image with their right eye.

Polarization-multiplexed displays [25, 47] achieve the same effect with glasses that have two orthogonal light filters for each eye, but uses two projectors with matching filters to project the stereo pairs. Because of the filters, each eye will only view the corresponding image from the projector with matching filter.

More recently there has been the development of Head-Mounted displays (HMD's) [47, 48] and of autostereoscopic displays [47]. Head mounted displays provide two small screens placed in front of each eye of the viewer, so that they always see only one image from the stereo pair. While the autostereoscopic displays use special screens with lenses or barriers to project multiple views from a screen, and as the viewer moves, each eye is limited to a corresponding view.

All of these display types come with tradeoffs from the requirement of glasses, both powered or un-powered, to the effects of the display on color, resolution, or brightness. Each of the display methods has its strengths and weaknesses, but the most important for panorama applications might be in cost, the number of simultaneous viewers, and color reproduction.

Beyond display technologies, recent work has focused on mobile and interactive stereo presentations of content. In particular, the use of stereo panoramas has been studied to provide viewers a comfortable disparity for stereoscopic viewing. Recently, Wither et al. [50] has studied how to comfortably present 3D panoramas as maps and pedestrian guidance. Cabral et al. [7] have studied the generation of 3D floor plans and their 3D visualization using multiple panoramas.

Evaluating good or bad stereoscopic content and its visualization is an important topic with ongoing research. Currently there is no standardized means of evaluating stereoscopic content for viewing due to the ongoing developments of generation and display methods. The following papers present recent developments in this area: In [19], Jensen et al. created a dataset for stereopsis evaluation under various lighting conditions, views, and with structured light models. A robotic arm rig was built to capture small table top sized objects. While this data was created for the evaluation of stereo reconstruction methods, it could also potentially be used for stereo viewing evaluation. Cardoso et al. [8] developed an objective evaluation graphical interface tool for stereo image quality, ImQET. Such tools are leading to the development of a standardized method for evaluating stereo content. As part of this work, recent research has also focused on optimizing the stereo content for the type of presentation display, especially with autostereoscopic displays [11]. In a related topic, Niu et al. [30] has studied where the visual attention of a viewer is drawn using stereo saliency.

# Chapter 3

## Panorama Generation Under Linear Motion

Linear camera motion is one of the most common models considered by many panorama generation methods. This motion is natural for satellite imaging, many aerial flights, and for ground vehicles following straight roads. Linear motion can be split into two ideal cases: (1) the case that the camera is viewing in a direction perpendicular to the motion direction, as shown in Fig. 1.1a. (2) In the second case, the camera is viewing along (parallel to) the motion direction (see Fig. 1.1b), either forward or backward.

During large scale captures of real world scenes with 3D structures, the imagery will contain considerable motion parallax. Given a camera with pinhole geometry, in case 1 the optical flow between frames is parallel to the motion direction, while in case 2 the optical flow between frames radiates circularly from a Focus of Expansion (FOE) (in the forward viewing case). Stereo mosaicking generates panoramas that encode the motion parallax for the exploitation of the 3D information.

The following sections describe the mosaicking methods developed to create panoramas for each of these situations. It is worth noting that in real world captures, the camera view will not be exactly aligned or perpendicular to the motion direction. In these cases the

imagery is simply treated with the ideal case it is closest to, after image rectification using the motion parameters either measured by external sensors such as IMU and GPS, or an image registration algorithm. Real world results show that this works well.

### 3.1 View Perpendicular to Camera Motion

A pinhole geometry camera viewing perpendicular to the motion path is common when imaging from an aerial or ground vehicles and achieved by placing the camera to view directly above, below, or out of the sides of the vehicle. When imaging a scene in this ideal manner, the optical flow between frames will be parallel to the motion direction. Figure 3.1 illustrates how the optical flow will be parallel regardless of the camera roll. (This roll can happen from either a camera that was not mounted aligned to the the vehicle body, or a vehicle that moves diagonally instead of directly forward.)

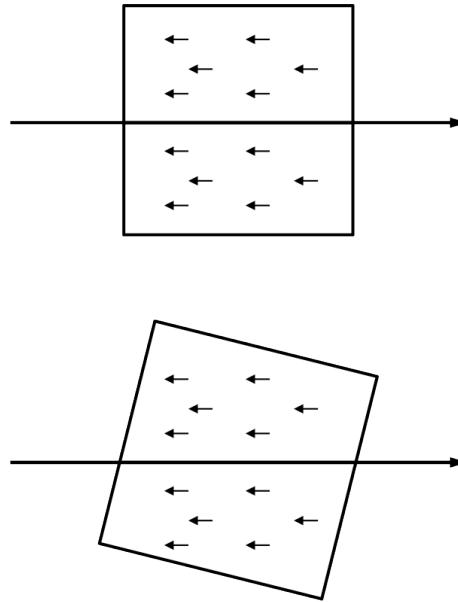


Figure 3.1: (top) The optical flow is parallel to the motion direction when the camera is viewing perpendicular to the motion direction. (bottom) The rotation (roll) of the camera about its optical axis does not affect the optical flow from remaining parallel to the motion.

Constructing mosaics requires that we first align the images of the sequence. This is typically done by taking the first frame as the reference frame to which the following frames

are aligned. This is done with a registration procedure that computes a global transformation (typically affine) between frames using matching feature points or correlation techniques [20, 29, 41].

### 3.1.1 Parallel-perspective geometry and panoramas

The panorama itself is constructed by taking a slit from each aligned frame (which has a perspective-perspective projection according to pinhole geometry) and stitching it onto the panorama. This slit is taken perpendicular to the optical flow from each frame at the same location (fig. 3.2). This procedure is similar to the pushbroom technique of scanning and mosaicking a 1D sensor translating over a scene. This produces a *parallel-perspective panorama* which has a perspective projection along the slit (since the image comes from a normal array imager assumed to have pinhole geometry) and it will have parallel projection along the motion direction, since all slits come from the same fixed location in each image frame (see fig. 3.3).

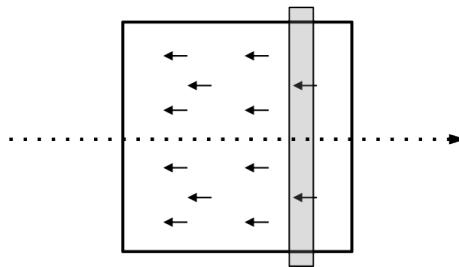


Figure 3.2: A slit taken from a frame.

### PRISM (Parallel Ray Interpolation for Stereo Mosaics)

Building a parallel-perspective panorama from a camera in motion is complicated by factors such as the frame capture rate, the speed of the camera, the camera motion path, and the amount of parallax due to the 3D features of the scene. This usually results in a scenario where the imagery is not dense enough between two captured frames to build a true parallel-perspective projection; where each panorama column comes from the same imaging angle.

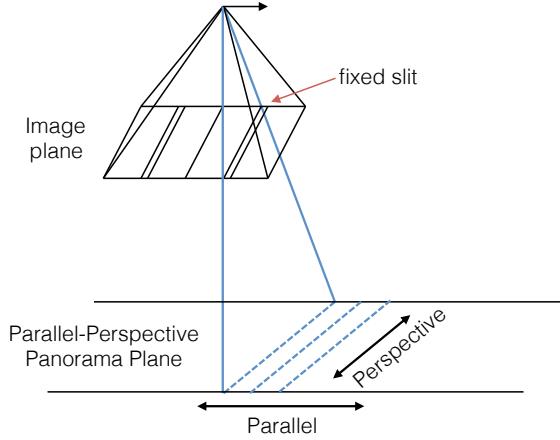


Figure 3.3: A parallel-perspective panorama constructed from a fixed slit.

For this reason, the PRISM (Parallel Ray Interpolation for Stereo Mosaicing) method [57] was developed to interpolate the missing parallel rays from neighboring rays. Figure 3.4 illustrates the basic problem: we have two consecutive frames (frame 1 and frame 2) which image the scene after some ideal translation. We can see here that a feature  $P$  in the scene is imaged by both frames, but not with parallel rays (the bold black lines in the figure). Instead we have a red and blue line depicting the differing angles (different slits) that would actually capture the feature. The PRISM algorithm finds these points along the slits in both frames and warps them by moving them along the dashed arrows so that the parallel-perspective panorama has a consistent parallel view on the scene and does not contain holes or wrong image data in-between consecutive views.

### 3.1.2 Stereo and multi-view panoramas

Stereo images are typically captured as two images taken of the same scene from a slightly offset distance called the baseline, similar to the displacement of the human eyes. Stereo panoramas are constructed by creating two parallel-perspective panoramas that capture the same scene from a different baseline. Since we are following the pushbroom imaging model, this actually means capturing two slits that imaged the same scene from some offset. Since our slits come from a normal array image plane, these two slits will actually correspond to

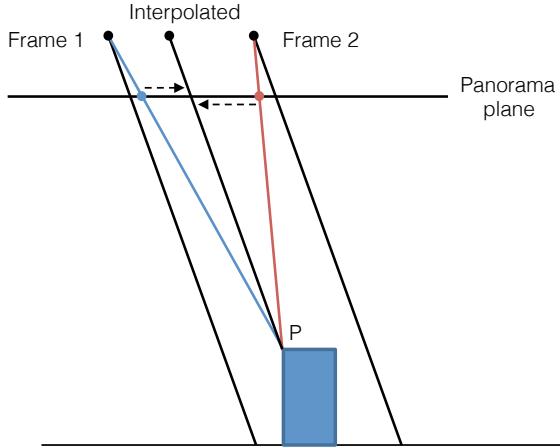


Figure 3.4: Interpolating parallel rays for parallel-perspective panoramas.

two different view angles upon the scene. Figure 3.5 illustrates that a moving camera will capture the same scene point from two locations at two different slits for some baseline, a forward viewing slit and backward viewing slit. We can then construct a panorama from each of these slits producing a stereo pair of panoramas. The depth,  $Z$ , of any point in the scene can then be calculated as:

$$Z = \frac{B}{2 \tan \beta} \quad (3.1)$$

where  $2\beta$  is the angle between the two viewing rays,  $B$  is the baseline of the displacement from where the two images were taken, as illustrated in fig. 3.5. Note that in a pair of stereo panoramas, the angle  $2\beta$  is fixed, whereas the baseline  $B$  varies from point to point. Having an adaptive baseline is actually advantageous in 3D reconstruction [57], as compared to a fixed baseline stereo camera.

We can extend this concept and construct a panorama from many slits from the original image frames (fig. 3.6). Each panorama will have a different view angle on the scene and can be used to adaptively pick the two view-angle panoramas that are optimal for a given application. If that application is 3D reconstruction we pick two panoramas that provide a large baseline, and if the application is 3D viewing we pick two panoramas with a small baseline for viewing comfort. Another benefit of this representation is that the epipolar

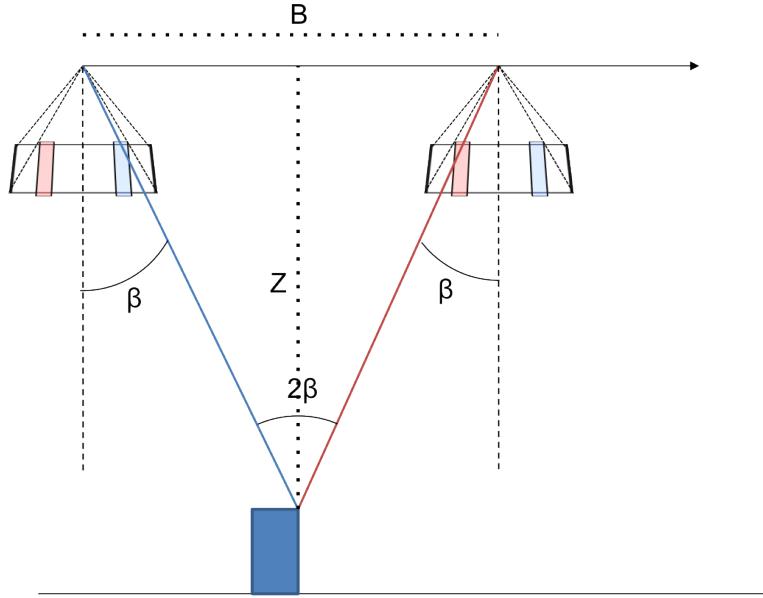


Figure 3.5: Stereo parallel-perspective projection geometry. A point imaged by a forward angle slit will be viewed at a later time in the frame sequence from a backward angle slit.

curves will be parallel lines along the camera motion path and will be aligned in all of the multi-view panoramas.

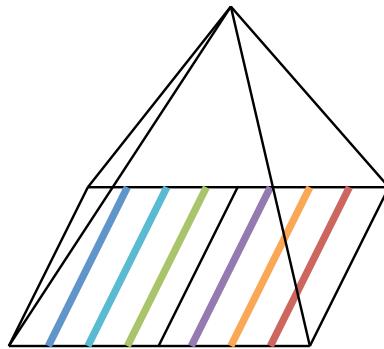
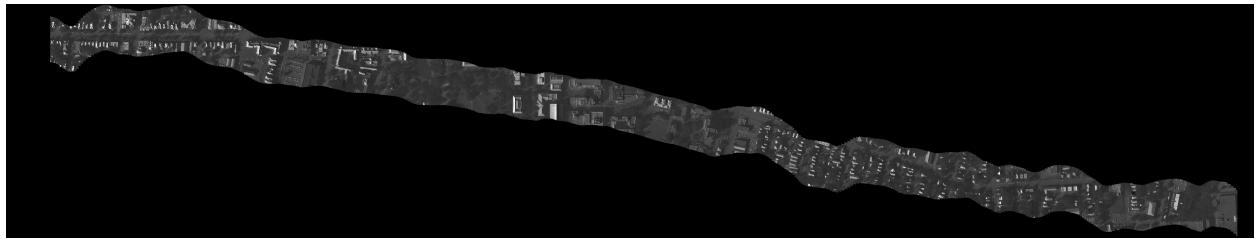


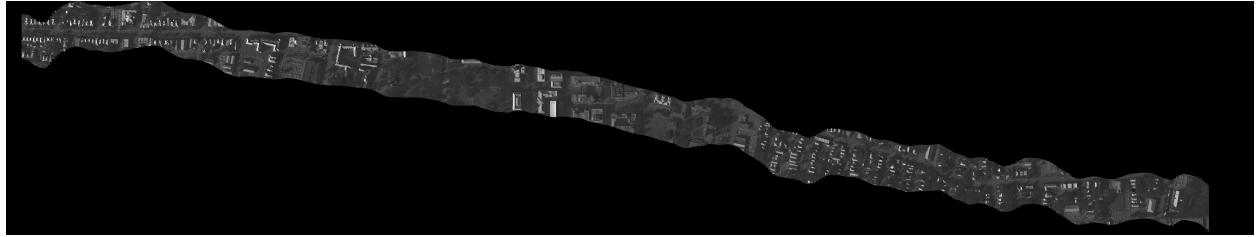
Figure 3.6: Multiple slits can be used to create panoramas. This enables the selections of adaptive baselines for improved 3D viewing and 3D reconstruction.

### 3.1.3 Results

The following are results from real world aerial captures from both EO and IR sensors, demonstrating the method extends to multi-modal imagery.

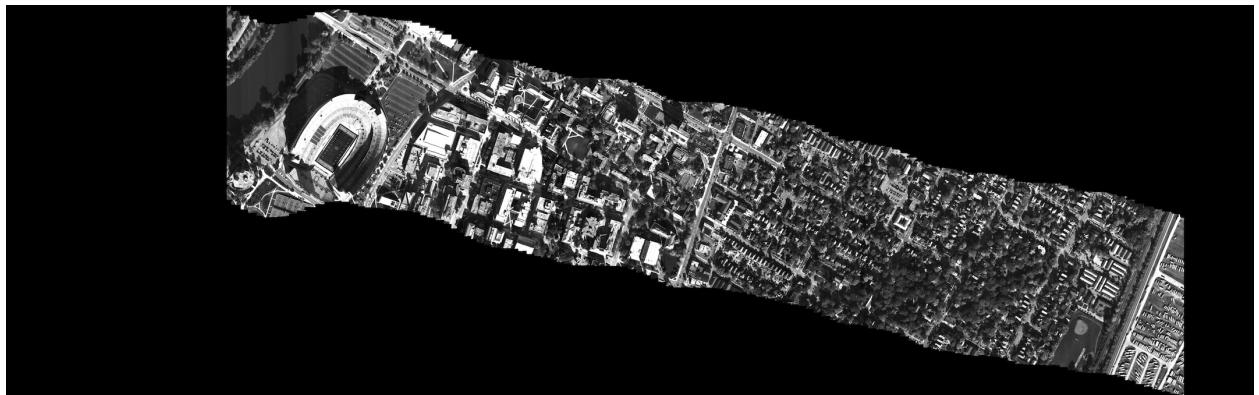


(a)

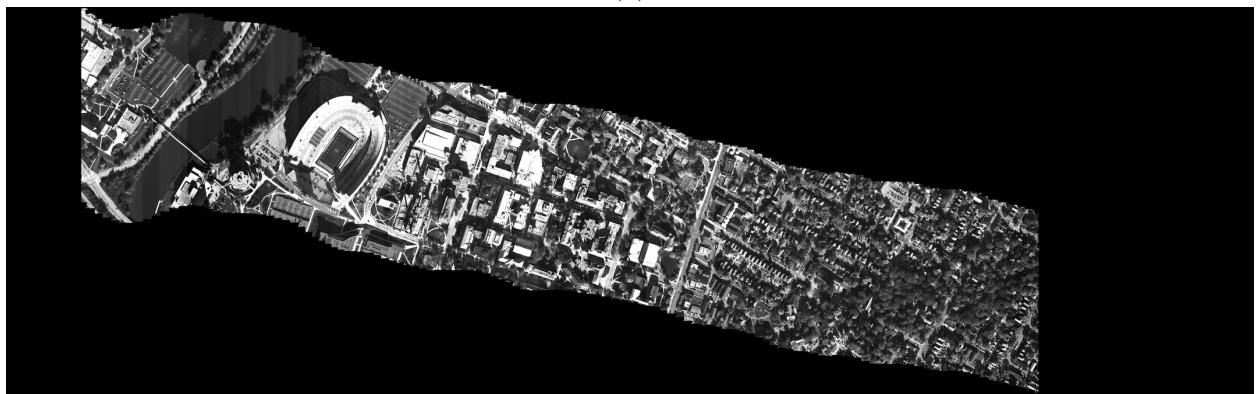


(b)

Figure 3.7: Two of 8 IR panoramas generated of a scene captured with linear motion.



(a)



(b)

Figure 3.8: Two of 8 EO panoramas generated of a scene captured with linear motion.

The data used in these results came from the publicly available Columbus Surrogate Unmanned Aerial Vehicles (CSUAV) dataset [13] (see appendix A). In this dataset, two UAV's were flown over Columbus, Ohio. The EO UAV contained a high-resolution EO sensor, capturing 5 frames per second, at an altitude of 6,500 feet. The other IR UAV contained a mid-wave IR sensor capturing 30 frames per second, at an altitude of 2,500 feet. Each UAV made multiple back and forth passes over their designated linear route. In the results presented in fig. 3.7 and fig. 3.8 we simply consider a single run along the route, and we used the PRISM algorithm to generate multi-view panoramas, of which we show two for each run, a left-eye and right-eye stereo pair.

The panoramas we constructed using the following procedure:

Step 1. *Motion Estimation and Image Rectification.* Frame-to-frame local motion parameters were estimated for the sequence using a pyramid-based correlation match.

Global motion parameters were then computed and frames are rectified so that the mosaics are aligned to linear motion.

Step 2. *Stereo View Selection.*  $2N$  ( $N = 4$  in the above examples) slits were selected for  $N$  pairs of stereo mosaics to provide multiple view angles on the scene.

Step 3. *Mosaic Generation.* A panorama is constructed for each slit across the entire sequence. The PRISM ray interpolation is applied when combining slits from neighboring frames to construct the dense panoramas without gaps or seams.

## 3.2 View Parallel to Camera Motion

When the camera view is parallel to the camera motion, there is a Focus of Expansion (FOE) point that is visible in the sequence of images. Under the ideal case that the optical camera axis is parallel to the motion direction, we will find the FOE in the center of the image. Rousso et al. [40] showed that optical flow of points in the scene will move from the

FOE, radiating outwards towards the edges of the image (see fig. 3.9). Rousso et al. [40] demonstrate that if a pipe is formed around the camera motion path and it intersects the sequence of images, then the optical flow will be parallel to the motion direction along the walls of the pipe. Therefore a panorama can be constructed by projecting frame pixels onto the pipe. The pixels that are projected are those pixels on the curve formed where the pipe intersects the image planes. In the ideal case this curve is a circle (the curve becomes an ellipse as the optical axis diverges from the motion path).

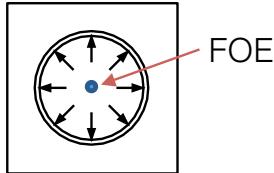


Figure 3.9: The optical flow radiating from the FOE when camera optical axis is parallel to motion direction.

If we wish to construct stereo or multi-view panoramas from this imagery, we identify the slits captured in the imagery to be concentric circles around the FOE. Two pipe panoramas can be constructed from two different circular slits, by mosaicking the circular slits onto a pipe projection. Because the circular slits have different radii, the pipe panoramas first have to be rescaled for stereoscopic viewing.

### 3.3 Unified Stereo Mosaic Framework

In our first step towards a *Unified Stereo Mosaic Framework*, we show that we can treat a view parallel to the camera motion (motion model L(b)) the same as we treat a view perpendicular to the camera motion (motion model L(a)).

If we consider how pipe panoramas are traditionally created for a view parallel to the camera motion, we first must pick two concentric circular slits. Each slit will create its own pipe panorama, and the radius of each pipe will be based on the radii of the circular

slits. Further, in [40] the pipe panorama constructed uses optical flow to interpolate pixels between slits being mosaicked onto a pipe panorama.

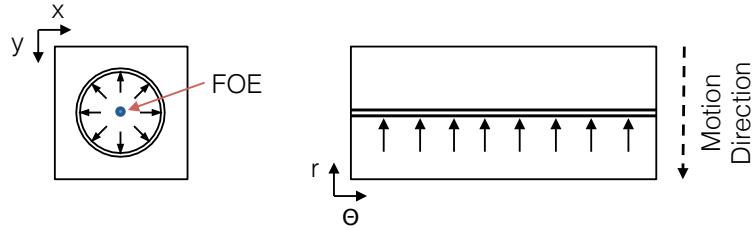


Figure 3.10: Illustrating how an FOE slit can be transformed into a linear slit with a polar transformation. The camera is moving towards the FOE, which is at the line  $r = 0$  in the polar image. Slits are perpendicular to the direction of camera motion in this polar image.

This process of extracting circular slits can be simplified if we first apply a polar conversion centered on the FOE to all of the image frames first. In this manner, the circular slits that must be extracted become linear slits, similar to the case of viewing perpendicular to the camera motion (see fig. 3.10), and the view direction is perpendicular to the linear slits. Therefore, after the polar conversion, we can apply the linear mosaicking procedure from section 3.1. Multiple slits can be identified in this image, to create linear panoramas from the converted imagery for multi-view panoramas. Further, this has the benefit of reusing the simpler linear stereo panorama model. The warped panoramas will represent the walls of the pipe projection used in [40], with the added benefit that the multi-view panoramas are all aligned, and do not require rescaling before stereo visualization. Further, the PRISM or Fast-Layering mosaicking method can be used that uses actual pixel data as opposed to optical-flow estimated pixels.

# Chapter 4

## Panorama Generation Under Circular Motion

Another common method for building panoramas is to constrain the camera motion to a circular path. As fig. 1.2 illustrates, there are three ideal ways the camera view is further constrained. Two of these views, viewing towards the inside or the outside of the circular path are very common and have been researched for many years. The third type of motion, viewing perpendicular to the circular paths plane is a less studied case and is primarily encountered in large scale aerial imaging.

### 4.1 Viewing In

A camera is usually moved on a circular path with a view towards the center of the circle to model an object inside of that path. Often, when this object is small enough, it is simpler to turn the object a full 360 degrees as the camera stays still. Figure 4.1 illustrates the geometry of two rays from a 1D camera capturing such an object.

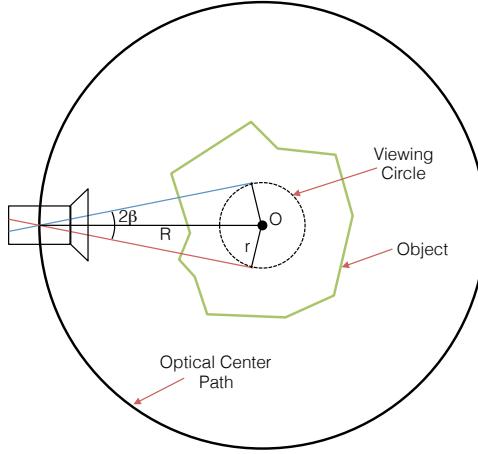


Figure 4.1: Top view of cyclograph geometry for a camera viewing in towards the center of the path.

## Cyclograph Geometry

Regardless of whether the camera or the object is in motion, the geometry remains. It is called a cyclograph geometry. The camera's optical center is rotating around an axis,  $O$  (it's rotation center), at a distance  $R$  from the rotation center, while viewing inside. The figure illustrates two rays being captured with an angular distance of  $2\beta$  that will represent the left eye and right eye stereo panoramas. Each of these rays will capture a 360 degree panorama on a viewing circle tangent to the ray, with radius  $r$ . The radius of the viewing circle can be determined with the following equation:

$$r = R \sin \beta \quad (4.1)$$

If we extend this geometry to a 2D pinhole geometry camera, we observe that the viewing circles epipolar lines will be horizontal and parallel. Therefore the construction of the panoramas can also be done with mosaicking and using the PRISM algorithm to interpolate between slits, as long as we convert the circular path of the camera into a straight line, which simply means that we take the angular position ( $\phi$ , from 0 to 360 degrees) of the camera as a linear position. The panoramic images generated will cover 360 degrees along the horizontal direction (the  $\phi$  axis) and will cover the same vertical pixels ( $v$ ) as the  $y$ -axis in

the original camera (see fig. 4.2). In the physical space, the two panoramas have a circular projection in the  $\phi$  direction, and a perspective projection in the  $v$  direction.

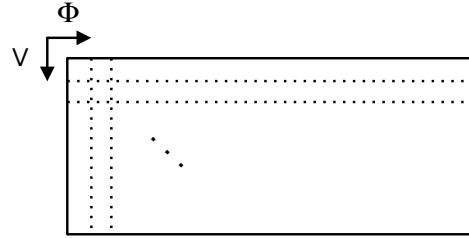


Figure 4.2: Image coordinates for circular projection panoramas

The depth of any point on the object surface can be computed with eq. 4.2. The angular distance,  $\phi$ , of the point is determined by matching the point in both the left and right eye panoramas, at locations  $I_L(\phi_1, v_1)$  and  $I_R(\phi_2, v_2)$ . The angular disparity is then computed as  $\phi = \phi_2 - \phi_1$  (see fig. 4.3). In the case of 2D cameras, the equation remains the same; since  $v_1 = v_2$ , both viewing circles are aligned and the epipolar lines are horizontal for corresponding stereo panoramas.

$$D = \frac{r}{\sin \frac{\phi}{2}} \quad (4.2)$$

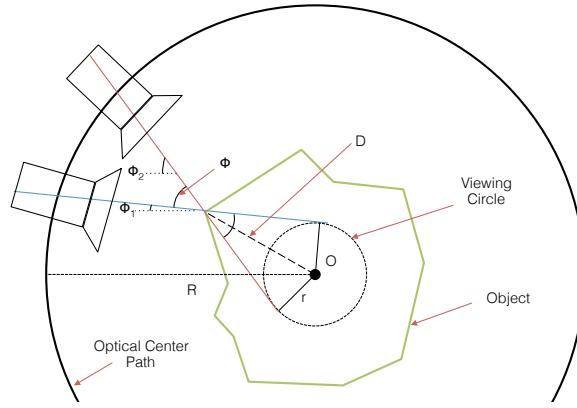


Figure 4.3: Computing depth  $D$  from cyclograph geometry.

In this imaging geometry,  $R$  and  $\beta$  are constrained such that the viewing circle must lie within the object. Note how similar the two equations eq. 3.1 and eq. 4.2 are. In the two equations,  $B$  and  $r$  are related to the baselines, and  $\beta$  and  $\phi$  are the “disparities”. However

there is a major difference. Whereas in eq. 3.1 the angle between two rays is kept fixed for any point and the baseline  $B$  changes, in eq. 4.2 the angular distance  $\phi$  (the disparity) changes and the “baseline” (the radius of the viewing circle  $r$ ) stays fixed.

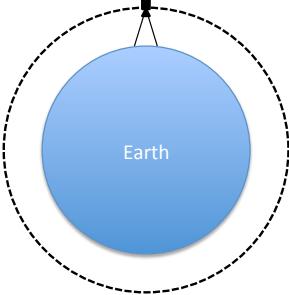


Figure 4.4: A camera flying on a circular path around the earth

For very large scale scenes, such as a plane flying over the earth, we note that it also falls under this geometry. In this case the earth is the object at the center of the circular path; fig. 4.4 illustrates that a plane is actually capturing images on a circular path, where  $R$  is *very large* compared to the distance moved by the plane. Therefore we can simplify these large scale circular cases by handling them with the linear geometry of chapter 3. We look at this in section 4.4.

## 4.2 Viewing Out

A camera moving along a circular path and viewing outward can also be used to capture a 360 degree panorama. 360 degree views are desirable to many wide area surveillance, robotics, and mapping applications (such as street-view and indoor mapping). Typically, this is a circular path with a small enough radius,  $R$ , that it can be implemented on a portable rotating platform that can be mounted to a moving platform that does not take up or miss much scenery. An added benefit of creating a small rotating platform is that it can support multiple cameras of different modalities, commonly these are EO and IR cameras in surveillance applications. In these applications it is useful to construct stereo panoramas for both 3D reconstruction as well as 3D visualization. Since the cameras are typically offset,

and they have different intrinsic parameters (focal length, resolution, etc.), it is necessary to not only build stereo panoramas from all modalities, but it is also important for analysis and exploitation that the two modalities are correctly aligned.

## Concentric Geometry

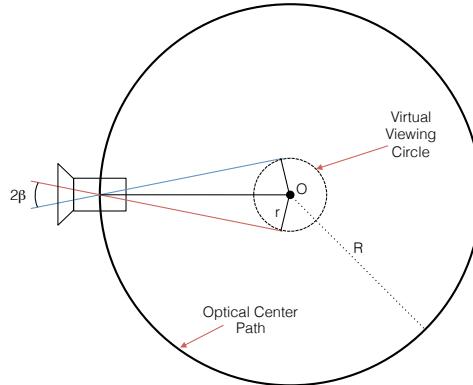


Figure 4.5: Top view of concentric geometry for a camera viewing outwards from the center of the rotation path.

Figure 4.5 illustrates the basic geometry of a single 1D camera on a rotating platform. The cameras optical center follows the circular path around rotation axis  $O$ , at a distance  $R$ , while viewing out from the rotation center. Like in the case of the cyclograph geometry, if we consider two rays with angular distance  $2\beta$ , we can construct a panorama for each ray on a virtual viewing circle, with radius  $r$ . The radius  $r$  can be computed using eq. 4.1. The images captured in this geometry are called concentric panoramas, which are essentially the same as those of the cyclograph geometry illustrated in fig. 4.2.

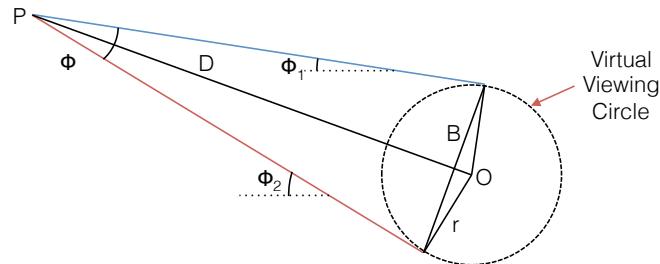


Figure 4.6: Computing depth of a scene point  $P$  using concentric stereo panoramas.

We can compute the depth of a scene point  $P$ , by matching the point in both stereo panoramas at their respective locations,  $I_L(\phi_1, v_1)$  and  $I_R(\phi_2, v_2)$ . Figure 4.6 illustrates how the corresponding rays from each panorama will converge. The depth  $D$  can then be computed with eq. 4.2, where  $\phi = \phi_2 - \phi_1$ . Once again, in the 2D case, we note that the epipolar lines of the stereo panoramas are aligned horizontal lines, therefore  $v1 = v2$  for all static features.

These 360-degree panoramas also have a circular projection in the horizontal direction and perspective projection in the vertical direction, but the flattened panoramas will be parallel-perspective projections, therefore, we can construct the panoramas using the PRISM algorithm as before. We are also able to construct multiple virtual viewing circles, except they will not have the perfect parallel epipolar line geometry as in the linear cases, since the radii of the viewing circles will be different causing misalignments.

## Multimodal alignment

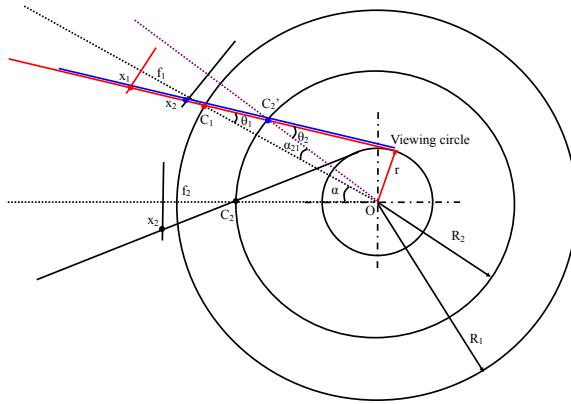


Figure 4.7: Multimodal alignment of two offset cameras on a rotating platform.

In surveillance applications where rotating platforms are used, it is often beneficial to include 2 or more multimodal sensors on the platform. Due to the availabilities and costs of multimodal sensors, such as IR cameras, it is often more practical and inexpensive to build a platform using off-the-shelf components. We can then apply the geometry outlined here to construct 360 degree panoramas for each sensor. But, when this is done, it is nearly

impossible to obtain both EO and IR (or other modality) sensors with the same intrinsic parameters (focal length, resolution, pixel size, etc.) and it is also difficult (or inconvenient due to platform size) to precisely place both sensors at the same radius  $R$ . This is problematic because the sensors will have viewing circles of different radii, and their epipolar lines will not coincide. Therefore, it is ideal to be able to calibrate and align the panoramas, such that the viewing circles of both sensors coincide. In our work [37], our aligned multimodal panorama generation method has the following four steps:

Step 1. *Sensor Alignment and Calibration.* We install the two cameras in such a way that both of them point away from the rotation center. We then calibrate the platform to obtain values for  $R_1, R_2, f_1, f_2$ , the distance from the rotation center and focal lengths of each camera.

Step 2. *Central Panorama Generation.* Since the cameras are placed next to each other at some distance, we must determine the angular distance,  $\alpha$ , between the two cameras. We arrive at this by rotating the platform a complete 360 degrees. We then build a *central panoramic view image* (CPVI) for each camera using the center slit, where the radius of the viewing circle is zero ( $r = 0$ ). We determine  $\alpha$  by finding feature points in both images, and shifting the second one so that the distances between points is set to 0 (in reality there will be some difference so we aim to minimize the distance by shifting by the average distance of all feature points).

Step 3. *Vertical Alignment.* The vertical direction is scaled since the sensors have different FOVs.

Step 4. *Off-center Panorama Generation.* We must align the panoramas for off-center slits by finding feature points in one sensors panorama, and finding the column for which they match in the second sensors image.

Figure 4.7 illustrates the geometry for off-center panorama alignment. Given column  $x_2$  in one camera, we can obtain the corresponding column  $x_1$  in the second camera and the additional shift  $\alpha_{21}$  as follows:

$$x_1 = \frac{f_1}{f_2} \frac{R_2}{R_1} x_2 \quad (4.3)$$

$$\alpha_{21} = \arctan \frac{x_2}{f_2} - \arctan \frac{R_2}{R_1} \frac{x_2}{f_2} \quad (4.4)$$

This procedure was developed to quickly align the multi-view panoramas generated from each of the multi-modal cameras. This method is particularly useful when a full intrinsic and extrinsic calibration is not possible, such as when the scene is texturally similar, which makes matching feature points difficult.

## Results

The following are results from our real platform, outfitted with an EO and IR camera on a controlled rotating platform with a constant speed. Each sensor took 3600 frames, with the EO resolution at 720x480 and the IR resolution at 640x480. The following images in fig. 4.8 show the alignment results by mixing the IR image into the red channel of the color EO images. Due to the channel mixing, cyan colors demonstrate cooler areas while the redder pixels show hotter areas. It can be observed that these hotter areas coincide with computer screens that have been running for hours. The images also illustrate that the vertical FOV of the two sensors was not equal, but the alignment procedure was able to align the two panoramas. This also shows the benefit of generating concentric panoramas. Whereas the original EO and IR images cannot be globally aligned due to the motion parallax, the generated panoramas can. Generating multi-view panoramas also allow multimodal stereoscopic imaging of the scene.

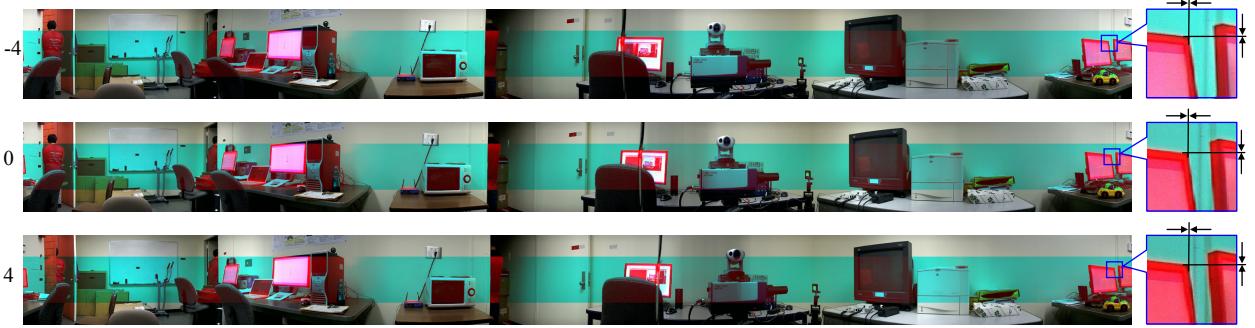


Figure 4.8: Multimodal alignment results for a platform with an EO and IR camera superimposed.

### 4.3 Viewing Perpendicular

In aerial imaging applications such as surveillance and monitoring, it is often the case that there is some area of interest and that the aerial platform is directed to continuously cover that area. A common way of doing this is to have the aerial platform circle an area at a set altitude, while its camera views down towards the ground, perpendicular to the circular path. Large aircraft must make circles with many miles of radius. While it is possible to capture all of that imagery and perform bundle adjustment and reconstruct the scenery, we have explored a method to create stereo mosaics from the scene that can be used for both 3D reconstruction and visualization in realtime. The following sections summarize our work [28] in this area.

### Circular Geometry

First we present the ideal geometry to generate circular mosaics from a pushbroom sensor on a circular camera path. A 1D camera with a single column off the center of an angle  $\beta$ , moves along a circular path with a center  $C$  and a radius  $R$ . The cameras optical axis  $Z$  is perpendicular to the circular path. A circular panoramic image is generated by this scanning camera as illustrated in fig. 4.9. The scanlines of the circular images are circles. Such an image is represented by  $I(\alpha, r)$ , where  $\alpha$  is the angle of the pixel along the circle measured from a starting point, and  $r$  is the distance along the column direction.

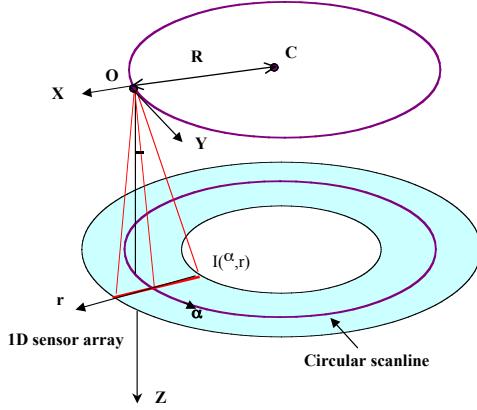


Figure 4.9: Circular flight geometry.

## Stereo Geometry

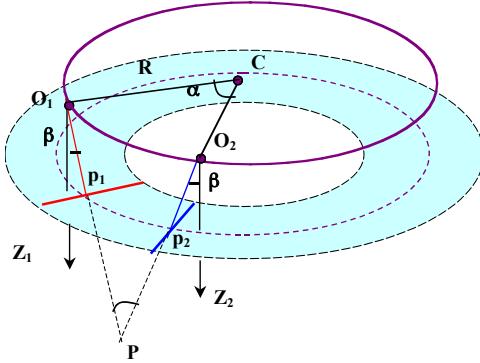


Figure 4.10: Stereo geometry for circular flight.

Given two such circular scanning cameras moving on the same circular path with a center  $C$  and a radius  $R$ , both with viewing angle  $\beta$ , one looking forward ( $O_1$ ) and the other looking backward ( $O_2$ ), a pair of circular stereo panoramas can be generated (fig. 4.10). For any 3D point  $P$ , its correspondences,  $p_1 = (\alpha_1, r_1)$  and  $p_2 = (\alpha_2, r_2)$ , in the two panoramas are (approximately) along a circular scanline. Therefore, we have  $r_1 = r_2$ , and the angular disparity

$$\alpha = \alpha_2 - \alpha_1 \quad (4.5)$$

The baseline  $B$  between the two views  $O1$  and  $O2$  can be calculated as

$$B = 2R \sin \frac{\alpha}{2} = R\alpha, \quad (4.6)$$

where  $\alpha$  is measured in radians, and the radius is much larger than the arc length  $B$  (this is typical of aerial captures and large circular paths). Then the distance from each view ( $O1$  or  $O2$ ) to the 3D point can be calculated as

$$D = R \sin \frac{\alpha}{2} / \sin \beta = R\alpha / (2\beta) \quad (4.7)$$

where  $\beta$  is also measured in radians. Hence the  $Z$  coordinate of the point  $P$  can be computed as

$$Z = D \cos \beta = R \sin \frac{\alpha}{2} / \tan \beta = 2R\alpha / \tan \beta. \quad (4.8)$$

We note here that pushbroom stereo mosaics under a circular motion path are different from multi-perspective stereo panoramas with circular projections. In stereo panoramas with circular projections (eq. 4.2), the optical axis of the camera points to (or away from) the center of the circular motion, while in pushbroom stereo panoramas with circular motion, the optical axis of the camera is perpendicular to the circular motion path. In fact, in all the cases where the optical axis is not pointing to (or away from) the center of the circular path, pushbroom stereo mosaics can be generated by applying image rectification before mosaicing, either to the former case (when the angle between the camera direction and the horizontal direction is smaller than 45 degrees), or to the latter case (when the camera direction and the nadir direction is smaller than 45 degrees).

Second, in circular pushbroom stereo mosaics, the depth error is independent of the depth in theory (eq. 4.8), which is the same as linear pushbroom stereo panoramas (eq. 3.1). Whereas the depth error in stereo panoramas (either cyclographic or concentric) is proportional to the square of the depth.

The two types of pushbroom stereo panoramas (for linear and circular viewing down) can be combined into one model for a more general motion, in that the motion is characterized as piecewise linear and circular. Then if a camera moves on a more general path, a generalized pushbroom panorama can be built along that path, in which the projection is perspective perpendicular to the direction of the motion. If the rates of changes of motion directions are slow, we can fit the motion parameters onto a smooth path that is piecewise linear and circular (with large radii), so that locally the epipolar geometry is still along scanlines.

## Using PRISM: Geometry and Results

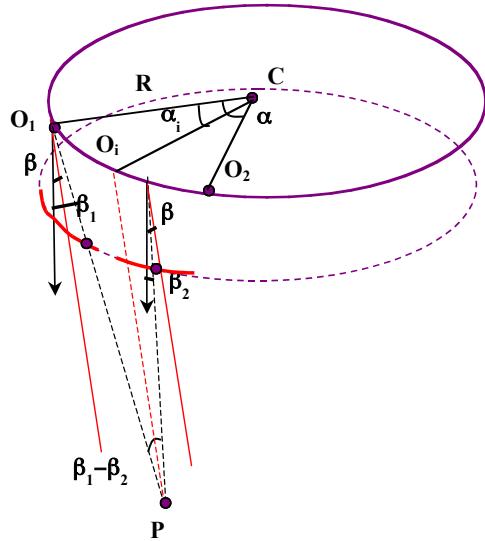


Figure 4.11: PRISM geometry for circular flights.

We have further observed that since matching image points in the stereo panoramas for 3D scene points are found on circles with the same radius,  $r$ , about the center of rotation, we can apply the PRISM algorithm to interpolate in between slits. Figure 4.11 illustrates how interpolation is performed. But we further observe, that if we apply a polar transformation to the imagery with a center at  $C$ , the circles become horizontal parallel lines just like in the linear parallel-perspective case. Below we demonstrate the application of this method to a simulation (fig. 4.12 and fig. 4.13) following an ideal circular path, and to a real world image capture, that follows a circular path (fig. 4.14).

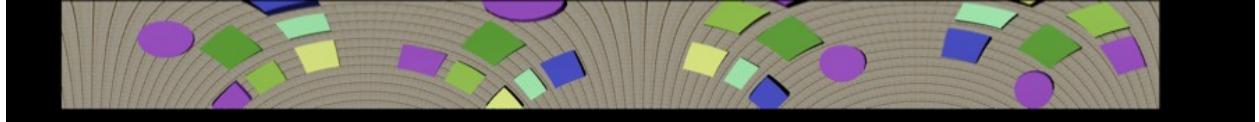


Figure 4.12: A 360 degree panorama for a circular simulation (in polar coordinates).

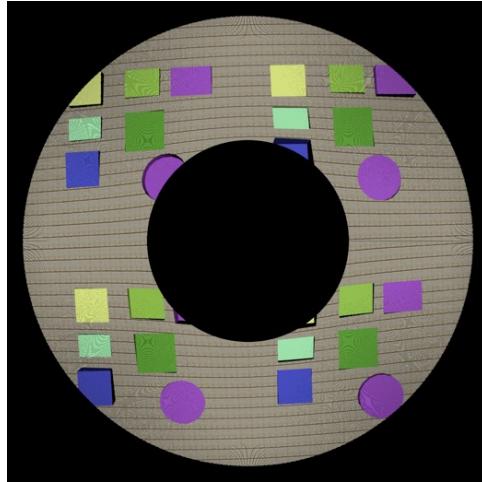


Figure 4.13: Complete circular panorama (in cartesian coordinates).

## 4.4 Unified Stereo Mosaic Framework

We now look at the various circular motion models, demonstrate their similarities to each other and to the linear motion model. This allows us to continue using the PRISM and Fast Layering mosaicking algorithms to create multi-view stereo mosaics. The motion models are: C(a) for viewing out, C(b) for viewing in, and C(c) for viewing perpendicular to the circular camera path.

We first look at how similar motion models C(a) and C(b) are. Both mosaic models produce 360 degree panoramas of the type illustrated in fig. 4.2. For one entire rotation of the circular path, the camera covers 360 degrees horizontally and the same vertical pixels

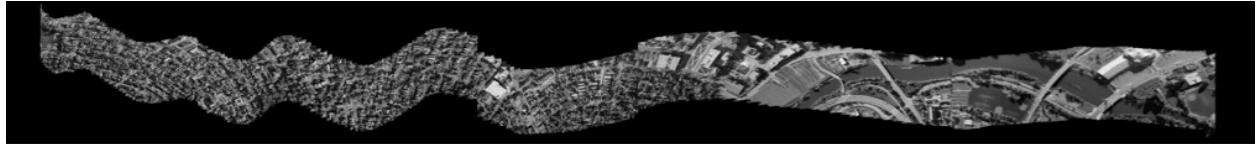


Figure 4.14: A circular panorama for a real scene capture (in polar coordinates).

(v) as the  $y$ -axis of the original camera. The radius of the viewing circles eq. 4.1 are the same as is the depth equation 4.2. If we only consider the imaging rays, we note that the only difference between the two models is the cameras viewing direction along the imaging ray, (see figures 4.3 4.5 4.6).

When we consider large scale scenes, we can show that the cyclograph, C(b), motion model is best approximated by the linear motion model, L(a). If we consider the case of a plane flying over the earth, it is actually on a very large circular path around the earth, to which we can apply the cyclograph geometry. Here we show why it is best to use the linear model instead in terms of depth resolution. This also simplifies the mosaicking model for the imagery, providing better stereoscopic depth visualization.

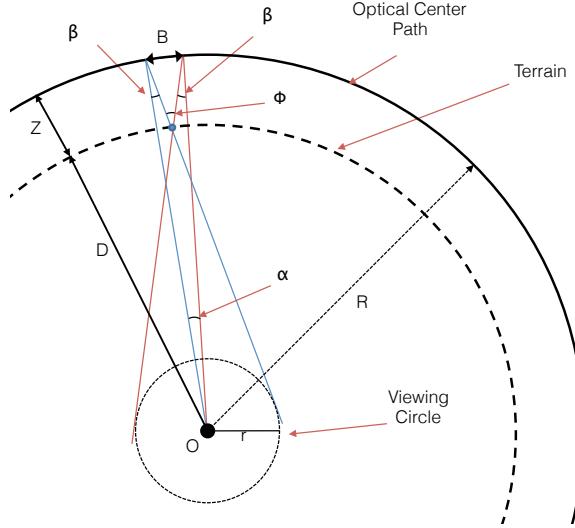


Figure 4.15: Cyclograph stereo geometry for very large scale scenes. A camera's optical center path moves around the earth's terrain.

Figure 4.15 illustrates two ray scanners imaging around the earth. We can model this with both cyclograph and with linear geometry. We note that in the linear case the imaging rays always form a fixed  $2\beta$  angle. However, in the cyclograph geometry, the angle between

the two rays is  $\phi$ , which is larger than  $2\beta$  or is:

$$\phi = 2\beta + \frac{B}{R} \quad (4.9)$$

where  $B$  is the baseline between the two viewpoints of the rays, and  $R$  is the radius of the circular camera path. If we combine equations 4.1 4.2 and 4.10, we get:

$$D = \frac{R \sin \beta}{\sin(\beta + 2B/R)} \quad (4.10)$$

The issue becomes that for a much larger radius  $R$  than baseline  $B$ , the term  $2B/R$  approaches 0, and  $D \approx R$ , as well as  $\phi \approx 2\beta$ . Therefore computing  $Z = R - D$  will result in an erroneous value for  $Z$ . Here we show that combining the linear and cyclograph depths are approximately  $R$ ,  $Z + D = R$  when  $R$  is much larger than  $B$ , we have:

$$\begin{aligned} & D + Z \\ &= \frac{R \sin \beta}{\sin(\beta + B/2R)} + \frac{B \cos \beta}{2 \sin \beta} \\ &\approx \frac{R \sin \beta}{\sin \beta + (B/2R) \cos \beta} + \frac{B \cos \beta}{2 \sin \beta}, [B \ll R \Rightarrow \cos(B/2R) \rightarrow 1, \sin(B/2R) \rightarrow B/2R] \\ &= R + \frac{B^2 \cos^2 \beta}{2(2R \sin \beta + B \cos \beta) \sin \beta} \\ &\approx R, [B \ll R] \end{aligned}$$

Finally, we consider the case of showing that the circular motion model when viewing perpendicular, C(c), can be similarly mosaicked if first converted to a linear case. We do this by first finding the center of the circular path rotation. This center is then used to convert each image frame into polar space. This converts the partial arc scan lines, into straight lines in polar space. Once all frames are converted, the motion path itself is also a straight

line, and the linear model can be applied. We use the PRISM or Fast Layer algorithms to construct the panoramas as seen in section 4.3.

# Chapter 5

## Panorama Generation Under General Motion

Cameras that move with general motion are not typically handled all at once. The typical approach taken by researchers [40, 59] is a divide and conquer approach, that is to segment the camera path into parts that conform to one of the previously reviewed constraints in linear or circular motion. Figure 5.1 illustrates a simple example of a camera moving around a structure, such as a L-shaped building, and how the path can be handled in three consecutive parts, a circular path viewing in, followed by a linear path looking perpendicular, followed by a circular path viewing out.

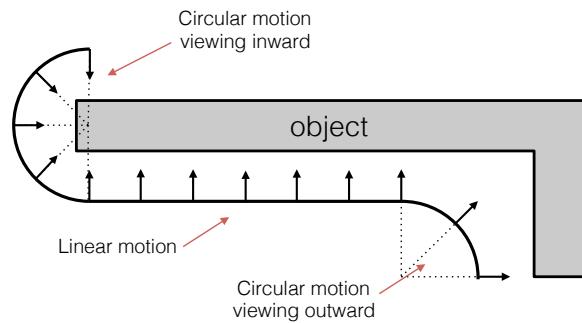


Figure 5.1: A general motion path can be segmented into three constrained paths.

Such methods have not been fully explored in the literature. Handling general motion paths requires that the camera motion path is first estimated, and then analyzed for circular and linear segments. The benefits of such an approach are that the PRISM and other mosaicking algorithms can be used for each segment to construct panoramas. But this will require either manual input or an automated approach that can fit the constrained motion models to the general path in order to be able to segment it optimally. This method also has two more problems: (1) the resulting panoramas for each segment will not be continuous, and (2) estimating the camera paths for long sequences will allow errors in motion estimation (drift) to significantly affect the results of the panoramas.

To deal with these problems in large-scale scenes we examined the applications where general motion is undertaken, particularly in aerial imaging. These paths are usually taken in surveillance and tracking applications, and typically these are all long and persistent captures (taking from hours to days), where the camera continuously cycles around an area of interest. The path is not necessarily circular, but it typically forms a cycle. We developed a new approach, the Direct Layering Approach, to constructing panoramas for general motion paths that can handle such long cyclical motion paths while producing continuous stereo panoramas that are in alignment for the entire sequence.

## 5.1 Direct Layering Approach

Our Direct Layering procedure [27] proceeds as follows:

1. *Motion Estimation:* First, the interframe motion parameters for the sequence of images is estimated using frame-to-frame registration using either correlation or feature point matching approach. The sequence of images can be low frame rate (the only constraint is that consecutive frames must contain sufficient overlap of the scenery for matching).

2. *Base Cycle Registration and Error Correction:* The next steps are to (a) compute the global motion path of the sequence, (b) identify where the cycle closes, and (c) correct drift errors from the base cycle path.

- (a) The global motion is computed by making the first frame the reference frame, and applying each subsequent interframe motion to the previous frame. This gives each frame a global position on the path relative to the reference frame (frame 1).
- (b) Identifying when the cycle closes can be done through the analysis of the motion path (although it contains some error) or simply by matching every new frame with the first frame to detect when the cycle closes.
- (c) Due to drift in the global registration process, it is necessary to perform some correction. This is done by computing the total error accumulated, which is determined at the cycle closing, the last frame will not align with the first frame and is taken to represent the global error. That total error is then redistributed in a weighted manner across all frames such that the global alignment is correct and the cycle closes. The weight for each frame is determined by computing the interframe error in registration using the SAD (sum of absolute difference), SSD (sum of squared difference), MSE, or RMSE metrics.

3. *Layering Approach for Fast Multi-view Mosaicking:* After computing the camera motion and correcting its global drift we can construct a set of multi-view panoramas for the video scene. We use a layered approach for fast multi-view mosaic generation. The basic principle is the following. Based on the global motion parameters, buffers for multiple empty layers are created, and are laid out in the order of the 1st layer, the 2nd layer, and so on, each below the previous one, as can be seen in fig. 5.3. All of the original frames are warped based on their global motion parameters and the warped frames are laid out in the panorama space. Starting from the first layer at the top,

pixels of each warped frame are placed down through all the layers until they hit the first empty layer and it gets drawn there. If it hits a layer (layer  $n$ ) with existing pixels at its location, it continues onto the next layer (layer  $n + 1$ ), until it falls on an empty layer.

The procedure `GENERATEPANORAMAS` (Figure 5.2) outlines our approach to constructing a set of multi-view panoramic mosaics. The procedure takes as input an ordered sequence of the video images (along with its computed parameters),  $Img_s$ , and a reference to the ordered sequence of multi-view panoramas,  $Pano_s$ .

Figure 5.3 shows an illustration of drawing three image frames onto the first two multi-view panoramas. Here we see that  $img[3]$  paints its leading slit onto  $pano[1]$ , and the following slit onto  $pano[2]$  and so on. Note that the slits contributing to a particular panoramic layer come from very similar perspective directions in the original images, thus minimizing misalignments between slits. In the ideal case, the camera performs a pure 1D translation in the  $Y$ -direction, and each image contributes a single  $X$  column to each panoramic layer and therefore it forms a perfect parallel-perspective (pushbroom) mosaic [57]. In more general cases, each layer approximates a multi-perspective panorama with similar viewing directions. Thus two layers can form a pair of multi-perspective stereo panoramas.

Our goal with the Direct Layering method is to provide 3D viewable results quickly without requiring the use of the PRISM algorithm. Warping and PRISM interpolation operations in particular can get computationally expensive as the imagery resolution keeps increasing. Blending operations can produce ghosting which can make it distracting for some users when viewing the imagery on a 3D display. Instead we rely on the fact that our drift correction method has produced results with very minor local misalignments that the viewer can cope with, in particular because we performed the layered construction approach, where similar views align into each layer. We understand that this is a subjective issue and will vary from user to user. Nevertheless, the multi-view panoramas thus generated naturally

**Require:** All *Panos* set to *empty*

```

1: procedure GENERATEPANORAMAS(Imgs, Panos)
2:    $N \leftarrow \text{COUNT}(\text{Img}s)$ 
3:    $L \leftarrow \text{COUNT}(\text{Panos})$ 
4:   for  $t \leftarrow 1$  to  $N$  do
5:      $\text{img} \leftarrow \text{Img}s[t]$ 
6:     for  $l \leftarrow 1$  to  $L$  do
7:        $\text{pano} \leftarrow \text{Panos}[l]$ 
8:       for all  $\text{img}(x, y) \neq \text{empty}$  do
9:          $(x', y') \leftarrow \text{PROJECT}(\text{img}, x, y)$ 
10:        if  $\text{pano}(x', y') \text{ is empty}$  then
11:           $\text{pano}(x', y') \leftarrow \text{img}(x, y)$ 
12:           $\text{img}(x, y) \leftarrow \text{empty}$ 

```

Figure 5.2: GENERATEPANORAMAS procedure.

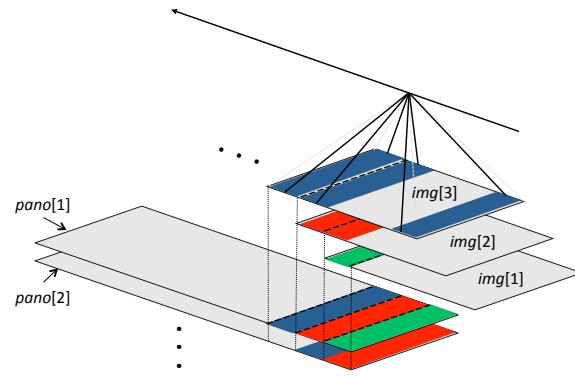


Figure 5.3: Illustration of the GENERATEPANORAMAS procedure. Under ideal conditions, each multi-viewpoint panorama  $\text{pano}[i]$  is constructed from a similar view angle on the scene.

represent varying viewing angles (fig. 5.3), and in an ideal case, such as under a linear motion, the panoramas would be parallel-perspective.

### 5.1.1 Multiple Runs and Results

In addition to creating multi-view panoramas for the base cycle, we use the base cycle as a reference panorama for subsequent cycles in the persistent imagery. For all of the subsequent cycles frames, we developed a *spatial-temporal weighing method* to globally register each frame from its previous frame and from the base cycle, and we weigh these two results once again using an error metric like the SAD metric. This weighed result will handle drift as new

frames arrive, allowing us to construct the panoramas in real time. Since the spatial-temporal method uses the base cycle, we are guaranteed to construct panoramas that are aligned across all cycles, and the panoramas are continuous. The layered approach is used to generate sets of multi-view mosaics from both the base cycle as well as the subsequent cycles. Again, the multi-perspective panoramic geometry here is the key factor that allows us to perform spatial-temporal alignment under obvious motion parallax; and the multi-view panoramas preserve the motion parallax for 3D reconstruction and visualization. Furthermore, alignment of multiple runs also provide an efficient and effective way to detect changes over time, caused by movers (e.g. vehicles), illumination, and changes of the scenes (such as new buildings). In fact multi-view mosaics can also be generated from aligned image frames that come from across all runs, providing dense imagery overlap for multi-view mosaics. This will be left for our continuing work.

Figure 5.4 demonstrates the global motion path estimation of multiple cycles without our method (note how much effect drift has on the path), while the right figure illustrates the path using our base cycle drift correction method and the spatial-temporal method applied to subsequent cycles. Figure 5.5 shows results for a base cycle and a second cycle after applying our method.

## 5.2 Unified Stereo Mosaic Framework

In this chapter we developed the Fast Layering algorithm to directly generate multi-view panoramas of a scene. When the overlap of the frames is substantial, the results are similar to the PRISM results. The procedure chooses slit size based on the amount of motion between frames, so it does not require interpolation. For visualization purposes, the results are good and without perceptible seams.

The slits collected in the Fast Layering method are perpendicular to the current position along the computed camera motion path. All multi-view panoramas will have that property

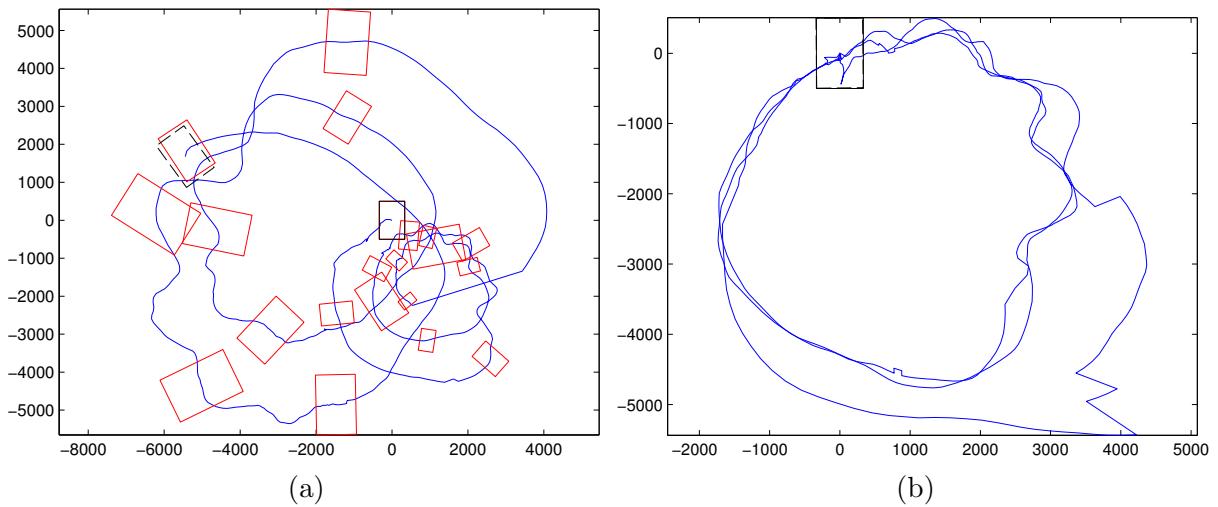


Figure 5.4: The CLIF 2007 sequence makes 4 approximately circular passes of an area. The units in both horizontal and vertical axes are in pixels. The dashed black box represents the last frame. Red boxes represent every 50th frame. (a) Image alignment without error correction. (b) Image alignment after error correction.

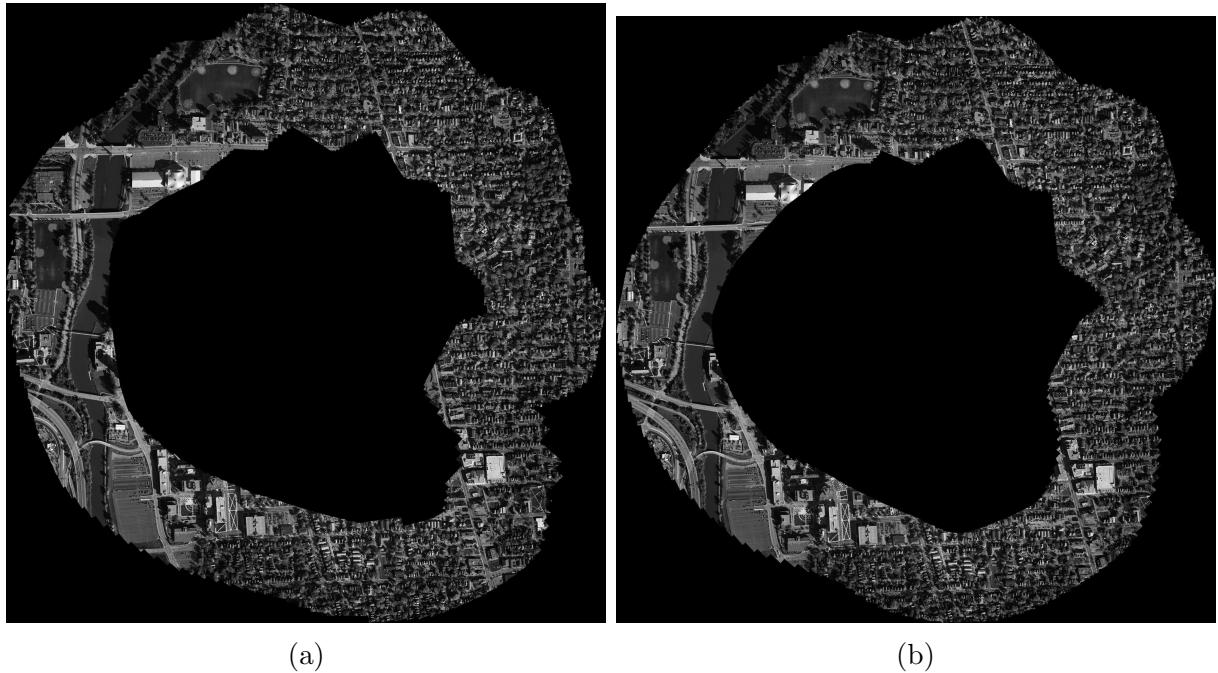


Figure 5.5: Panoramas for (a) base cycle and (b) cycle 2.

for any given frame. As incoming frames are added to the panoramas the path remains approximately straight for a few frames. If we only consider a window of frames at a time, their resulting stereo panoramas are approximately linear, and their epipolar lines are also approximately horizontal. This property allows us to view these panoramas stereoscopically via a window, and we can pan along the cameras path, by rotating the imagery, such that the epipolar lines and path are locally horizontal so that they align with the human visual system. These panoramas cannot be viewed as a whole stereoscopically, as only a small part will satisfy our human visual system.

# Chapter 6

## Stereoscopic Visualization using Multi-view Panoramas

One of the benefits of constructing multi-view panoramas is that it allows users to visualize large scale scenes stereoscopically in real or near real time, even before a full 3D reconstruction operation is performed. For users in surveillance, search and rescue, and other time sensitive applications, the speed of 3D analysis is crucial. Therefore, we have addressed many issues that arise in visualizing large scale scenes with multi-view panoramas.

### 6.1 Stereo Viewing and Motion Path Alignment

In order to view a scene stereoscopically we need a left-eye and right-eye image pair. The stereo panorama methods discussed in the previous chapters all generate left-eye and right-eye pairs from two slits and using the PRISM or Fast Layering methods. If we consider the parallel-perspective panoramas generated for linear motion, we know that they have parallel epipolar lines that follow the linear motion path. For stereo viewing we must simply combine the two panoramas into an anaglyph or an appropriate stereo format to view stereoscopically, while maintaining the epipolar lines horizontally so that they align with the human visual

system. This is easily done by rotating the entire panoramas such that the estimated motion path is horizontal and aligned with the eyes.

We have developed a multi-view stereo viewer application, MVSV, which can load in a multi-view panoramas of a scene, it automatically rotates the panoramas into horizontal alignment, formats the stereo pair for the available stereo display, allows the user to adaptively choose the baseline for the stereo pair they want to view, allows the user to pan and scale the panorama, and allows the user to pick an angular view of the scene.

## Non-linear motion alignment

For the circular and general motion path panoramas, the alignment is a bit more complex. In chapters 4 and 5 we saw that the epipolar curves followed the motion path when building multi-view panoramas of the scenes. Since these epipolar curves are no longer linear, the panoramas are no longer viewable as a whole, since only parts of the motion curve will align horizontally with the viewers eyes.

Our MVSV application can still produce a stereo view by loading in the motion trajectory for the loaded multi-view panoramas, and modifies the left and right panning motions along the panorama to follow the motion curve, fig. 6.1, applying the necessary rotations to the panorama, such that the local portion of the viewable motion curve fits closely to a horizontal line. This ensures that the viewable portion of the panorama has epipolar curves that align well with the human eyes. With this addition, all of the other features, such as display, scaling, adaptive baseline, and view angle are all still useable.

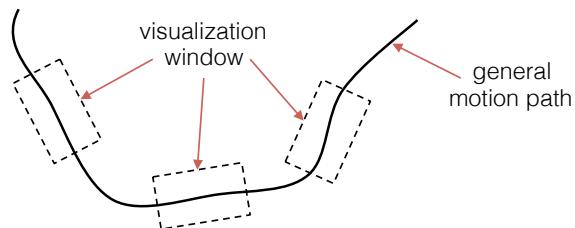


Figure 6.1: The visualization window follows along the motion path, keeping the local windows epipolar lines approximately horizontal.

## 6.2 Multi-modal / Multi-pass Alignment

In our circular path work with a rotating platform and multi-modal sensors, we generated multi-view panoramas for each sensor, the EO and IR sensor. We described in section 4.2 how we aligned the two cameras even when their intrinsic parameters differed. After aligning the two sensors, we used these parameters to scale the smaller resolution sensor onto the sensor with the larger resolution. For these applications the MVSV application allows the user to load the multi-view panoramas for both cameras, and then allows the user to pan, and scale as usual. In addition to the usual behavior, it also allows the user to interactively switch between the two modalities, all while viewing in stereo.



Figure 6.2: (a) MVSV displaying EO in stereo. (b) MVSV displaying IR in stereo. (c) MVSV displaying IR overlay on EO (not in stereo).

In addition to switching between the two modalities, it was also found helpful to allow one modality to be overlaid on the other in the red or green channel. While these overlays are

not in stereo, it was found helpful in analyzing the two modalities simultaneously. Figure 6.2 shows an example of this feature.

In our general motion work, particularly those cases with multiple passes over an area of interest, we generate multi-view panoramas for each individual pass. Since all passes are aligned in their motion path, we also allow the user to interactively switch between the different available passes no matter which section of the pass they are currently viewing. The implementation of this is similar to the multi-modal case, in that each pass can be considered a different modality that must be maintained in alignment as the user interacts with the scene.

### 6.3 Interactive Viewing Choices

In order to improve the user’s immersive experience, we extended the MVSV application to provide interactivity by allowing for an adaptive baseline/disparity, interactive view angles, and automatic viewer tracking.

Before describing the feature details, we briefly describe how the human visual system (HVS) perceives depth. The HVS uses many pictorial clues such as size, occlusion, perspective, and focus among many others, to determine relative depth, as well as using parallax and stereopsis to estimate distance [22, 38]. When we present a viewer a stereoscopic image, we are presenting the HVS system a corresponding image to each eye such that the HVS uses stereopsis to fuse the scene and perceive depth. But this presentation can often lead to viewer discomfort, primarily because the pictorial clues may mismatch with the information from stereopsis.

In particular, two main factors lead to viewer discomfort, a mismatch between *accommodation* and *convergence*. Accommodation is each individual eye’s ability to focus light at some distance, and convergence is the ability of both eyes converging onto the same point. As an example see figure 6.3, it represents a top-down view of a viewer looking at a stereo

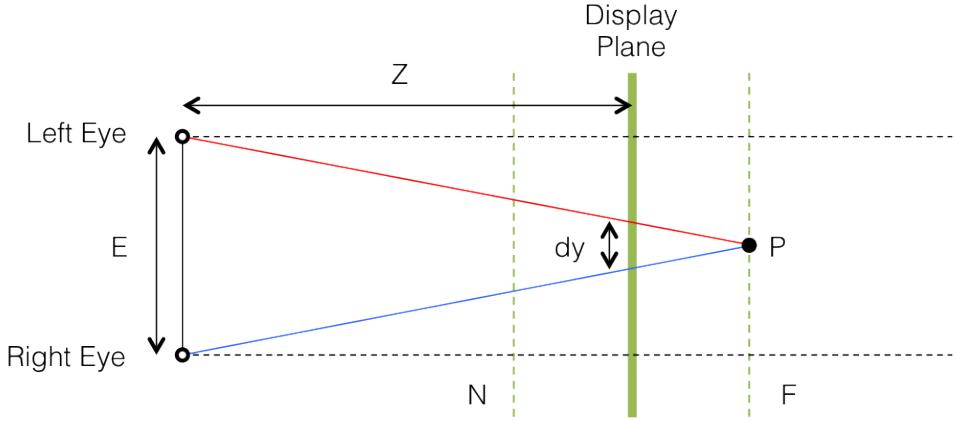


Figure 6.3: A viewer perceiving depth beyond a stereo display.

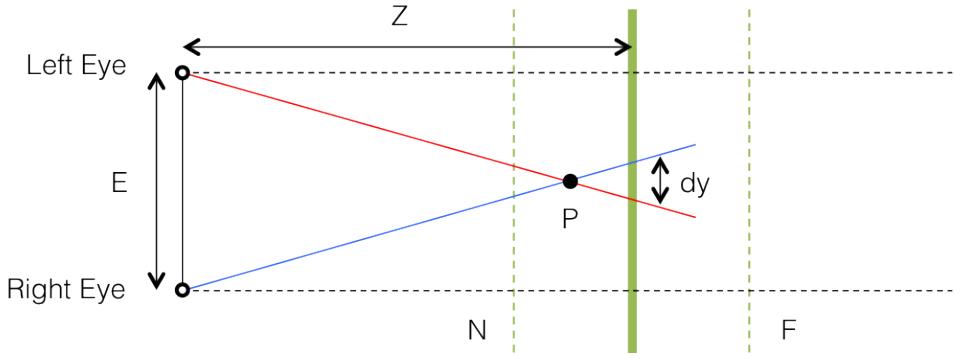


Figure 6.4: A viewer perceiving depth in front of a stereo display.

display. In the figure, the viewers eyes are converging on a point  $P$  that is beyond the distance of the screen (that is controlled through the disparity), but each eye will focus on the first object it sees, which is the screen at distance  $Z$ . The larger this mismatch grows, the more strain it adds to the HVS. Note also, that the farther away the point is from the viewer, each eyes viewing direction begins to approach the parallel lines, which will also cause discomfort as the eyes are focused much closer.

Jones et al. [22] conducted a human factors study to determine what the comfortable depth range was, and determined it to be dependent on the viewers eye separation, their distance to the display, and the display size. Jones et al. [22] arrived at a range of 50mm in front to 50mm behind the screen for comfortable viewing. Battiatto et al. [4] used a maximum

disparity range instead of half of the viewers eye separation measurement to constrain the allowable depth range in front and behind the screen.

### 6.3.1 Adaptive Baseline/Disparity

When we construct stereo panoramas, they are built from two slits that are a disparity of  $dy$  pixels apart, or  $dy/2$  pixels to the left and right of the center (this  $dy$  is different from the visualization  $dy$  from fig. 6.3 and fig. 6.3). To construct multi-view panoramas we pick various  $dy$  distances to generate a series of panoramas all with different view angles upon the scene. This enables us to adaptively pick the baseline, depending on whether we want to reconstruct the scene or view it. Our MVSV application allows the user to interactively pick the disparity which is best suited to viewing the scene. The optimal viewing disparity is dependent on the viewing conditions, such as the display size, viewer distance to the display, and depth range in the scenery. In particular, if we use the method proposed in [4] to compute the max pixel disparity allowable, we need to compute the pixels/inches of a given display. If the eye separation is 3 inches, than the max disparity should be no more than 1.5 inches. We translate that to pixels as follows:

$$Disparity_{max} = E/2(ScreenPixels/ScreenWidth)$$

Some common max disparities for displays are:

- 65" (1920pix): **50.9pixels** =  $1.5 * (1920/56.5)$
- 15" (1680pix): **193.8pixels** =  $1.5 * (1680/13)$
- 24" (1920pix): **140.5pixels** =  $1.5 * (1920/20.5)$

### 6.3.2 Interactive View Angle

Once the user has picked an optimal disparity with which to view the scenery, it is not necessary to keep the view centered around the center of the frame. Since there are multi-view mosaics, the user can choose to slide their view along the multi-view mosaics to change

the angle of the view on the scene. In some cases, the user could have the impression that they are seeing around occluded regions (such as a car under a bridge, ground under a tall tree, and the side of a building) by doing this. Figure 6.5 demonstrates how a scene being viewed with a constant disparity, changes when the view angle slides left, to center, and all the way right in the multi-view framework.



Figure 6.5: MVSV allowing the view angle to be changed in stereo. (a) MVSV displaying the left most view angle. (b) Viewing through the center angle. (c) MVSV displaying the right most view angle.

### 6.3.3 Automatic Viewer Tracking

We developed an automated viewer tracker that can set the disparity and view angle based on the viewers position relative to the screen. We know that optimal disparity is a function of the display size, and viewer distance to the display, and that it gets bigger the farther away from the screen the viewer is. By tracking the users position, the disparity can be adjusted automatically. In addition, for large displays where the viewer is free to move about the display, we use the viewers position as they move from the left to the right of the displays center to adjust the view angle.

We can track the viewer's distance and position either using an RGB-D sensors or a face tracking algorithm with a regular 2D camera. In this way, we set the disparity and position parameters automatically as the user moves in front of the display. In this work we used a

face tracking algorithm, where the centroid's  $x$ -position is used to set the position. We can map the users  $X$ -position in the real world using a known feature, such as the eye separation in inches as follows:

$$Z = f \frac{\text{eye\_sep}_{in}}{\text{eye\_sep}_{pix}} \quad (6.1)$$

$$X = x \frac{Z}{f} \quad (6.2)$$

By combining these two we don't have to compute the focal length ( $f$ ) of the camera, instead we can use an estimated eye separation for the viewer for a close approximation.

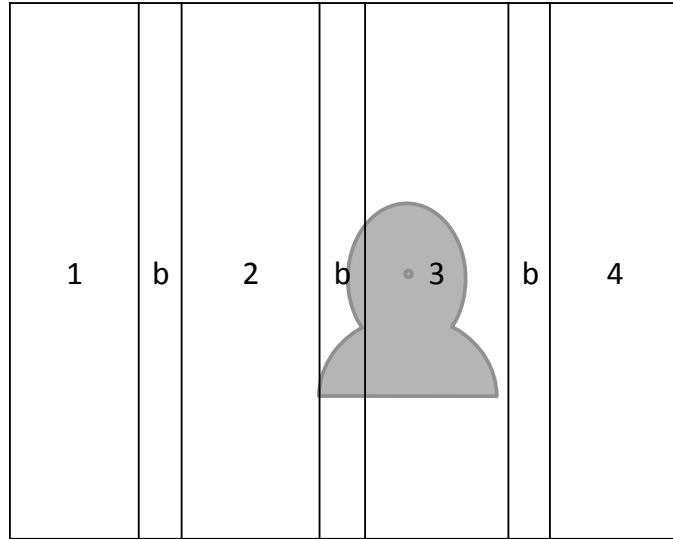


Figure 6.6: Interactive view angles are changed when viewer's detected face enters a new numbered view zone.

Once we know the users position, we also need to know how many view angles are available. Given  $N$  views, we split the camera's view into  $2N - 1$  columns ( $N$ -view zones and  $(N - 1)$ -buffer zones, see fig. 6.6). As the user moves parallel to the screen, the user enters numbered or buffer regions. The view ONLY changes when the face's centroid enters a numbered region. In this way, if the user is on the edge of any two zones, rapid changes

will not be triggered. This is especially necessary with face tracking, which can contain a few pixels of variation in detection size and centroid.

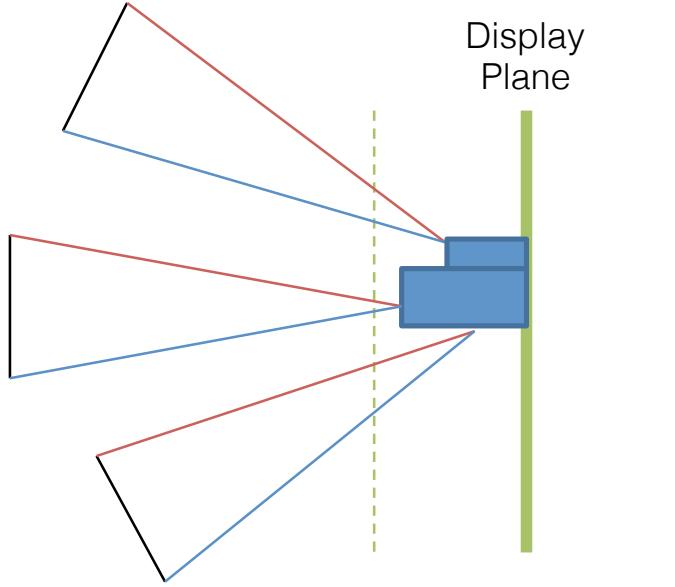


Figure 6.7: As the viewer moves parallel to the screen, the view angle is changed to provide corresponding views.

With the user tracked, the view angle is automatically changed so that the viewers view corresponds to their location parallel to the screen, as illustrated in fig. 6.7.

## 6.4 Stereoscopic Display Types

The MVSV application was developed to support a wide array of stereoscopic display methods. The simplest display method it supports is color-multiplexed methods. These are anaglyphs, such as the common red-cyan images which take their red channel from the left-eye image, and the blue and green channels from the right-eye image. It can also support other color combinations, although they are not implemented at this time. This method can be used on any display and requires cheap passive glasses with color filters.

Another common method of displaying stereoscopically is active shutter glasses. This requires special displays that can time-multiplex two images to the corresponding eye. The MVSV application supports these displays by combining the left-eye and right-eye images

into the appropriate formats. The common formats are row-interlaced, check board interlaced, and side-by-side formatted. These formats are also used by displays that perform polarization multiplexing with passive polarized glasses, and by some auto-stereoscopic displays.

## 6.5 Memory Management

Memory is one of the biggest constraints the MVSV application has to deal with. When we generate panoramas of a large scale scene, they are usually taken with high resolution sensors which can easily produce panoramas that are  $20k \times 20k$  pixels or more. It is usually not a problem to load one or two very large panoramas into memory for most computers. But once we start operating on multi-modal or multi-pass imagery that has around 10 multi-view panoramas per modality or pass, we quickly run out of memory. Instead, the MVSV application produces  $1k \times 1k$  tiles for each panorama at various scales that can be quickly loaded from disk as needed, and allows the program to unload tiles for any section of the scene not currently in view. This memory management process is not only useful for the maintaining speed during visualization but is also used when constructing very large panoramas as was described in section 5.1.

# Chapter 7

## Conclusion and Discussion

### 7.1 Unified Stereo Mosaic Framework

In this dissertation we developed a stereo panorama generation framework for large scale scenes captured under various constrained motion models. We first explored how to generate stereo panoramas for each motion model and at the end of each chapter, we showed how they are similar, to show that our general motion method, through some conversions, is equivalent to the panorama generation methods for constrained motion models. In summary we showed:

- that the linear motion  $L(b)$  is equivalent to linear motion  $L(a)$  through a cartesian to polar conversion. This allows us to use either the PRISM or Fast Layering algorithms for stereo panorama construction. (In section 3.3)
- that the circular motion  $C(a)$  and  $C(b)$  are equivalent motion models sharing the same stereo panorama generation methods. (In section 4.4)
- that for large enough circular motion, such as imaging around the globe, the circular motion  $C(b)$  is best approximated by the motion model  $L(a)$ , producing better results. (In section 4.4)

- that the circular motion  $C(c)$  can be converted to the linear model  $L(a)$  by a cartesian to polar conversion of the imagery about the rotation axis. (In section 4.4)
- that the general motion model developed here  $G(b)$ , is constructed so that any local section of the camera path is treated as linear model  $L(a)$ . (In section 5.2)
- that by using the general or linear motion model for any of the motion models, we are able to maintain the local epipolar geometry of the stereo mosaics as horizontal parallel lines. This allows us to directly stereoscopically visualize the stereo panoramas for any general motion. (In section 5.2)

## 7.2 Large Scale Scene Issues

In this dissertation, we also considered other problems that arise in applications with large scale scenes. We considered how to generate panoramas for general motion, how to align imagery coming from multiple-runs or multi-modality sensors over the same scene, and how to interactively visualize the data stereoscopically.

- *Stereo Mosaic Layering:* In section 5.1 we developed a Fast Layering algorithm as an alternative to the PRISM algorithm, that allows for faster panorama generation of general motion scene captures, while not greatly sacrificing quality.
- *Multi-Run and Multi-Modal Mosaic Alignment:* In section 4.2 we developed a method for the alignment of 2 multi-modal cameras with different intrinsic parameters on a rotation platform. The method ensures that the multi-view panoramas generated for each camera are in alignment to the other camera's multi-view panoramas. In section 5.1.1 we developed a method for the alignment of multi-view panoramas created for multiple passes over the same scene with a general motion path. The method identifies a base cycle in the imagery and then proceeds to align multiple passes in

a spatial-temporal manner. When multi-view panoramas are generated for each pass over the scene, the panoramas will be in alignment.

- *Intelligent Stereo Visualization:* Finally, in chapter 6 we present our work on the MVSV program, which interactively visualizes the panoramas to a user or operator. The MVSV program is able to visualize general motion panoramas by following the motion path, and presenting the user a windowed view to the imagery that maintains the epipolar lines locally horizontal for the viewer (this permits stereo viewing). Further, the program is able to track the users face to provide and immersive and interactive experience. The program automatically switches the view from the multi-view panoramas so that it corresponds to the viewers position to the left or right of the screen, and it is able to set the stereo disparity so that it is appropriate based on the users distance from the screen.

## 7.3 Discussion

In the development of this work we implemented the multi-view stereo panorama generation methods and the MVSV visualization program in C++ and Python. Here we discuss the various implementation and data related issues we faced in developing this work for large-scale scenes. In particular we look at the computation costs, memory requirements, and evaluation issues.

### Computation Costs

In our work we aimed to develop methods that could be applied for the realtime generation and visualization of multi-view panoramas. During the generation step, the most costly operations were the frame-to-frame registration. Our primary method to compute the planar motion parameters for a sequence of images was a pyramidal correlation method. We initially chose a large search window (a 10% scale of the image resolution), but this takes up a long

time especially with high-resolution imagery. While the imagery we tested on had a regular capture rate between 2Hz and 30Hz, the vehicle speed was not constant due to wind and path conditions. Therefore, we maintained a mean and standard deviation of the inter-frame translation, so that we could dynamically set the correlation search window, and we also maintained a mean and standard deviation of the match error using the sum of absolute difference error metric. With the error metric, we could determine if a match was valid, or if we had to expand the search window for a given frame. With this implementation, we were able to process 5 frames per second at a 50% resolution of the CLIF 2007 data ( $1336 \times 2008$  pixels).

Our implementation was a single thread program, running on a laptop with a dual core 2.66GHz Intel Core i7 CPU. The image matching step can be considerably improved by reimplementing the method in parallel to take advantage of multiple threads or a GPU (Graphical Processing Unit) architecture. This would allow the generation step to run in realtime.

The MVSV visualization program is not computationally costly when it is used to stereoscopically display the imagery. But, when using the interactive features of viewer tracking, performed with face tracking, the application uses up CPU resources. In our implementation, we used the Qt toolkit to graphically develop our interface and we used the OpenCV [5] implementation of Face Tracking using the Haar Classifier. On the same laptop machine, this algorithm ran in realtime, where it would take about .5 seconds to initially find a face, and it can then track the face at 20Hz+. The OpenCV implementation was not compiled with GPU acceleration, only with multi threading support.

On mobile platforms with power constraints, CPU's are constrained as well. Implementing the generation and visualization components of our work, should still be possible. In particular, mobile GPU's are both more powerful than CPU's and use up less power, and our algorithms, such as the Fast Layering mosaicking method is not computationally expensive, it is much more dependent on large amounts of memory.

## Memory Costs

When we construct and visualize large-scale scenes, the biggest cost is memory. Our methods generate multi-view panoramas of the entire scene, which get very large when the original imagery is of high resolution. Even with a high-end desktop or laptop that has 8GB or 16GB, it is not possible to allocate enough memory for all of the panoramas being used. And, camera resolutions continue to increase, so it is not feasible to simply keep adding more memory. Instead, we implemented a tiling approach for both generating and visualizing the multi-view panoramas. With this implementation, each panorama was sliced and indexed into tiles of size 1024 pixels. Each tile has both an  $X$  and  $Y$  index, starting with  $(0, 0)$  on the top left of each panorama, and each tile is saved as a separate image on the hard drive. This allows both the generation and visualization programs to only load into memory tiles that are being written to or are being displayed, and releasing tiles that are not in use. In this implementation, the read and write operations to disk are slower than the use of RAM (Random Access Memory). It is particularly slow for mechanical spinning disk drives, but not as slow when implemented on flash-based Solid State Drives (SSD). Memory use can be further improved by asynchronously writing to the disk drive on a separate thread than the one using the tiles. On a mobile device with limited memory, the same tiling approach can be applied using tiles of smaller size, depending on the availability of memory.

## Evaluation

Evaluating stereoscopic content is a new and unstandardized topic of research. Recently, researcher [8] have released a quantitative tool for the evaluation of stereoscopic content for good and comfortable depth viewing. The majority of evaluation research has focused on quantitative surveys of viewers to determine whether content produces a good depth experience or not.

For our multi-view panoramas, we can determine correctness based on the registration, if we know the ground truth. Another method, we will explore in future work, is to use

the multi-view and multi-modal panoramas for change detection and object tracking. In this way, we will know if the stereo panoramas produce a comparable result to traditional tracking methods.

## 7.4 Future Work

The focus of this thesis work has been panorama generation and stereoscopic visualization of large scale scenes. Previous work, including ours, has shown how we handle various types of constrained camera motion (linear and circular) so that we can construct panoramas, and our more recent work handles general camera motion. While we tackled many problems in this dissertation, there is additional work left for the future. The following are topics we will continue to pursue:

- *Extend the visualization application.* We leave further development of interactive features in our MVSV program for the future. There are many additional viewer based movements and gestures that can be implemented so that a viewer can interact with the data quicker than with a mouse and keyboard commands. In particular, we plan to apply image and depth scaling to the user tracking feature. The viewer tracking can also be improved with the use of a RGB-D sensor.
- *Mobile device implementation.* An area of large interest is the application of both the panorama generation and visualization applications on mobile platforms. For generation, mobile platforms will allow the user to more quickly generate panoramas, but there are many computing and particularly low memory constraints that such platforms have that must be dealt with in the implementation of the programs. For visualization, small autostereoscopic screens are available, but many currently have very low resolutions. As we saw in chapter 6, this can lead to a reduced depth resolution and to viewer discomfort. Therefore the panoramas must be adjusted to the mobile screen constraints.

- *Evaluate multi-view / multi-modal panoramas for Change Detection and Object Tracking.* With the multiple run and modality panorama alignment, we will be able to evaluate how change detection and tracking algorithms perform using stereo panoramas versus globally aligned image frames. Since the stereo panoramas use parallax, it is our opinion that many of the errors that occur due to parallax in these applications may be improved. The one drawback we foresee, is the slight distortion of independently moving objects in the stereo panoramas.

# Appendix A

## Data Sources

The following sections provide details about the datasets used in this dissertation.

### A.1 CLIF 2006

The Columbus Large Image Format (CLIF) 2006 Dataset was publicly released by the Air Force Research Laboratory. The data is available here: [https://www.sdms.afrl.af.mil/index.php?collection=CLIF\\_2006](https://www.sdms.afrl.af.mil/index.php?collection=CLIF_2006)

The data was collected in April 2006 and it captured multiple circular passes above Columbus, Ohio, specifically the Ohio State University area. A 6 camera configuration captured the scene.

#### A.1.1 Data details

- # of cameras: 6
- Type of camera: Monochromatic EO
- Image resolution:  $2672 \times 4016$
- Frame rate:  $\sim 2$  Hz
- # of Frames: 24564 (4095 images x 6 cameras) raw files

- Altitude:  $\sim$ 15,000 feet

### A.1.2 Issues

The data also captured IMU data, but it was not used in this dissertation. The IMU data was not accurate enough for registration. Also, this dataset contained corrupted frame data. Only one portion of the capture was used. During development, this data was only used at 25% and 50% resolution for performance reasons.

## A.2 CLIF 2007

The Columbus Large Image Format (CLIF) 2007 Dataset was also publicly released by the Air Force Research Laboratory. The data is available here: <https://www.sdms.afrl.af.mil/index.php?collect>

The data was collected in October 2007 and it captured multiple circular passes above Columbus, Ohio, specifically the Ohio State University area. A 6 camera configuration captured the scene.

### A.2.1 Data details

- # of cameras: 6
- Type of camera: Monochromatic EO
- Image resolution:  $2672 \times 4016$
- Frame rate:  $\sim$ 2 Hz
- # of Frames: 96,702 (16117 images x 6 cameras) raw files
- Altitude:  $\sim$ 15,000 feet

### A.2.2 Issues

The data also captured IMU data, but it was not used in this dissertation. The IMU data was not accurate enough for registration. The 2007 data did not have the errors that the 2006 data contained. We only used one camera of the data, and we used the data at 25%, 50%, and 100% scale resolution. The smaller scales were used to test the methods faster.

## A.3 CSUAV

The Columbus Surrogate Unmanned Aerial Vehicle (CSUAV) Dataset was publicly resealed by the Air Force Research Laboratory. The data is available here: <https://www.sdms.afrl.af.mil/index.php?>

The data was collected in October 2007 with two separate UAV systems flying straight flight paths back and forth multiple times, the two UAV's flew approximately perpendicular to each other. One UAV was capturing monochromatic EO imagery, and the other Mid Wave Infrared (MWIR) imagery.

### A.3.1 Data details

The monochromatic EO camera:

- # of cameras: 1
- Altitude: 6,500 ft.
- Sensor: Illunix XMV-11000
- Frame Size:  $4004 \times 2672$  (12bit resolution)
- FOV ( $H^\circ \times V^\circ$ ):  $23.9^\circ \times 16.1^\circ$
- Focal Length: 85mm
- Frame Rate:  $\sim 5Hz$

The MWIR camera:

- # of cameras: 1
- Altitude: 2,500 ft.
- Sensor: FLIR Systems ThermoVision SC6000
- Frame Size:  $650 \times 512$  (14bit resolution)
- FOV ( $H^\circ \times V^\circ$ ):  $18.2^\circ \times 14.6^\circ$
- Focal Length: 50mm
- Frame Rate: 30 Hz

# Appendix B

## Publications

### Journals

- E. Molina and Zhigang Zhu. Persistent Aerial Video Registration and Fast Multi-View Mosaicing. *Image Processing, IEEE Transactions on*, 23(5):2184–2192, May 2014
- Edgardo Molina, Alpha Diallo, and Zhigang Zhu. Visual noun navigation framework for the blind. *Journal of Assistive Technologies*, 7(2):118–130, 2013

### Book Chapters

- Zhigang Zhu and Edgardo Molina. Panostereo: Panoramic Stereoscopic Imaging. In Phil Laplante, editor, *Encyclopedia of Image Processing*. Taylor and Francis Group (invited and under review), 2014
- Z. Zhu, W. Li, E. Molina, and G. Wolberg. LDV Sensing and Processing for Remote Hearing in a Multimodal Surveillance System. In Z. Zhu and T. Huang, editors, *Multimodal Surveillance: Sensors, Algorithms and Systems*. Artech House Publisher, 2007

### Conferences

- Edgardo Molina, Zhigang Zhu, and Yingli Tian. Visual Nouns for Indoor/Outdoor Navigation. In *Proceedings of the 13th International Conference on Computers Helping People with Special Needs - Volume Part II*, ICCHP’12, pages 33–40, Berlin, Heidelberg, 2012. Springer-Verlag
- E. Molina, Zhigang Zhu, and C.N. Taylor. A Layered Approach for Fast Multi-view Stereo Panorama Generation. In *Multimedia (ISM), 2011 IEEE International Symposium on*, pages 589–594, Dec 2011

- Yufu Qu, W.L. Khoo, E. Molina, and Zhigang Zhu. Multimodal 3D panoramic imaging using a precise rotating platform. In *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on*, pages 260–265, July 2010
- Edgardo Molina, Zhigang Zhu, and Olga Mendoza-Schrock. Mosaic-based 3D scene representation and rendering of circular aerial video. In *SPIE Defense, Security, and Sensing*, pages 77040K–77040K. International Society for Optics and Photonics, 2010
- E. Molina, Hao Tang, Zhigang Zhu, and O. Mendoza. Mosaic-based Modeling and Rendering of Large-Scale Dynamic Scenes for Internet Applications. In *Aerospace and Electronics Conference, 2008. NAECON 2008. IEEE National*, pages 322–329, July 2008
- Zhigang Zhu, Weihong Li, E. Molina, and G. Wolberg. LDV Sensing and Processing for Remote Hearing in a Multimodal Surveillance System. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–2, June 2007

## Patents Pending and Provisional

- Zhigang Zhu, Tony Ro, Lei Ai, Wai Khoo, Edgardo Molina, and Frank Palmer. Wearable navigation assistance for the vision-impaired, December 27 2013. US Patent App. 14/141,742 (pending)

# Bibliography

- [1] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. *ACM Trans. Graph.*, 25(3):853–861, 2006.
- [2] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23:294–302, August 2004.
- [3] Manoj Aggarwal and Narendra Ahuja. High dynamic range panoramic imaging. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 2–9. IEEE, 2001.
- [4] Sebastiano Battiato, Alessandro Capra, Salvatore Curti, and Marco La Cascia. 3D stereoscopic image pairs by depth-map generation. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 124–131. IEEE, 2004.
- [5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [6] Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 2:217–236, October 1983.
- [7] Ricardo Cabral and Yasutaka Furukawa. Piecewise Planar and Compact Floorplan Reconstruction from Images. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 628–635. IEEE, 2014.
- [8] Jose Vinicius de Miranda Cardoso, Carlos Danilo Miranda Regis, and Marcelo Sampaio de Alencar. ImQET: Objective Stereoscopic Image Quality Evaluation Tool. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, WebMedia ’14, pages 25–30, New York, NY, USA, 2014. ACM.
- [9] Yaron Caspi and Michal Irani. Aligning non-overlapping sequences. *International Journal of Computer Vision*, 48(1):39–51, 2002.
- [10] Jin-Xiang Chai and Heung-Yeung Shum. Parallel projections for stereo reconstruction. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 493–500. IEEE, 2000.

- [11] Alexandre Chapiro, Simon Heinze, Tunç Ozan Aydin, Steven Poulakos, Matthias Zwicker, Aljosa Smolic, and Markus Gross. Optimizing Stereo-to-Multiview Conversion for Autostereoscopic Displays. In *Computer Graphics Forum*, volume 33,2, pages 63–72. Wiley Online Library, 2014.
- [12] Junguk Cho, Joon Hyuk Cha, Yong Min Tai, Young-Su Moon, and Shihwa Lee. Stereo Panoramic Image Stitching with a Single Camera. In *Consumer Electronics (ICCE), 2013 IEEE International Conference on*, pages 256–257, Jan 2013.
- [13] CSUAV. Columbus Surrogate Unmanned Aerial Vehicle (CSUAV) Dataset. <https://www.sdms.afrl.af.mil/index.php?collection=csuav>. Accessed: 2015-01-13.
- [14] J. Gao, S.J. Kim, and M Brown. Constructing Image Panoramas using Dual-Homography Warping. *CVPR*, 2011.
- [15] Joshua Gluckman, Shree K Nayar, and Keith J Thoresz. Real-time omnidirectional and panoramic stereo. In *Proc. of Image Understanding Workshop*, volume 1, pages 299–303. Citeseer, 1998.
- [16] Google. Google Street View. <https://www.google.com/maps/views/u/0/streetview?gl=us>. Accessed: 2015-01-13.
- [17] Benjamin Heiner and Clark N. Taylor. Creation of geo-referenced mosaics from MAV video and telemetry using constrained optimization bundle adjustment. In *IROS*, pages 5173–5178, 2009.
- [18] Ho-Chao Huang and Yi-Ping Hung. Panoramic stereo imaging system with automatic disparity warping and seaming. *Graphical Models and Image Processing*, 60(3):196–208, 1998.
- [19] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and Aanæs H. Large Scale Multi-view Stereopsis Evaluation. In *CVPR’14*, 2014.
- [20] Yekeun Jeong, D Nister, D Steedly, R Szeliski, and In-So Kweon. Pushing the Envelope of Modern Methods for Bundle Adjustment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.
- [21] Jiaya Jia and Chi-Keung Tang. Image Stitching Using Structure Deformation. *IEEE Trans. PAMI*, 30:617–631, 2008.
- [22] Graham R Jones, Delman Lee, Nicolas S Holliman, and David Ezra. Controlling perceived depth in stereoscopic images. In *Photonics West 2001-Electronic Imaging*, pages 42–53. International Society for Optics and Photonics, 2001.
- [23] Rakesh Kumar, Harpreet S Sawhney, Jane C Asmuth, Art Pope, and Steve Hsu. Registration of video to geo-referenced imagery. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 1393–1400. IEEE, 1998.

- [24] Yuping Lin, Qian Yu, and Gerard Medioni. Map-Enhanced UAV Image Sequence Registration. *WACV*, 0:15, 2007.
- [25] Paul May. A survey of 3-D display technologies. *Information Display*, 32:28–33, 2005.
- [26] Microsoft. Bing Street Side. <http://www.microsoft.com/maps/streetside.aspx>. Accessed: 2015-01-13.
- [27] E. Molina and Zhigang Zhu. Persistent Aerial Video Registration and Fast Multi-View Mosaicing. *Image Processing, IEEE Transactions on*, 23(5):2184–2192, May 2014.
- [28] Edgardo Molina, Zhigang Zhu, and Olga Mendoza-Schrock. Mosaic-based 3D scene representation and rendering of circular aerial video. In *SPIE Defense, Security, and Sensing*, pages 77040K–77040K. International Society for Optics and Photonics, 2010.
- [29] M Montemerlo, S Thrun, D Koller, and B Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *AAAI/IAAI*, pages 593–598, 2002.
- [30] Yuzhen Niu, Yujie Geng, Xueqing Li, and Feng Liu. Leveraging Stereopsis for Saliency Analysis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 454–461, June 2012.
- [31] NVIDIA. 3D Vision 2: Full HD Stereoscopic glasses for your PC. <http://www.nvidia.com/object/product-geforce-3d-vision2-wireless-glasses-us.html>. Accessed: 2014-06-25.
- [32] Clark F. Olson, Adnan I. Ansar, and Curtis W. Padgett. Robust Registration of Aerial Image Sequences. In *Proc. ISVC(2)*, ISVC ’09, pages 325–334, Berlin, Heidelberg, 2009. Springer-Verlag.
- [33] Taragay Oskiper, Han-Pang Chiu, Zhiwei Zhu, Supun Samarasekera, and Rakesh Kumar. Stable vision-aided navigation for large-area augmented reality. In *VR*, pages 63–70, 2011.
- [34] Taragay Oskiper, Zhiwei Zhu, Supun Samarasekera, and Rakesh Kumar. Visual Odometry System Using Multiple Stereo Cameras and Inertial Measurement Unit. In *CVPR*, 2007.
- [35] S. Peleg, M. Ben-Ezra, and Y. Pritch. Omnistereo: panoramic stereo imaging. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):279–290, Mar 2001.
- [36] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. *CVPR*, 0:338, 1997.
- [37] Yufu Qu, W.L. Khoo, E. Molina, and Zhigang Zhu. Multimodal 3D panoramic imaging using a precise rotating platform. In *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on*, pages 260–265, July 2010.

- [38] Stephan Reichelt, Ralf Haussler, Norbert Leister, Gerald Flatterer, Hagen Stolle, and Armin Schwerdtner. Holographic 3-D Displays-Electro-holography within the Grasp of Commercialization, Chapter 29, in Advances in Lasers and Electro Optics, edited by Nelson Costa and Adolfo Cartaxo. *InTech, ISBN*, pages 978–953, 2010.
- [39] Christian Richardt, Yael Pritch, Henning Zimmer, and Alexander Sorkine-Hornung. Megastereo: Constructing High-Resolution Stereo Panoramas. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1256–1263, June 2013.
- [40] B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha. Universal mosaicing using pipe projection. In *Computer Vision, 1998. Sixth International Conference on*, pages 945–950, Jan 1998.
- [41] D Scaramuzza and F Fraundorfer. Visual Odometry [Tutorial]. *Robotics & Automation Magazine, IEEE*, 2011.
- [42] Yoav Y Schechner and Shree K Nayar. Generalized mosaicing. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 17–24. IEEE, 2001.
- [43] Wang Shengchun, Luo Xiaoyue, Huang Yaping, and Yin Hui. Monocular stereo imaging using bi-slit projections from Forward motion video. In *Signal Processing (ICSP), 2014 12th International Conference on*, pages 1301–1305, Oct 2014.
- [44] Heung-Yeung Shum and Richard Szeliski. Stereo reconstruction from multiperspective panoramas. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 14–21. IEEE, 1999.
- [45] Neil G Smith, Steve Cutchin, Robert Kooima, Richard A Ainsworth, Daniel J Sandin, Jurgen Schulze, Andrew Prudhomme, Falko Kuester, Thomas E Levy, and Thomas A DeFanti. Cultural heritage omni-stereo panoramas for immersive cultural analyticsFrom the Nile to the Hijaz. In *Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium on*, pages 552–557. IEEE, 2013.
- [46] Hao Tang and Zhigang Zhu. Content-Based 3-D Mosaics for Representing Videos of Dynamic Urban Scenes. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(2):295–308, 2012.
- [47] Hakan Urey, Kishore V Chellappan, Erdem Erden, and Phil Surman. State of the art in stereoscopic and autostereoscopic displays. *Proceedings of the IEEE*, 99(4):540–555, 2011.
- [48] Oculus VR. Oculus Rift Development Kit 1 Order Page. <https://www.oculusvr.com/order/dk1/>. Accessed: 2014-06-25.
- [49] Shengchun Wang, Siwei Luo, Yaping Huang, JiangYu Zheng, Peng Dai, and Qiang Han. Railroad online: acquiring and visualizing route panoramas of rail scenes. *The Visual Computer*, 30(9):1045–1057, 2014.

- [50] Jason Wither, Carmen E. Au, Raymond Rischpater, and Radek Grzeszczuk. Moving Beyond the Map: Automated Landmark Based Pedestrian Guidance Using Street Level Panoramas. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 203–212, New York, NY, USA, 2013. ACM.
- [51] AJ Woods. Understanding crosstalk in stereoscopic displays. In *Keynote Presentation at the Three-Dimensional Systems and Applications Conference, Tokyo, Japan*, pages 19–21, 2010.
- [52] Andrew J Woods. How are crosstalk and ghosting defined in the stereoscopic literature? In *IS&T/SPIE Electronic Imaging*, pages 78630Z–78630Z. International Society for Optics and Photonics, 2011.
- [53] Andrew J Woods and Chris R Harris. Comparing levels of crosstalk with red/cyan, blue/yellow, and green/magenta anaglyph 3D glasses. *Proceedings of SPIE stereoscopic displays and applications XXI*, 7253:0Q1–0Q12, 2010.
- [54] Tao Yan, Zhe Huang, Rynson W. H. Lau, and Yun Xu. Seamless Stitching of Stereo Images for Generating Infinite Panoramas. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, VRST '13, pages 251–258, New York, NY, USA, 2013. ACM.
- [55] Cheng Zhang, Chunping Hou, Xiaoyan Wang, and Zhiyuan Wang. A setup for panoramic stereo imaging. *Multimedia Tools and Applications*, pages 1–18, 2015.
- [56] J.Y. Zheng. Digital route panoramas. *MultiMedia, IEEE*, 10(3):57–67, July 2003.
- [57] Z. Zhu, A.R. Hanson, and E.M. Riseman. Generalized parallel-perspective stereo mosaics from airborne video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):226–237, Feb 2004.
- [58] Zhigang Zhu, Allen R Hanson, Howard Schultz, and Edward M Riseman. Generation and Error Characterization of Pararell-Perspective Stereo Mosaics from Real Video. In *Video Registration*, pages 72–105. Springer, 2003.
- [59] Zhigang Zhu and Edgardo Molina. Panostereo: Panoramic Stereoscopic Imaging. In Phil Laplante, editor, *Encyclopedia of Image Processing*. Taylor and Francis Group (invited and under review), 2014.
- [60] Zhigang Zhu, Edward M Riseman, and Allen R Hanson. Parallel-perspective stereo mosaics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 345–352. IEEE, 2001.
- [61] Zhigang Zhu, Edward M. Riseman, Allen R. Hanson, and Howard J. Schultz. An efficient method for geo-referenced video mosaicing for environmental monitoring. *Mach. Vis. Appl.*, 16(4):203–216, 2005.

- [62] Zhiwei Zhu, Taragay Oskiper, Supun Samarasekera, Rakesh Kumar, and Harpreet S. Sawhney. Real-time global localization with a pre-built visual landmark database. In *CVPR*, 2008.