EXPLOITING ADVANTAGES OF NON-CENTRIC IMAGING FOR COMPUTER VISION

by

Wei Yang

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Fall 2017

EXPLOITING ADVANTAGES OF NON-CENTRIC IMAGING FOR COMPUTER VISION

by

Wei Yang

Approved: _____

Kathleen F. McCoy, Ph.D.
Chair of the Department of Computer and Information Science

Approved: _____

Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____

Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Jingyi Yu, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Chandra Kambhamettu, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Christopher Rasmussen, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Li Liao, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Gordon Wetzstein, Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Jingyi Yu for his support of my research and for his motivation, advise and immense knowledge. Dr. Yu has taught me how to keep focus on big picture, how to approach a problem and how to present my work. I have learned a lot more from him. And I would also like to thank Professor Chandra Kambhamettu, Professor Christopher Rasmussen, Professor Li Liao and Professor Gordon Wetzstein for serving as my dissertation committee members and providing insightful suggestions.

My sincere thanks also go to collaborators Yu Ji, Jinwei Ye and Haiting Lin. Thanks to Yu Ji for his help on construction of the directional encodable light source. Thanks to Jinwei Ye for her help on improving the language of my papers. Thanks to Haiting Lin for his useful inputs on the phase technique for image processing and light field stereo.

Many thanks to all members of the Graphic and Imaging Lab at the University of Delaware, Yang Yang, Mingyuan Zhou, Zhong Li, Zhan Yu, Xinqing Guo, Can Chen and Nianyi Li, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had together. I also would like to thank several students at ShanghaiTech University, Yingliang Zhang, Peihong Yu and etc., for their collaborations on the ICCP and ICCV papers.

I gratefully acknowledge the funding sources that made my Ph.D work possible. My work is partially supported by National Science Foundation under grants IIS-CAREER-0845268, IIS-1218156, IIS-1513031 and IIS-1422477. Part of my research(Chapter 6) was conducted when I was an intern at Plex-VR, which provided hardware(a dome system) that has made the research possible.

Finally it's my privilege to thank my parents and my bother. I would like to thank my parents for their support and sacrifice. They are always encouraging me to focus on my

education and tolerating my stubborn and irresponsibility. I would like to thank my brother. Without him, I wouldn't have a chance to study in the U.S and fulfill my dream today.

I would like to thank my wife Ling Qian, and my newborn son Vincent. Thanks to my wife for her sacrifice on career in order to accompany me. And thanks to my son Vincent, please be strong and grow healthy. I love you with all my heart.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Employing image features pertaining to scene geometry for reliable scene understanding and reconstruction is an important task in computer vision. The pinhole camera follows the perspective projection principle strictly, i.e project lines to lines and objects in the distance appear smaller than objects close by. Though being identical to the human vision, perspective images seem to lack effective features that can provide cues about the scene structure. In contrast, images captured by non-centric cameras are generally distorted (e.g.project lines to curves). The multi-perspective distortions produce some unique geometric features that will facilitate scene understanding tasks.

In this thesis, I comprehensively exploit the advantages of general non-centric cameras, the XSlit Camera in particular, in scene understanding context. In addition to vanishing point (VP), I first show that another geometric feature exists in non-centric cameras, called the coplanar common point (CCP). A CCP is a point in the image plane corresponding to the intersection of the projections of all lines lying on a common 3D plane. I explore the existence of CCP in general non-centric cameras and show its potential in scene recovery tasks. I show that CCP generally exists in non-centric cameras and derive the necessary and sufficient conditions for CCP to exist. Specifically, I conduct a comprehensive analysis from the perspective of ray-space and caustics and show how to determine the existence of CCP for a general non-centric camera. Experiments show that the CCP analysis provides useful insights on planar structure localization.

Another useful feature exhibited in non-centric images is the depth-dependent aspect ratio (DDAR): aspect ratio (AR) of an object in the image changes according to its depth to the camera. I first conduct a comprehensive analysis to characterize DDAR, infer object depth from its AR, and model recoverable depth range, sensitivity, and error. I show that repeated shape patterns in real Manhattan World scenes can be used for 3D reconstruction

using a single XSlit image. I also extend the analysis to model slopes of lines. Specifically, parallel 3D lines exhibit depth-dependent slopes (DDS) in image which can also be used to infer their depths. I validate the analyses using real XSlit cameras, XSlit panoramas, and catadioptric mirrors. Experiments show that DDAR and DDS provide important depth cues and enable effective single-image scene reconstruction.

Finally, I prove that structure-from-motion(SfM) via XSlit camera automatically avoid the scale ambiguity that plagues the perspective camera based solutions. I demonstrate that viewpoint transforms under XSlit camera can also be derived using the fundamental matrix analogous to the perspective case. To address non-linearity and mitigate depth-dependent distortions in XSlit images, I further develop a novel feature matching algorithm based on non-uniform Gaussian kernels. I also extend the bundle adjustment to XSlit images to refine the estimated camera poses. Experiments demonstrate that our XSlit-based SfM approach can reliably estimate camera motion and scene geometry while avoiding ambiguity.

Chapter 1

INTRODUCTION

The camera model describes how a camera collects light rays and converts them into pixels in image. The pinhole model serves as the workhorse for many tasks in computer vision as it is simple, effective and almost identical to the human vision. The relations between a 3D point and its pinhole projection can be characterized as a simple linear transformation, which can be described as a 3×4 homogeneous matrix. Though being very popular, the pinhole model still has many defects. For example, it's well known that pinhole cameras suffer from the scale ambiguity when applied for scene reconstruction. If we scale the entire scene by some factor k along with its distance to the pinhole camera, the image remains exactly the same. Furthermore, it's difficult for a pinhole camera to achieve very wide field of view (FoV). The angular resolution decreases rapidly as the pixel moves further away from the image center.

To overcome the defects, researchers have designed various more general cameras, such as the XSlit camera, catadioptric camera, multi-camera rig system and the light field camera and etc. These cameras do not obey the single viewpoint constraint, i.e.the collected rays no longer pass through a common point. They fall into the non-centric camera category. Non-centric cameras have advantages over the pinhole camera in some aspects. For example, the catadioptric cameras are able to acquire images with much wider Field of View (FoV) compared to perspective cameras. And a light field camera directly samples the 4D ray space and allows post-shot refocusing.

In this thesis, I give a comprehensive exploration to the advantages of general non-centric cameras. I present several unique features that only exist in non-centric cameras such as the Coplanar Common Points (CCP) and Depth Dependent Aspect Ratio (DDAR). I derive the necessary and sufficient conditions for these features to exist in general non-centric

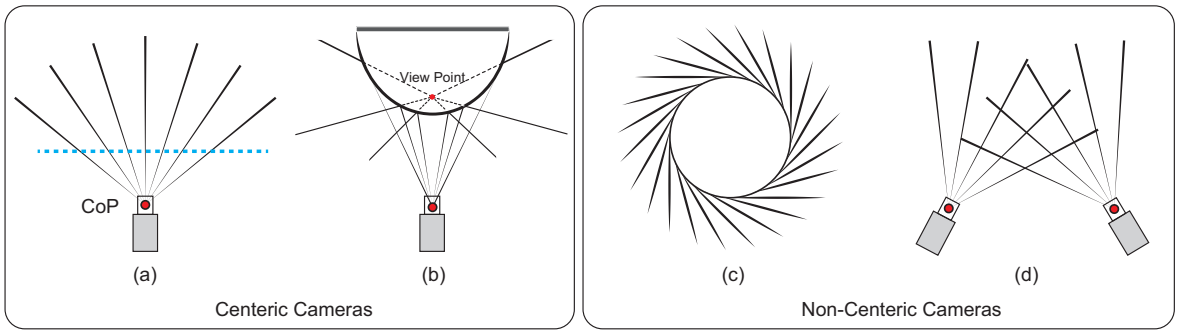


Figure 1.1: General cameras can be modeled as light ray bundles. (a) pinhole camera, (b) single view catadioptric camera. (c) con-centric mosaics, (d) stereo pair. (a)-(b) are central devices which collected light rays passing through a common point. (c)-(d) are non-central devices which don't follow the single view point rule.

cameras. I propose several key applications, including plane localization and Manhattan scene analysis, for demonstration. I further show how non-centric cameras can resolve the scale ambiguity when employed for Structure from Motion (SfM).

1.1 Non-centric Cameras

The pinhole imaging is a natural optical phenomenon, which can be effectively viewed as a light-proof box with a small hole on one side. Because of the rectilinear propagation of light, all collected rays will pass through the hole, which is also the point of view. We call imaging devices that follow the single view point rule as "centric cameras". The centric model is dominating in computer vision for its simplicity. However, the pinhole model is not the only valid camera model. A imaging model is valid as long as it satisfies the sampling, continuity and unique projection properties [85]. There are many valid and more general camera models.

General Linear Cameras: One type of non-centric cameras slightly relaxes the ray constraints of the pinhole model. For instance, a Pushbroom camera [27] collects rays along parallel planes from points swept along a linear trajectory. All rays of a XSlit camera [90] pass through two non-coplanar lines. And in oblique cameras [57] no two rays can intersect or be parallel. Yu and McMillan [85] introduced the General Linear Camera (GLC) model that unifies many previous multi-perspective cameras with a single framework. It provides

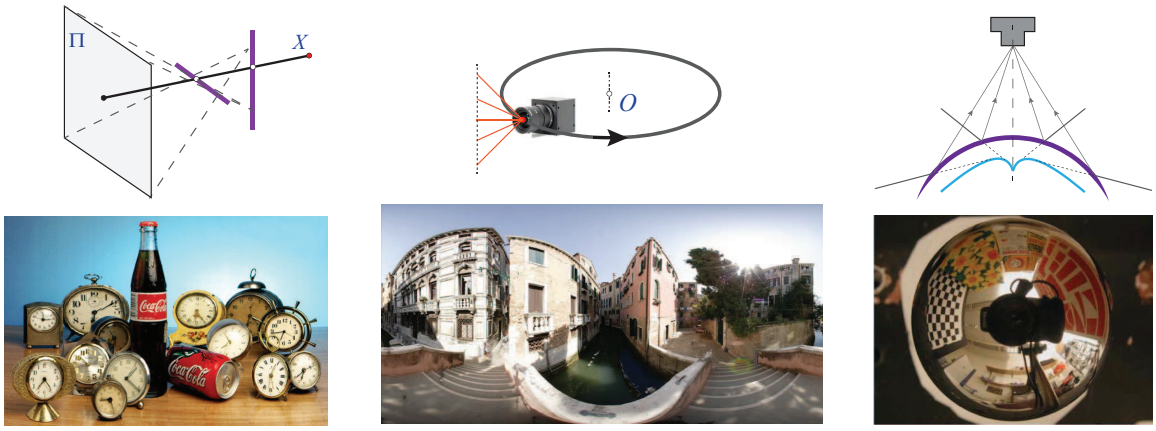


Figure 1.2: Non-centric images captured by real non-centric cameras or synthesized from light field. Courtesy of Tomas Pajdla and Shree Nayar.

an analytical representation of multi-perspective cameras as 2D affine subspaces embedded in the 4D light field space. The multi-perspective camera model is primitive and can be used to represent other more sophisticated cameras.

The GLC cameras are widely used to synthesize novel views and generate panoramas [63]. Centric cameras, such as the fisheye camera or the single viewpoint catadioptric camera, usually sacrifice the angular resolution to achieve wide FoV, which yields strong distortions around the image boundaries. Pushbroom and XSlit images exhibit relatively less distortions and are far more consistent over different image regions.

Non-centric Mosaics: The non-centric mosaics are synthesized from images captured at different viewing positions. Normally the camera motion is constrained to a path, e.g. a line or a circle. Novel views are rendered through composing images taken at different positions. The concentric mosaics [75, 71] and circular XSlit [89] are two typical examples of this category. They are both synthesized from rotational panoramas. To acquire a con-centric mosaics, it is common practice to rotate a camera off-axis on a circle. For each camera position, a column is sampled according to its angle to the optical axis. Then all selected columns are stitched together to form a panorama. The concentric mosaic allows users to move freely in a circular region and observe significant parallax and lighting changes without recovering the geometric and photometric scene models.

Catadioptric Systems: Another commonly used class of real non-centric cameras are catadioptric cameras [7, 74] in which refraction or reflection are integrated into an imaging system. Catadioptric combinations have been used in many early optical systems, such as lighthouse reflectors, microscope and telescopes. Nayar [52] proposed to place a orthographic camera in front of a paraboloid for acquiring images with much wider FoV. Bake [7] derived the entire class of catadioptric systems with a single effective viewpoint constructed from a single conventional camera and a single mirror. Such imaging systems require subtle configuration and engineering. Slightly deviating from the single viewpoint constraint yields more flexible and general non-centric cameras [74].

The catadioptric systems have the ability to achieve ultra wide FoV and cover vast scene region with a single camera. Due to this property, they are widely used in robotic vision and surveillance systems. Also, catadioptric images are able to store information of the surroundings very efficiently and hence are used for environment mapping in computer graphics.

Light Field Cameras: More advanced non-centric camera samples the complete 4D ray space. A simple way to achieve this is to move a camera on a plane and capture the images accordingly [33, 39]. The plenoptic function [4, 39, 46] is measured separately through groups of collected ray bundles. This method is easy to carry out but only applicable for static scenes. A more sophisticated way to acquire the light field is to build a 2D camera array [78]. However, building such camera array requires substantial amount of engineering and efforts. The recent development of light field camera is to put a mirco-lens array in front of a conventional sensor to sense the intensity, color and directional information [53]. This design also leads to the commercialization of the light field camera. The Lytro and Raytrix cameras can capture a few hundred of views of the scene at a single shot. From those views, one can conduct post-shot refocusing and extract depth information of the scene.

Above mentioned non-centric cameras are developed in different contexts and aimed at specific applications. For example, the Pushbroom camera and con-centric mosaics are designed to synthesize new views and generate panoramas, while the catadioptric cameras

are developed for acquiring images with ultra wide FoV. When applied for scene interpretation, the advantages of non-centric cameras are far less clear. This thesis considers the non-centric cameras in general by parameterizing them in ray space. With the mathematical basis, I present several geometric features that commonly exist in non-centric cameras and demonstrate that they are able to facilitate the scene understanding tasks.

1.2 Thesis Statement

The non-centric images have a lot advantages in analyzing scene geometry. Non-centric images are able to describe the particulars of a scene that inaccessible from a single view simultaneously. This property introduces several features, such as the Coplanar Common Points (CCP) and Depth Dependent Aspect Ratio (DDAR), that can help to understand the scene geometry. Structure from motion (SfM) via non-centric cameras can naturally resolve the scale ambiguity that plagued the perspective camera based solutions. These features convincingly demonstrate the advantages of non-centric cameras over single view cameras for scene understanding.

Coplanar Common Points: A CCP corresponds to the intersection of the curved projections of all lines lying on a common 3D plane in non-centric images. CCPs generally exist in a broad range of non-centric cameras such as the GLC and catadioptric cameras. The perspective camera is the single exception that does not have CCP. In contrast to the VP in perspective images, which is the characteristics of line directions, the CCP, in essence, is characteristics of positions. Detecting and identifying CCPs can facilitate 3D plane localization tasks, which is crucial to Manhattan scene reconstruction.

Depth Dependent Aspect Ratio: Unlike perspective camera that preserves aspect ratio (AR) under depth variations, aspect ratio changes monotonically with respect to depth in a XSlit camera and catadioptric camera. Similar to AR variations, the slopes of projected 3D lines will change with respect to depth. This property leads to new depth-from-AR scheme using a single XSlit image.

XSlit Structure from Motion: Similar to the perspective case, there exists a fundamental matrix to correlate two XSlit images captured at different viewpoints. The fundamental matrix can be reduced to 4×4 such that absolute translation and rotation matrices can be solved from a linear system. Hence SfM via XSlit camera can estimate the camera motion and scene geometry with an absolute scale.

1.3 Contributions

The thesis makes the following contributions:

• **CCP Existence Analysis:** I conduct a comprehensive analysis in ray space and show that finding the CCP of a 3D plane is equivalent to solving an array of ray constraint equations. I then derive the necessary and sufficient conditions for CCP to exist in general non-centric cameras such as concentric mosaics and catadioptric cameras. The analysis further reveals the relationship between the CCP and the caustic (focal) surfaces of rays. I further propose a simple but effective recipe for determining CCP existence using caustic analysis for catadioptric cameras.

• **CCP Detection and Application:** I develop robust algorithms for fitting curved images of 3D lines, locating the CCPs, and mapping them back to 3D planes. To detect the CCPs, I first adopt a point-aggregate strategy to find all segments of the line on the image plane. Then the CCPs are identified through circular constraints. Finial, I use the circular constraints to filter out false intersections and identify the CCPs. The analysis provides useful insights on scene structure for catadioptric cameras.

• **Depth Dependent Aspect Ratio and Slope Analysis:** I demonstrate that non-centric cameras such as the XSlit camera and catadioptric camera exhibit a different DDAR property that can help to resolve scale ambiguity. Then I conduct a comprehensive analysis to characterize DDAR, infer object depth from its AR, and model recoverable depth range, sensitivity, and error. I further show that repeated shape patterns in real Manhattan World scenes can be used for 3D reconstruction using a single XSlit image. At last, I extend the DDAR analysis to model the slopes of lines, which leads to a more effective depth inference scheme.

• **Depth Recovery From A Single XSlit Image:** I propose a simple but effective graph-cut based scheme to recover object depths from a single XSlit image and an effective formulation to model recoverable depth range, sensitivity and errors. In particular, I show how to exploit repeated shape patterns exhibiting in real Manhattan World scenes to conduct 3D reconstruction.

• **XSlit Structure from Motion:** I prove that structure-from-motion (SfM) via XSlit camera automatically avoid the scale ambiguity that plagues the perspective camera based solutions. I demonstrate that viewpoint transforms under XSlit camera can also be derived using the fundamental matrix analogous to the perspective case. I then further reduce the degree of freedom in the fundamental matrix by applying the XSlit constraints and solve the viewpoint transformation from a linear system. Finally, I extend the bundle adjustment to XSlit images to refine the estimated camera poses and scene geometry.

• **XSlit Feature Matching:** I develop a robust feature matching algorithm for XSlit images by applying multiple non-uniform Gaussian kernels to sample the affine SIFT feature space to mitigate XSlit distortions. My method can generate substantial amount of correspondences, and is more accurate than previous methods.

1.4 Thesis Overview

This thesis is organized as follows:

Chapter 2 discusses development of centric and non-centric cameras. It also reviews works in camera development and scene analysis. It discusses the advantages and limitations of both centric and non-centric cameras.

Chapter 3 models each ray using two parallel planes parametrization (2PP). Then it discusses the ray constraints for general cameras. The geometric properties of XSlit camera, as a special case, is further analyzed.

Chapter 4 presents the Coplanar Common Points (CCP) in general non-centric cameras. It discusses the necessary and sufficient conditions for CCP to exist in general non-centric cameras such as concentric mosaics and catadioptric cameras. Then several key applications, including plane localization, are proposed.

Chapter 5 first discusses scale ambiguity that enwinds scene interpretation tasks. Then it demonstrates that non-centric cameras such as the XSlit camera exhibit a different DDAR property that can help to resolve scale ambiguity. This chapter further shows how to employ the DDAR property for depth inference from a single XSlit image.

Chapter 6 proposes the XSlit structure from motion framework. It first introduces the XSlit fundamental matrix analogous to the perspective case under viewpoint transforms. We then further reduce the degree of freedom in the fundamental matrix by applying the XSlit constraints and solve the viewpoint transformation from a linear system. Finally, the bundle adjustment is extended to XSlit camera to refine the estimated camera poses and scene geometry.

Chapter 7 concludes the thesis and discusses the future works. Some open questions that remain are also included.

Chapter 2

BACKGROUND AND PREVIOUS WORK

This chapter discusses the background and previous work on both centric and non-centric cameras. I first review the history, physical structure, imaging process and limitations of the pinhole camera model. Then I discuss the development of various non-centric cameras and their applications in computer vision. Finally I compare the pinhole camera model and non-centric camera models in scene interpretation context.

2.1 Cameras

The first descriptions of pinhole images appeared in Mozi writings and the Aristotelian Problems. Arab polymath Ibn al-Haytham discovered that light rays travel in straight lines and built the first camera obscura through digging a hole on a wall in 1040AD. The French inventor Niepce (1765-1833) developed heliography, and created the world's oldest image using a lens based camera obscura. Following the predecessors' footprint, researchers and engineers have designed and constructed numerous cameras, including DSLR, mobile phone cameras, surveillance cameras and etc., for specific purposes.

The development of non-centric cameras is much earlier than modern digital cameras. French photographer Ducos du Hauron (1837-1920) designed and constructed a crossed-slit anamorphoser by replacing the pinhole of a camera obscure with two slits spaced apart along the camera axis. The crossed-slit anamorphoser is a valid camera: for every scene point, it determines two planes with the two slits, the intersection line of the two planes corresponds to a valid light ray that will be captured by the camera. Apparently, the crossed-slit anamorphoser falls into the non-centric camera category.

The crossed-slit anamorphoser inspired the exploring of more general camera models. The Pushbroom scanners are regularly used for passive remote sensing from space. Gupta

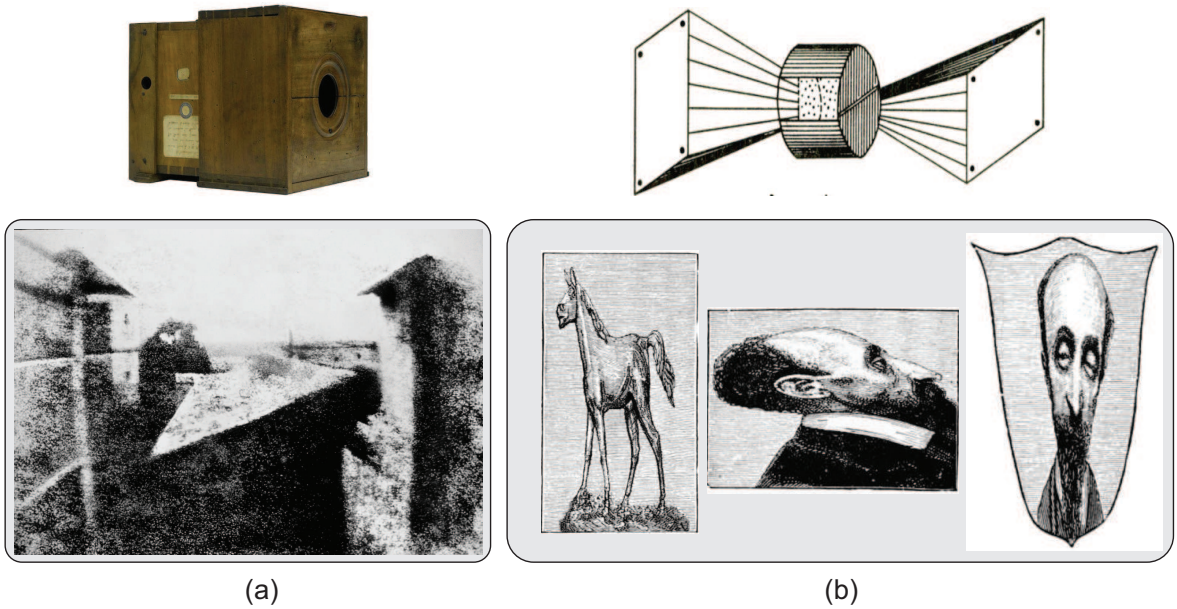


Figure 2.1: (a). Niepce's *View from the Window*, the earliest surviving photograph of a real world scene, made using a camera obscura(top left); (b) Huaron's crossed-slit anamorphoser(top right) and the anamorphic images.

and Hartley [57] introduced the simplified Pushbroom model that ignored the nonlinearity of the mathematical model involving orbital dynamics. Zomet et al. [90] generated XSlit images by stitching linearly varying columns across a sequence of perspective images and derived a close-form projection model for the XSlit camera. Yu et al. [86] used GLC to model a broad class of multi-perspective cameras, including Pushbroom, XSlit, Oblique camera and etc.

Instead of moving camera along a line to generate new views, Shum [71] moved a perspective camera around planar concentric circles, and create concentric mosaics by composing slit images taken at different locations. Under the same setup, Zomet et al. [90] proposed to extract the columns that pass through a common slit. This yielded a circular XSlit image. The concentric mosaics provide a much richer user experience by allowing the user to move freely in a circular region and observe significant parallax and lighting changes.

Another commonly used class of non-centric cameras are catadioptric camera systems, in which a regular pinhole camera is put in front a curved mirror. Nayar and Baker [8, 7] proposed all possible profiles of the quadric mirrors that can be used to construct

single viewpoint catadioptric camera system. And all projection rays after projection passing through CoP would have passed through virtual viewpoint before reflected by the mirror surface. If the perspective camera in catadioptric camera system slightly deviates from its desired position, we have a non-centric catadioptric camera. Swaminathan et al. [74] used the Jocobi matrix to derive the caustic surface from envelop of the reflection rays. As we mentioned before, the primitive multi-perspective camera models can also be used to parameterize non-centric catadioptric cameras. Yu and McMillan [87] viewed the mirror surface as piecewise triangle patches and represent each reflection patch as a GLC. The catadioptric camera system has a larger FoV compared to the pinhole camera, and can benefit many applications such as video surveillance, autonomous navigation and obstacle avoidance.

Recent progress in non-centric camera is the light field camera which samples the 4D light field directly. Isaksen, McLevoy and et al. [33, 39] moved a camera on a plane and captured the images accordingly. This method is easy to carry out, but it is only applicable for static scenes. Wilburn et al. [78] built a 2D camera array to capture the light field. However, building such camera array requires substantial amount of engineering and efforts. Recent development of light field camera is to put a micro-lens array in front of a conventional sensor to sense the intensity, color and directional information [53]. This design also leads to the commercialization of the light field camera. The Lytro and Raytrix cameras can capture a few hundred of views of the scene at a single shot. From those views, one can conduct post-shot refocusing and extract depth information of the scene.

Non-centric cameras are developed in different contexts and aimed at specific applications. For example, the Pushbroom camera and con-centric mosaics are designed to synthesize new views and generate panoramas, while the catadioptric cameras are developed for acquiring images with ultra wide FoV. The light field images allow post shot refocusing.

2.2 Non-centric Camera Applications

Non-centric cameras are widely used in graphics and vision.

Figure 2.2: Generated panorama using Pushbroom camera model

2.2.1 Panorama

One application of the non-centric cameras is to generate panoramas. Centric cameras, such as the fisheye camera or the single viewpoint catadioptric camera, usually sacrifice the angular resolution to achieve wide FoV, which yields strong distortions around the image boundaries. While non-centric images, such as Pushbroom and XSlit images, exhibit relatively less distortions and are far more consistent over different image regions. The concentric mosaics are also very effective panorama generation method. It provides a much richer user experience by allowing the user to move freely in a circular region and observe significant parallax and lighting changes.

2.2.2 Non-photorealistic Rendering

Non-centric cameras change the viewpoint across the imaging plane, it is possible to illustrate more details of the scene than that could be seen from a single point of view. Fig. 2.3 top compares one of Picassos and M.C Escher's paintings with images synthesized using the GLC framework [86]. With subtle adjustment of viewing directions across different regions, we can use multi-perspective rendering to create faux animations. Zomet et al. [90] used a single XSlit camera synthesize approach to achieve out-open effects. Mei et al. [47] introduced an occlusion camera that samples both the visible surface and hidden surface in the reference view.

2.2.3 Scene Reconstruction

Similar to perspective cameras, stereoscopic parallax also exists in non-centric geometry. Seitz [70] and Pajdla [58] proposed all possible conditions for a non-centric camera

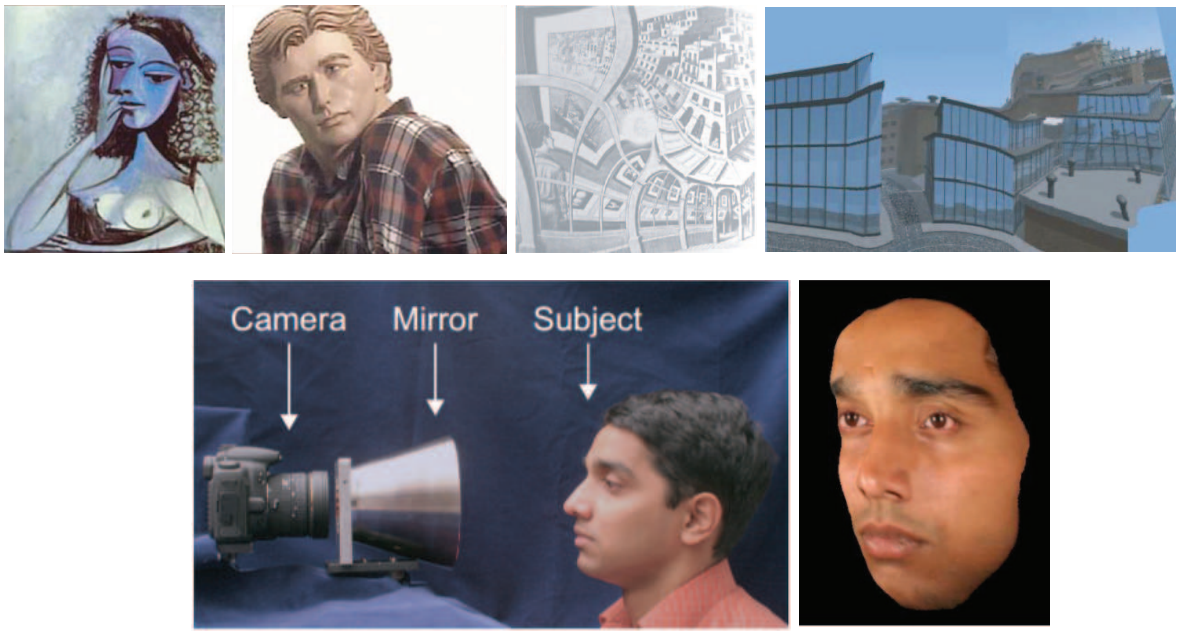


Figure 2.3: Non-centric camera applications. Top: Non-photorealistic rendering, Bottom: 3D reconstruction using radial catadioptric camera.

to have epipolar geometry: that is the epipolar surfaces should be planes, hyperboloids, or hyperbolic-paraboloids, which are also doubly ruled surfaces. In the same way with generating concentric mosaics, Peleg et al. [59] selected the center column and generate a circular Pushbroom, and then created a stereo panorama. A pair of XSlit cameras can have valid epipolar geometry, the necessary condition is that the two XSlit camera share one slit or their slits intersect in four distinct points [21].

Some non-centric cameras do not satisfy the epipolar geometry constraint, Ding and Yu [19] proposed the epsilon stereo pairs which allow a slight vertical parallax. With this model, they then introduced a warping method to minimize stereo inconsistencies. Another method to generate near stereoscopic is through cutting the light field [34]. Kim et al. [34] generated the stereoscopic view from light field through piecewise continuous cuts, minimizing an energy reflecting prescribed parameters, maximum disparity gradient and desired stereoscopic baseline and etc.

2.3 Scene Understanding

One of the most important applications of cameras is scene understanding. According to the number of cameras employed, the techniques to analyze the scene structure can be very different. Scene understanding through a single image usually is a ill-posed problem and requires certain priors, such as planar scene structure or camera is up-right and et al. When two cameras are involved, one can analyze the epipolar geometry and use stereo matching method to estimate the depthmap. If we have more than two camera, the scene understanding task becomes a SfM problem. Accordingly, feature matching, camera motion estimation and bundle adjustment need to be analyzed in order to recover the scene structure.

2.3.1 Perspective Cameras

There are some nice properties of perspective projections that suitable for scene understanding tasks. For example, the images of lines are still lines and parallel lines converging at a VP. With this property, the artists of Renaissance developed the famous linear perspective technique for painting in 1400s. And today in computer vision community, VP is widely used in the field of camera calibration [42], scene understanding [32, 17, 15] and etc.

A major task of computer vision is to infer 3D geometry of scenes using as fewer images as possible. Tremendous efforts have focused on recovering a special class of scene called the Manhattan World (MW) [14]. MW is composed of repeated planar surfaces and parallel lines aligned with three mutually orthogonal principal axes and fits well to many man-made (interior/exterior) environments. Under the MW assumption, one can simultaneously conduct 3D scene reconstruction [18, 24] and camera calibration [69]. MW generally exhibits repeated line patterns but lacks textures and therefore traditional stereo matching is less suitable for reconstruction. Instead, prior-based modeling is more widely adopted. For example, Furukawa et al.[24] assign a plane to each pixel and then apply graph-cut on discretized plane parameters. Other monocular cues such as the vanishing points [15] and the reference planes (e.g.the ground) have also been used to better approximate scene geometry. Hoime et al.[32, 31] use image attributes (color, edge orientation, etc.) to label image

regions with different geometric classes (sky, ground, and vertical) and then "pop-up" the vertical regions to generate visually pleasing 3D reconstructions. Similar approaches have been used to handle indoor scenes [18]. Machine learning techniques have also been used to infer depths from image features and the location and orientation of planar regions [64, 65]. Lee et al.[38] and Flint et al.[23] search for the most feasible combination of line segments for indoor MW understanding.

When multiply views are allowed, we are in the field of photogrammetry. The stereo method involves two perspective cameras and by comparing information about a scene from two vantage points, 3D information can be extracted by examination of the relative positions of objects in the two panels. The stereo matching methods can be divided into local algorithms and global ones. The local methods compare a block of pixels around each pixel between the stereo images [67]. Global methods compute the disparity map through minimizing a global energy functional [35]. [30] suggested a efficient semi-global matching approach which compare candidate blocks on the epipolar line along with dynamic programming. With more than two cameras, we have a Structure from Motion problem. SfM has been a well-studied problem in computer vision and great success has been achieved in robotics [16], autonomous navigation [50], large-scale 3D reconstruction [5, 72] etc. The very early root of SfM can be traced back to 1980s, where Higgins introduced a relative orientation estimation technique. After decade of development, it evolves into the current iterations of algorithms [25]. Modern SfM has shown great success in obtaining extremely realistic models. With immerse computational powers, SfM can now be used to recover very large scale 3D models, e.g., from community photo collections shared on the internet.

A typical SfM pipeline now includes feature detection and matching, camera pose estimation, triangulation and bundle adjustment. Reliable feature matching is crucial for the success of SfM. SIFT feature [43] has been proved to be very reliable. However it's unsatisfiable to match images with very large view change. Several affine invariant detectors are designed, e.g. Harris detectors and Hessian points. J. M. Morel [84] proposed ASift, which follows affine transformation parameters to correct images, performs best in a comparative

study of SIFT variants [79]. But we observe that ASIFT generally loses the subpixel accuracy of SIFT, which may introduce error into camera pose estimation. With corresponded feature points across views, one can use the classical 8 point algorithm to estimate the camera poses. Finally, the camera poses are optimized through minimize the retrojection error of matched feature points. The absolute scale estimation remains to be a challenging problem for SfM tasks. Standard approach for scale estimation is to use a stereo camera setup with known baseline [54, 16] where the scale factor is determined by triangulating feature points in the stereo pair. Clipp et al.[13] recover scale by tracking features on two non-overlapping cameras. For the single perspective camera case, prior knowledge on the camera motion or the scene has been used to recover the scale factor. Scaramuzza et al.[66] use the camera-to-ground distance to keep track of the camera motion in order to estimate the scale. Davison et al.[16] use a pattern of known size to compute the absolute scale of the entire scene. Pollefeys et al.[61] adopt an additional GPS sensor to acquire exact dimension. Scaramuzza et al.[66] use the nonholonomic constraints to estimate scale factor for cameras mounted on a moving vehicle.

2.3.2 Non-centric Cameras

Unlike the perspective cameras, it's possible to extract the scene information from a single non-centric image. Based on the singular value decomposition(SVD) of the Plücker coordinates of four points on a line, Lanman et al. [37] directly locate the line from a single non-centric catadioptric image. Ye et al.[82] used line curvatures in XSlit images for Manhattan scene reconstruction and developed a rotational stereo based on XSlit camera and defined the corresponding disparity for depthmap estimation.

The problem of SfM for a general camera has attracted much attention in the past decades. Pless [60] represented the generalized camera as a set of raxel and gave the epipolar constraints under Plücker coordinate. In this framework, Sturm [73] unified the multi-view geometry for general camera models. Li [40] analyzed the degenerated cases of the GEC and proposed a linear algorithm as solution. Cardoso [12] discussed the problem of finding the closest generalized essential matrix from a given 6×6 matrix. Other methods such as

[55, 76] estimated the relative motions of the multi-camera rig without using the GEC. The projection geometry of an XSlit camera has been widely studied in scene understanding. Seitz and Kim [70] and Pajdla [21] independently classified all possible stereo pairs in terms of their epipolar geometry. Sturm [70] analyzed the multi-view geometry in general non-centric camera.

Chapter 3

XSLIT CAMERA GEOMETRY

In this chapter, I first discuss how to parameterize a ray in space through two parallel planes parameterization (2PP). Then I introduce the ray space and represent a general camera as a 2D manifold of light rays. The camera model essentially is the constraints imposed on the manifold of rays. Finally I discuss geometric properties of non-centric cameras, especially the XSlit camera.

3.1 Ray Representation

Light essentially is electromagnetic radiation within a certain portion of the electromagnetic spectrum. A camera captures the radiation along each collected ray. Generally, the radiance will weaken as it travels further in space due to the absorbtion effect of transmission median. To simplify the analysis, most researchers do not take the fade of rays into count and hence light rays can be equivalently represented as lines in space. In this thesis, we also adopt this assumption for its simplicity. It's crucial for many applications to have an efficient and precise representation of the rays, and there are several mendelizing methods.

3.1.1 Plenoptic Function

The most straight forward way to represent a ray is to use a point on the ray, ray's direction and its radiation. Hence we can model the light ray as function $L(x, y, z, \theta, \phi)$ that describes the amount of radiation flowing in the light direction $[\theta, \phi]$ start from the point $[x, y, z]$ in space. $L(x, y, z, \theta, \phi)$ is the ideal 5D plenoptic function [4] which can express the image of a scene from all possible viewing positions at all possible viewing angle. Similarly, Gershun defined the light field as an infinite number of vectors for each point with lengthes representing the radiances.

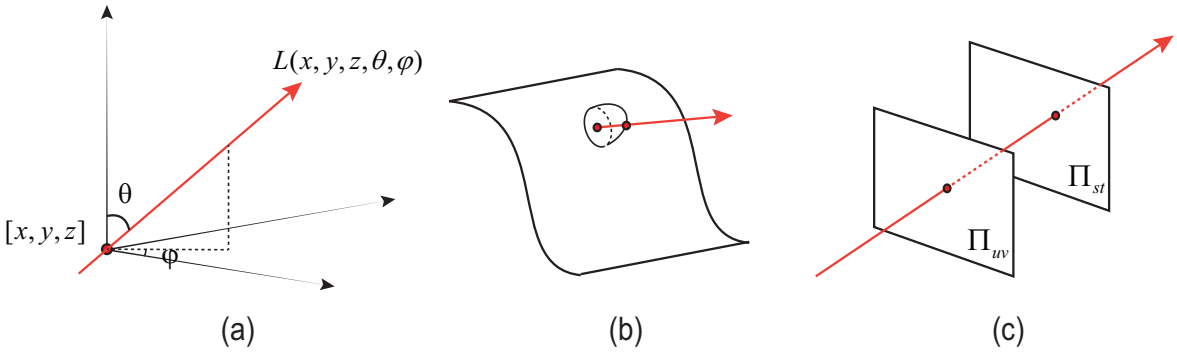


Figure 3.1: (a) the 5D plenoptic function. (b)(c) two example ways to parameterize the 4D light field

If we consider the radiance along a ray remains constant along its transmission path, we have a four-dimensional function which called the light field [39] or Lumigraph [26]. The set of rays in a light field can be parameterized in a many ways. For example, we can model each ray as its intersection point on surface and its emit direction, as shown in Fig. 3.1(b). Another way is using a pair of points on the surface of a sphere to represent the ray. The most common is the two parallel planes parameterization (2PP) shown at Fig. 3.1(c).

Ray parameterizations methods is important in computer vision as it serves as the analytic basis for many problems. For example, novel views can be generated by extracting 2D subset from the 4D light field of a scene [39, 26]. Orthographic, XSlit [90], General Linear Cameras [86], perspective or another type of projection can be created by appropriate parameterization of the light field. We can further synthesize the aperture and focus of the view by integrating an appropriate 4D subset of the samples in a light field captured by a light field camera [53].

3.1.2 Two Parallel Planes Parameterization

The most common approach to represent the 4D light field is 2PP. In 2PP, each ray is parameterized by its intersections with two parallel plane $[u, v, s, t]$ where $[s, t]$ is the intersection with the first plane Π_{st} and $[u, v]$ the second Π_{uv}. Though the 2PP cannot represent rays parallel to the two planes, it has some nice properties. A linear combination of the

$[u, v, s, t]$ coordinate of any two rays is still a valid representation of some ray. In some articles, the 4 dimensional vector $[u, v, s, t]$ is further written as $[u, v, \sigma, \tau]$, where $\sigma = s - u$ and $\tau = t - v$. In this thesis, we use 2PP to represent general non-centric cameras, derive the ray constraints and explore their advantages for scene understanding task.

3.1.3 Plucker Coordinates

Plücker coordinates, introduced by Julius Plücker in the 19th century, represent a line with its direction and moment, which lead to a six dimension homogenous vector. A line L in 3-dimensional Euclidean space is determined by two distinct points on the line: $\mathbf{x} = [x_1, x_2, x_3]$ and $\mathbf{y} = [y_1, y_2, y_3]$. The displacement vector from $\mathbf{x}$ to $\mathbf{y}$ is $\mathbf{d} = \mathbf{y} - \mathbf{x}$. $\mathbf{d}$ represents the direction of the line. The moment of the line is $\mathbf{m} = \mathbf{x} \times \mathbf{y}$, where $\times$ denotes the vector cross product. L can be uniquely determined by $\mathbf{d}$ and $\mathbf{m}$, the Plücker coordinate of L is:

$$[\mathbf{d} : \mathbf{m}] = [d_1, d_2, d_3, m_1, m_2, m_3] \tag{3.1}$$

Compared to 2PP parameterization, the Plücker coordinates have the ability to represent all possible rays. In contrast, rays that parallel to the uv and st planes can not be parameterized by 2PP. Furthermore, Plücker coordinates can represent the line geometry concisely in 3-dimensional space, especially for those involving incidence. For example, the constraint for two rays L and L' intersection in Plücker coordinate is $\mathbf{d}^\mathsf{T} \cdot \mathbf{m}' + \mathbf{m}^\mathsf{T} \cdot \mathbf{d}' = 0$. While the line intersection constraint in 2PP is a bilinear equation, which involves nonlinearity. The major drawback of Plücker coordinate is that they don't form a vector space. The linear combination of the Plücker coordinates of two rays is not guaranteed to be a valid ray parametrization.

3.2 Camera as Ray Space

A general camera, centric or non-centric, correspond to a 2D manifold of rays in ray space. Hence, a general imaging process entails the mapping of 3D geometry onto a 2D

manifold of rays, i.e., each pixel $[i, j]$ maps to a ray $[u, v, s, t]$ in 3D space. We can represent the camera as 2D ray manifold $\mathbf{\Sigma}$:

$$\mathbf{\Sigma}(i, j) = [u(i, j), v(i, j), s(i, j), t(i, j)] \tag{3.2}$$

A camera model imposes certain constraints on $\mathbf{\Sigma}$ to define which set of rays are collected. Consider the pinhole model as a study case, for each ray $r = [u, v, s, t]$ collected by a pinhole camera with CoP $[o_x, o_y, o_z]$, there exist some λ that:

$$\lambda \cdot [u, v, 0] + (1 - \lambda) \cdot [s, t, 1] = [o_x, o_y, o_z] \tag{3.3}$$

Eliminating λ we have: $\begin{bmatrix} 1 - o_z & 0 & o_z & 0 \\ 0 & 1 - o_z & 0 & o_z \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ s \\ t \end{bmatrix} = \begin{bmatrix} o_x \\ o_y \end{bmatrix}$. The equations is the constraints on the rays that collected by the pinhole camera.

Sophisticated general cameras have more complicate ray collection behavior, and hence will yield more complicate constraints. The Pushbroom and XSlit cameras collect rays that pass through one or two common slits. This yields linear constraints on ray space. Non-centric catadioptric cameras collection rays through reflections of mirror with quadric surface, the ray constraints are high ordered non-linear equations.

To analyze the ray geometry in this case, one can choose space varying parallel planes to model the light rays. Specifically, we can choose the local tangent planes defined by the derivative of $[u, v, s, t]$, i.e.planes with span $[u_x, v_x, s_x, t_x]$ and $[u_y, v_y, s_y, t_y]$, where subscription means the partial derivative. In this way, the local ray constraints become linear.

3.3 XSlit Geometry Analysis

In this section, I derive the constraints for a special type of non-centric camera, the XSlit camera, in ray space.

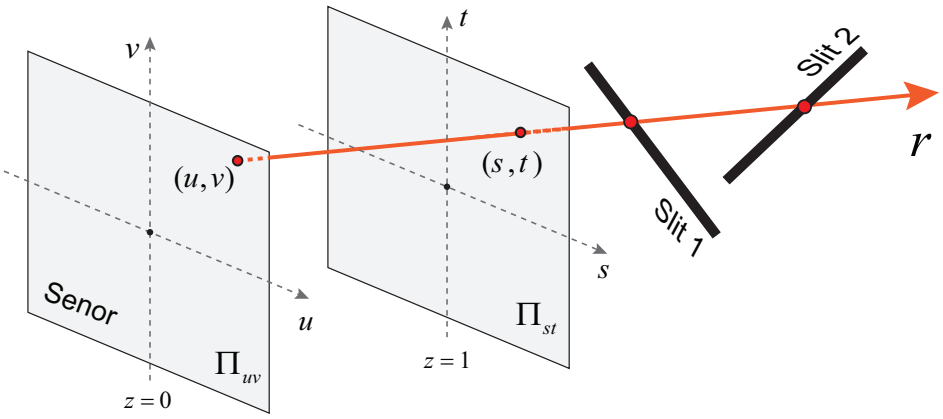


Figure 3.2: XSlit camera geometry: rays collected by the camera should simultaneously pass through two slits at different depths.

3.3.1 Ray Constraints

Geometrically, an XSlit camera collects rays that simultaneously pass through two oblique slits (neither coplanar nor parallel) in 3D space. For simplicity, we assume the sensor plane is parallel to the two slits' planes and use it as the $x-y$ plane. Furthermore, we use the intersection of the two slits' orthogonal projections on the sensor plane as the origin of the coordinate system. As shown in Fig. 3.3, the two slits lie on depth z_1 and z_2 and form angle θ_1 and θ_2 w.r.t the x-axis, where $z_2 > z_1$ and $\theta_1 \neq \theta_2$. This configuration is consistent with the real XSlit construction [82] and the XSlit panoramas [90].

To conduct ray geometry analyze on XSlit, we adopt the Two-Plane Parametrization (2PP) that represents a ray by its intersections with two parallel planes Π_{uv} and Π_{st}. To simply our analysis, we choose the sensor plane ($z = 0$) to be Π_{uv} and the plane at unit distance ($z = 1$) to be Π_{st}. If a ray intersects the two planes at [u,v,0] and [s,t,1], the ray direction can be represented as $[\sigma, \tau, 1] = [s-u, t-v, 1]$. We then uniquely represent each 3D ray using a four-tuple $[u, v, \sigma, \tau]$. Under this representation, the XSlit camera geometry can be formulated as two linear constraints on the ray coordinates:

$$\sigma = (\mathsf{A}u + \mathsf{B}v)/\mathsf{E}, \quad \tau = (\mathsf{C}u + \mathsf{D}v)/\mathsf{E} \tag{3.4}$$

where

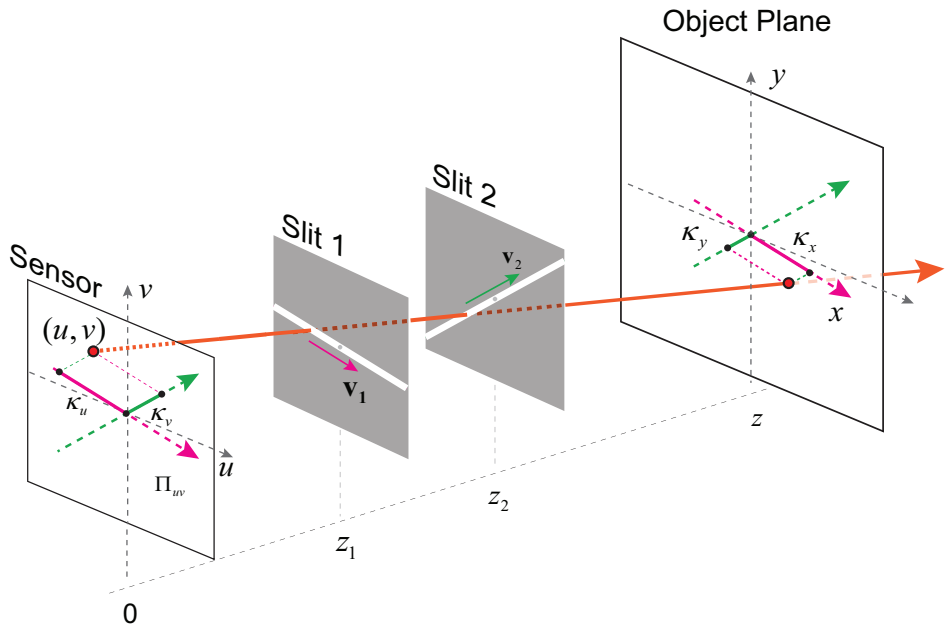


Figure 3.3: XSlit projection can be described as the combination of two projection components along the slits directions. Each individual component can be viewed as pinhole projection as they are parallel to either slits.

$$\mathsf{A} = z_2 \cos\theta_2 \sin\theta_1 - z_1 \cos\theta_1 \sin\theta_2, \qquad \mathsf{B} = (z_1 - z_2) \cos\theta_1 \cos\theta_2$$

$$\mathsf{C} = (z_1 - z_2) \sin\theta_1 \sin\theta_2, \qquad \mathsf{D} = z_1 \cos\theta_2 \sin\theta_1 - z_2 \cos\theta_1 \sin\theta_2$$

$$\mathsf{E} = z_1 z_2 \sin(\theta_2 - \theta_1).$$

We call Eqn. 3.4 the XSlit constraints. Previous studies reached similar conclusions in various forms [86, 90, 62].

3.3.2 A Geometric Perspective

Another way to analyze the XSlit camera is from the geometric perspective. We can decompose the XSlit camera into two pinhole projections along the two slits directions. Specifically, we project along the two slits directions individually and then combine the components after projection as the final result.

Under the XSlit setup, the z components along the two slits are 0. And the x-y directions are $\mathbf{v_1}[\cos\theta_1, \sin\theta_1]$ and $\mathbf{v_2}[\cos\theta_2, \sin\theta_2]$ that spans $\mathbf{R}^2$ space.

Previous approaches study projection using XSlit projection matrix [90], light field parametrization[86], and linear oblique[58]. We introduce a simpler projection model analogous to pinhole projection. Consider a 3D point p to p'. The process can be described as follows: first decompose the x-y components of p into two basis vectors, $\mathbf{v_1}$, $\mathbf{v_2}$ and write it as $[\kappa_x, \kappa_y, z]$. Next project individual component to $[\kappa_u, \kappa_v]$. Each component can be viewed as pinhole projection as they are parallel to either slits. Finally obtain the mapping from p to p'.

We first represent p on the basis of $\mathbf{v_1}$ and $\mathbf{v_2}$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \kappa_x \mathbf{v_1} + \kappa_y \mathbf{v_2}$$

We then project $\kappa_x \mathbf{v_1}$ and $\kappa_y \mathbf{v_2}$ independently. Notice the two components are at depth z. And $\kappa_x \mathbf{v_1}$ is parallel to slit 1 and $\kappa_y \mathbf{v_2}$ is parallel to slit 2. Their projections imitate the pinhole projection except that the focal lengths are different:

$$\kappa_u = -\frac{z_2}{z - z_2}\kappa_x, \kappa_v = -\frac{z_1}{z - z_1}\kappa_y \tag{3.5}$$

Notice the XSlit mapping is linear, we can combine κ_u and κ_u to compute p'.

$$p' = \kappa_u \mathbf{v_1} + \kappa_v \mathbf{v_2}$$

κ_u and κ_v are also the linear representations of p' on basis of $\mathbf{v_1}$ and $\mathbf{v_2}$.

3.4 XSlit Camera Construction

For the former, we use an XSlit lens [82] to construct a real XSlit camera. The design resembles the original anamorphoser proposed by Ducos du Hauron that replaces the pinhole in the camera with a pair of narrow, perpendicularly crossed slits. Similar to the way of using a spherical thin lens to increase light throughput in a pinhole camera, the XSlit lens relay perpendicular cylindrical lenses, one for each slit. In this thesis, we use two cylindrical lenses with focal lengths 2.5cm (closer to the sensor) and 7.5cm (farther away

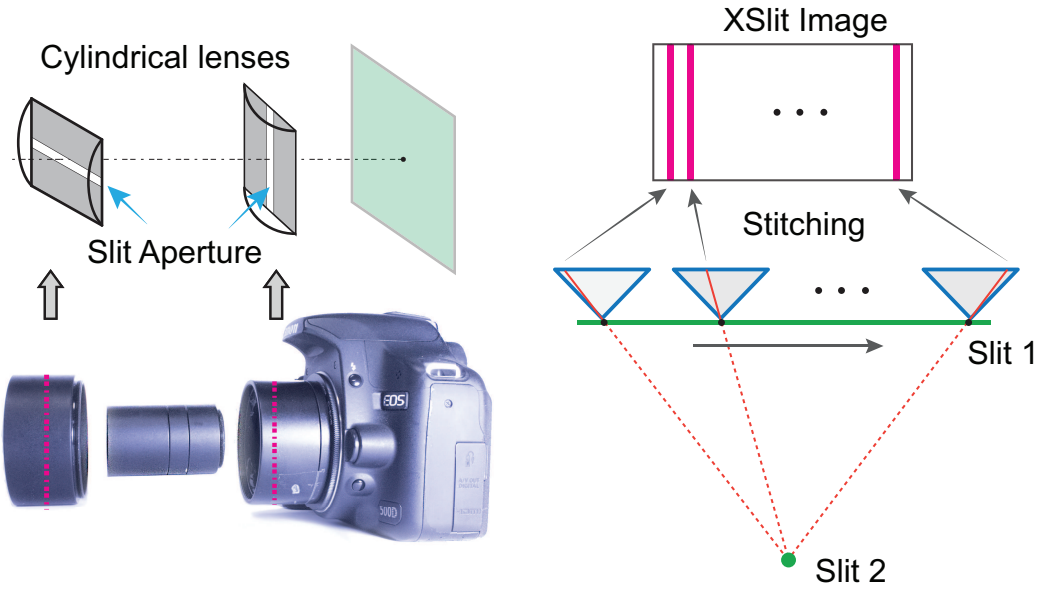


Figure 3.4: XSlit images can be captured by a real XSlit lens (left) or by stitching linearly varying columns from a 3D light field (right).

from the sensor) respectively. The distance between the two slits is adjustable between 5cm and 12cm and the slit apertures have a width of 1mm.

Another way to generate XSlit images is to capture a sequence of images by translating a pinhole camera along a linear trajectory at a constant velocity. In a similar vein, Seitz and Adams et al. acquire the image sequence by mounting the camera on a car facing towards the street. Additional registration steps [6] can be applied to rectify the input images. Next, linearly varying columns across the images are selected and stitched together. Fig. 3.4 shows the procedure of generating a XSlit image using a regular camera.

Chapter 4

COPLANAR COMMON POINTS

In this chapter, I explore the existence of Coplanar Common Points (CCP) feature in a broad range of non-centric cameras and its applications for scene understanding.

I first derive the necessary and sufficient conditions for CCP to exist in a general non-centric camera. I show that finding the CCP of a 3D plane is equivalent to solving an array of ray constraint equations in ray space. For certain types of non-centric cameras, e.g catadioptric imaging system, the ray space constraints can be highly complex. I show that the caustics provides simple and effective solution for determining CCP existence in these camera models. To demonstrate that CCP can potentially benefit the 3D reconstruction tasks, I then show some key applications of CCP. I show that with solely CCPs, we still can localize the planes in rotationally symmetric mirrors. Experiments on both synthetic and real data show that the CCP based solution provides effective and reliable solution for scene understanding.

4.1 Background

Employing image features pertaining to scene geometry for 3D reconstruction and scene understanding is an important task in computer vision. The classical pinhole camera, also referred as perspective camera, collects rays passing through a common CoP and exhibit some unique features. The most notable one is the existence of the Vanishing Point(VP). The Florentine artist and architect Brunelleschi first demonstrated its principles in early 1400s. When a set of parallel lines in space is not parallel to the picture/imaging plane, their projections on the picture/imaging plane will intersect at a common point, i.e the VP. Based on this rule, the artists of Renaissance developed the famous linear perspective technique for

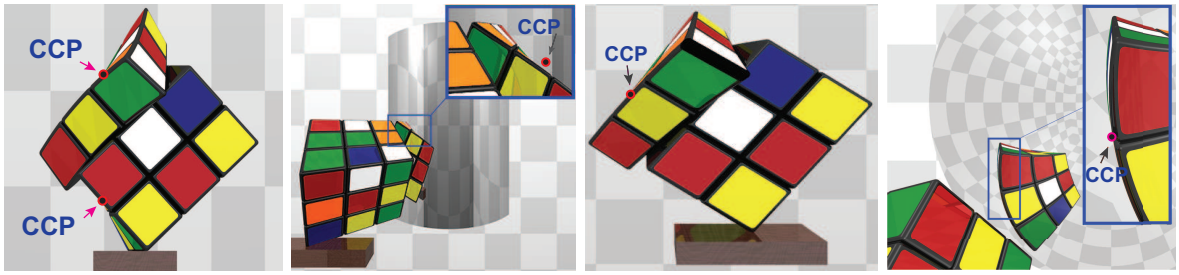


Figure 4.1: Different with the pinhole camera, lines map into curves in non-centric cameras. Some CCPs are directly observable.

painting. And today in computer vision community, tremendous efforts [17, 11] have been focused on utilizing the VP for scene understanding in perspective images.

The question this chapter aims to address is whether there exists some unique image features analogous to VPs in non-centric cameras that can be used for scene reconstruction. I demonstrate one such feature called the Coplanar Common Points (CCPs). CCP is the characteristics of positions (as an opposite of VP, which imply co-directionality): for a set of (oblique or parallel) lines lying on a 3D plane Π, will their images still intersect at a common pixel in the image plane? This is equivalent to the question: does there exist a common ray originating from the camera that will intersect with all lines on the plane. It is easy to verify that CCP does not exist in pinhole cameras: suppose such ray exists for plane Π, then the center of projection (CoP) must lies the plane since every ray will passes through the CoP. The plane will degenerate into a line and the CCP existence becomes trivial.

Though absent in pinhole camera, CCP generally exists for a broad range of non-centric cameras, ranging from well known Pushbroom and XSlit cameras [90], to the more general general linear cameras (GLCs), and to non-centric catadioptric mirrors [74]. I first observe that 3D lines map to curves (e.g., hyperbolaes in the XSlit cameras) as shown in Fig. 4.1, these curves intersect at the CCP as far as they lie on the same plane. In fact, the CCP and the 3D plane forms an one-to-one mapping, a highly useful property for 3D scene reconstruction and understanding.

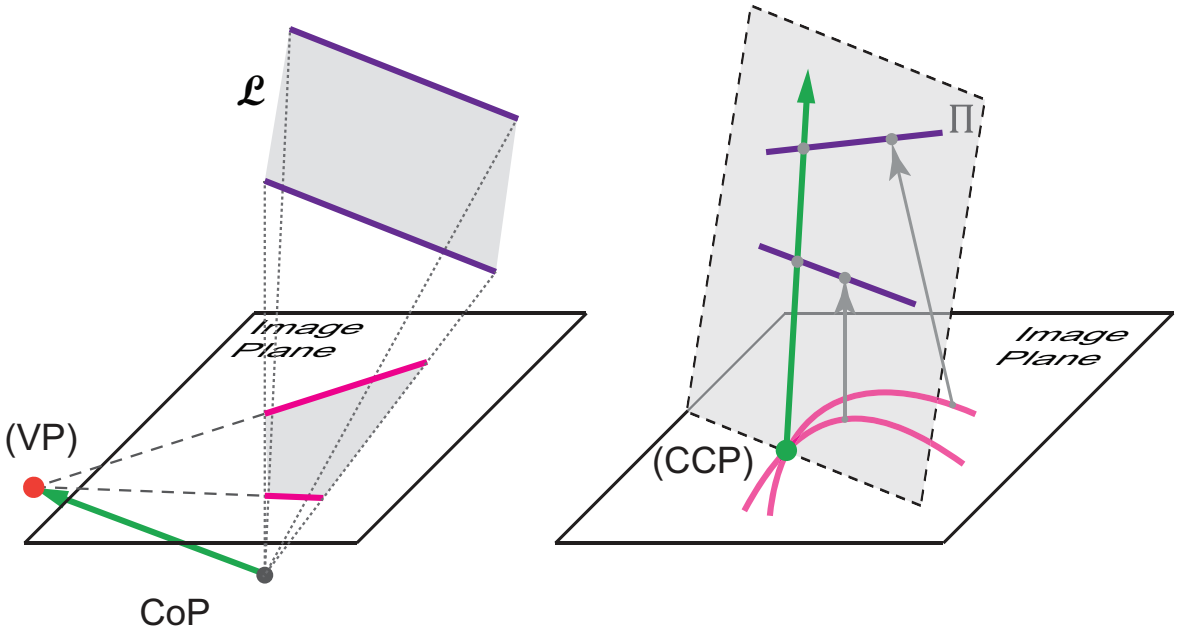


Figure 4.2: CCP and VP. The ray connects CoP and VP shares the common direction with the parallel lines. CCP is a point in the image plane corresponding to the intersection of the projections of all lines lying on a common 3D plane. Same to VP, the ray generate CCP shares the same plane.

4.2 Ray Space Analysis

Though the existence of VP is intuitive, all parallel lines that not parallel to image plane will converge at VP, the existence of CCP is camera and plane dependent. In this section, I explore how to explain the existence of CCP in the ray space. In general, a light ray is the propagation of radiance and it may vary along the path in presence of participating media and occlusions. To simplify the analysis, I assume that there is no participating media or occlusions so that the rays can be equivalently represented as lines. To represent the ray space, I adopt the two parallel planes paramterization (2PP). According to sec 3.1.2. All rays are parameterized as the 4-tuple $[u, v, \sigma, \tau]$. More complicated representation is possible and the derivation should be equivalent. The major reason to choose 2PP model is that, under 2PP, rays form a vector space, whereas other ray/line parameterizations such as the Plucker coordinates, do not.

A camera should correspond to a 2D manifold of rays as each pixel (i, j) on the image maps to a unique ray $[u, v, s, t]$ in 3D space, as shown in Eqn. 4.1. To answer the question

whether a CCP exists, I first investigate constraints imposed by a 3D plane. Given a 3D plane with normal $[n_x, n_y, n_z]$ can be parameterized as $\Pi : n_x x + n_y y + n_z z + d = 0$, any ray $r[u, v, \sigma, \tau]$ lying on Π should satisfy two constraints: 1) r's origin must lie on the plane and 2) r's direction is orthogonal to Π's normal. Therefore, we can derive the *ray-on-plane constraints* as:

$$\begin{cases} n_x u(i,j) + n_y v(i,j) + d = 0 \\ n_x \sigma(i,j) + n_y \tau(i,j) + n_z = 0 \end{cases} \tag{4.1}$$

To find whether CCP exists in a general non-centric camera, I set out to combine the *ray-on-plane constraints* with the camera's ray constraints and determine if there exists a solution that satisfies all constraints. This is equivalent find the point (i, j) in image that satisfy the constraints. The ray that generates point (i, j) should lie on the plane Π, and hence (i, j) is a CCP.

4.2.1 General Linear Camera

Yu and McMillan [86] introduced a class of primitive non-centric camera called the general linear cameras or GLC. They correspond to 2D affine subspaces embedded in the 4D light field space [88] and they can be used to describe a broad range of commonly used non-centric cameras including Pushbroom [27], XSlit, and linear oblique cameras.

A GLC is constructed by three generator rays r_1, r_2, r_3 so that all rays that collected by GLC are affine combinations of these three rays:

$$GLC = \{r : r = \alpha r_1 + \beta r_2 + (1 - \alpha - \beta) r_3, \forall \alpha, \beta\} \tag{4.2}$$

where α, β are affine coefficients.

Without loss of generality, we can pick three special generator rays originating from $[1, 0]$, $[0, 1]$ and $[0, 0]$ on Π_{uv} and rewrite the GLC equation as two linear constraints:

$$\begin{cases} \sigma = u\sigma_1 + v\sigma_2 + (1 - u - v)\sigma_3 \\ \tau = u\tau_1 + v\tau_2 + (1 - u - v)\tau_3 \end{cases} \tag{4.3}$$

where $[\sigma_i, \tau_i]$, $i = 1, 2, 3$ are the directions of the three ray generators.

Now that given a 3D plane Π, to determine if it has a CCP in the GLC, we can simply set out to find if there exists a ray that simultaneously satisfy the *ray-on-plane constraints* (Eqn. 4.1 and the GLC constraints (Eqn. 4.3). Notice that combining the two sets of equations result in a 4x4 linear system in $[u, v, \sigma, \tau]$:

$$\begin{bmatrix} -n_x & -n_y & 0 & 0 \\ 0 & 0 & -n_x & -n_y \\ \sigma_3 - \sigma_1 & \sigma_3 - \sigma_2 & 1 & 0 \\ \tau_3 - \tau_1 & \tau_3 - \tau_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ \sigma \\ \tau \end{bmatrix} = \begin{bmatrix} d \\ n_z \\ \sigma_3 \\ \tau_3 \end{bmatrix} \tag{4.4}$$

Whether the linear system has a solution depends on the determinant J:

$$J = n_x^2(\sigma_2 - \sigma_3) + n_x n_y(\tau_2 - \tau_3 - \sigma_1 + \sigma_3) - n_y^2(\tau_1 - \tau_3)$$

Hence the condition for CCP existence of GLC is that the determinant $J \neq 0$.

	Pinhole	XSlit	Pushbroom	Pencil	Bilinear
CCP Existence	$\times$	$\surd$	$\surd$	$\surd$	$\surd$

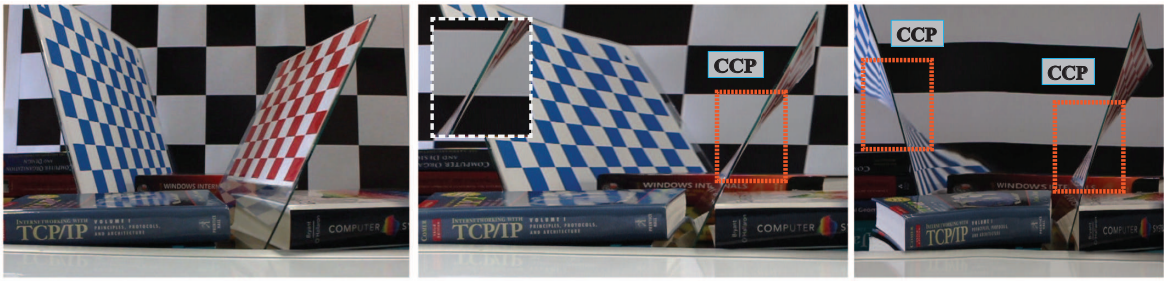
Table 4.1: CCP existence in popular GLCs.

Now that let us look at specific types of GLCs. To simplify our analysis, we assume that we translate the uv plane so that the third generator ray passes both the origins of the st and uv plane, i.e., $r_3 = [0, 0, 0, 0]$.

Pinhole: Assume the camera's CoP is at $[0, 0, f]$, by using the similitude relationship, we have $\sigma_2 = \sigma_3 = 0$, $\tau_1 = \tau_3 = 0$, $\sigma_1 - \sigma_3 = \tau_2 - \tau_3 = -1/f$. Therefore, we have $J = 0$ for any plane Π. Hence, CCPs do not exist in a pinhole camera.

Pushbroom: A pushbroom camera collects rays that passing through a common slit and parallel to a plane. We assume that the slit is parallel to Π_{uv} at distance Z. It is easy to see that we have $\sigma_2 = \sigma_3 = 0$, $\tau_1 = \tau_3 = 0$, $\sigma_1 - \sigma_3 = 0$, $\tau_2 - \tau_3 = -1/Z$. Therefore, $J \neq 0$ for general a 3D plane and CCPs exist in a pushbroom camera.

XSlit: We assume that Π_{uv} is parallel to both two slits. Z_1 and Z_2 are distances between Π_{uv} and the two slits. $[0, 0, 0]$ is the intersection point when orthogonally project both slits onto Π_{uv}. We further assume that the projections of two slits on Π_{uv} are perpendicular.



(a) Perspective (b) Pushbroom (c) XSlit

Figure 4.3: Synthesized GLC images through stitching specific rows or columns from a row of pinhole images. (a) The perspective view of the scene. (b) Pushbroom Image. (c) XSlit Image. (d) Pencil Image. Notice all lines on the 3D plane coverage at a CCP in Pushbroom, XSlit and Pencil cameras.

For the non vertical case, see [82]. In this model, we have $\sigma_2 = \sigma_3 = 0$, $\tau_1 = \tau_3 = 0$, $\sigma_1 - \sigma_3 = -1/Z_2$, $\tau_2 - \tau_3 = -1/Z_2$. $J \neq 0$ for general planes. XSlit camera has CCP.

Pencil and Bilinear: If we assume that the slit is parallel to $\Pi_u v$ at depth Z, we have $\sigma_2 = 1/Z$, $\sigma_3 = 0$, $\tau_1 = \tau_3 = 0$, $\sigma_1 - \sigma_3 = -1/Z$, $\tau_2 - \tau_3 = -1/Z$. Therefore, $J \neq 0$ for a general 3D plane and therefore CCPs exist. A similar conclusion holds for the bilinear (linear oblique) [57] cameras. The complete results are showed in Table. 4.1.

To prove the existence of CCPs in GLCs, I synthesize several popular GLC images by stitching specific rows/columns from a row of pinhole images. I mount a Cannon 60D SLR with 50mm F1.8 lens on translation track. Two planes are placed in front of the camera and the planes intersect with the camera path. I record a video while the camera is moving at a constant velocity. The resolution of the captured video is 1280×720. From each frame, I choose a specific column or row and stitch them together to form a new image. For push-broom, I choose column 480 in all frames, as showed in Fig. 4.3(b). I linearly increase the column index in terms of frame number and stitch these columns to form an XSlit image, as showed in Fig. 4.3(c). Finally, I linear increase the row index in terms of the frame number and stitch these rows to form a pencil camera, as showed in Fig. 4.3(d). The highlighted rectangles illustrate where CCPs occur in pushbroom, XSlit and pencil cameras, which is consistent with our theory.

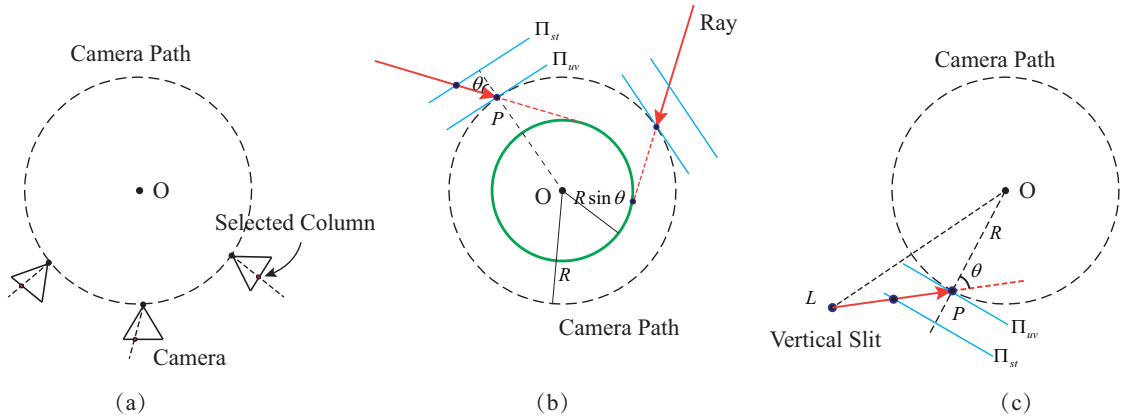


Figure 4.4: Concentric mosaics are synthesized from a sequence of images captured by a perspective camera moving along a circular path. (a) We define 2PP tangent to the camera path and rotating with the camera. (b) Circular XSlit panorama.

4.2.2 Concentric Mosaics

There are other widely used non-centric cameras, for example, concentric mosaics [75, 71] or circular XSlit [89]. These camera models are generally synthesized from rotational panoramas. To acquire a concentric mosaics, it is common practice to rotate a camera off-axis on a circle. For each camera position, a column is sampled according to its angle from the optical axis. Then all selected columns are stitched together to form a panorama, as showed in Fig. 4.4(a).

To investigate the existence of the CCP, we set the origin of the coordinate system as the rotation axis. Assume the xy plane is the the camera path plane and we can also adopt 2PP model for ray parametrization. However instead of using two fixed planes, we use two parallel planes that rotate along with the camera. We set Π_{uv} tangent to the camera path at P and vertical to the xy plane. Π_{st} is parallel to Π_{uv} with distance 1. At each position, the column with angle θ is sampled. The set of all collected ray intersect Π_{st} at $[x + (x - y\tan\theta)/R, y + (y + x\tan\theta)/R]$. This allows us to map rays collected by the camera as a 2D manifold defined by x, y and θ:

$$[u, v, \sigma, \tau] = [x, y, (x - y\tan\theta)/R, (y + x\tan\theta)/R] \tag{4.5}$$

Notice the z value of the intersection on Π_{st} is not necessarily 1 as in the conventional 2PP

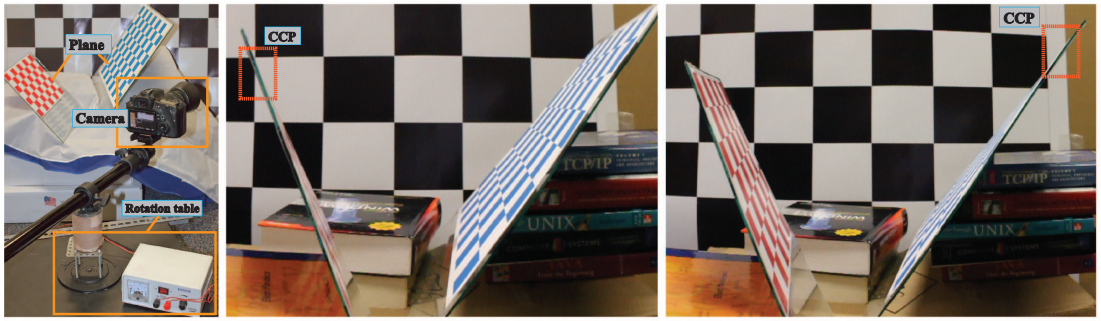


Figure 4.5: Captured CCP in concentric mosaics.

case. Assume $z = \lambda$, we can rewrite the *ray-on-plane constraints* as:

$$\begin{cases} n_x x + n_y y + d = 0 \\ \sin\theta(\sin\theta x + \cos\theta y) \cdot n_x + \sin\theta(\sin\theta y - \cos\theta x) \cdot n_y + \lambda n_z + d = 0 \end{cases} \tag{4.6}$$

Notice though that the solution to Eqn. 4.6 is actually the intersection of Π and a cylinder centered at origin and with radius $R\sin\theta$. Therefore Eqn. 4.6 essentially indicates that the plane should have intersections both with the camera path circle and the cylinder, as showed in Fig. 4.4(a). This analysis is consistent with the geometric interpretation: the cylinder is the inner viewing surface and all rays the collected by the camera should be tangent to the cylinder. This type of concentric mosaics resembles the pushbroom camera where the central columns are stitched together.

A different way to construct concentric mosaic is propose in [89], analogous to stitching an XSlit panorama from a translational array of images. Using the same acquisition setup, we select, at each camera position, the column with the ray that passes a predefined vertical slit, as showed in Fig 4.4(b). The result is a circular XSlit model with one vertical slit and one circular slit where the circular slit is the trajectory of the camera. In this set up, θ is a nonlinear function of x and y. The intersections of the rays collected by P and Π_{st} is along PL. Hence $\sigma = \widetilde{\lambda}(s_x - x), \tau = \widetilde{\lambda}(s_y - y)$, s_x, s_y are x and y coordinates of the vertical slit. Eqn. 4.6 now becomes: $n_x\widetilde{\lambda}(s_x - x) + n_y\widetilde{\lambda}(s_y - y) + \lambda n_z = 0$. Since λ, $\widetilde{\lambda}$ are both

scalers, we can eliminate $\widetilde{\lambda}$ as:

$$\begin{cases} n_x x + n_y y + d = 0 \\ n_z s_x + n_z s_y + \lambda n_z + d = 0 \end{cases} \tag{4.7}$$

This indicates that for a plane Π to have a CCP, it should intersect with both the vertical slit and the camera path.

To construct a concentric mosaic, I mount a Cannon 60D SLR with 50mm F1.8 lens on a rotation table. I align the optical axis to pass through the rotation axis. Two planes are placed in front of the camera and the planes intersect the camera path. I record a video as the camera rotates. The system setup is showed in Fig. 4.5(a). From the recorded view, weI select specific columns from different frames and stitch them to form a concentric mosaic image. Fig. 4.5(b) shows the images formed by stitching column 560 from all frames. Fig. 4.5(c) shows the result by stitching column 840. The highlighted regions show where the CCPs occur.

4.3 Caustic Perspective

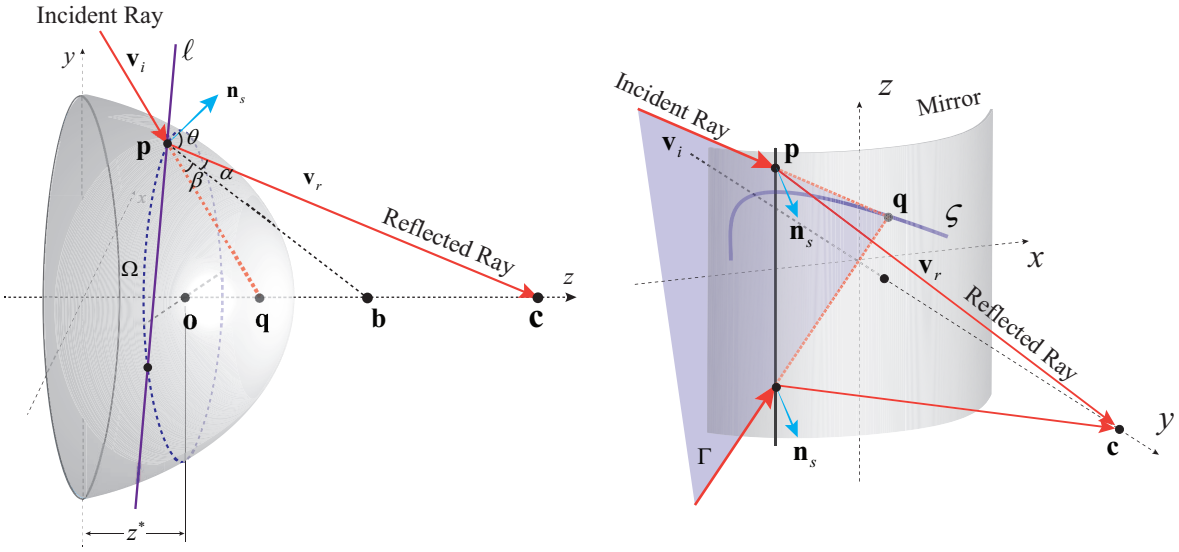
A commonly used class of real non-centric cameras are catadioptric mirrors [7, 74] in which a regular pinhole camera is positioned in front of a curved mirror for acquiring images with a much wider field-of-view (FoV).

Recall that our goal is to determine if we can find an incident ray collected by the camera that lie on Π. Notice that each point $P(x, y, z)$ on the mirror surface corresponds to a reflection ray. Hence we can also potentially map the CCP problem into the ray space similar to the GLC and concentric mosaic case: assume the mirror surface is in form $z(x, y)$, the incident ray $\mathbf{v}_i = [i^x, i^y, 1]$ can be computed as:

$$\mathbf{v}_i = \mathbf{v}_r - 2\frac{\mathbf{n}_s \cdot \mathbf{v}_r}{\| \mathbf{n}_s \|^2}\mathbf{n}_s \tag{4.8}$$

$\mathbf{n}_s$ is the surface normal and $\mathbf{v}_r$ is the reflection ray. Intersect this ray with Π_{uv} and Π_{st}, we can get the 4D representation:

$$[u, v, \sigma, \tau] = [x - z \cdot i^x, y - z \cdot i^y, i^x, i^y] \tag{4.9}$$



(a) Rotationally Symmetric Mirror (b) Rotationally Symmetric Mirror

Figure 4.6: (a) Ray geometry in rotationally symmetric mirror. The condition for a plane to have CCP: Intersect with z-axis at a point $\mathbf{q}$ and have intersections with the Ω that determined by $\mathbf{q}$. (b) Ray geometry in cylinder mirror. The condition for a plane to have CCP: Intersect with the caustic ζ and not perpendicular to the xy plane.

u, v, σ, τ are functions in x and y. The set of rays collected by the mirror surface form a ray-space parametric manifold in x and y.

$$\Sigma(x, y) = [u(x, y), v(x, y), \sigma(x, y), \tau(x, y)] \tag{4.10}$$

We have the *ray-on-plane constraints*:

$$\begin{cases} u(x, y)n_x + v(x, y)n_y + d = 0 \\ \sigma(x, y)n_x + \tau(x, y)n_y + n_z = 0 \end{cases} \tag{4.11}$$

For a given plane Π, we have two equations, two unknowns. In theory, we can determine if Π has a CCP by testing if Eqn. 4.11 has a solution. In reality, $\Sigma(x, y)$ can become highly complex and searching for the solution is a challenging algebraic problem.

A different and more intuitive solution is view the problem from the caustic perspective. Caustic is a curve or surface where the light rays light concentrate [74, 87]. Caustic surfaces always appear in pairs and locally they can be interpreted as XSlit cameras. If a plane has a CCP, then the plane has to intersect with the two caustic surfaces, a necessary

condition for the CCP to exist. However, the condition is insufficient: the resulting intersections are two curves on the caustic surfaces. Since the caustic surfaces need to appear in pairs, the two curves do not necessarily form valid correspondences. Therefore, we would need to check for, every ray originating from the first curve, whether it will pass through the second curve. If there exists such a ray, then the CCP exists. Otherwise, it does not. The procedure above provides a simple but effective recipe for determining CCP existence. In the following sections, I analyze several commonly used catadioptric mirrors.

4.3.1 Rotationally Symmetric Mirrors

A rotationally symmetric mirror is formed by rotating a quadric curve about its symmetric axis. Assume the symmetric axis is aligned with the z-axis, the mirror surface can be parameterized as

$$r^2 + Az^2 + 2Bz - C = 0, x^2 + y^2 = r^2 \tag{4.12}$$

A, B and C are the curve parameters that determine the mirror shape. In particular, $A = 1, B = 0, C > 0$, the mirror is a sphere; $A > 0, C > 0$, elliptical mirror; and $A < 0, C > 0$, hyperbolic mirror. We assume the the pinhole viewing camera is on the symmetric axis of the mirror. The only singular case is when the pinhole coincides with curve's foci that emulates a virtual pinhole, in which the CCP does not exist. Therefore, in order to observe the CCP, we need to place the viewing camera off the foci.

By the symmetry, all reflection rays that collection by the view camera should intersect the symmetric axis. Hence the the symmetric axis is one of the two caustic surfaces.

CCP Condition 1: *the 3D plane must have an intersection with the symmetric axis.*

The projection under a rotationally symmetric mirror is shown in Fig. 4.6(a). Given an incident ray $\mathbf{v}_i$ from the scene towards the mirror surface, to produce image in the viewing camera, its reflection ray $\mathbf{v}_r$ must pass through the CoP $\mathbf{c} = [0, 0, c]$, lying on the z-axis. Since $\mathbf{v}_i$ and $\mathbf{v}_r$ are coplanar, $\mathbf{v}_i$ also have an intersection with the z-axis. Therefore, a valid CCP projection must intersect with the z-axis, and hence the plane.

As mentioned above, the plane that intersects with the symmetric axis does not necessarily have a CCP. Assume $\mathbf{q} = [0, 0, d']$, $d' = -d/n_z$ is the intersection point between the

common plane and symmetric axis. We now only need to check the set of rays determined by $\mathbf{q}$. Assume the incident ray is reflected at $\mathbf{p}$ on the mirror surface and we have $\mathbf{v}_i = \mathbf{p} - \mathbf{q}$. It is important to note that ruled by the law of reflection, there only exists one circle of $\mathbf{p}$ on the mirror whose resulting reflection ray can enter $\mathbf{c}$. Assume the circle is $\Omega : x^2 + y^2 = r^{*2}$ for $\mathbf{p}$, we can obtain the set of rays as a cone that connects $\mathbf{q}$ and Ω.

To compute Ω, we orthogonally project $\mathbf{p}$ onto the z-axis and obtain $\mathbf{o} = [0, 0, z^*]$. By Eqn. 4.12, the surface normal at $\mathbf{p}$ can be computed as $[x^*, y^*, Az^* + B]$. Thus the tangent plane at $\mathbf{p}$ is $x^*x + y^*y + (Az^* + B)z + (Bz^* - C) = 0$. Hence we can compute the intersection point $\mathbf{b}$ of the tangent plane and z-axis. Since $\forall p \in \Omega$, the corresponding reflection rays pass through $\mathbf{c}$, the tangent plane bisect the angle formed by $\mathbf{v}_i$ and $\mathbf{v}_r$, i.e, $\alpha = \beta = 90° - \theta$, by the law of reflection, as shown in Fig. 4.6. Consider the triangle formed by $\mathbf{q}$,$\mathbf{p}$ and $\mathbf{c}$, we can formulate the following equation by applying the angle bisector theorem to solve for z^*

$$\frac{\sqrt{r^{*2} + (d' - z^*)^2}}{\sqrt{r^{*2} + (c - z^*)^2}} = \frac{C - Bd' - (B + d'A)z^*}{(cA + B)z^* + cB - C} \tag{4.13}$$

The solution to Eqn. 4.13 corresponds to valid reflection points on the mirror surface.

CCP Condition 2: *the 3D plane must have intersection with* Ω.

Recall that not all the planes contain $\mathbf{q}$ will intersect with the circle Ω. To test the plane-circle intersection, we compute the distance from the plane to the z-axis at $z = z^*$ as:

$$D = \frac{|n_z(z^* - d')|}{\sqrt{n_x^2 + n_y^2}} \tag{4.14}$$

If $D > r^*$, the plane will have no intersection with the circle and thus has no CCP. If $D = r^*$, the plane has one intersection with the circle and thus has a single CCP; When $D < r^*$, the plane has two intersections with the circle and will have two CCPs.

4.3.2 Cylinder Mirrors

Another commonly used class of catadioptric mirrors is cylinder mirrors. Given a quadratic curve on the xy-plane, instead of rotating it about its symmetric axis, we extrude

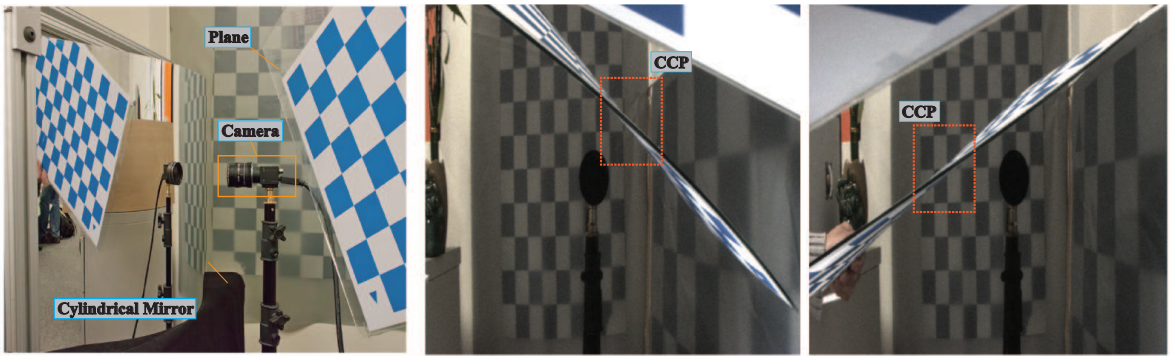


Figure 4.7: Captured CCP in cylinder mirror.

the curve along the z-axis to form a cylinder. By aligning the y-axis with symmetric axis, a cylinder mirror can be parameterized as

$$x^2 + Ay^2 + 2By - C = 0, z = t \tag{4.15}$$

Same as the rotational symmetric mirror, A, B and C are the quadratic curve parameters. When $A = 1, B = 0, C > 0$, we have a cylindrical mirror; $A > 0, C > 0$, elliptical cylinder mirror; and $A < 0, C > 0$, hyperbolic cylinder mirror. We place the camera on the y-axis such that the CoP can be written as $\mathbf{c} = [0, c, 0]$.

Whether the cylinder mirrors have one dimensional caustic is not as clear as the rotationally symmetric mirrors. We start with considering the forward projection problem, i.e, finding the incident ray constraints whose corresponding reflection ray will pass through the CoP, as shown in Fig. 4.6(b). Assume $\mathbf{p} = [x, y, z]$ is a point on the mirror surface where reflection occurs. First, we can determine the direction of the reflection ray $\mathbf{v}_r$ by connecting $\mathbf{p}$ and $\mathbf{c}$. Thus we have $\mathbf{v}_r = [-x, -y, c - t]$. The surface normal at $\mathbf{p}$ can be computed as $\mathbf{n}_s = [x, Ay + B, 0]$. By specular reflection, we can compute the incident ray from Eqn. 4.8 as:

$$\mathbf{v}_i = [\frac{2g - f}{f}x, \frac{2Ag - f}{f}y + (\frac{2g}{f}B + c), -t] \tag{4.16}$$

Where $f(x) = (1 - A)x^2 + (B^2 - AC), g(y) = -(B + Ac)y + (Bc - C)$. The incident ray can be parameterized in the point-direction form as $\mathbf{p} + \lambda\mathbf{v}_i$. Let $\lambda = 1$, we have

the intersection point of the incident ray and the xy-plane ($z = 0$) as $\mathbf{q} = [2gx/f, 2g(Ay + B)/f + c, 0]$. Notice that $\mathbf{q}$ is independent of the z component of $\mathbf{p}$. This indicates that for all $\mathbf{p}$ on a vertical line (parallel to z-axis) on the mirrors surface, the corresponding effective incident ray will pass through the same point $\mathbf{q}$ on xy-plane. From the geometric perspective, $\mathbf{q}$ is actually the reflection point of the CoP w.r.t the tangent plane of the mirror surface at $\mathbf{p}$ and hence it is equivalent to a virtual CoP. Since all points on a vertical line share the same tangent plane, the CoP reflection $\mathbf{q}$ remains the same. Further, by sliding the vertical line on the mirror surface, we obtain a set of $\mathbf{q}$ that form a curve ζ. ζ is the one dimensional caustic of cylinder mirrors and can be derived in $\mathbf{q}$ as:

$$\zeta(x, y) = \Sigma\{[\frac{2gx}{f}, 2\frac{g}{f}(Ay + B) + c, 0]\} \tag{4.17}$$

For all x, y on the cylinder mirror surface.

CCP Condition: *the 3D plane must intersect with* ζ.

As mentioned before, each point on ζ determines a set of rays on a vertical plane Γ, as show in Fig. 4.6(b). If only the common plane does not perpendicular to the x-y plane, i.e $n_z \neq 0$, there will be one valid CCP projection ray which is the intersection line of the common plane and Γ. Similar to the rotationally symmetric mirror example, the number of intersections between the common plane and ζ determines the number of CCPs.

For validation, I place a PointGrey FL2-08S2C camera in front of a cylindrical mirror and align the optical axis with the cylindrical axis, as showed in Fig. 4.7(a). The resolution of the captured images is 1024×768. I place a plane in front of the mirror and the catadioptric images of the plane is shown in Fig. 4.7(b) and (c). Notice that the plane exhibits the CCP. To understand why this is the case, recall that our analysis shows that as far as the plane has an intersection with the caustic circle, the plane generally has a CCP. In our case, I position the plane near the center of the mirror and tilt it so that it is guaranteed to intersect with the circle and hence has a CCP.

4.3.3 Validation

I render catadioptric mirror images using the POV-Ray ray tracer (www.povray.org). My first experiment is performed on a cylindrical mirror $x^2+(y-10)^2 = 16, z = t$ where the viewing camera is place at the origin $(0, 0, 0)$ facing towards the cylindrical mirror. Our scene consists with two planes: $\Pi_1 : 2.7475y + z - 12 = 0$ and $\Pi_2 : 3.9153y + z - 8.9378 = 0$. Each plane consists of five lines, among which three are parallel. The camera and scene setups are shown in Fig. 4.8(a). As shown in Fig. 4.8(b), Π_1 has one intersection with ζ while Plane Π_2 has two. Fig. 4.8(c) shows the captured catadioptric images of Π_1 and Π_2 and we can observe that Π_1 has one CCP and Π_2 has two. Our results are hence consistent with the theoretical prediction.

Next, I test on a hyperbolic mirror $z^2/16 - r^2/9 = 1$ with the viewing camera the origin. The scene consists of two planes: $\Pi_3 : 0.6608y - z + 5.1530 = 0$ and $\Pi_4 : 0.7908y+z-5.4184 = 0$. Same as our first experiment, I place five lines on each plane. The experimental setup is shown in Fig. 4.9(a). As showed in Fig. 4.9(a), the intersections between the planes and circle Ω. Fig. 4.9(b) shows the captured catadioptric image of the two planes and the numbers and position of CCP are consistent with the intersection points between the plane and Ω as predicted by our theory. We can further map the CCPs back to 3D space, showed in Fig. 4.9(c). The recovery of the 3D planes are highly accurate.

4.4 Application

In perspective images once the VP of a set of parallel lines is identified, the lines' common direction can be directly recovered with calibrated camera parameters: the set of parallel lines are also parallel to the light ray that passes through the CoP and VP. Notice that the exact line can not be recovered since we still need to know at least one point the line passes through.

A CCP determines only one on-plane ray in space. Similar to the VP, with one CCP we can only recover to a set of planes that pass through the ray. This ambiguity exists in GLCs, concentric mosaics and cylinder mirrors. In certain cases, one plane can have two CCPs, e.g. in rotationally symmetric mirrors, a plane can be directly located from its CCPs.

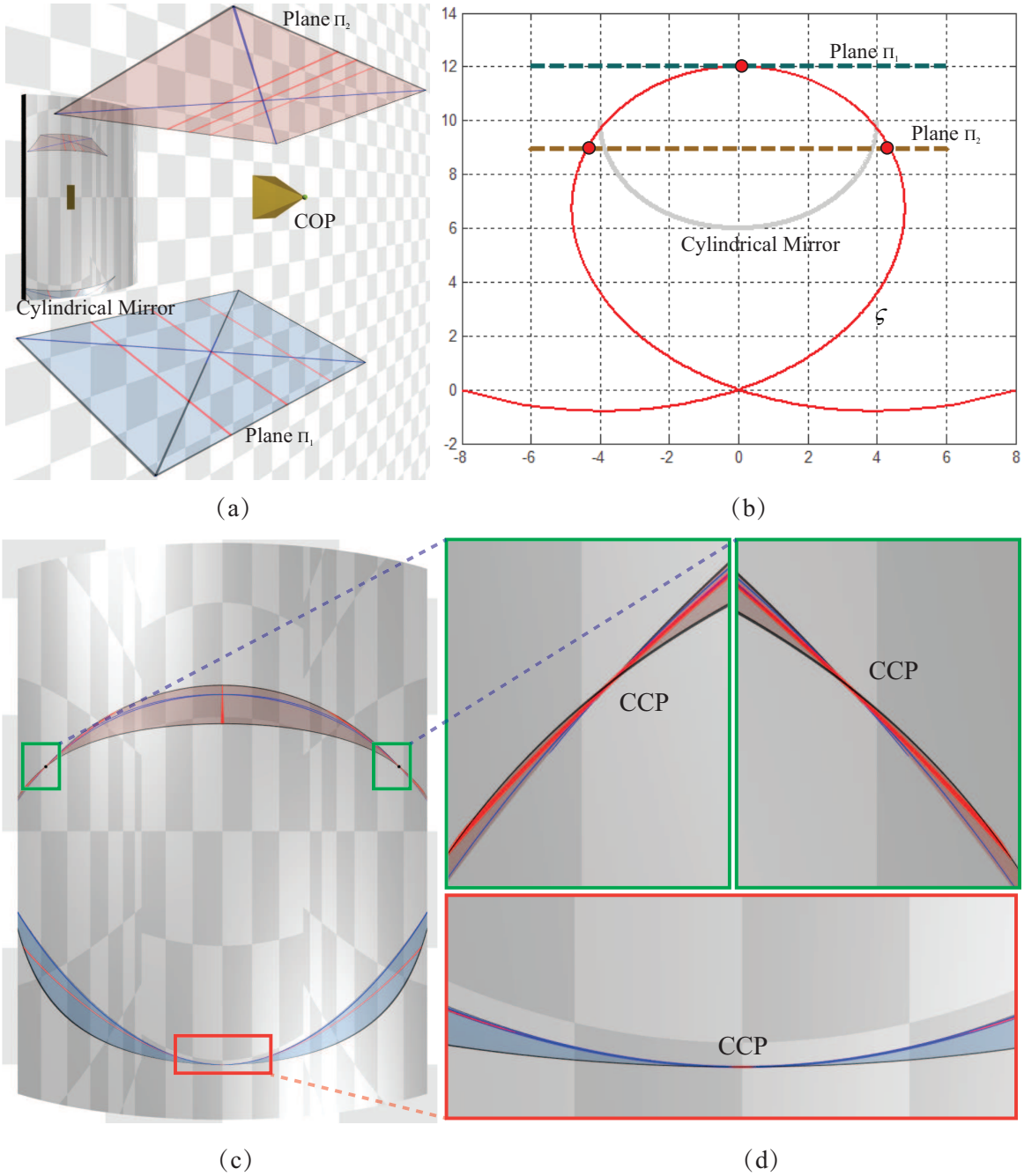


Figure 4.8: Experiments on a cylindrical mirror. (a) Experimental setup; (b) Intersections between each plane and ζ; (c) Rendered catadioptric images; (d) Close-up views at each CCP.

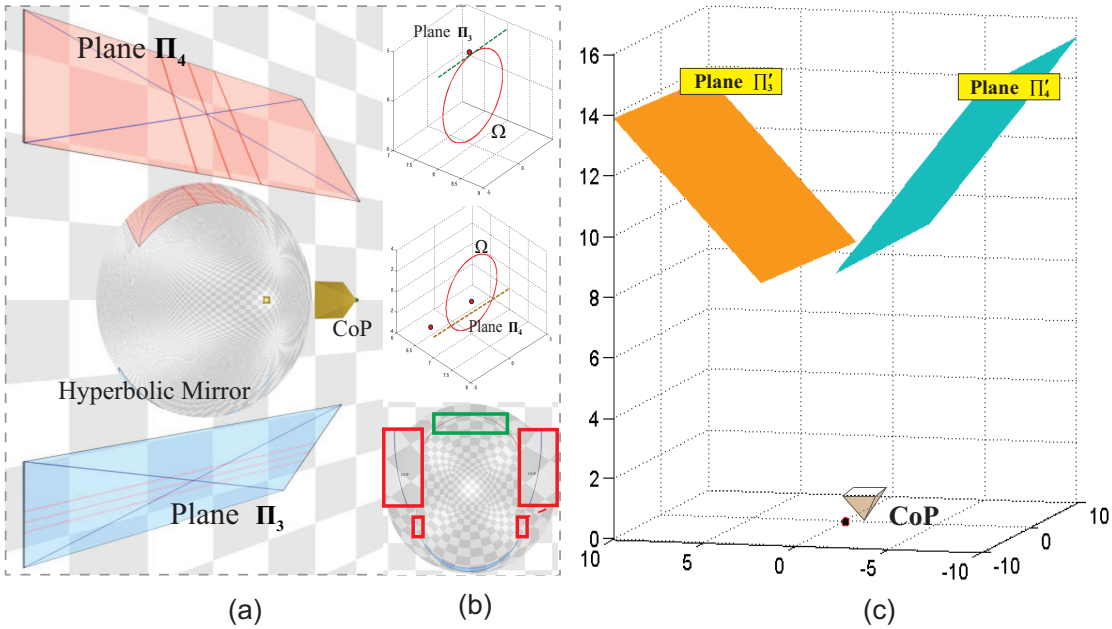
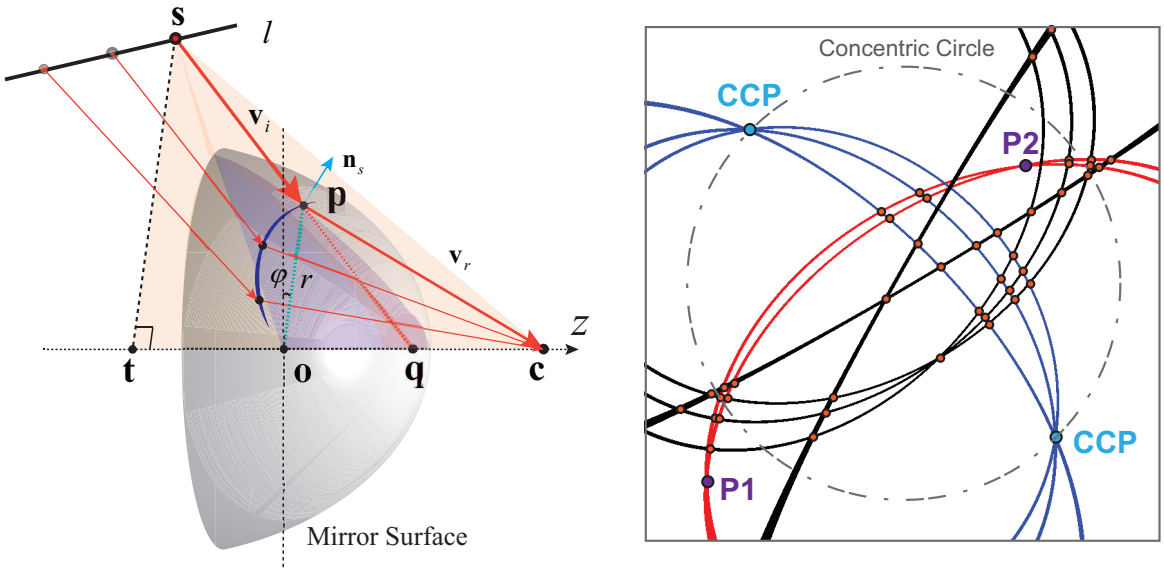


Figure 4.9: Experiments on a hyperbolic rotationally symmetric mirror. (a) Experimental setup; (b) Intersections between each plane and Ω; (c) Rendered catadioptric image; (d) Close-up views at each CCP.



(a) Line Projection Geometry in Rotationally Symmetri Mirror

(b) Identify CCPs in Catadioptric Images

Figure 4.10: (a) Line projection in a symmetric catadioptric mirror. The point $\mathbf{s}$ on line must lay on a cone determined by $\mathbf{p}, \mathbf{q}$. (b) CCPs of a common plane must lie on a circle, this constraint can effectively rules out interferences such as $P1$ and $P2$.

Recall that a plane may have one or two CCPs for rotationally symmetric mirrors. We consider the two cases separately.

Case I: One CCPs. If there is only one CCP, the plane must be the tangent plane of the circle Ω. Therefore, we can first find the tangent line l at $z = z^*$ of Ω. The plane can be reconstructed through the l and the CCP's corresponding $\mathbf{q}$.

Case II: Two CCPs. If two CCPs exist, we can instantly recover the plane by three points on it, i.e, the two CCPs and their corresponding $\mathbf{q}$.

Complete the image of 3D lines: since the CCP may lie out of the image plane, we need to complete the image of 3D lines on the mirror in rotationally symmetric mirrors. Recall that the line images in rotationally symmetric mirrors are high-order curves. Therefore, it is infeasible to directly fit the curves using a parametric model. Instead, we adopt a point-aggregate strategy to find all possible pixels on the image plane that can be the image of the 3D line. The complete set is then the line image.

Give a 3D point $\mathbf{s} : [x_0, y_0, z_0] + \lambda[dx, dy, dz]$ on line l and its reflection point $\mathbf{p} : [x, y, z]$ on mirror surface $\mathbf{p}$. We first establish a constraint on $\mathbf{s}$ and $\mathbf{p}$. Notice the $\mathbf{s}$,$\mathbf{p}$ and $\mathbf{c}$ are coplaner. So we have the following equation: $(x_0 + \lambda dx)/(y_0 + \lambda dy) = \tan\varphi$, φ is the angle between $\mathbf{op}$ and y-axis. The incident ray $\mathbf{v}_i$ intersects z-axis at $\mathbf{q} = [0, 0, z^*]$. Line $\mathbf{st}$ is perpendicular to the z-axis, we have its length as $\dfrac{z_0 + \lambda dz - z^*}{z - z^*} r$. Since $\mathbf{s}$ is on the cone that defined by $\mathbf{p}, \mathbf{q}$, we obtain a constraint on the line equation and its possible reflection point position on the mirror:

$$(x_0 + \lambda d_x)^2 + (y_0 + \lambda d_y)^2 = \frac{r^2}{(z - z^*)^2}(z_0 + \lambda d_z - z^*)^2$$

Substituting λ with $\tan\varphi$, we have:

$$a \cdot \tan\varphi z^* + b \cdot \tan\varphi + c \cdot z^* + d = \sqrt{1 + \tan^2\varphi}\frac{(z - z^*)}{r} \tag{4.18}$$

where $a = d_y/e, b = (y_0 d_z - z_0 d_y)/e, c = -d_x/e, d = (z_0 d_x - x_0 d_z)/e, e = y_0 d_x - x_0 d_y$.

Since a,b,c,d and e are uniquely determined by the line parameters, they can be used to identify the line image. In our point-aggregate fitting algorithm, we first estimate the a, b, c

and d using the observed line image. We then in turn use a, b, c and d to locate possible pixels on the line image. Given a catadioptric mirrors, we first establish a lookup table T that contains surface point $\mathbf{p}[r, z]$ and its corresponding $\mathbf{q}[0, 0, z^*]$. For each pixel $\mathbf{p}'$ on the observed line image, we compute r and φ corresponding to $\mathbf{p}$. We then consult the lookup table T to find the corresponding z^* of $\mathbf{p}$. By Eqn. 4.18, we can use φ, r and z^* to solve for a, b, c and d. Recall that we can use a, b, c and d to trace out all the points on the line image. **Identify CCPs:** after complete the image of 3D lines, we intersect every two lines and obtain a set of CCP candidates through clustering and voting. We group these CCP candidates if they are generated by the same set of lines. Recall our CCP condition analysis in Sec. 4.3.1, the CCPs are generated by rays with refection points lie on a circle Ω. Hence in the image, CCPs of a common plane should also lie on a circle, as shown in Fig. 4.10(b). With this CCP candidates constraint, for each candidate group we test each two intersection points to see whether they are on a circle. If yes, then we find a CCP pair.

For validation, I mount a spherical mirror on a vertical reference plane. The radius of the mirror is 51.64mm. I place the PointGrey FL2-08S2C camera with focal length 7.85mm in front of the spherical mirror. The camera is pre-calibrated. Next, I align the optical axis to pass through the center of sphere using the reference plane, as showed in Fig. 4.11(b). I set the center of the spherical mirror to be the origin of the coordinate system and connect it with the camera's CoP as the z axis. As a result, the CoP is at $[0, 0, 182.37]$ in our coordinate system.

I attach three parallel white stripes on to a black plane and place it in front of the mirror. The captured image is showed in Fig. 4.11(a). I then apply our curve fitting algorithm of the white stripes and the fitted results are showed in Fig. 4.11(c). Our results reveal that the images of these stripes (lines) intersect at two CCPs in addition to the vanishing points. The 3 fitted curves, however do not exactly intersect at the same CCPs due to errors in curve fitting. I therefore average the estimation as the final detected CCPs. In this example, the two CCPs have pixel coordinate as $[332.5, 207.1]$ and $[809.45, 232.5]$. I map them back to their reflection points on the mirror at $z^* = 33.675$ and finally locate the plane from the two CCPs and z^*. The plane reconstructed is $x + 6.2806y - 3.8944z + 131.14 = 0$, showed in

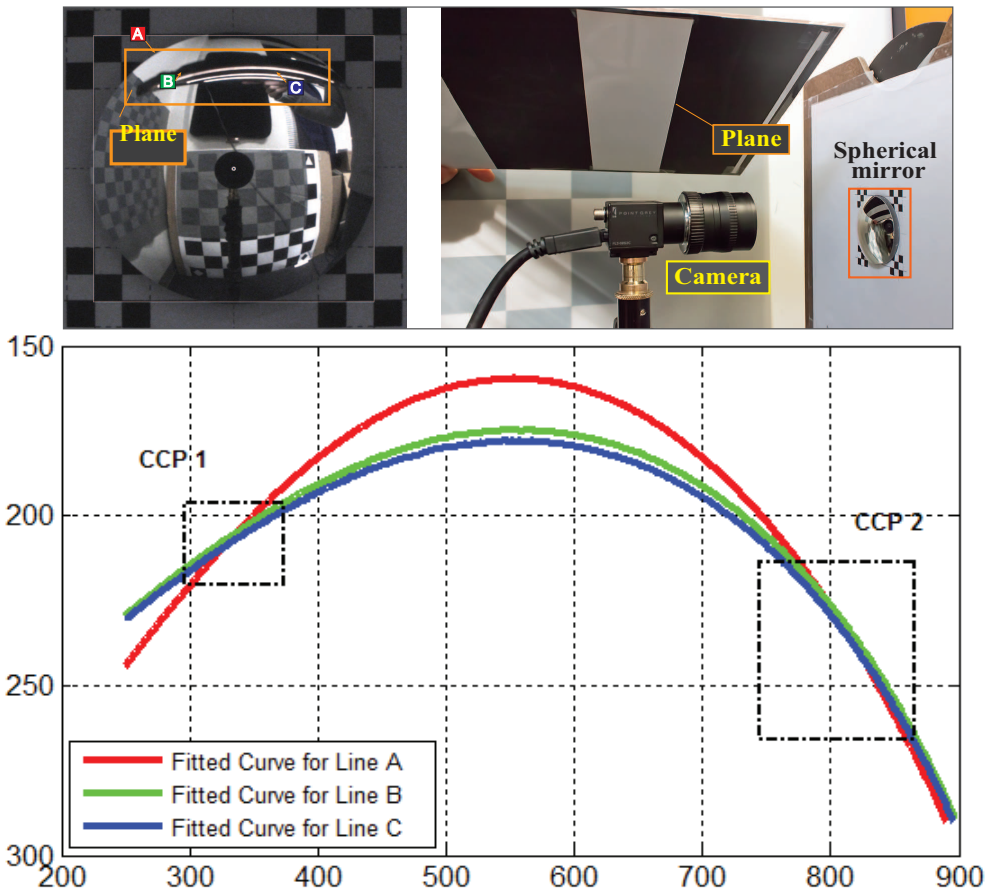


Figure 4.11: Line projection in a symmetric catadioptric mirror. Left: We show the line image and the mirror profile; Middle: Located CCPs by curve fitting; Right: Reconstructed plane by using CCPs.

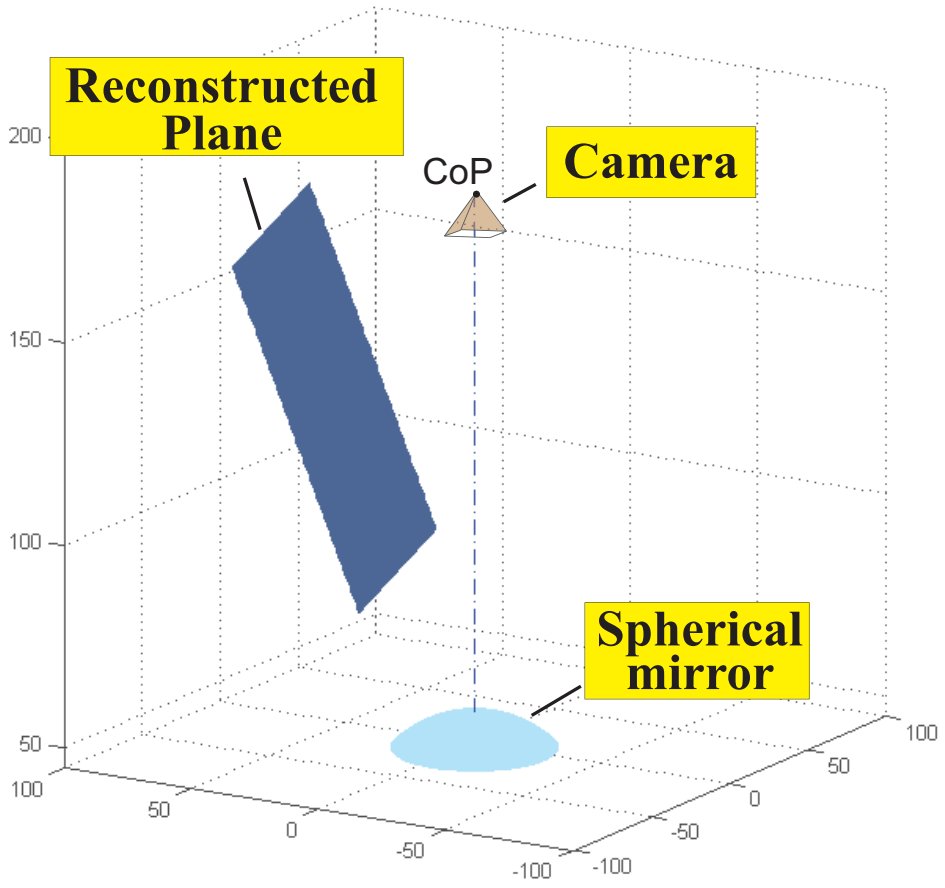


Figure 4.12: Line projection in a symmetric catadioptric mirror. Left: We show the line image and the mirror profile; Middle: Located CCPs by curve fitting; Right: Reconstructed plane by using CCPs.

Fig. 4.12.

4.5 Discussion

I have explored a new type of image features called the coplanar common point or CCP in general non-centric cameras. A CCP corresponds to the intersection of the curved projections of all lines lying on a common 3D plane. I have shown that CCPs generally exist in a broad range of non-centric cameras such as the general linear camera, and the perspective camera is the single exception that do not have CCP. I have further derived the necessary and sufficient conditions for a plane to have CCP in an arbitrary non-centric camera such as non-centric catadioptric mirrors. I have demonstrated that with CCPs, we can conduct scene

understanding tasks, such as plane localization and Manhattan scene understanding. CCP is one of the most advantageous feature of non-centric over centric cameras, e.g the inherit coplanar ambiguity in perspective imaging can be naturally resolved with a non-centric camera and employ CCPs to distinguish different planes. Experiments on both synthetic and real data show that the CCP based solution provides effective and reliable cues for scene understanding.

Chapter 5

SCALE AMBIGUITY

In this chapter, I explore the Depth Dependent Aspect Ratio(DDAR) feature in XSlit camera [90]. I show that the XSlit camera exhibits DDAR that can help to resolve the scale ambiguity that plagued perspective cameras. In contrast with the invariant aspect ratio(AR) in perspective cameras, the observed AR of a frontal parallel object in XSlit image changes according to its depth, as shown in Fig. 5.1. I first develop a comprehensive analysis to characterize DDAR in the XSlit camera. This derivation leads to a simple but effective graph-cut based scheme to recover object depths from a single XSlit image and an effective formulation to model recoverable depth range, sensitivity, and errors. In particular, I show how to exploit repeated shape patterns exhibiting in real Manhattan World scenes to conduct 3D reconstruction.

The DDAR analysis can further be extended to model the slopes of lines. Specifically, for parallel 3D lines of a common direction, I show that as far as the direction is different from both slits, their projections will exhibit depth-dependent slopes or DDS, i.e., the projected 2D lines will have different slopes depending on their depths. DDS and DDAR can be combined to further improve 3D reconstruction accuracy. I validate the theories and algorithms on both synthetic and real data. For real scenes, I experiment on different types of XSlit images including the ones captured by the XSlit lens [83] and synthesized as stitched panoramas [68]. In addition, my scheme can be applied to catadioptric mirrors by modeling reflections off the mirrors as XSlit images. Experiments show that DDAR and DDS provide important depth cues and enable effective single-image scene reconstruction.

The GLC theory [85] has shown that the XSlit camera can describe a broad range of non-centric cameras. In fact, the Pushbroom, orthographic and perspective cameras can all

be viewed as special XSlit entities. Hence, the DDAR property of XSlit can be easily applied to more general non-centric cameras in the future.

5.1 Background

A single perspective image exhibits scale ambiguity: 3D objects of difference sizes can have images of an identical size under perspective projection. In photography and architecture, the forced perspective technique employs this optical illusion to make an object appear farther away, closer, larger or smaller than its actual size while preserving the aspect ratio. An example is in the film "the Lord of the Rings" where characters apparently standing next to each other would be displaced by several feet in depth from the camera. For computer vision, however, such an invariance provides little help, if not harm, to scene reconstruction.

Prior approaches on resolving the scale ambiguity range from imposing shape priors [10, 24], extracting local descriptors [56] to analyzing the vanishing points [36]. I approach the problem from a different angle: I analyze aspect ratio changes of an object with respect to its depth. Consider a frontal-parallel rectangle R of size $l_h \times l_v$ located d away from the sensor and $d > f$ where f is the camera's focal length. Under perspective projection, its image is an rectangle R' similar to R of size $[l'_h, l'_v] = \frac{f}{d-f}[l_h, l_v]$. This implies that the aspect ratio $r = l_v/l_h$ of R and R' remain the same. The property can termed as aspect-ratio invariance (ARI). ARI is an important property of perspective projection. ARI, however, no longer holds under non-centric projections, exhibiting depth-dependent aspect-ratio (DDAR).

5.2 Related Work

This chapter explores a different and previously overlooked properties of MW: the scene contains multiple objects with an identical aspect ratio or size (e.g., windows) but lie at different depths. In a perspective view, these patterns will map to 2D images of an identical aspect ratio. In contrast, I show that the aspect ratio changes with respect to depth if one adopts a non-centric or multi-perspective camera. Such imaging models widely exist in nature, e.g., a compound insect eye, reflections and refractions of curved specular surfaces, images seen through volumetric gas such as a mirage, etc. Rays in these cameras generally do

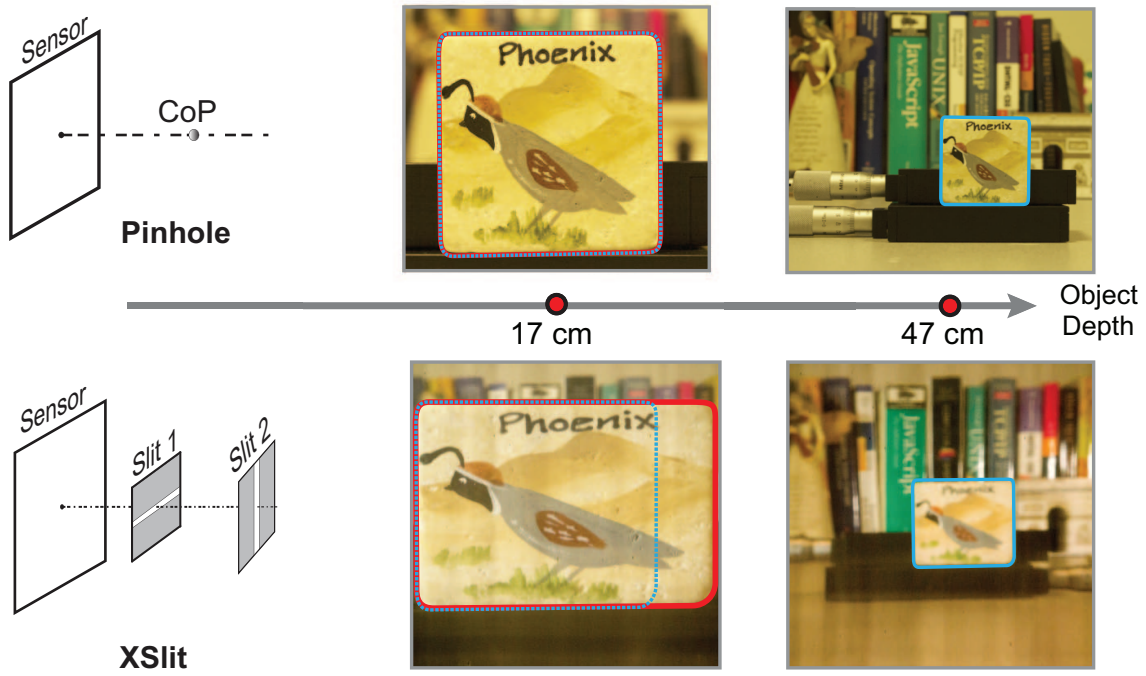


Figure 5.1: Images of the same object lying at different depths have an identical aspect ratio (AR) in a perspective camera (Top) but have very different ARs in an XSlit image (Bottom).

not pass through a common CoP and hence do not follow pinhole geometry. Consequently, they lose some nice properties of the perspective camera (e.g., lines no longer project to lines); at the same time they also gain some unique properties such as the coplanar common points [80], special shaped curves [82], etc. I focus on the depth-dependent aspect ratio (DDAR) property for inferring 3D geometry.

The special non-centric camera I employ here is the crossed-slit or XSlit camera. An XSlit camera collects rays simultaneously passing through two oblique lines (slits) in 3D space. The projection geometry of an XSlit has been examined in various forms in previous studies, e.g., as projection model in [90], as general linear constraints in [86], and as ray regulus in [62]. For long the XSlit camera has been restricted to a theoretical model as it is physically difficult to acquire ray geometry following the slit structure. The only exception is the XSlit panoramas [70, 58] where an XSlit panorama can be stitched from a translational sequence of images or more precisely a 3D light field [39]. Recently, Ye et al.[83] presented

a practical XSlit camera. Their approach relays two cylindrical lenses with perpendicular axes, each coupled with a slit shaped aperture to achieve in-focus imaging.

5.3 Depth Dependent Aspect Ratio

I first analyze how aspect ratio of an object changes with respect to its depth in an XSlit camera. I call this property Depth-Dependent Aspect Ratio or DDAR.

5.3.1 Aspect Ratio Analysis

Equation 3.5 reveals that κ_x and κ_y are projected to κ_u and κ_v with different scale on the two directions parallel to the slits. In other words, with the change of depth, the ratio will be change accordingly. Specifically, we can compute the ratio as:

$$\frac{\kappa_u}{\kappa_v} = \frac{z_2(z - z_1)}{z_1(z - z_2)} \frac{\kappa_x}{\kappa_y} \tag{5.1}$$

This is fundamentally different from the pinhole/perspective case where the ratio remains static across depth. To understand why it is the case, recall that the pinhole camera can be viewed as a special XSlit camera where the two slits intersect, i.e., they are at the same depth $z_1 = z_2$. In that case, Eqn. degenerates to $\frac{\kappa_x}{\kappa_y} = \frac{\kappa_u}{\kappa_v}$, i.e., the aspect ratio is invariant to depth.

We use $r_o = \frac{\kappa_x}{\kappa_y}$ to represent the base aspect ratio and $r_i = \frac{\kappa_u}{\kappa_v}$ represents the aspect ratio after XSlit projection. From Eqn. 5.1, we can derive the depth from the aspect ratio as:

$$z = \frac{z_1 z_2 (r_i - r_o)}{z_1 r_i - z_2 r_o} \tag{5.2}$$

5.3.2 Monotonicity:

Given a fixed XSlit camera, Eqn. 5.2 reveals that the AR monotonically decreases with respect to z. In fact, we can compute the derivative of z with respect to r_i:

$$\frac{\partial z}{\partial r_i} = \frac{z_1 z_2 (z_1 - z_2) r_o}{(z_1 r_i - z_2 r_o)^2} \tag{5.3}$$

Since $z_1 < z_2$, we have $\frac{\partial z}{\partial r_i} < 0$, i.e., the depth z decrease monotonically with r_i. In fact the minimum and the maximum ARs correspond to:

$$r_i^{\min} = r_i|_{z\to\infty} = \frac{z_2}{z_1} r_o, r_i^{\max} = r_i|_{z\to z_2} = \infty \tag{5.4}$$

5.3.3 Depth Sensitivity:

Another important we address here is depth sensitivity. We compute the partial derivative of r_i respect to z for z ranging from z_2 to ∞ and we have:

$$\frac{\partial r_i}{\partial z} = \frac{z_2(z_1 - z_2)}{z_1(z - z_2)^2} r_o \tag{5.5}$$

The sensitivity is the absolute value of $\frac{\partial r_i}{\partial z}$ and it decrease monotonically for $z > z_2$. This implies that as objects get further away, the depth accuracy recoverable from the AR also decreases. According to Eqn. 5.5, the sensitivity is positively related to $\frac{z_2}{z_1}$ and $z_1 - z_2$. Farther separated slits and greater ratio between two slits distances corresponds to higher sensitivity. This phenomenon resembles classical stereo matching using two perspective cameras where the deeper the object, the smaller the disparity and the less accuracy that stereo matching can produce.

5.3.4 Depth Range:

We can further compute the maximum discernable depth $z^{\max}$. To do so, we first compute r_i when $z \to \infty$ as $r_i^{\infty} = \frac{z_2}{z_1} r_o$. Next we change r_i^{∞} with ϵ, the smallest ratio change that is discernable in image. We have $r_i^* = \frac{z_2}{z_1} r_o + \epsilon$. The lower bound of ϵ is $1/L$, L is the image width or height, without considering subpixel accuracy. Sine the depth changes monotonically with r_i, the maximum discernable depth is correspond to r_i^*. Finally we compute the depth use Eqn. 5.2:

$$z^{\max} = \frac{z_2}{z_1}[1 + (z_2 - z_1)\frac{r_o}{\epsilon}] \tag{5.6}$$

Eqn. 5.6 indicates that the larger slit distance ratio $\frac{z_2}{z_1}$ and bigger separating distance of two slits $z_2 - z_1$ correspond to a larger discernable depth range.

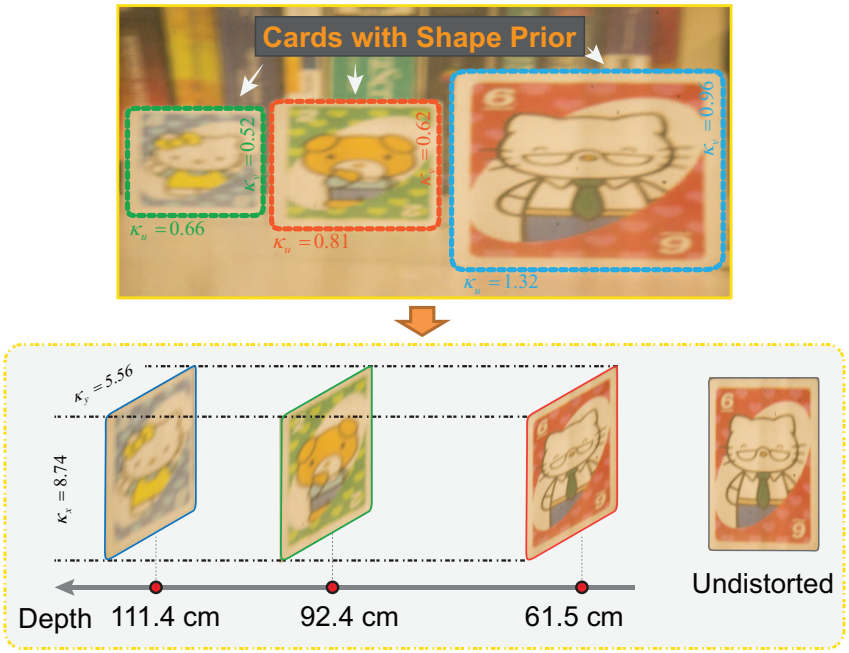


Figure 5.2: Depth-from-DDAR: Top shows a scene that contains multiple cards of an identical but unknown size. Bottom shows their recovered depths and original size using the proposed scheme from this single image.

5.4 Depth Inference

The analysis reveals that if we know r_o in prior, i.e., the base aspect of the object, we can directly infer the object's depth from its aspect ratio in the XSlit camera. A typical example is using an Parallel-Orthogonal XSlit camera (PO-XSlit) to capture an up-right rectangle. In a PO-XSlit camera, the slits are orthogonal and axis aligned. In this case, r_o directly corresponds to the aspect ratio of the rectangle and r_i corresponds to the observed AR of the project rectangle. The simplest case is to capture a up-right square whose aspect ratio $r_o = 1$. From the AR change, we can directly infer its depth using Eqn. 5.2. In practice, we do not know the AR of the object in prior. However, many natural scenes contain (rectangular) objects of identical sizes (e.g., windows of buildings) and we can infer their depth even without knowing their ground truth AR.

5.4.1 Shape Prior

Specifically, consider K rectangles of an identical but unknown sizes and hence ARs. Assume they lie at different depths z^j. According to Eqn. 3.5, we have two equations for each rectangle:

$$\begin{aligned} \kappa_u^j z^j + z_2 \kappa_x &= z_2 \kappa_u^j \\ \kappa_v^j z^j + z_1 \kappa_y &= z_1 \kappa_v^j \end{aligned} \tag{5.7}$$

Where $j = 1..K$, z^j, κ_x and κ_y are unknowns. And κ_u and κ_v are computed from the image. For K identical rectangles, we have $K + 2$ unknowns and $2K$ equations. The problem can be solved using SVD when two or more identical rectangles are present. Fig. 5.2 shows several examples using the proposed technique recovering depth of multiple cards of an identical size. The depth along with the exact scale can be extracted from a single XSlit image under the shape prior.

5.4.2 Depth Prior

If the objects are of identical aspect ratios but of different sizes, still exhibit ambiguity. Then according to Eqn. 5.1, there are K equations and $K + 1$ unknowns (assume K objects). One useful prior that can be imposed here is the distribution of depth of objects. In real scenes, objects are likely to br evenly distributed. For example, if we assume that these rectangles are with equal distance along the z direction.

In this scenario/case, we obtain the AR equation for each object:

$$z^j r_o - r_i^j \frac{z_1}{z_2} z^j - z_1 r_o = -z_1 r_i, \quad j = 1..K \tag{5.8}$$

Furthermore, the equal distance prior gives us the constraint $z^j - z^{j-1} = z^{j+1} - z^j$, for $j = 2...(K-1)$. For K objects in the scene, we have $2K-2$ equations, and $K+1$ unknowns. The problem is determined if we have 3 rectangles in the scene. And it's over-determined if we have more than 3 objects.

It is very important to note that inferring depth under the same setting is not possible in the perspective camera case. In pinhole image $z_1 = z_2$ and $r_i = r_o$, hence Eqn. 5.7

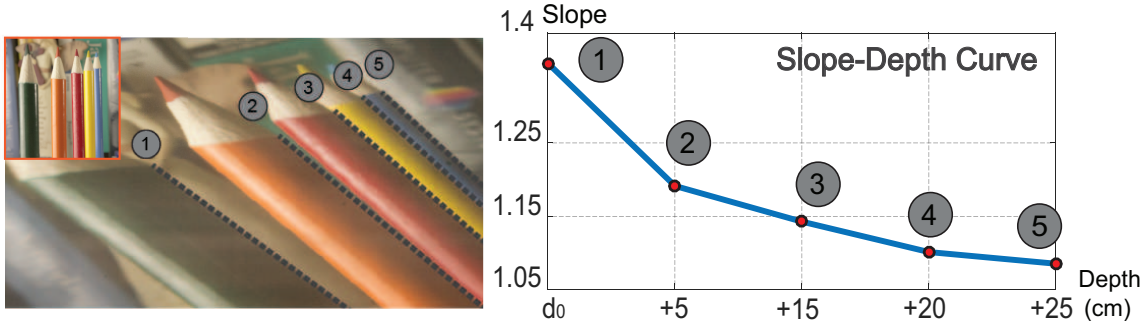


Figure 5.3: Extending DDAR to DDS. Top: parallel 3D lines map to 2D lines of different slopes in an XSlit image. Bottom: the slopes can be used to infer the depths of the lines.

and Eqn. 5.8 degenerate. As shown in the introduction, scaling the scene and adjusting the distance from the scene to the pinhole camera accordingly will result in a same projected image as the ground truth scene dose.

5.4.3 Line Slope Analysis

Section 5.4 reveals that inferring depth from DDAR is that we need to obtain some prior knowledge of either the base AR r_o or the depth distribution of multiple identities. Further, the rectangular shape needs to be in the up-right position to align with the two slits. In this section, we extend the AR analysis to study the slope of lines and we show that this analysis leads to a more effective depth inference scheme.

We treat a line frontal parallel to the XSlit camera as the diagonal of a parallelogram (rectangle in PO-XSlit case), whose sides are along the two slits directions. Given a line with slope s and a point $p_1[x_1, y_1, z]$ on it, then we have $p_2[x_1 + 1, y_1 + s, z]$ of is on the line. We can map it to a line with slope s' on XSlit image, which p_1 and p_2 map to points $p_1'(u_1, v_1)$ and $p_2'(u_1 + c, v_1 + cs')$ respectively. According to definition of r_o, we can decompose the segment p_1-p_2 onto two slits direction and take the ratio of the two component to get r_o:

$$r_o = \frac{\sin\theta_2 - s\sin\theta_2}{s\cos\theta_1 - \cos\theta_2} \tag{5.9}$$

r_i is computed using Eqn. 6.15 too, only substitute s with s'. Reuse Eqn. 5.2, we can get the depth.

Eqn. 6.15 and 5.2 reveals that we can directly infer the depth of the line from its slope. Similar to the aspect ratio case, such inference cannot be conducted in the pinhole camera since the frontal parallel line slope is invariant to depth.

The analysis above applies only to lines parallel to XSlit camera. For lines unparallel to the camera, previous studies have shown that they map to curves, or more precisely hyperbolas [82]. However, the analysis can still be applied by computing the tangent direction on the hyperbolas, where each tangent direction can be mapped to a unique depth. This can be viewed as approximating a line as piecewise segments frontal-parallel to the camera where each segment's depth can be computed from its projected slope. The complete derivation is included in the supplementary materials.

5.4.4 Scene Reconstruction

Based on the proposed theories, I present a new framework on single-image Manhattan scene reconstruction using the XSlit camera. The main idea here is to integrate depth cues from DDAR (for up-right rectangle objects) and from line slopes (for other lines and rectangles) under a unified depth inference framework. Further, the initial depth estimation scheme can only infer depths on pixels lying on the boundaries of the objects, it is important to propagate the estimation to all pixels in order to obtain the complete depth map.

The proposed approach is to first infer the depth for the lines or repeat objects from DDAR. Next I cluster pixels into small homogenous patches or superpixels [22]. The use of superpixels not only reduce the computational cost and but also preserves consistency across the regions, i.e the pixels in a homogeneous region such as walls of a building tend to have a similar depth. Finally, I model optimal depth estimation/propagtion as a Markov Random Field (MRF). The initial depth value V_i for superpixel S_i is computed by blending the depths inferred from DDAR according to their geodesic distance to S_i. And then I the smooth out V based on distance variations and color consistency. This procedure can be modeled as a Markov Random Field (MRF), where the data term: $E_d(S_i) = U_i - V_i$. And the smoothness term is: $E_s(S_i, S_j) = w_{ij}(U_i - U_j)$, w_{ij} is the weight account for distance variations and color consistency. Finally I estimate the depth map U by optimizing the energy function:

Figure 5.4: An XSlit image of the arch scene that contains 3D concentric circles (left). Their images correspond to ellipses of different aspect ratios (right).

$E(U) = \sum_{S_i} E_d(Si) + \lambda \sum_{S_i,S_j \in N} E_s(S_i, S_j)$, N represents the superpixel neighborhood. The problem can be solved using the graph-cut algorithm [9].

5.5 Experiments

I experiment the proposed approach on both synthetic and real scenes. For synthetic scenes, I render images using 3ds Max. For real scenes, I acquire images using the XSlit lens as well as synthesize XSlit panoramas from video sequences.

5.5.1 Synthetic Results.

I first render an XSlit images of a scene containing repeated shapes (Fig. 5.4). The architecture consists of concentric arches of depths ranging from 900cm to 2300cm. I assume that the actual aspect ratio of the arches is 1, i.e., a circle. I position a PO-XSlit camera with $z_1 = -3.2$cm and $z_2 = -346.7$cm frontal parallel to the arches and the images of the arches are ellipses of different aspect ratios. Notice that in the pinhole case, they will be map to circles. I first detect ellipses using Hough transform and then measure their aspect ratios using the major and minor axes. Finally, I use the ratios to recover their depths using

Eqn. 5.2. The recovered depths for the near and far arches are 906.6cm and 2281.0cm, i.e., the errors are less than 2%.

Next I render two XSlit panoramas, one for the corridor and the second for the facade. Both scenes exhibit strong linear structures with many horizontal and vertical lines. Our analysis shows that for lines to exhibit DDS, they should not align with either slit. Therefore, I rotate the POXSlit, i.e., $\theta_1 = 45°$ and $\theta_2 = 135°$. For the corridor scene, the XSlit camera has a setting of $z_1 = -3.6$cm, $z_2 = -717.9$cm and for the facade scene, $z_1 = -3.1$cm, $z_2 = 4895.9$cm. I first use the LSD scheme[77] to extract 2D lines from the XSlit images and cluster them into groups of horizontal and vertical (in 3D) lines. This is done by thresholding their aspect ratios Eqn. 5.4. For lines in each group, I compute their depths using Eqn. 6.15 and 5.2. This results in a sparse depth map. To recover the full depth map, I apply the MRF (Sec. 5.4.4) and the final result is shown in Fig. 5.6. My technique is able to recover different depth layers while preserving linear structures. For comparison, I render a single perspective image and apply the learning-based scheme Make3D [65]. Make3D can detect several coarse layers but cannot detect fine details as mine since these linear structures appear identical in slope in a perspective image but exhibit different slopes in an XSlit image.

5.5.2 Real Results.

I explore several approaches to acquire XSlit images of a real scene: by a real XSlit lens and through panorama synthesis. For the former, I use an XSlit lens [82]. The design resembles the original anamorphoser proposed by Ducos du Hauron that replaces the pinhole in the camera with a pair of narrow, perpendicularly crossed slits. Similar to the way of using a spherical thin lens to increase light throughput in a pinhole camera, the XSlit lens relay perpendicular cylindrical lenses, one for each slit. In my experiments, I use two cylindrical lenses with focal lengths 2.5cm (closer to the sensor) and 7.5cm (farther away from the sensor) respectively. The distance between the two slits is adjustable between 5cm and 12cm and the slit apertures have a width of 1mm.

I first capture a checkerboard at known depths and compare the measured AR and our predicted AR using Eqn. 5.2. I test three different slit configurations, $z_2/z_1 = 1.3$,

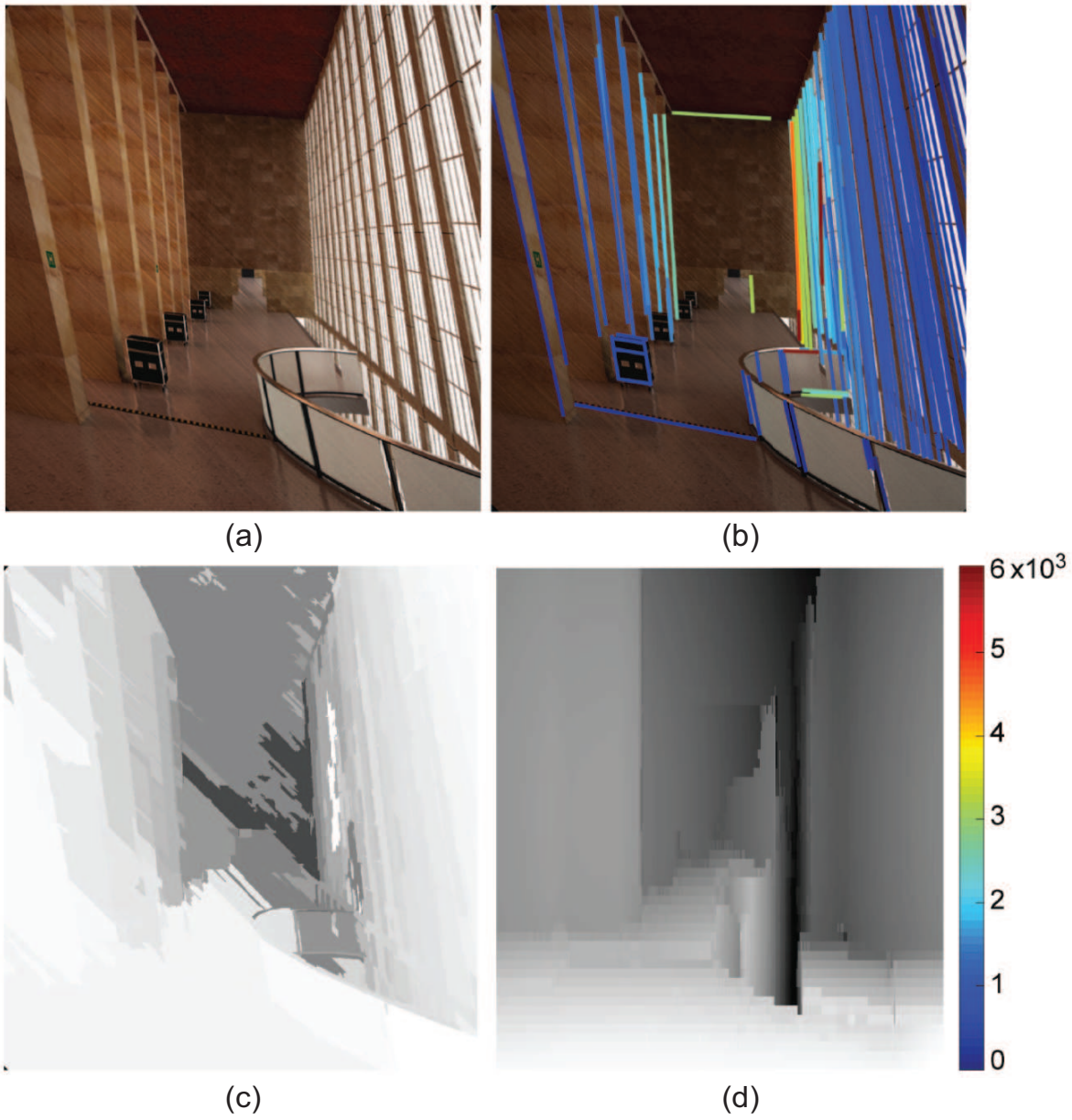


Figure 5.5: (a) An XSlit image of a scene containing parallel 3D lines, (b) the detected lines and their estimated depth using DDS, (c) the depth map recovered using our scheme, and (d) the one recovered using Make3D [65] by using a single perspective image.

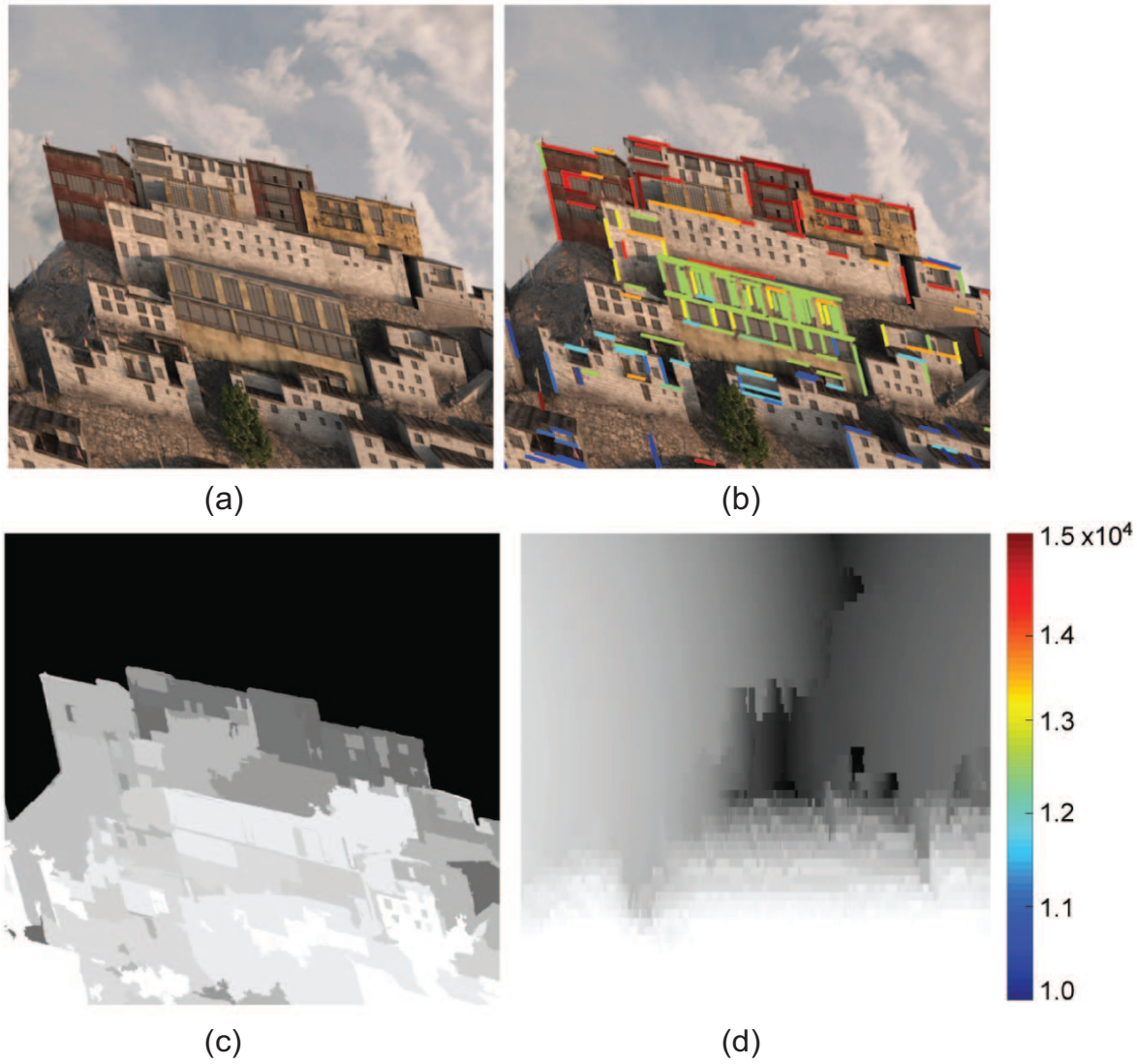


Figure 5.6: (a) An XSlit image of a scene containing parallel 3D lines, (b) the detected lines and their estimated depth using DDS, (c) the depth map recovered using our scheme, and (d) the one recovered using Make3D [65] by using a single perspective image.

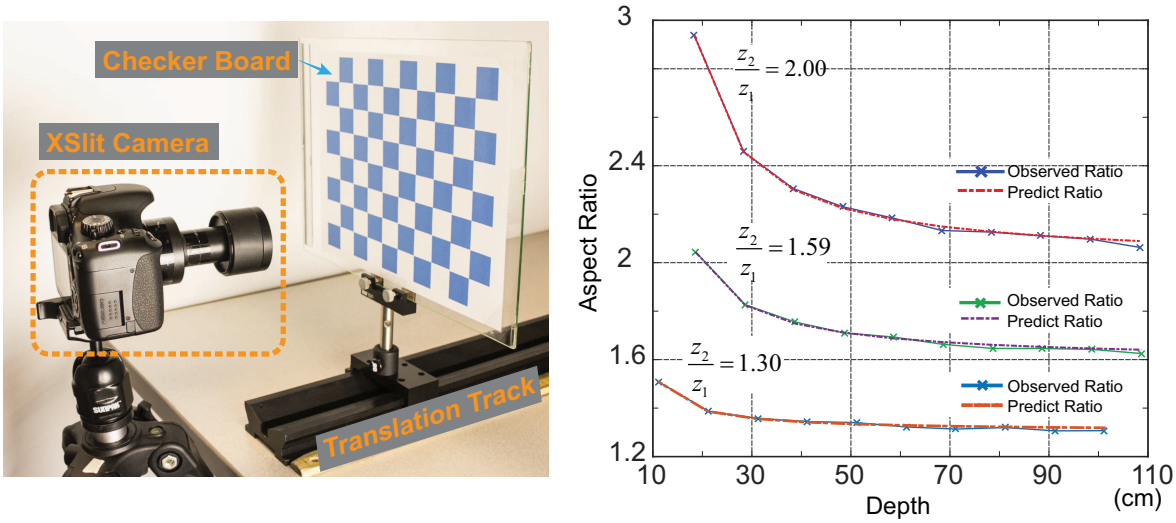


Figure 5.7: Experimental validations of the analysis. I place checker board in front of the XSlit camera and move it away(Left). The comparisons of measured AR and predict AR with different silts configurations(Right).

$z_2/z_1 = 1.59$ and $z_2/z_1 = 2.0$. Fig. 5.7 shows that the predicted AR curve fits well with the ground truth. In particular, as an object gets farther away from the sensor, its AR also changes slower. Further, the larger the baseline z_2/z_1 is, the larger the aspect ratio variations across the same depth range, as predicted by the theory.

Next, I verify our DDS analysis using images captured the XSlit camera. In Fig. 5.8, I position a Lego® house model in front of the XSlit camera ($z_1 = 6.12$cm and $z_2 = 11.81$cm). I rotate the XSlit camera by 45 degrees so that the 3D lines on the house will not align with either slit. Fig. 5.8(a) shows the acquired image. Next, I conduct line fitting and slope estimation similar to the synthetic case for estimating the depths of the detected lines. Fig. 5.8(a) highlights the detected lines and their depths (using color) and Fig. 5.8(b) shows the complete depth map using the MRF solution. The results shows that major depth layers are effectively recovered. The error on the top-right corner is caused by the lacking of line structures.

A major limitation using the XSlit camera is its small baseline (between the two slits). My analysis shows that the maximum recoverable depth range depends on this baseline. Further, since images captured by the XSlit camera exhibits noise and strong defocus blurs, the actual recoverable depth range is even smaller. For example, my analysis shows that

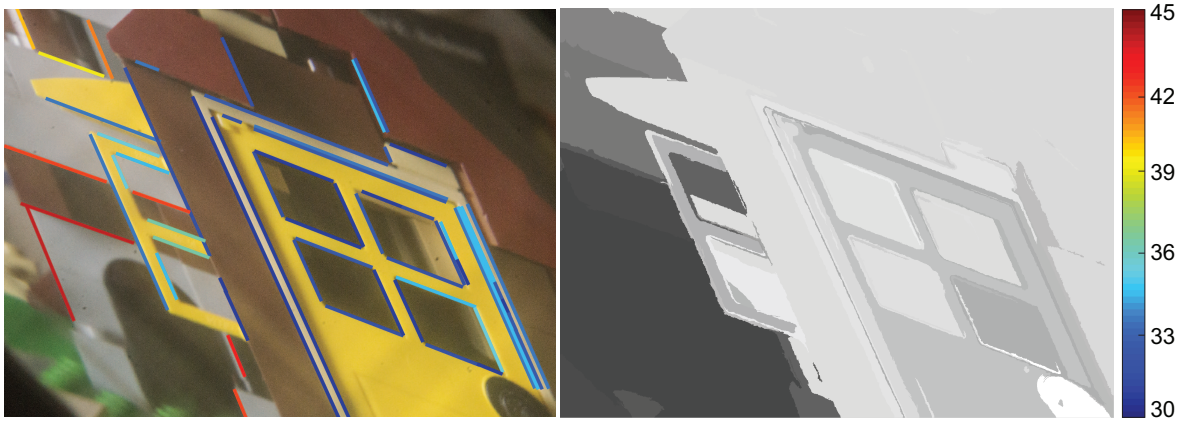


Figure 5.8: Real result on a Lego® house scene. (a) an XSlit image of the scene captured by the XSlit camera. Detected lines are highlighted in the image. (b) the recovered depth map using our slope and aspect ratio based scheme.

with baseline $z_2/z_1 = 2$, two cards are placed at $30m$ and $35m$ will have undistinguishable ARs. Their ratio difference reach the lower bound that determined by pixel size. For outdoor scenes, I resort to XSlit panorama synthesis.

To produce XSlit panoramas, Zomet et al. [90] capture a sequence of images by translating a pinhole camera along a linear trajectory at a constant velocity. In a similar vein, Seitz and Adams et al. acquire the image sequence by mounting the camera on a car facing towards the street. Additional registration steps [6] can be applied to rectify the input images. Next, linearly varying columns across the images are selected and stitched together. Fig. 3.4 shows the procedure of generating a XSlit image using a regular camera.

Fig. 5.9 shows the XSlit panorama synthesized from an image sequence captured by a moving camera. I linearly increase the column index in terms of frame number and stitch these columns to form an XSlit image. The moving path of the camera is 55cm long. And the camera is tilt with 20° angle. The resulting two slits are at -1.8cm and 41cm respectively.

Recent ray geometry studies [20] show that reflections off certain types of catadioptric mirror can be approximated as an XSlit image. In Fig. 5.11, I position a perspective camera facing towards a cylindrical mirror and Fig. 5.11(b) shows that DDAR can both be observed on the acquired image. In particular, I put multiple cubes of an identical size at

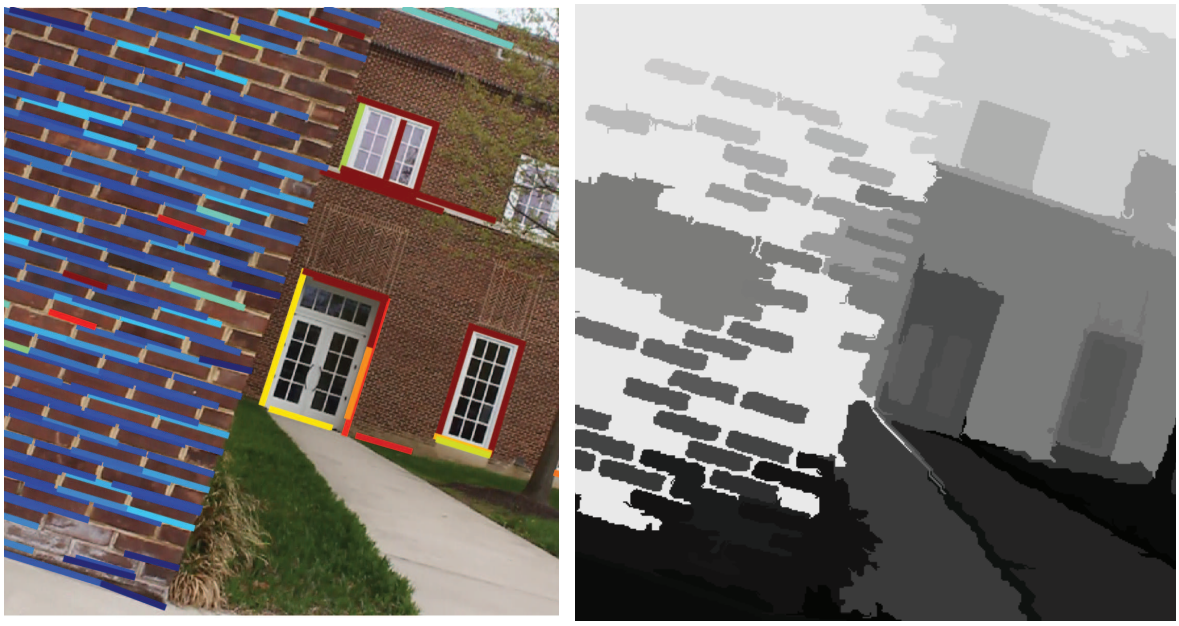


Figure 5.9: The XSlit image of an outdoor scene. Left: An XSlit panorama and the detected lines. Right: The recovered depth map.

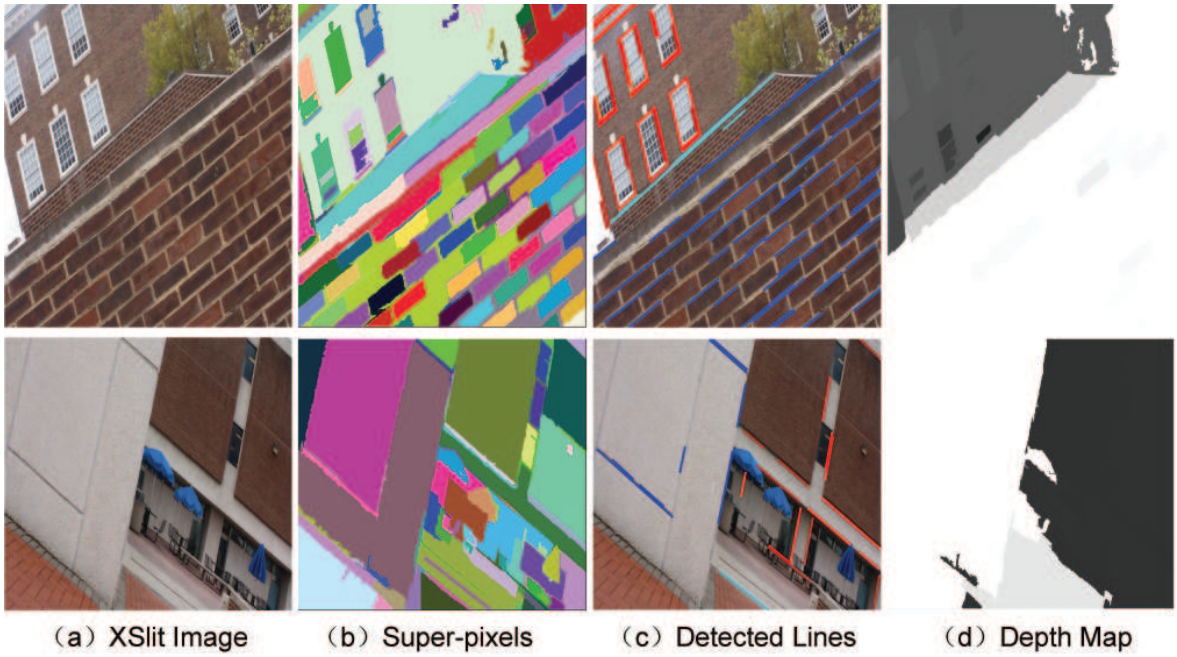
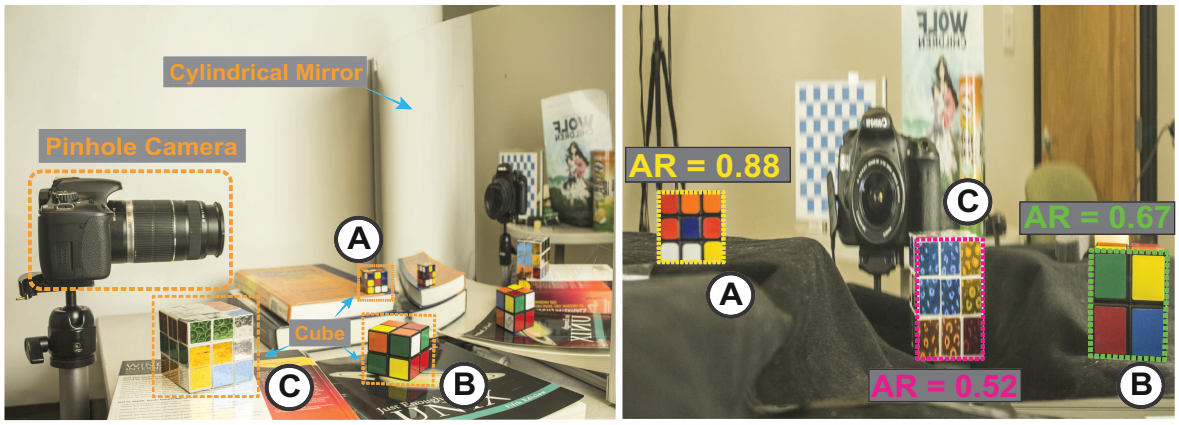


Figure 5.10: More result of depth reference for XSlit panoramas



(a) Scene Setup (b) Captured Image

Figure 5.11: Results on catadioptric mirrors. Left: I capture the scene using a cylindrical catadioptric mirror. Right: the aspect ratios of cubes change with respect to their depthes.

different depths and their aspect ratios change dramatically. This is because two virtual slits of the catadioptric mirror are separated far away where DDAR is more significant than the XSlit camera case.

5.5.3 Discussion

I have comprehensively studied the aspect ratio (AR) distortion in XSlit cameras and exploited its unique depth-dependent property for 3D inference. The studies have shown that unlike perspective camera that preserves AR under depth variations, AR changes monotonically with respect to depth in an XSlit camera, i.e., 3D objects of an identical size will exhibit significantly different AR under different depths. This has led to new depth-from-AR schemes using a single XSlit image even if the original AR of an object is unknown. I have further shown that similar to AR variations, the slope of projected 3D lines will also vary with respect to depth, and I have developed theories to characterize such variations based on AR analysis. Finally, AR and line slope analysis can be integrated for 3D reconstruction and I have experimented on real XSlit images captured by an XSlit camera, synthesized from panorama stitching, and captured using a catadioptric mirror to validate the proposed framework.

Chapter 6

XSLIT STRUCTURE FROM MOTION

Perspective camera emulates human eyes whereas multi-perspective models are more common in insect eyes (e.g., fly's compound eyes). Although classical computer vision problems, such as structure-from-motion (SfM), have been well studied using perspective camera, little attention has been paid to multi-perspective cameras. In this chapter, I study the SfM problem using a generalized multi-perspective camera, the crossed-slit (XSlit) camera. I demonstrate that XSlit SfM can automatically avoid scale ambiguity due to depth-dependent distortions. To conduct SfM, I first derive the fundamental matrix in XSlit images. To address non-linearity and handle distortions in XSlit images, I further develop a novel feature matching algorithm based on non-uniform Gaussian kernels. Finally, I propose a novel error metric based on depth-dependent aspect ratio for bundle adjustment to refine the estimated camera poses. Experiments demonstrate that the proposed XSlit-based SfM approach can reliably estimate camera motion and scene geometry while avoiding ambiguity.

6.1 Background

A perspective camera collects rays passing through a common 3D point (i.e., the CoP) and its images resembles what would be seen by human eyes. However, this model is rare in insect eyes. For example, many insects have compound eyes, which consist of thousands of individual eye units or ommatidia. These ommatidia are located on a convex surface and viewing towards different directions. Compound eyes hence have a very large field-of-view that greatly help detect fast movement. Notice that a compound eye does not have a common CoP and no longer follows the perspective camera model. Instead, they follow the *multi-perspective* model that combines rays from different viewpoints. Despite the incongruity of view, a multi-perspective image is able to preserve spatial coherence and

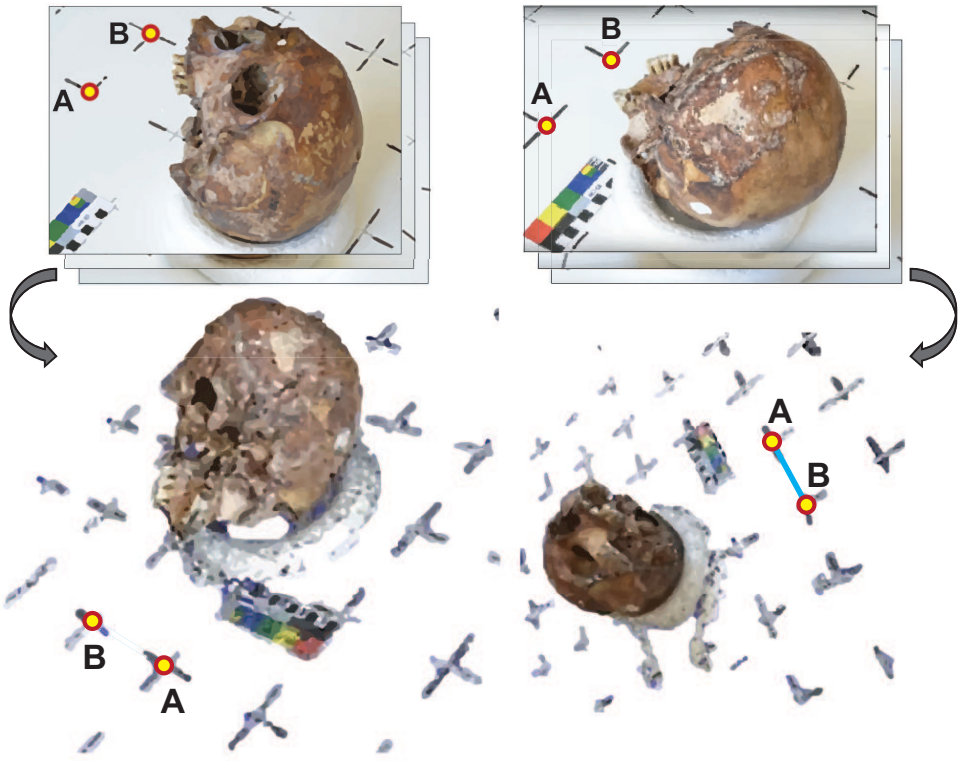


Figure 6.1: The scale ambiguity in traditional SfM introduce align problem: the front and back side of the skull are reconstructed individually and hence have different scale, it's a common practice to use additional marks for alignment.

depict details of a scene that are simultaneously inaccessible from a single view within a single context.

Structure-from-Motion (SfM) is the problem that recovers both the camera motion and 3D scene geometry from a sequence of images [44]. The problem is well studied using the motion of a perspective camera. However, little attention is paid to using multi-perspective cameras. It is well known that compound eyes have much higher motion sensitivity than human eyes. Therefore, studying SfM using multi-perspective cameras may lead to new classes of algorithms that can apply to fast moving objects. Early works on multi-perspective stereo matching lay out the theoretical foundation for studying SfM using

multi-perspective camera. The seminal work of Seitz [70] characterizes all possible multi-perspective stereo pairs and concludes that the epipolar geometry, if exists, has to be a doubly ruled surface. Pajdla [57] independently reached similar results. Pless [60] further derives the Generalized Epipolar Constraint (GEC) for generic camera models. To characterize the properties of multi-perspective cameras, Yu et al.[86] propose the General Linear Camera (GLC) and conclude that many multi-perspective cameras, such as catadioptric cameras, are special cases of the XSlit because the two slits provide a special set of surface ruling that determines the ray manifold of the local GLC. I hence adopt the XSlit as a generalized multi-perspective camera to study the SfM problem. The proposed solution developed using XSlit can be extended to a broad range of multi-perspective cameras.

Performing SfM using an XSlit camera, however, is challenging for two reasons: 1) camera poses are difficult to estimate since the XSlit projection is non-linear and a projection matrix from the perspective case cannot be generalized; and 2) feature matching is non-trivial in XSlit images due to distortions. In this chapter, I develop a novel and robust XSlit SfM framework that can estimate both camera poses and 3D scene geometry at *an absolute scale*. I first show that, similar to the perspective case, there exists a fundamental matrix to correlate two XSlit images captured at different poses. I further reduce the degree of freedom in the fundamental matrix such that absolute translation and rotation matrices can be solved from a linear system. I then develop a robust feature matching algorithm for XSlit images by applying multiple non-uniform Gaussian kernels to sample the affine SIFT feature space to mitigate XSlit distortions. Finally, I propose a novel error metric based on depth-dependent aspect ratio for bundle adjustment to iteratively refine the estimated camera parameters and scene geometry. It is worth noting that the depth-dependent distortions in XSlit images enables the algorithm to automatically avoid the scale ambiguity that plagues SfM in the perspective case. Synthetic and real experiments demonstrate that the proposed XSlit-based SfM approach can estimate the camera motions and the scene geometry with an absolute scale with high fidelity and reliability.

6.2 Related Work

The work is closely related to SfM and multi-perspective (especially the XSlit) stereo matching. In this section, I discuss the most relevant works.

SfM is a well-studied problem in computer vision and great success has been achieved in robotics [16], autonomous navigation [50], large-scale 3D reconstruction [5, 72] etc. Most existing works are performed using a perspective camera. I refer the readers to [44] for a comprehensive survey. It is well known that SfM via a perspective camera suffers from the scale ambiguity [28]. This is because the perspective projection is subjective to a scale factor: objects of different sizes can map to the same perspective images with identical scale by placing the objects at different depths. Standard approach for resolving the scale ambiguity is to use a stereo camera setup with known baseline [54, 16] where the scale factor is determined by triangulating feature points in the stereo pair. Clipp et al.[13] recover scale by tracking features on two non-overlapping cameras. For a single perspective camera case, constraints or priors on either the camera motion or the scene geometry has to be imposed to recover the scale. Scaramuzza et al.[66] use the camera-to-ground distance to keep track of the camera motion and estimate the scale. Davison et al.[16] use a pattern of known size to compute the absolute scale of the entire scene. Pollefeys et al.[61] adopt an additional GPS sensor to acquire exact dimension. I found that by using a multi-perspective camera, the scale ambiguity is automatically resolved due to depth-dependent distortions.

There have been significant advances on the theory of multi-perspective stereo in the past decade. Seitz [70] and Pajdla [21] independently studied all possible multi-perspective stereo pairs that can have valid eipolar geometry. Sturm [70] analyzed the multi-view geometry in general multi-perspective camera. Ding and Yu [19] introduced a new near stereo model called epsilon stereo pairs for multi-perspective images that do not satisfy the epipolar geometry constraints. Pless[60] derived the Generalized Epipolar Constraint (GEC) in generic cameras for motion estimation. Hee et al.[29] applied GEC on a multi-camera setup. Mičušík and Pajdla [48] studied the calibration of para-catadioptric camera by applying the multi-perspective epipolar geometry. However, using multi-perspective cameras for SfM is more challenging than stereo matching because the camera parameters also need to be

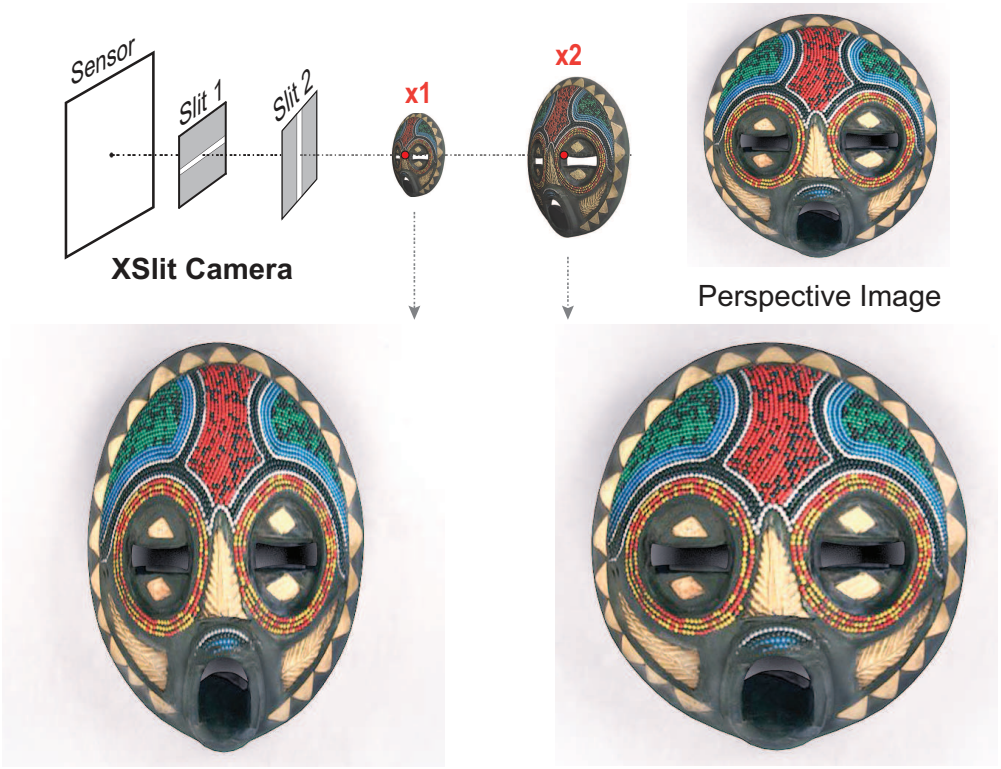


Figure 6.2: Unlike the perspective camera, objects at different depths are distorted differently in an XSlit image.

estimated on the fly.

Yu et al.[86] used the General Linear Camera (GLC) to model a broad range of multi-perspective cameras and their result reveal that many multi-perspective cameras are special cases of the XSlit. An XSlit camera collects rays passing through two oblique (neither coplanar nor parallel) slits in 3D. Zomet et al.[90] derived a close-form projection model for the XSlit. Ponce [62] proposed ray regulus to model the camera ray space. The proposed work is also related to XSlit stereo and 3D reconstruction. Feldman et al.[21] proved that a pair of XSlit cameras can have valid epipolar geometry if they share a slit or the slits intersect in four pairwise distinct points. Ye et al.[83] constructed a new form of stereo pair by rotating the slits. Ye et al.[82] used line curvatures in XSlit images for Manhattan scene reconstruction. More recently, Li et al.[41] adopted XSlit camera to sample the 4D light field.

6.3 Camera Pose Estimation

In this section, I show how to estimate the camera poses under XSlit projection in order to solve SfM. I perform the analysis using the ray space geometry. I first demonstrate that similar to the perspective case, there exists a fundamental matrix to correlate two XSlit images taken at different viewpoints. I then further reduce the degree of freedom in the fundamental matrix by applying the XSlit constraints and solve the viewpoint transformation from a linear system.

6.3.1 XSlit Fundamental Matrix

Given a reference XSlit image $\mathbb{X}$ and a target XSlit image $\mathbb{X}'$ captured at a different viewpoint, our goal is to align $\mathbb{X}'$ to $\mathbb{X}$ via a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$. Let's consider a 3D scene point $\mathbf{P}$. As shown in Fig. 6.3, $\mathbf{P}$ is projected into $\mathbb{X}$ and $\mathbb{X}'$ by two rays $\mathbf{r}[u, v, \sigma, \tau]$ and $\mathbf{r}'[u', v', \sigma', \tau']$ respectively. Assume the world coordinate is set under the reference image $\mathbb{X}$, we first transform $\mathbf{r}'$ into the world coordinate as $\mathbf{r}^*[u^*, v^*, \sigma^*, \tau^*]$. Since $\mathbf{r}$ and $\mathbf{r}^*$ pass through a common 3D point $\mathbf{P}$, their ray coordinates satisfy a bilinear constraint [87]: $\frac{u - u^*}{v - v^*} = \frac{\sigma - \sigma^*}{\tau - \tau^*}$. It's vector form can be written as:

$$\mathbf{d}^\mathsf{T} \cdot \mathbf{m}^* + \mathbf{m}^\mathsf{T} \cdot \mathbf{d}^* = 0 \tag{6.1}$$

where $\mathbf{d} = [\sigma, \tau, 1]^\mathsf{T}$, $\mathbf{m} = [-v, u, \chi]^\mathsf{T}$, $\chi = v\sigma - u\tau$ and $\mathbf{d}^* = [\sigma^*, \tau^*, 1]^{\mathsf{T}^*}$, $\mathbf{m}^* = [-v^*, u^*, \chi^*]^\mathsf{T}$, $\chi^* = v^*\sigma^* - u^*\tau^*$.

Similarly, we define $\mathbf{d}'$ and $\mathbf{m}'$ for $r'[u', v', \sigma', \tau']$. Since the two image coordinates in $\mathbb{X}$ and $\mathbb{X}'$ are correlated by transformation matrices $\mathbf{R}$ and $\mathbf{t}$, the relationship between $\mathbf{d}'$, $\mathbf{m}'$ and $\mathbf{d}^*$, $\mathbf{m}^*$ can be derived as:

$$\mathbf{d}^* = \mathbf{R} \cdot \mathbf{d}', \quad \mathbf{m}^* = \mathbf{R} \cdot \mathbf{m}' - [\mathbf{t}]_\times \mathbf{R} \cdot \mathbf{d}' \tag{6.2}$$

By substituting Eqn. 6.2 into Eqn. 6.1, we have:

$$\begin{bmatrix} \mathbf{d}^\mathsf{T} & \mathbf{m}^\mathsf{T} \end{bmatrix} \mathbf{F}_{6\times 6} \begin{bmatrix} \mathbf{d}' \\ \mathbf{m}' \end{bmatrix} = 0 \tag{6.3}$$

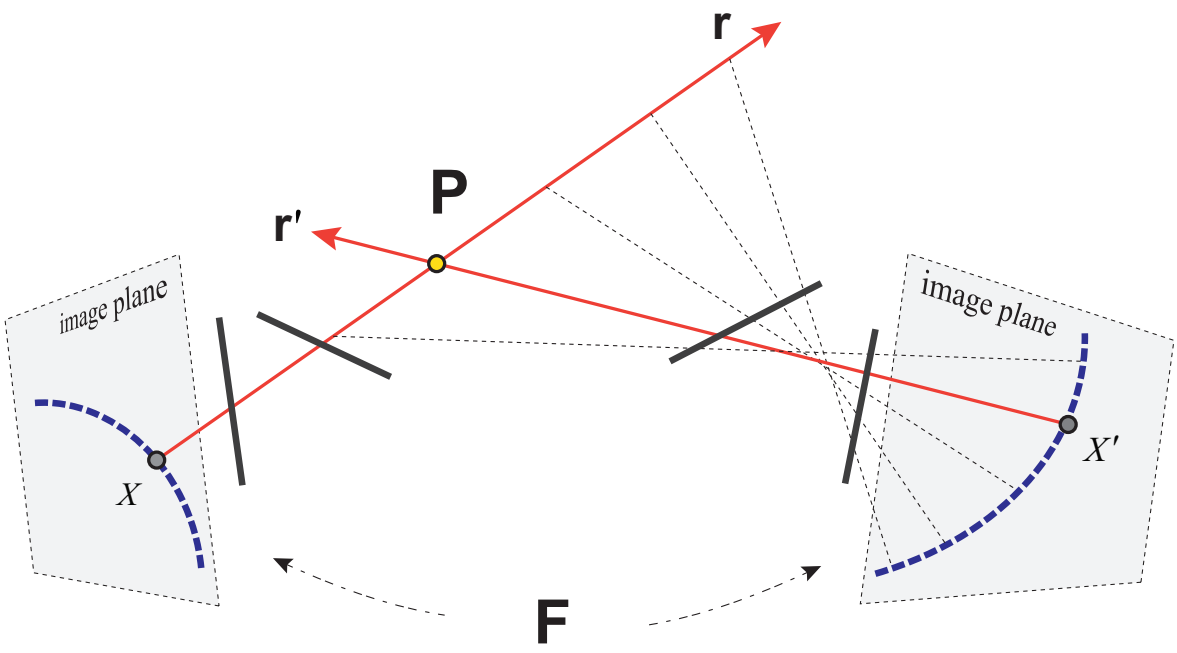


Figure 6.3: XSlit images captured from different viewpoints are correlated by a fundamental matrix $\mathbf{F}$.

where $\mathbf{F} = \begin{bmatrix} -[\mathbf{t}]_\times \mathbf{R} & \mathbf{R} \\ \mathbf{R} & 0 \end{bmatrix}$

Since both $\mathbf{d}$ and $\mathbf{m}$ are uniquely determined by image pixel coordinate $[u, v]$, Eqn. 6.3 indicates that there exists a fundamental matrix $\mathbf{F}$ similar to the perspective case to register two XSlit images captured at different viewpoints. $\mathbf{F}$ is a 6×6 matrix. A linear solution may treat $-[\mathbf{t}]_\times \mathbf{R}$ and $\mathbf{R}$ as two independent unknowns and suggest solving it with 17 pair of correspondences. However, there are ambiguities in the linear equation system because of the enforced XSlit ray constraint. Non-linear methods can not solve the ambiguity either.

In order to reduce the degree of freedom in $\mathbf{F}$, we apply the XSlit constraints (Eqn. 3.4) to decompose $[\mathbf{d}\ \mathbf{m}]^\mathsf{T}$ into two matrices:

$$\begin{bmatrix} \mathbf{d} \\ \mathbf{m} \end{bmatrix} = K p^\mathsf{T} \tag{6.4}$$

where $K = \begin{bmatrix} 0 & -\mathsf{B} & \mathsf{A} & 0 \\ 0 & -\mathsf{D} & \mathsf{C} & 0 \\ \hdashline & & I_{4\times4} & \end{bmatrix}$ and $p = [1, -v, u, \chi]$. Notice that K is only related to the two slits' configuration and we call it the XSlit intrinsic matrix, while p is determined by pixel coordinates $[u, v]$.

By substituting K and p into Eqn. 6.3, we can rewrite the equation as follow:

$$p^{\mathsf{T}}\tilde{\mathbf{F}}p' = 0, \quad \text{where } \tilde{\mathbf{F}} = K^{\mathsf{T}}\mathbf{F}K \tag{6.5}$$

Since p is a 1×4 vector, our new fundamental matrix $\tilde{\mathbf{F}}$ is a 4×4 matrix with its last element to be zero. As a result, we are able to solve the unknown elements in $\tilde{\mathbf{F}}$ with 14 pairs of corresponding points between $\mathbb{X}$ and $\mathbb{X}'$ by applying SVD.

Pless[60] studied the multi-view geometry for generic cameras under the Plücker coordinate. The Generalized Epiplor Constraint (GEC) is derived to characterize the epipolar geometry. Our derivation of XSlit fundamental matrix performed in the ray space is consistent with GEC.

6.3.2 Pose Transformation Estimation

Once we have the fundamental matrix $\tilde{\mathbf{F}}$, we can use it to solve the camera pose transformation matrices $\mathbf{R}$ and $\mathbf{t}$. However, we cannot solve $\mathbf{R}$ and $\mathbf{t}$ directly from Eqn. 6.5 since the XSlit intrinsic matrix K is under-determined and cannot be inverted. To address this problem, we apply the QR matrix decomposition on K and convert K into the multiplication of an orthogonal matrix $\mathcal{Q}$ and an upper triangular matrix $\mathcal{R}$, i.e., $K = \mathcal{Q}\mathcal{R}$. Substituting K into Eqn. 6.5, we have

$$\hat{\mathbf{F}} = \mathcal{R}_4^{-\mathsf{T}} * \tilde{\mathbf{F}} * \mathcal{R}_4^{-1} = \left(\mathcal{Q}^{\mathsf{T}} \begin{bmatrix} -[\mathbf{t}]_\times \mathbf{R} & \mathbf{R} \\ \mathbf{R} & 0 \end{bmatrix} \mathcal{Q} \right)_4 \tag{6.6}$$

where the subscript 4 means 4×4 sub-matrix from the lower left corner. By substituting K's formulation in Eqn. 6.4, we can rewrite Eqn. 6.6 as

$$\hat{\mathbf{F}} = \begin{bmatrix} \mathrm{M} & \mathrm{V} \\ \mathrm{U} & 0 \end{bmatrix} = \begin{bmatrix} \mathcal{Q}_1^{\mathsf{T}} & \mathcal{Q}_3^{\mathsf{T}} \\ 0 & \mathsf{e}_3^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} -[\mathbf{t}]_\times \mathbf{R} & \mathbf{R} \\ \mathbf{R} & 0 \end{bmatrix} \begin{bmatrix} \mathcal{Q}_1 & 0 \\ \mathcal{Q}_3 & e_3 \end{bmatrix} \tag{6.7}$$

where M is a 3×3 sub-matrix of $\hat{\mathbf{F}}$, U and V are 3×1 vectors, $\mathcal{Q}_1$ and $\mathcal{Q}_3$ are the 3×3 sub-matrices of $\mathcal{Q}$, and $\mathsf{e}_3 = [0, 0, 1]^{\mathsf{T}}$. From Eqn. 6.7, we can derive the following constraints on $\mathbf{R}$ and $\mathbf{t}$:

$$\mathrm{U} = \mathsf{e}_3^{\mathsf{T}} \cdot \mathbf{R}\mathcal{Q}_1 \tag{6.8a}$$

$$\mathrm{V} = \mathcal{Q}_1^{\mathsf{T}}\mathbf{R} \cdot \mathsf{e}_3 \tag{6.8b}$$

$$\mathrm{M} = \mathcal{Q}_1^{\mathsf{T}}(-[\mathbf{t}]_\times \mathbf{R})\mathcal{Q}_1 + \mathcal{Q}_3^{\mathsf{T}}\mathbf{R}\mathcal{Q}_1 + \mathcal{Q}_1^{\mathsf{T}}\mathbf{R}\mathcal{Q}_3 \tag{6.8c}$$

From Eqn. 6.8b and Eqn. 6.8a, we have:

$$\mathbf{R} \cdot [\mathcal{Q}_1^{-\mathsf{T}}\mathrm{U}^{\mathsf{T}} \times \mathsf{e}_3] = \mathsf{e}_3 \times \mathcal{Q}_1^{-\mathsf{T}}\mathrm{V} \tag{6.9}$$

Combining Eqn. 6.8b, Eqn. 6.8a and Eqn. 6.9, we can solve the rotation matrix $\mathbf{R}$. Then we can solve the translation matrix $\mathbf{t}$ from Eqn. 6.8c by plugging in $\mathbf{R}$.

Our derivation can also be degenerated to formulate the perspective projection and explain why there exists the scale ambiguity. In the case of perspective camera, the high order element $\chi = v\sigma - u\tau = 0$. Hence Eqn. 6.8b and Eqn. 6.8a becomes independent. Eqn. 6.8c will be degenerated and becomes $\mathrm{M} = \mathcal{Q}_1^{\mathsf{T}}(-[\mathbf{t}]_\times \mathbf{R})\mathcal{Q}_1$. Therefore, we can only recover the direction of $\mathbf{t}$ and the absolute translation vector cannot be computed. By using a multi-perspective camera, this scale ambiguity is automatically resolved.

6.4 XSlit Feature Matching

In order to obtain correspondences for camera pose estimation, it is critical to find robust and invariant features in XSlit images. An XSlit image, however, exhibits various distortions, making it difficult for correspondence matching.

Many invariant features, such as SIFT [43], Harris and Hessian Affine [49], and MSER [45] are designed on perspective images. All these feature detectors cannot handle the XSlit distortions properly. The only exception is the Affine SIFT (ASIFT) feature

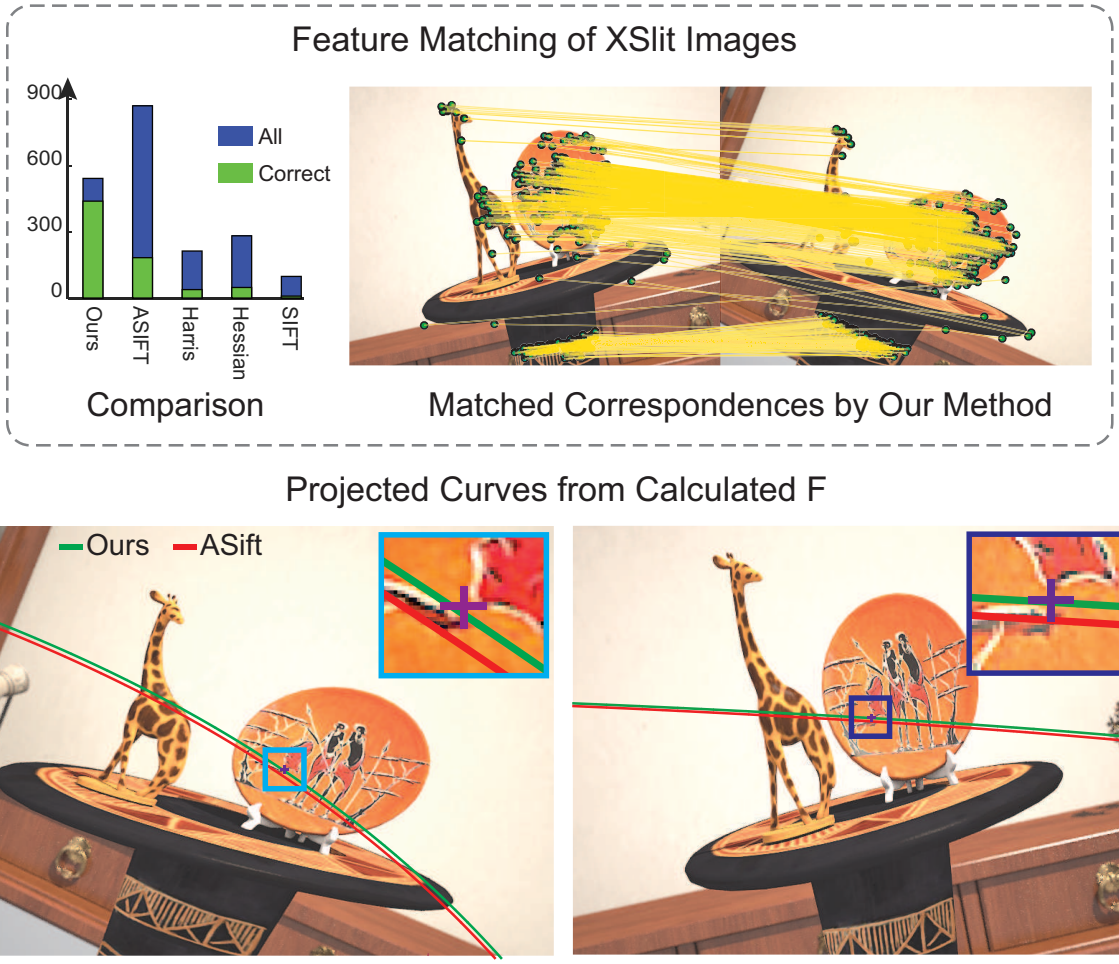


Figure 6.4: Comparison of feature matching methods on a pair of XSlit images. Top left: the total and correct numbers of matches reported by feature detectors. Top right: matched correspondences by our method. Bottom: Projected curves using the fundamental matrix calculated from correspondences produced by our method and ASift.

[51]. ASIFT can generate substantial amount of correspondences when applied on XSlit images since the affine transformation can approximate local XSlit distortion. However, there still exists a lot of mismatches.

To properly handle the distortions in XSlit images, we develop a new feature matching algorithm based on non-uniform Gaussian kernels. Similar to ASIFT, we sample the SIFT feature under different subspaces in order to undistort the XSlit image patch. However, what is different though, we use non-uniform Gaussian kernels to sample the subspaces instead of the perspective warping that is used in ASIFT.

Specifically, an affine transformation can be defined by a rotation angle θ, a shear factor s and a scale factor r:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & s_x \\ s_y & 1 \end{bmatrix} \begin{bmatrix} r_x & 0 \\ 0 & r_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6.10}$$

We then apply affine transformation matrix (Eqn. 6.10) to a Gaussian kernel to obtain non-uniform Gaussian kernels g as

$$g(x, y, \sigma, \theta, s, r) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \tag{6.11}$$

Finally, we apply the non-uniform Gaussian kernels g to transform the XSlit image patch I to feature subspaces.In our feature subspace, various distortions can be properly compensated. We can then perform SIFT feature detection in the subspace. However, mismatched correspondences can still occur occasionally. To eliminate the mismatched features, we perform a bi-directional search for valid correspondences. In particular, we first perform feature matching from the reference image to the target image and we then reverse the search direction and match features from the target to the reference. We only keep those correspondences that exist in both rounds.

Fig. 6.4 illustrates the effectiveness our feature matching algorithm in comparison with the state-of-the-arts. The top-left bar chart shows that our algorithm outperforms other methods in terms of both number of detected feature points and mismatch rate. Although ASIFT detects more feature points than our algorithm, its mismatch rate is extremely high. We also show the "epipolar curves" from the fundamental matrix calculated from our features

and ASIFT features. Our curves apparently establish more accurate correspondences and we can achieve sub-pixel accuracy.

6.5 Scene Reconstruction

Once we have correspondences and camera poses, we can recover the 3D scene geometry (i.e., 3D point cloud) by triangulating the camera projection rays. Let's consider a 3D point $\mathbf{P}[x, y, z]$, in a camera view with pose transformation matrices $\mathbf{R}$ and $\mathbf{t}$. Assume $\mathbf{P}$ is projected by ray $\mathbf{r}[u, v, \sigma, \tau]$ onto the image plane. The point projection in XSlit can be formulated as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \lambda \cdot \mathbf{R}^{-1} \begin{bmatrix} \sigma \\ \tau \\ 1 \end{bmatrix} = \mathbf{R}^{-1} \cdot \left(\begin{bmatrix} u \\ v \\ 0 \end{bmatrix} - \mathbf{t} \right) \tag{6.12}$$

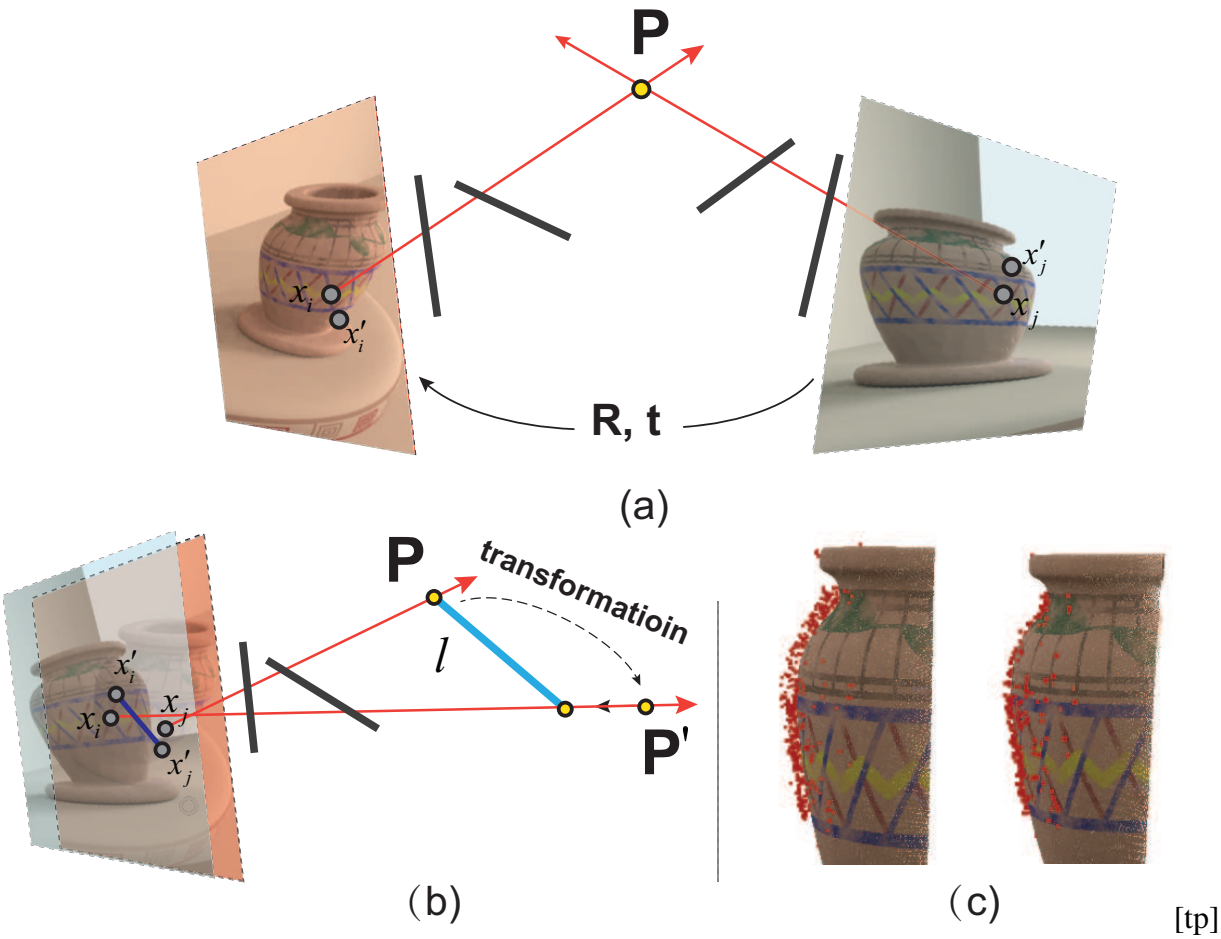
where λ is the ray propagation factor.

Our goal is to solve the 3D coordinate (x, y, z) of the scene point $\mathbf{P}$. We treat λ as independent variable and Eqn. 6.12 becomes a linear constraint. Given N views, we can formulate three equations per viewpoint using Eqn. 6.12 and we have 3N equations in total. We then stack these equations into a linear system and solve the N+3 unknowns (x, y, z and N λ values) by applying SVD.

Bundle Adjustment by Depth-Dependent Slope. We perform bundle adjustment on XSlit images to further refine the camera poses and the 3D scene geometry. Classical approach minimizes the re-projection errors of detected feature points. Similarly, we also adopt the re-projection error for optimization. The re-projection error metric can be written as:

$$E_d = \sum_{i}^{m} \sum_{j}^{n} w_i^j \left\| \Phi \left[K, \mathcal{T}\left(\mathbf{P}_j, \langle \mathbf{R}_i, \mathbf{t}_i \rangle\right) \right] - \mathrm{x}_{ij} \right\|^2 \tag{6.13}$$

where w_i^j is a binary variable indicating the visibility of the jth 3D point in camera i (1 means visible). K is the XSlit intrinsic matrix; $\mathcal{T}$ transforms the 3D point $\mathbf{P}$ into the camera coordinate using $\mathbf{R}$, $\mathbf{t}$; and Φ is the XSlit projection function defined as:



[tp]

Figure 6.5: Two error metrics for XSlit bundle adjustment. (a) Re-projection error; (b) Depth-dependent slope (DDS) error; (c) Bundle adjustment (BA) comparison without and with DDS error metric.

$$\Phi(K, \mathbf{P}) = \mathsf{E} \cdot \begin{bmatrix} \mathsf{E} + \mathsf{A}z & \mathsf{B}z \\ \mathsf{C}z & \mathsf{E} + \mathsf{D}z \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6.14}$$

Since the re-projection error is insensitive to deviations along the projection ray, we adopt the depth-dependent slope (DDS) [81] as an additional constraint. DDS indicates that the slope of a frontal parallel line segment in XSlit image changes according to its depth to the camera.

Let's consider the ith view $\mathbb{X}_i$ as reference and the jth view $\mathbb{X}_j$ as the target with rotation $\mathbf{R}_{ij}$ and translation $\mathbf{t}_{ij}$. Point $\mathbf{P}_k$ projects to x_{ik} in $\mathbb{X}_i$, and x_{jk} in $\mathbb{X}_j$. Alternatively, we assume the camera is static and $\mathbf{P}_k$ moves reversely. Consider a frontal parallel line segment ℓ passing through $\mathbf{P}_k$ and intersecting with the two projection rays. We denote the slope of line segment connecting x_{ik} and x_{jk} as s_α. The observed aspect ratio of ℓ in image is:

$$r_\alpha = \frac{\sin\theta_2 - s_\alpha \sin\theta_1}{s_\alpha \cos\theta_1 - \cos\theta_2} \tag{6.15}$$

After transforming $\mathbf{P}_k$, we obtain a new point $\mathbf{P}'_k = \mathbf{R}_{ij}\mathbf{P}_k + \mathbf{t}_{ij}$, where $\mathbf{R}_{ij}$ and $\mathbf{t}_{ij}$ is the transformation between $\mathbb{X}_i$ and $\mathbb{X}_j$. Since ℓ is front parallel, we can easily compute its slope s_β by trace $\mathbf{P}'_k$ along its projection ray to $\mathbf{P}_k$'s depth z_k. Similarly, we can also compute the aspect ratio r_β. The error metric based on DDS can be written as:

$$E_r = \sum_i^m \sum_j^n v_i^j \| z_1(z_k - z_2) r_\alpha - z_2(z_k - z_1) r_\beta \|^2 \tag{6.16}$$

We combine both the re-projection error and the DDS error as our final objective function for optimizing the viewpoint transformation matrices $\mathbf{R}$, $\mathbf{t}$ and the 3D point coordinate $\mathbf{P}$.

$$\mathbf{P}, \mathbf{R}, \mathbf{t} \leftarrow \arg\min_{\mathbf{P},\mathbf{R},\mathbf{t}} (E_d + E_r) \tag{6.17}$$

In Fig. 6.5, I show illustrations of our two error metrics and the bundle adjustment results. We can see that the DDS error metric effectively improves the reconstruction accuracy.

6.6 Experiments

In this section, I perform experiments to evaluate the proposed algorithms.

6.6.1 Feature Matching

I first evaluate the non-uniform Gaussian based feature matching algorithm. I show that the proposed feature detector is able to handle drastic viewpoint changes as well as large geometric distortions exhibited in XSlit images. I test our algorithm on the Graffiti dataset [1] which contains images captured with viewing angles ranging from 20° to 60°. Fig. 6.6(a) shows a subset of matched feature points produced by our proposed algorithm. The viewing angle difference between the two images is 60°. The overlayed parallelograms illustrate the affine transformations that are used to generate the feature points. This example shows that our algorithm is able to handle images large viewpoint change.

To quantitatively evaluation the performance, I employ the recall-precision curve where recall refers to the ratio between the number of valid features and all matched features. I compare the proposed method with state-of-the-art feature detectors, such as SIFT, Harris, Hessian, MSER and ASIFT. Valid feature points are defined as a pair of corresponding points within 1.5 pixel difference after being warped by the estimated homography. In Fig. 6.6(b), the left image shows the recall curve w.r.t the viewing angle change and the right image shows the number of valid feature and all matched features. While the recall curves of all other feature detectors descend rapidly when the viewing angle variation increases, the ratio remains high for our method. This is because the non-uniform Gaussian kernels adopted in our approach are effective in handle large distortions. Although in some cases the ASIFT detects more feature points in total, its mismatch rate is very high.

6.6.2 Camera Pose Estimation

Next I evaluate our pose estimation algorithm through simulation. In our experiment, I set up the XSlit camera as $z_1 = 1$, $z_2 = 2$ and $\theta_1 = 0$, $\theta_2 = 90°$. I render images at resolution 800×600. The camera is moved by a rotation matrix with Euler angles $[30°, 30°, -30°]$ and translation vector $[2, 3, 0]$.

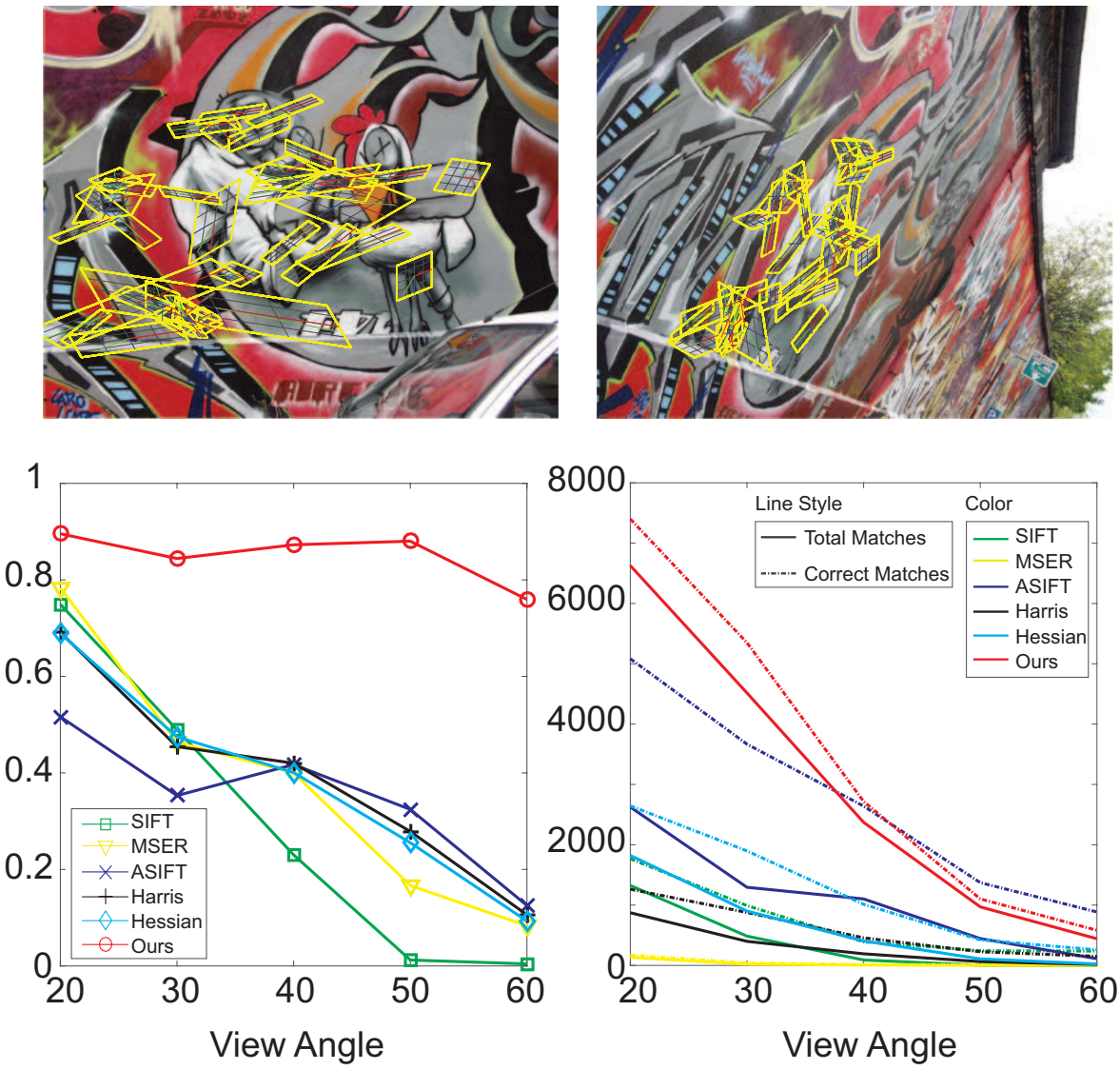


Figure 6.6: Feature matching evaluation. Top: Subset of feature points generated by our algorithm. The overlayed parallelograms illustrates the affine transformations that are used to produce the features; Bottom left: Precision-recall curves in comparison with state-of-the-arts; Bottom right: Numbers of valid features (correctly matched) and all matched features w.r.t. different viewing angles.

I compare our method with [40]. I randomly generate 100 3D points and project those points into the XSlit camera. I then add Gaussian noise with std = 1 to the projected pixels. I feed these noisy data to our algorithm and Li's method [40]. I use the angular difference between two rotation matrices as the rotation error. Given two rotation matrices $\mathbf{R_1}$ and $\mathbf{R_2}$, their angular difference is computed as $\cos^{-1}[(\mathrm{tr}(\mathbf{R_1} * \mathbf{R_2}^{\mathrm{T}}) - 1)/2]$. The translation error is directly measured by the Euclidean distance between the two translation vectors. We simulate 1000 random tests. Fig. 6.7 shows the histogram of the rotation errors and translation errors for our algorithm and [40]. Li et al.'s method [40] does not work well on XSlit because it assumes the camera is locally central or axial.

I further evaluate the robustness of the proposed algorithm w.r.t the noise ratio and the point-to-camera distance. The point-to-camera distance is measured by the XSlit camera depth sensitivity $r_z = z_2(z_2 - z_1)/z_1$, where z_1 and z_2 are the two slits' distances. The results are shown in Fig. 6.8 .

6.6.3 Point Cloud Reconstruction

Finally, I evaluate our XSlit SfM framework on both synthetic and real data.

Synthetic Data. I use ray tracing to render synthetic XSlit images. Specifically, I implement an XSlit camera model in the open source ray tracer POV-Ray [3].

I first test on a simple ladybug scene which contains very few feature points. I use a XSlit camera with $z_1 = 1$ and $z_2 = 3$ to capture the ladybug image. The size of the ladybug is $9 \times 5 \times 5$. I place the XSlit camera about 15 unit away from the ladybug and rotate around it. I render an image in every 10° and use 6 images in total. The image resolution is 800×600. I follow the incremental SfM pipeline. I first perform feature matching and pose estimation for camera 1 and 2 and triangulate rays from the matched feature points. I then perform the same operations for camera 2 and 3, and so on until all cameras are considered. After pose estimation and triangulation for all five camera pairs, I merge and transform all recovered point clouds into the camera 1's coordinate. Finally, I perform the proposed bundle adjustment algorithm to refine both the camera poses and point clouds. The results (both point cloud and camera poses) are shown in Fig. 6.9 (top row). I superimpose

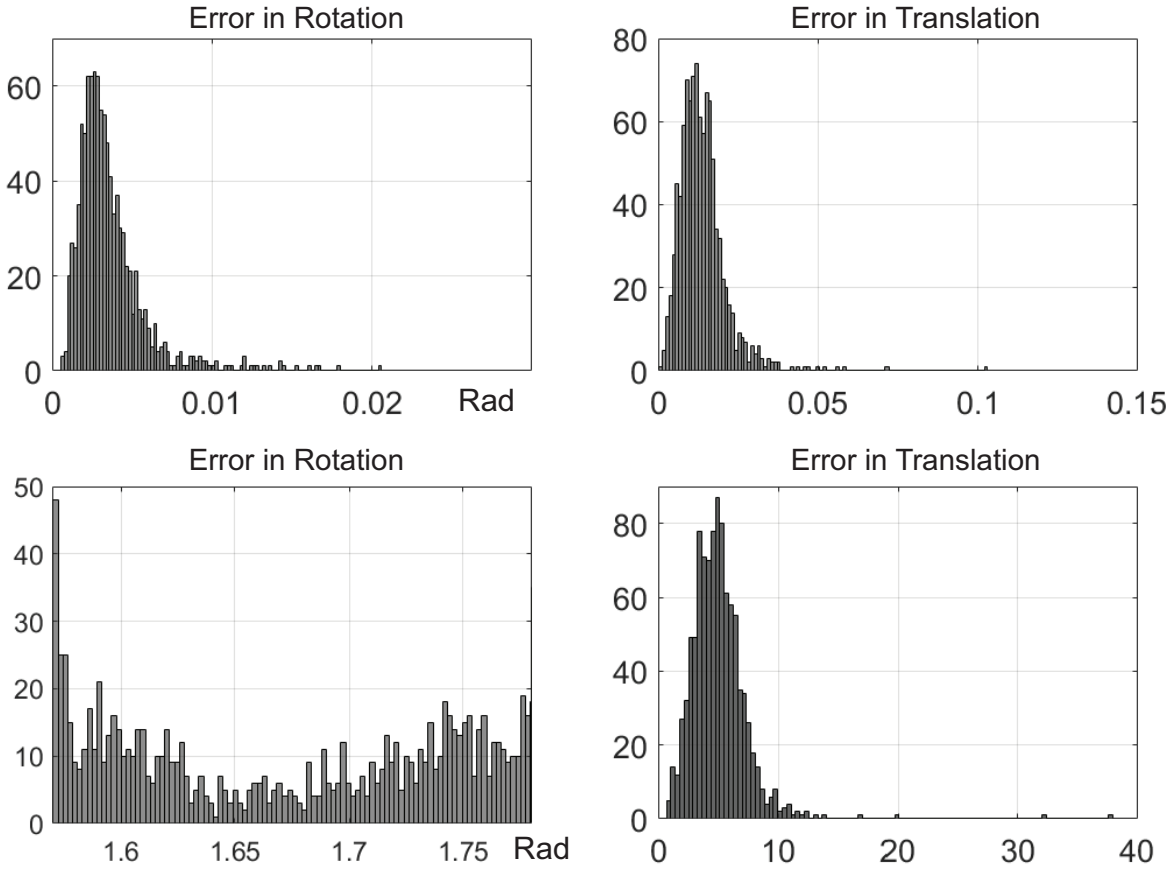


Figure 6.7: Histogram of translation error and rotation error in comparison with [40]

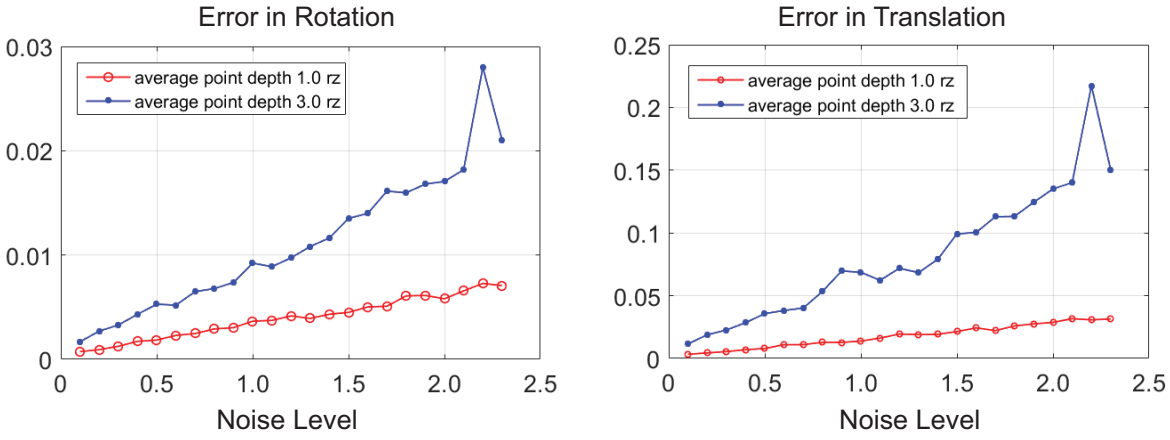


Figure 6.8: Translation error and rotation error under different noise levels and point-to-camera distances.

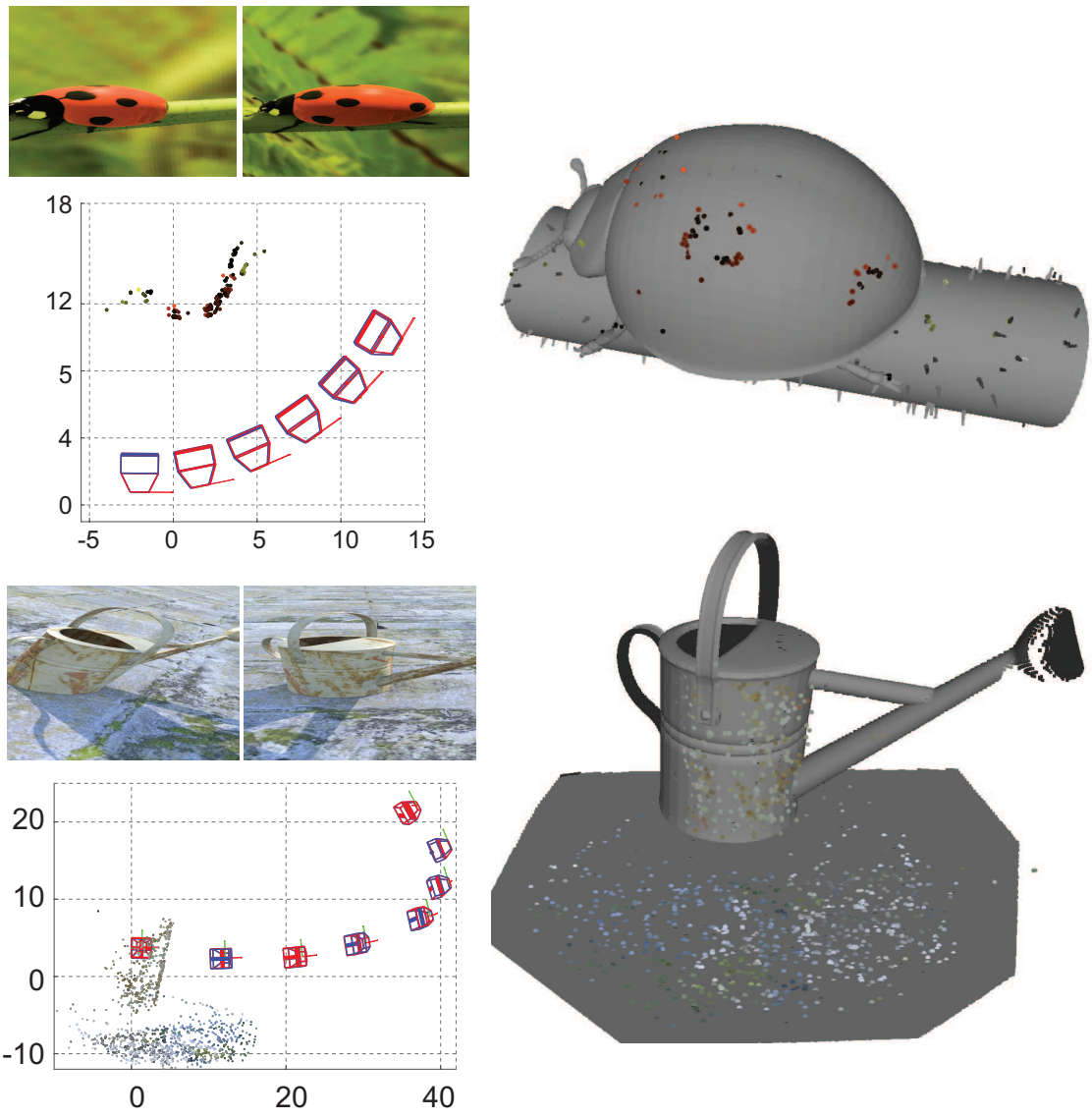


Figure 6.9: Results on two synthetic data. Top row shows the ladybug scene and bottom row shows the water pot scene. For each scene, I show two sample XSlit images, recovered point clouds, estimated camera positions (blue), ground truth camera positions (red), and our point cloud superimposed on the ground truth mesh.

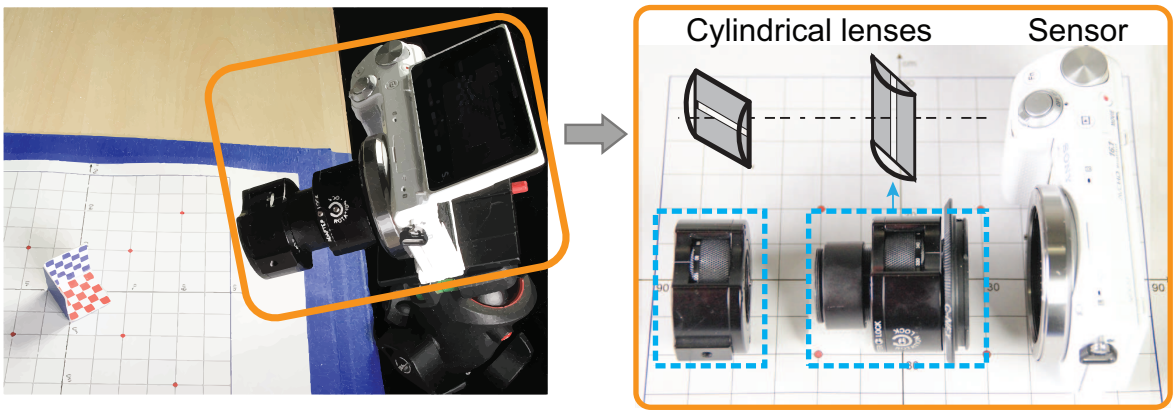


Figure 6.10: Left: Our experimental setup; Right: Our real XSlit camera construction.

the recovered point cloud onto the ground truth mesh. Although the point cloud is sparse due to the few number of available feature points, the recovered points fit well on the ground truth and the 3D points are recovered with absolute scales.

I then test on a more complex water pot scene. The size of the water pot is $20\times10\times20$. The pot has high resolution texture with fine details that can provide more feature points. I use an XSlit camera with $z_1 = 3$ and $z_2 = 9$. I place the XSlit camera about 35 unit away from the pot and then rotate around the object. I render 8 images in total with 15° step. The image resolution is 800×600. Our reconstruction process is the same as the ladybug scene. Our results are shown in Fig. 6.9 (bottom row). I recovered a denser point cloud this time and our reconstruction also fits the ground truth well.

Real Data. To capture real data, I construct an XSlit lens using two cylindrical lenses [82]. In particular, the two cylindrical lenses are with focal lengths 25mm (closer to the sensor) and 75mm (farther away from the sensor) respectively. The principal axis of the two lenses are orthogonal and I use two slit apertures with 1mm width to form sharp images. The distance between the two lenses is adjustable between 5cm and 12cm. Our camera setup is shown in Fig. 6.10.

I first test our method on a simple checker scene (as shown in Fig. 6.10). I use a cube as our reconstruction target and put checkerboards on the cube faces to provide feature points. In this experiment, I manually extract the checker corners and use them to estimate camera poses. I use triangulation to recover the 3D points. The estimated camera poses and

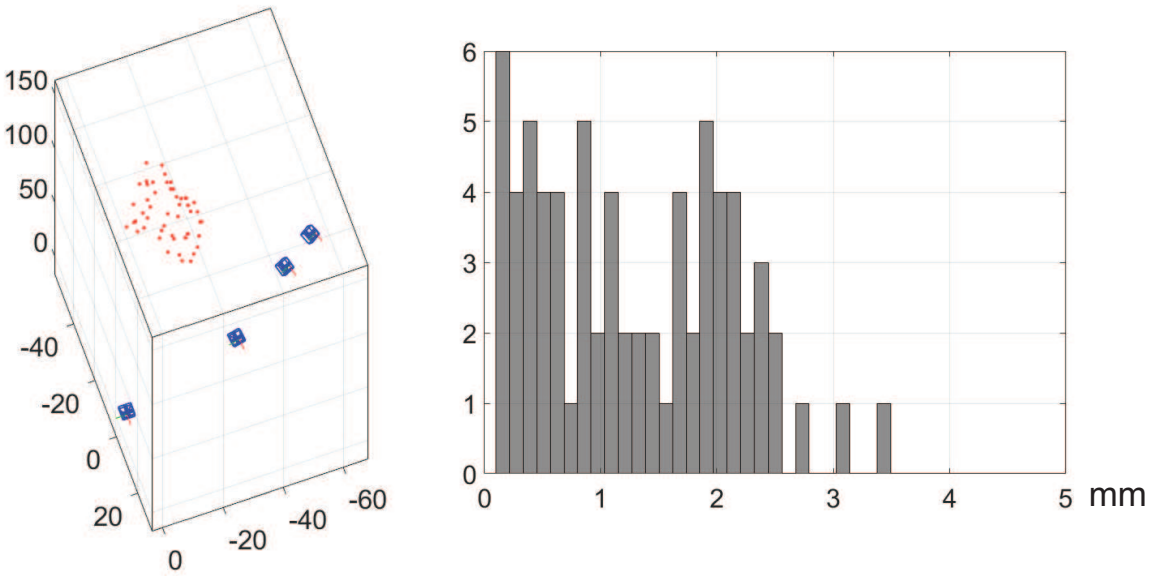


Figure 6.11: Results of the checker cube scene. Left: Our recovered camera poses and 3D points; Right: Histogram of distance errors.

3D points are shown in Fig 6.11 left. I use the distance error between neighboring corners to evaluate our reconstruction since the checker corners are all equally spaced by 5mm. Fig 6.11 right shows the histogram of the distance errors.

I then construct another scene by placing two toys on a printed coordinate grid, as shown in Fig. 6.12(a). I perform the proposed feature matching algorithm on captured XSlit images. Fig. 6.12(b) shows the feature matching result for one XSlit image pair. I then estimate the camera poses and triangulate the 3D point cloud. I also take 12 images using a perspective camera and compute a surface mesh using a SfM software AgiSoft [2]. I treat the mesh as ground truth. I resolve the scale ambiguity in the perspective case using the coordinate grid. I align the two reconstructions and superimpose our point cloud on the ground truth mesh. As shown in Fig. 6.12(c), the two reconstructions are consistent and our XSlit SfM estimates the 3D point cloud with absolute scale.

6.7 Discussions

In conclusion, I have presented a novel SfM framework using an XSlit camera. XSlit SfM is free from the scale ambiguity due to depth-dependent distortions. I first derived the

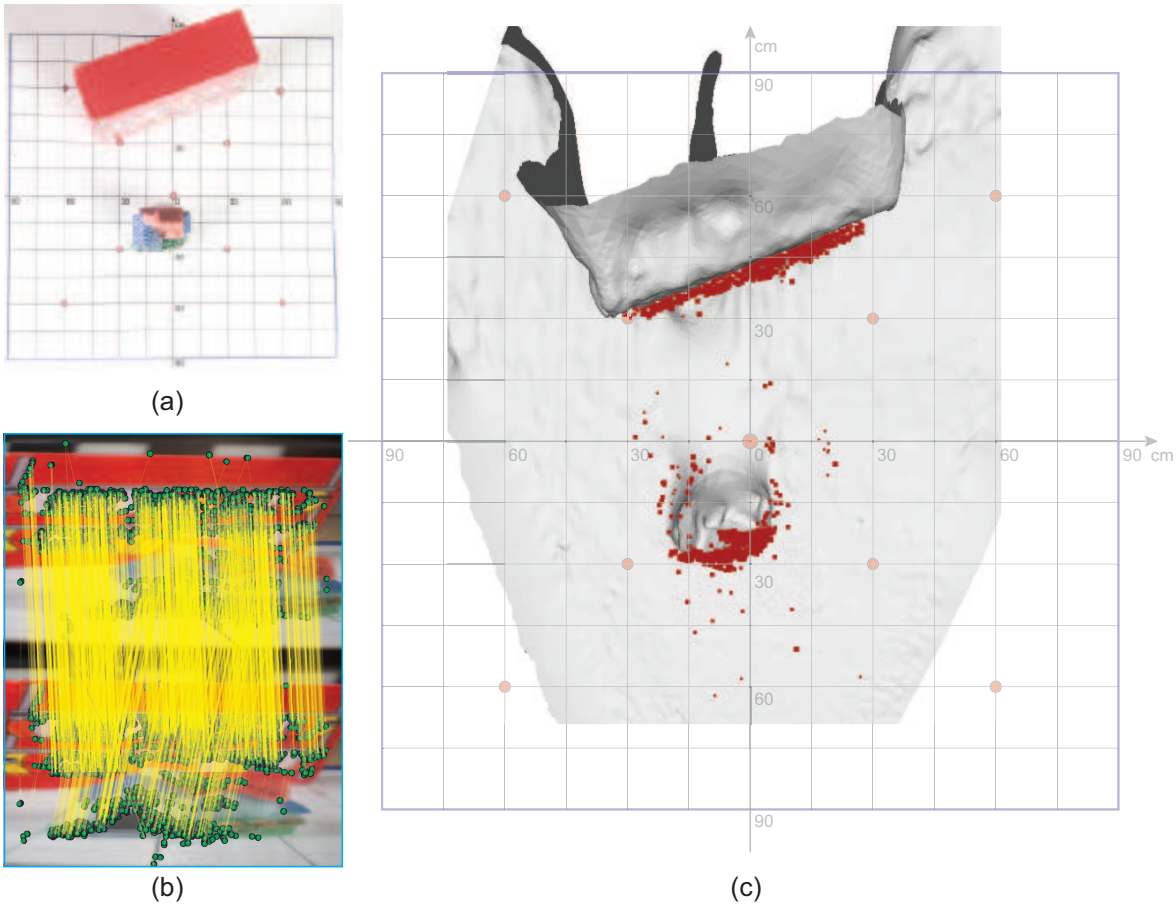


Figure 6.12: Results of the toy scene. (a) Scene setup; (b) Matched feature points; (c) Recovered point cloud superimposed on the ground truth mesh.

fundamental matrix to correlate XSlit images captured at different viewpoints. I reduced the degree of freedom in the fundamental matrix and use it to estimate the viewpoint transformation. Furthermore, to perform accurate feature matching in XSlit images where often exhibits large distortions, I developed a non-uniform Gaussian based feature algorithm to handle distortions. Finally, I extend the bundle adjustment on XSlit images to refine the estimated camera pose and the 3D scene structure. Through both synthetic and real experiments, I show that the proposed XSlit SfM framework is able to estimate the camera pose accurately and recover the 3D scene geometry with an absolute scale.

Chapter 7

CONCLUSIONS AND FUTURE WORK

This thesis has focused on the exploration of advantages of general non-centric cameras. The specific context I interest in is scene understanding, including single camera and multi-cameras based solutions. This chapter first summarize the approaches I use and the contributions made in the thesis. Then I discuss the limitations and open questions. Finally, I explore the future work implied by this work.

7.1 Summary

Here I summarize the approaches and contributions of the thesis. In summary, the thesis have explored the advantages of general non-centric cameras. Chapter 4 presented the CCP feature that exists only in non-centric cameras. The necessary and sufficient conditions for CCP to exist in general non-centric cameras are identified. I propose several key applications, including plane localization, for demonstration. Chapter 5 explored the DDAR property in a special type of non-centric cameras, the XSlit camera. With repeat pattern or vertical line assumptions about the scene, I propose a depth inference framework from a single XSlit image for Manhattan scenes. Chapter 6 rebuilt the SfM framework for XSlit camera, and show how non-centric cameras can resolve the scale ambiguity that enwinds the perspective cameras.

7.1.1 CCP

In chapter 4, I have explored a new type of image features called the coplanar common point or CCP in general non-centric cameras. A CCP corresponds to the intersection of the curved projections of all lines lying on a common 3D plane. I have shown that CCPs generally exist in a broad range of non-centric cameras such as the GLCs and catadioptric

cameras, and the perspective camera is the single exception that do not have CCP. I have further derived the necessary and sufficient conditions for a plane to have CCP in an arbitrary non-centric camera such as non-centric catadioptric mirrors. Finding the CCP of a 3D plane is equivalent to solving an array of ray constraint equations in ray space. For certain types of non-centric cameras, e.g catadioptric imaging system, the ray space constraints can be highly complex. The caustics provides simple and effective solution for determining CCP existence in these camera models. To demonstrate that CCP can potentially benefit the 3D reconstruction tasks, I then show some key applications of CCP. I show that with solely CCPs, we still can localize the planes in rotationally symmetric mirrors. Experiments on both synthetic and real data show that the CCP based solution provides effective and reliable solution for scene understanding. Experiments have validated our theories and the detected CCPs can facilitate 3D plane localization tasks, which is crucial to 3D scene reconstruction.

7.1.2 DDAR

I have comprehensively studied the aspect ratio (AR) distortion in XSlit cameras and exploited its unique depth-dependent property for 3D inference in chapter 5. The studies have shown that unlike perspective camera that preserves AR under depth variations, AR changes monotonically with respect to depth in an XSlit camera, i.e., 3D objects of an identical size will exhibit significantly different AR under different depths. This has led to new depth-from-AR schemes using a single XSlit image even if the original AR of an object is unknown. I have further shown that similar to AR variations, the slope of projected 3D lines will also vary with respect to depth, and I have developed theories to characterize such variations based on AR analysis. Finally, AR and line slope analysis can be integrated for 3D reconstruction and I have experimented on real XSlit images captured by an XSlit camera, synthesized from panorama stitching, and captured using a catadioptric mirror to validate our framework.

7.1.3 XSlit Structure from Motion

In chapter 6, I have presented a novel XSlit SfM framework that directly solve the scale ambiguity. I first demonstrate that there exists a fundamental matrix to correlate XSlit images captured at different viewpoints. I reduce the degree of freedom in the fundamental matrix and use it to estimate the viewpoint transformation. Furthermore, to perform accurate feature matching in XSlit images where often exhibits large distortions, I develop a non-uniform Gaussian based feature algorithm to handle distortions. Finally, I extend the bundle adjustment on XSlit images to refine the estimated camera pose and the 3D scene structure. Through both synthetic and real experiments, I show that the proposed XSlit SfM framework is able to estimate the camera pose accurately and at the same time, reconstruct 3D scene geometry with absolute scale.

7.2 Limitations

In the thesis, I show that non-centric cameras exhibit several nice properties that can benefit the scene understanding tasks. However, there are still many limitations for proposed solutions.

CCPs generally exist in non-centric cameras. However, the existence is camera and plane dependent. For example, for the non-centric catadioptric cameras the CCPs existence is limited to planes that intersect with the rotation axis and a corresponding reflection circle on the mirror surface. We can see that the experiments used a slant plane to show the CCP existence in chapter 4. The non-generality of CCP in catadioptric cameras limits its applicable scope since catadioptric cameras are commonly used in robotic vision. Furthermore, identifying CCPs from images is difficult since the images of lines are generally curved in non-centric cameras. The conic curve fitting is less robust and reliable then line fitting. And also there are a lot of false intersections. Separating CCPs from false intersections is another problem that need to be addressed. Though being lack of generality in catadioptric system, the CCPs generally exist in Pushbroom and XSlit cameras. Exploration of CCPs in XSlit and Pushbroom for planar structure recovery is promising.

Though DDAR provides useful information about the scene geometry in XSlit images, the depth inference problem is still undetermined, as we still require the original ratio of the object to compute the depth. The solution illustrated in chapter 6 is assuming we position the XSlit camera front parallel and up right. The projections of non-front parallel lines in XSlit image are hyperbolas and hence it's hard to perform depth analysis. For more general usage of DDAR, we need to develop a depth inference scheme without explicit assumption about the scene geometry.

In XSlit SfM, the depth sensitivity reduces as the object moves away from the XSlit camera. Hence the accuracy of the XSlit camera motion estimation reduces accordingly. The recoverable range of the XSlit camera is determined by the two slits distance and their distances to the sensor. Generally the working distance is relatively smaller compared to that of perspective cameras. To achieve larger depth recoverable range, we need larger XSlit cameras.

7.3 Future Work

Though have answered many questions, the work in this thesis also have opened many doors for future exploration about non-centric cameras in scene understanding.

First, the accuracy of CCP detection largely depends on the curve fitting. The curves in catadioptric mirrors are highly non-linear, the current solution is to search for the optimal solution from a set of basis functions using a look-up table. In particular, I do not consider the mirror geometry and as a result it can be sensitive to discretization errors. In the future, I plan to develop tailored curve fitting algorithm by imposing mirror geometry as constraints. In addition, although our caustic-based analysis is applicable to arbitrary catadioptric mirrors, we have by far only studied in depth the cylindrical and rotationally symmetric mirrors. In the future, I will explore efficient testing schemes for general catadioptric mirrors. Finally, I intend to integrate the VP and CCP analysis under a unified geometric framework. Conceptually VPs present directions and CCPs represent positions. A unified representation under projective geometry [62] may sufficiently address both problems via a more elegant model.

The cylindrical lens based XSlit camera has a small baseline(i.e., the distance between the two slits) and therefore can only acquire AR changes within a short range. Constructing a large baseline XSlit camera will be costly as it is difficult to fabricate large form cylindrical lens. A more feasible solution would be adopt a cylindrical catadioptric mirror where the reflection image can be approximated as an XSlit image. In the future, I will explore effective schemes for correcting both geometric distortion and blurs due to imperfect mirror geometry. I will also investigate integrating our AR based solution into prior based frameworks to enhance reconstruction quality. For example, a hybrid XSlit-perspective camera pair can be constructed. Finally, since AR distortions commonly exhibit in synthesized panoramas as shown in the thesis, I plan to study effective image-based distortion correction techniques to produce perspectively sound panoramas analogous to [6].

The reconstructed point cloud by the proposed XSlit SfM framework is rather sparse. This is because lack of texture on the object surface for feature matching. In the future, I plan to extend the approach to dense point cloud reconstruction by triangulating textureless regions using camera transformation matrices. On the other hand, the proposed approach only works well within a limited depth range. This is due to the limitation in slit size. With future advances on camera construction, I expect to apply the proposed approach to large scale scenes. In the future, I also plan to extend the SfM framework to a broad range of multi-perspective cameras.

BIBLIOGRAPHY

[1] Affine covariant regions. http://www.robots.ox.ac.uk/~vgg/research/affine/.

[2] Agisoft photoscan. http://www.agisoft.com/.

[3] Pov-ray. http://povray.org/.

[4] Edward H Adelson and James R Bergen. The plenoptic function and the elements of early vision. 1991.

[5] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[6] Aseem Agarwala, Maneesh Agrawala, Michael F. Cohen, David Salesin, and Richard Szeliski. Photographing long scenes with multi-viewpoint panoramas. *ACM Trans. Graph.*, 25(3):853–861, 2006.

[7] Simon Baker and Shree K. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2), 1999.

[8] Simon Baker and Shree K Nayar. Single viewpoint catadioptric cameras. *Panoramic vision*, pages 39–71, 2001.

[9] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[10] Ricardo Cabral and Yasutaka Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 628–635. IEEE, 2014.

[11] V. Caglioti and S. Gasparini. "how many planar viewing surfaces are there in noncentral catadioptric cameras?" towards singe-image localization of space lines. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1266–1273, June 2006.

[12] João R Cardoso and Pedro Miraldo. Fitting generalized essential matrices from generic 6x6 matrices and its applications. *arXiv preprint arXiv:1709.06328*, 2017.

[13] Brian Clipp, Jae-Hak Kim, Jan-Michael Frahm, Marc Pollefeys, and Richard Hartley. Robust 6dof motion estimation for non-overlapping, multi-camera systems. In *Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on*, pages 1–8. IEEE, 2008.

[14] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 941–947. IEEE, 1999.

[15] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000.

[16] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *IEEE International Conference on Computer Vision(ICCV)*, volume 3, pages 1403–1410, 2003.

[17] Erick Delage, Honglak Lee, and Andrew Ng. Automatic single-image 3d reconstructions of indoor manhattan world scenes. *Robotics Research*, pages 305–321, 2007.

[18] Erick Delage, Honglak Lee, and Andrew Y Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2418–2428. IEEE, 2006.

[19] Yuanyuan Ding and Jingyi Yu. Epsilon stereo pairs. In *BMVC*, volume 2, page 6, 2007.

[20] Yuanyuan Ding, Jingyi Yu, and Peter Sturm. Recovering specular surfaces using curved line images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2326–2333. IEEE, 2009.

[21] Doron Feldman, Daphna Weinshall, et al. On the epipolar geometry of the crossed-slits projection. In *null*, page 988. IEEE, 2003.

[22] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[23] Alex Flint, Christopher Mei, David Murray, and Ian Reid. A dynamic programming approach to reconstructing building interiors. In *European Conference on Computer Vision(ECCV)*, pages 394–407. Springer, 2010.

[24] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Reconstructing building interiors from images. In *IEEE International Conference on Computer Vision(ICCV)*, pages 80–87. IEEE, 2009.

[25] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010.

[26] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of Conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1996.

[27] R. Gupta and R.I. Hartley. Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 963–975, 1997.

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

[29] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2746–2753. IEEE, 2013.

[30] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 807–814. IEEE, 2005.

[31] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. *ACM Transactions on Graphics (TOG)*, 24(3):577–584, 2005.

[32] Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *IEEE International Conference on Computer Vision(ICCV)*, volume 1, pages 654–661. IEEE, 2005.

[33] Aaron Isaksen, Leonard McMillan, and Steven J Gortler. Dynamically reparameterized light fields. In *Proceedings of Conference on Computer graphics and interactive techniques*, pages 297–306. ACM, 2000.

[34] Changil Kim, Alexander Hornung, Simon Heinzle, Wojciech Matusik, and Markus Gross. Multi-perspective stereoscopy from light fields. *ACM Transactions on Graphics (TOG)*, 30(6):190, 2011.

[35] Junhwan Kim et al. Visual correspondence using energy minimization and mutual information. In *IEEE International Conference on Computer Vision(ICCV)*, pages 1033–1040. IEEE, 2003.

[36] Jana Košecká and Wei Zhang. Video compass. In *European Conference on Computer Vision(ECCV)*, pages 476–490. Springer, 2002.

[37] Douglas Lanman, Megan Wachs, Gabriel Taubin, and Fernando Cukierman. Reconstructing a 3d line from a single catadioptric image. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 89–96. IEEE, 2006.

[38] David C Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2136–2143. IEEE, 2009.

[39] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[40] Hongdong Li, Richard Hartley, and Jae-hak Kim. A linear approach to motion estimation using generalized camera models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[41] Nianyi Li, Haiting Lin, Bilin Sun, Mingyuan Zhou, and Jingyi Yu. Rotational crossed-slit light field. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 4405–4413, 2016.

[42] David Liebowitz and Andrew Zisserman. Combining scene and auto-calibration constraints. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 293–300. IEEE, 1999.

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

[44] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012.

[45] Jiri Matas, Ondra Chum, Martin Urban, and Tomas Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press, 2002. doi:10.5244/C.16.36.

[46] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings of Conference on Computer graphics and interactive techniques*, pages 39–46. ACM, 1995.

[47] Chunhui Mei, Voicu Popescu, and Elisha Sacks. The occlusion camera. In *Computer Graphics Forum*, volume 24, pages 335–342. Wiley Online Library, 2005.

[48] Branislav Micusik and Tomas Pajdla. Autocalibration & 3d reconstruction with noncentral catadioptric cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE, 2004.

[49] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, European Conference on Computer Vision(ECCV), pages 128–142, London, UK, UK, 2002. Springer-Verlag.

[50] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.

[51] Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.*, 2(2):438–469, April 2009.

[52] Shree K Nayar and Simon Baker. Catadioptric image formation. In *Proceedings of the 1997 DARPA Image Understanding Workshop*, pages 1431–1437, 1997.

[53] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11):1–11, 2005.

[54] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.

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

[56] John Novatnack and Ko Nishino. Scale-dependent 3d geometric features. In *IEEE International Conference on Computer Vision(ICCV)*, pages 1–8. IEEE, 2007.

[57] T. Pajdla. Stereo with oblique cameras. In *IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 85–91, 2001.

[58] Tomáš Pajdla. Geometry of two-slit camera. *Rapport Technique CTU-CMP-2002-02, Center for Machine Perception, Czech Technical University, Prague*, 2002.

[59] Shmuel Peleg and Moshe Ben-Ezra. Stereo panorama with a single camera. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 395–401. IEEE, 1999.

[60] Robert Pless. Using many cameras as one. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–587. IEEE, 2003.

[61] Marc Pollefeys, David Nistér, J-M Frahm, Amir Akbarzadeh, Philippos Mordohai, Brian Clipp, Chris Engels, David Gallup, S-J Kim, Paul Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008.

[62] Jean Ponce. What is a camera? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1533. IEEE, 2009.

[63] Augusto Roman, Gaurav Garg, and Marc Levoy. Interactive design of multi-perspective images for visualizing urban landscapes. In *Proceedings of the Conference on Visualization'04*, pages 537–544. IEEE Computer Society, 2004.

[64] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, pages 1161–1168, 2005.

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

[66] Davide Scaramuzza, Friedrich Fraundorfer, Marc Pollefeys, and Roland Siegwart. Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1413–1419. IEEE, 2009.

[67] Daniel Scharstein, Richard Szeliski, and Ramin Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV)*, pages 131–140. IEEE, 2001.

[68] Yoav Y Schechner and Shree K Nayar. Generalized mosaicing. In *IEEE International Conference on Computer Vision(ICCV)*, volume 1, pages 17–24. IEEE, 2001.

[69] Grant Schindler and Frank Dellaert. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–203. IEEE, 2004.

[70] Steven M Seitz and Jiwon Kim. The space of all stereo images. *International Journal of Computer Vision*, 48(1):21–38, 2002.

[71] Heung-Yeung Shum and Li-Wei He. Rendering with concentric mosaics. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 299–306. ACM Press/Addison-Wesley Publishing Co., 1999.

[72] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[73] Peter Sturm. Multi-view geometry for general camera models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 206–212. IEEE, 2005.

[74] Rahul Swaminathan, Michael D. Grossberg, and Shree K. Nayar. Non-single viewpoint catadioptric cameras: Geometry and analysis. *International Journal of Computer Vision*, 66(3):211–229, 2006.

[75] Hynek Bakstein Tom. Rendering novel views from a set of omnidirectional mosaic images. In *In Proceedings of Omnivis 2003: Workshop on Omnidirectional Vision and Camera Networks*. IEEE Press, 2003.

[76] Jonathan Ventura, Clemens Arth, Gerhard Reitmayr, and Dieter Schmalstieg. A minimal solution to the generalized pose-and-scale problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 422–429, 2014.

[77] R Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: a line segment detector. *Image Processing On Line*, 2(3):5, 2012.

[78] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 765–776. ACM, 2005.

[79] Jian Wu, Zhiming Cui, Victor S Sheng, Pengpeng Zhao, Dongliang Su, and Shengrong Gong. A comparative study of sift and its variants. *Measurement Science Review*, 13(3):122–131, 2013.

[80] Wei Yang, Yu Ji, Jinwei Ye, S Susan Young, and Jingyi Yu. Coplanar common points in non-centric cameras. In *European Conference on Computer Vision(ECCV)*, pages 220–233. Springer, 2014.

[81] Wei Yang, Haiting Lin, Sing Bing Kang, and Jingyi Yu. Resolving scale ambiguity via xslit aspect ratio analysis. In *IEEE International Conference on Computer Vision(ICCV)*, December 2015.

[82] Jinwei Ye, Yu Ji, and Jingyi Yu. Manhattan scene understanding via xslit imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 81–88. IEEE, 2013.

[83] Jinwei Ye, Yu Ji, and Jingyi Yu. A rotational stereo model based on xslit imaging. In *IEEE International Conference on Computer Vision(ICCV)*, pages 489–496, Dec 2013.

[84] Guoshen Yu and Jean-Michel Morel. Asift: An algorithm for fully affine invariant comparison. *Image Processing On Line*, 1:11–38, 2011.

[85] Jingyi Yu. *General linear cameras: theory and applications*. PhD thesis, Massachusetts Institute of Technology, 2005.

[86] Jingyi Yu and Leonard McMillan. General linear cameras. In *European Conference on Computer Vision(ECCV)*, pages 14–27. Springer, 2004.

[87] Jingyi Yu and Leonard McMillan. Modelling reflections via multiperspective imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 117–124. IEEE, 2005.

[88] Jingyi Yu, Leonard McMillan, and Peter Sturm. Multiperspective modeling, rendering, and imaging. In *ACM SIGGRAPH ASIA 2008 Courses*, pages 14:1–14:36, 2008.

[89] Assaf Zomet, Doron Feldman, Shmuel Peleg, and Daphna Weinshall. Non-perspective imaging and rendering with the crossed-slits projection. Technical report, Leibnitz Center, Hebrew University of Jerusalem, 2002.

[90] Assaf Zomet, Doron Feldman, Shmuel Peleg, and Daphna Weinshall. Mosaicing new views: The crossed-slits projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):741–754, 2003.