Université de Montréal

Comparing of Radial and Tangencial Geometric for Cylindric Panorama

par Faezeh Amjadi

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Août, 2016

Résumé

Les caméras ont généralement un champ de vision à peine assez grand pour capturer partie de leur environnement. L'objectif de l'immersion est de remplacer virtuellement un grand nombre de sens, de sorte que l'environnement virtuel soit perçu comme le plus réel possible. Une caméra panoramique est utilisée pour capturer l'ensemble d'une vue 360°, également connue sous le nom d'image panoramique. La réalité virtuelle fait usage de ces images panoramiques pour fournir une expérience plus immersive par rapport aux images sur un écran $2D$. Cette thèse, qui est dans le domaine de la vision par ordinateur, s'intérèsse à la création d'une géométrie multi-caméras pour générer une image cylindrique panoramique et vise une mise en œuvre avec les caméras moins chères possibles. L'objectif spécifique de ce projet est de proposer une géométrie de caméra qui va diminuer au maximum les problèmes d'artefacts liés au parallaxe présent dans l'image panoramique. Nous présentons une nouvelle approche de capture des images panoramiques cylindriques à partir de plusieurs caméras disposées uniformément autour d'un cercle. Au lieu de regarder vers l'extérieur, ce qui est la configuration traditionnelle "radiale", nous proposons de rendre les axes optiques tangents au cercle des caméras, une configuration "tangentielle". Outre une analyse et la comparaison des géométries radiales et tangentielles, nous fournissons un montage expérimental avec de vrais panoramas obtenus dans des conditions réalistes.

Mots clés:

Vision par ordinateur, Imagerie panoramique, Panoramas cylindriques, Calibrage, Capture immersive, Parallaxe, Géométrie multi-caméra tangentielle, Géométrie multi-caméra radiale

Summary

Cameras generally have a field of view only large enough to capture a portion of their surroundings. The goal of immersion is to replace many of your senses with virtual ones, so that the virtual environment will feel as real as possible. Panoramic cameras are used to capture the entire 360° view, also known as panoramic images. Virtual reality makes use of these panoramic images to provide a more immersive experience compared to seeing images on a 2D screen. This thesis, which is in the field of the Computer vision, focuses on establishing a multi-camera geometry to generate a cylindrical panorama image and successfully implementing it with the cheapest cameras possible. The specific goal of this project is to propose the camera geometry which will decrease artifact problems related to parallax in the panorama image. We presents a new approach of cylindrical panoramic images from multiple cameras which its setup has cameras placed evenly around a circle. Instead of looking outward, which is the traditional ”radial” configuration, we propose to make the optical axes tangent to the camera circle, a ”tangential” configuration. Beside an analysis and comparison of radial and tangential geometries, we provide an experimental setup with real panoramas obtained in realistic conditions.

Keywords:

Computer vision, Panoramic imaging, Cylindrical panoramas, Calibration, Immersive capture, Parallax, Tangential camera model, Radial camera model

Table des matières

Table des figures

List of Abbreviations

3D	3 Dimension
2D	2 Dimension
FOV	Focal of View
VR	Virtual Reality
HMD	Head-Mounted Display

Acknowledgments

First of all, I would like to express my heartfelt thanks towards my supervisor, Prof. Sebastien Roy, for his guidance, collegiality, unwavering support, and extreme patience through my research. His suggestions and advice have provided me with knowledge and experience beyond the scope of my thesis and for reading through my thesis, and gave valuable advice.

For the ancestors who paved the path before me, upon whose shoulders I stand. This is also dedicated to my parents Ali and Masoumeh, to my siblings Zahra and MohammadReza.

1 Introduction

Nowadays computer graphics is applied in many domains of our life. At the end of the 20th century it is hard to assume an architect, engineer, or interior designer working without a graphics workstation. In the last years, the stormy development of microprocessor technology makes faster and faster computers to the market. The screen resolution, the number of colors available per pixel, and other features have expanded and improved, but user graphics have mostly been displayed via a simple two-dimensional display. These machines are implemented with sufficient and faster graphics boards and their prices have fallen down significantly. It becomes desirable even for a general user, to move into the world of computer graphics. This graphics power has become a new reality which started with computer games but now impacts everything. It enables to see the surrounding world in other dimension and to feel things that are not accessible in real life or even not yet created, as shown in figure(1.1). People always desire to move into this world and communicate with it - instead of only watching a picture on the monitor. This technology which becomes overwhelmingly attractive and fashionable in this decade is called Virtual Reality (VR). Virtual Reality (VR) is the application of computer technology to generate a simulated environment. Unlike traditional user interfaces, VR puts the user inside an experience. Instead of viewing a screen in front of them, users are immersed and able to interact with $3D$ worlds. By stimulating as many senses as possible, such as vision, hearing, touch, even smell, the computer is transformed into a doorkeeper to this artificial world. The only boundary to near-real VR experiences is the availability of content and inexpensive computing power. The very first idea of it was presented by Ivan Sutherland in 1965:

“make that (virtual) world in the window look real, sound real, feel real, and respond realistically to the viewer’s actions”.

An additional VR definition by Weiss et al. (2004) states:

”Virtual reality typically refers to the use of interactive simulations created with computer hardware and software to present users with opportunities to engage in

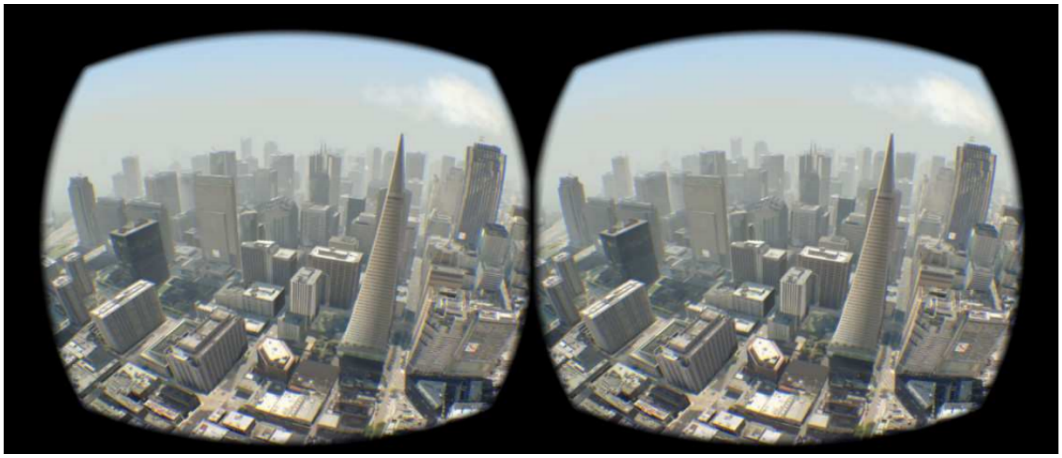


Figure 1.1 – Left and right eye views are created inside the headset, while also taking into account the user's looking direction.

environments that appear to be and feel similar to real-world objects and events. ”

VR typically requires a head-mounted display (HMD) that freely moves the display in tandem with the head. The goal of VR/AR is true immersion, nothing less than creating an illusion so real that it convinces the human brain, the world's finest computer.

Depending on input and display devices, VR systems can be separated into fully immersive and non-immersive setups. The advantage of fully immersive systems is the user's strong "sense of presence" which has been connected to the convergence of the system's multisensory input Jack et al. [2001]. In non-immersive systems, the VR system often consists of a computer monitor, mouse, keyboard and possibly joysticks, haptic devices and force sensors.

Despite the fast advance of computer graphics software and hardware in the past, most virtual reality systems still face the principal problems. First, creating the geometrical entities is a difficult manual process. Second, because the walk-through requires being performed in real-time, the rendering engine usually puts a boundary on scene complexity and rendering quality. Third, the demand for a specific target rendering engine has limited the availability of virtual reality for most people since the essential hardware is not widely available.

Traditionally, virtual reality systems use 3D computer graphics to model and render virtual environments in real-time. This approach usually requires laborious modeling and expensive special purpose rendering hardware. The rendering quality

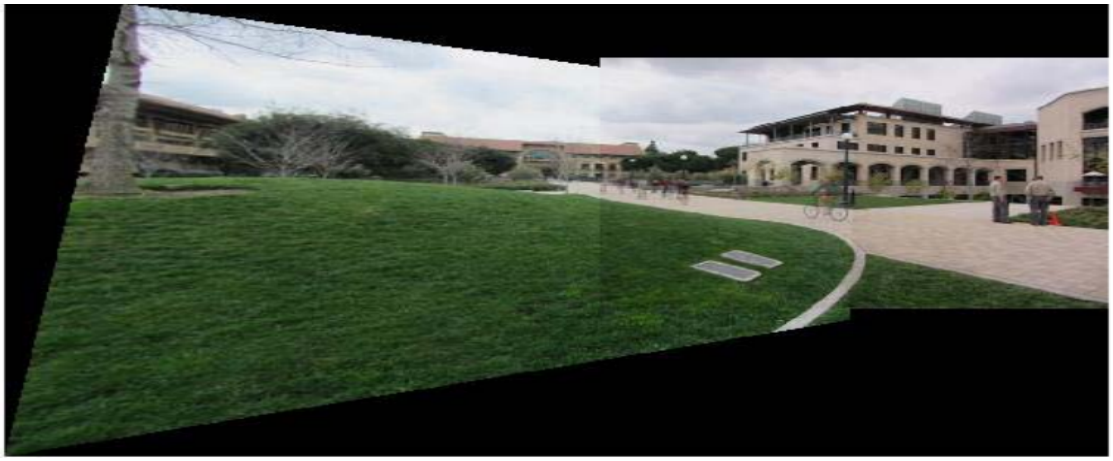


Figure 1.2 – Sample of two stitched images with a camera. The center of camera is fixed and only rotate camera to capture second image

and scene complexity are often limited because of the real-time constraint. As an alternative, we can use 360-degree cylindrical panoramic images to compose a virtual environment. The panoramic images can be created with computer rendering, specialized panoramic cameras or by ”stitching” together overlapping photographs taken with a regular camera. Walking in a space is currently accomplished by ”hopping” to different panoramic points. The image-based approach has been used in the commercial product QuickTime VR, a virtual reality extension to Apple Computer’s QuickTime digital multimedia framework. In this thesis, we work on panorama images that are high quality and low cost.

1.1 Panoramic Images

The term virtual reality (VR) was used to describe computer presentation of digital still images in a 360° panorama format. Common virtual reality software allows users to develop their own virtual environments by “stitching together” a number of images in such a way that the computer can present them as a seamless 360° panoramic scenes or environments.The images that we currently use are cylindrical panoramas. The panoramas are orientation independent because each of the images contains all the information needed to look around in 360°.

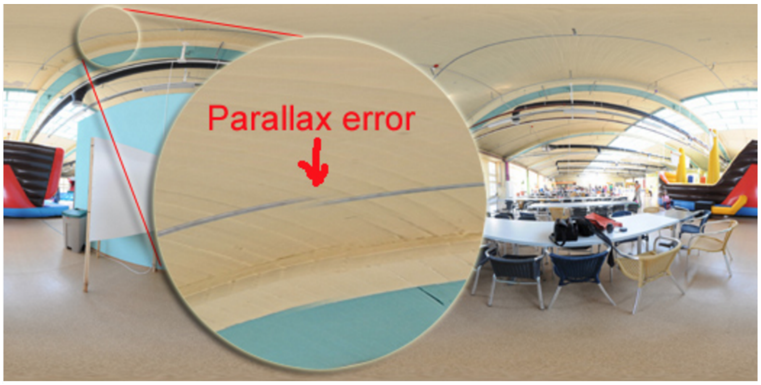


Figure 1.3 – Sample of Parallax error

1.1.1 Immersion

immerse capture is difficult to achieve with a single camera when it can be done with expensive fish eye lens or catadioptric cameras (combination of lens and mirrors) and low resolution, so we have to use multiple cameras or a single camera with rotation and translation to obtain a panorama image. In our work, we use multiple cameras and assemble for a 360° cylindrical panorama image. The number of cameras and their field of view (fov) has effects on the final image resolution.For example, if the number of cameras is twenty and fov are small, we obtain higher resolution than with fewer cameras with wider fov.

1.1.2 Visual Perception

Vision is generally considered the most dominant sense, and there is evidence that human cognition is oriented around vision. High-quality visual representation is critical for virtual environments. The major aspects of the visual sense that have an impact on display requirements are the following:

- **depth perception** stereoscopic viewing is a primary human visual mechanism for perceiving depth. However, because human eyes are located only on average 6.3 centimeters apart, the geometric benefits of stereopsis are lost for objects more distant than 30 meters, and it is most effective at much closer distances. Other primary cues (eye convergence and accommodation) and secondary cues (e.g. perspective, motion parallax, size, texture, shading, and shadows) are essential for far objects and of varying importance for near

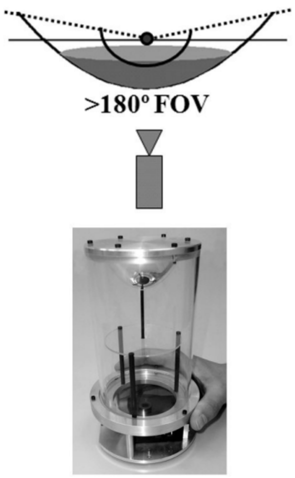


Figure 1.4 – catadioptric camera.

ones. In our work, we don't use depth perception but it is possible in the future.

- **accuracy and field-of-view** the total horizontal field of vision of both human eyes is about 180 degrees without eye/head movement and 270 with fixed head and moving eyes. The vertical field of vision is typically over 120 degrees. While the total field is not necessary for a user to feel immersed in a visual environment, 90 to 110 degrees are generally considered necessary for the horizontal field of vision. When considering accuracy, the central fov of a human eye has a resolution of about 0.5 minutes of arc.The camera resolution must be chosen to match human vision resolution. In practice, this is not always feasible sens the user is seeing the image through a screen. Therefore the screen resolution determines the minimum camera resolution.

1.2 Stitching Panorama Problem

Conventional image mosaicing techniques involve three basic steps. First, image registration is used to get the geometric relationship between input images. Second, images are warped into a common reference, so that their regions of overlap match each other. Third, these warped images are stitched together into a general mosaicing surface.

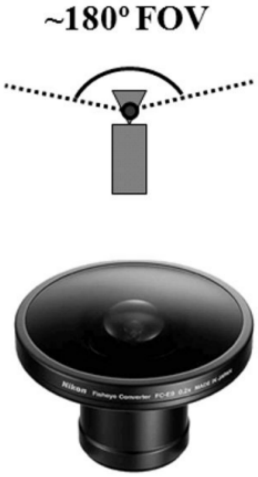


Figure 1.5 – Dioptric camera (e.g. fisheye).

Algorithms for adjusting images and stitching them into seamless photomosaics are some of the oldest and most generally used in computer vision. Image stitching is a well-studied subject Szeliski [2006]. Stitching multiple images concurrently to generate high-resolution panoramas is one of the most famous consumer applications of image registration and blending. Its first step is to align input images. Image alignment algorithms can determine the large-scale correspondence similarity among images with varying degrees of overlap. Image stitching algorithms get the alignment estimates provided by such registration algorithms and blend the images in a seamless manner, taking care to deal with possible problems such as blurring or ghosting caused by parallax and scene movement as well as alter image exposures.

To bypass the parallax problem, the input images should be taken from the same viewpoint as shown in figure 1.2 or the scene should be roughly planar. The problem of parallax is more visible when camera centers are more separated, as shown in figure 1.3. If we have multiple cameras then the center of all camera can not be in the single point, which means that there will always be parallax. To overcome the parallax problem,we can either apply an algorithm such as direct or feature based, or modify the cameras geometry.

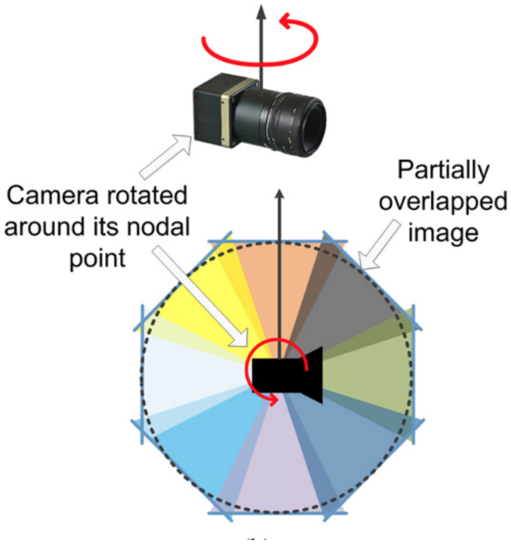

Camera rotated around its nodal point
Partially overlapped image

Figure 1.6 – rotating a camera about its nodal point to acquire multiple perspective projections with a common projection center also produces singular viewpoint panoramas

1.3 Prior Art

We can generate panoramic images with a help of single special panoramic camera or with the help of multiple standard cameras in conjunction with mosaicing algorithms to merge standard images into a panoramic image. If we want to generate mosaic 360 degrees panoramic images we have to position the cameras on a closed path, which is in most cases a circle. This section presents the cameras that are currently available to produce a panoramic image.

There are two main models for the acquisition of monoscopic panoramas: a sin-

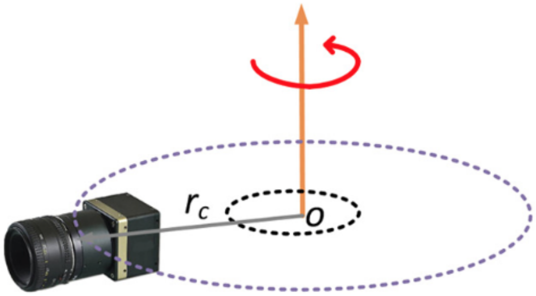


Figure 1.7 – circular non-central panorama

gular viewpoint model and a nonsingular viewpoint model. Any camera or addition technique available to produce omnidirectional imagery can be categorized into one of these two models. In the singular viewpoint acquisition model, there is a unique projection center that marks a single convergence point for all incident light rays. In this model, the catadioptric camera consists of a mirror (or mirrors) and a camera, as shown in figure 1.4. The camera captures the image which is reflected from the mirror. A dioptric camera is using a special type of lens, fish-eye lens, which increases the size of the camera's field of view, as shown in figure 1.5. In these cameras provide a single image which has a low resolution but can be used in real time.It is also possible to simulate single view point camera by rotating a camera around its nodal point, as shown in figure 1.6. This will provide high resolution but will be limited to the static scene where no object is moving. These panoramas are easily created by acquiring planar images to be mosaiced since they don't have any parallax problem.

In the case of a non-singular viewpoint model, the panoramic image is rendered using a centrally symmetric set of projection centers which are not spatially collocated. Cameras based on the non-singular viewpoint paradigm are more common than those based on a singular viewpoint model because the physical dimension of multiple camera configurations prevents sampling the scene from a single viewpoint. One of the methods for creating mosaiced panoramic images is capture images while rotating an off-center camera, as shown in figure 1.7 or by multiple cameras as illustrated by Google Jump Camera in figure 1.8.

Another way for generating panorama is using a pushbroom camera. When a moving camera captures a general static scene, the optical flow depends on the

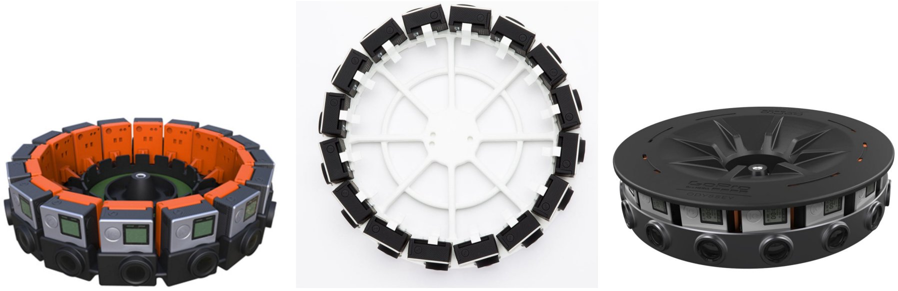


Figure 1.8 – circular non-central panorama by multiple cameras.they are constructed by JUMP-google

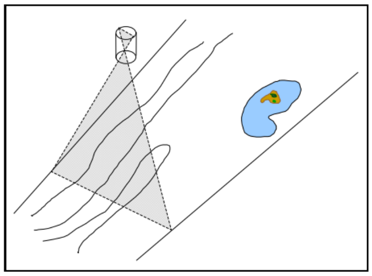


Figure 1.9 – An Pushbroom Camera

scene depth, making mosaicing difficult. A technique for mosaicing such scenes is the “slit camera”, or the “pushbroom camera”, used in aerial photography as shown in figure 1.9. This camera can be created as a 1-D sensor array which collects strips by “sweeping” the scene. The camera is modeled as a pin-hole camera moving along a linear trajectory in space with constant velocity and fixed orientation. Furthermore, the camera is constrained so that at any moment in time it images only points lying in one plane, called the view plane, passing through the center of the camera. Thus, at any moment of time, a $2D$ projection of the view plane into an image line takes place. The orientation of the view plane is fixed, and it is assumed that the motion of the camera does not lie in the view plane as described in Gupta and Hartley [1997].

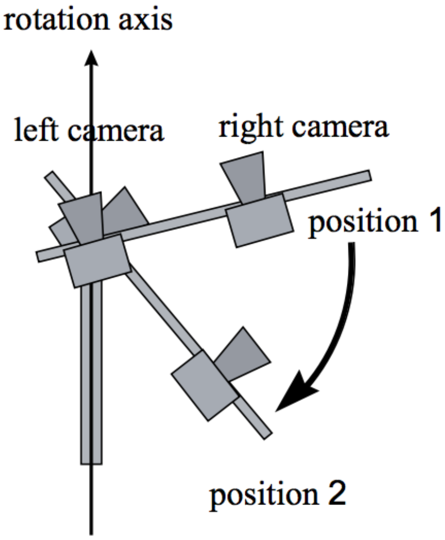


Figure 1.10 – Illustration of the rotation of the stereo camera set

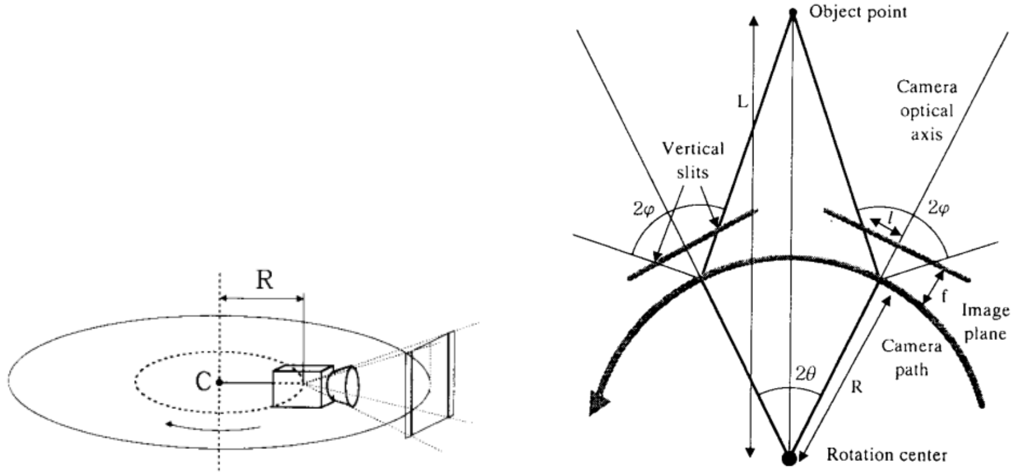


Figure 1.11 – Left:omni-directional stereo images are generated from two slit images taken by a camera whose focuse is at a distance of R from the rotation axis. Right:By measuring the angle of rotating the feature point from one slit to the other in the image, its range L can be estimated

Multiple viewpoint projections use different viewpoints for different viewing directions and is used mostly for special mosaicing applications. A multiple-center-of-projection image is an extension of a conventional image, characterized by having a set of cameras contributing to it, rather than only a single camera. Individual pixels or sets of pixels are acquired by different cameras, subject to certain constraints. A multiple-center-of-projection image consists of a $2D$ image and a parameterized set of cameras, meeting the following conditions: first, the cameras must lie on either a continuous curve or a continuous surface, second, each pixel is acquired by a single camera, third, viewing rays vary continuously across neighboring pixels and finally, two neighboring pixels must either correspond to the same camera or to neighboring cameras, are discussed in Wood et al. [1997], Rademacher and Bishop [1998]

All methods mentioned so far are used only for visualization purposes since the authors did not try to reconstruct the scene.

Panoramas can be made stereoscopic to provide depth perception to a human viewer. When two panoramic images are captured from two different viewpoints, the disparity, and the stereo perception will degrade as the viewing direction becomes closer to the baseline until no stereo will be apparent. Generation of image-based stereo panoramas by rotating a stereo head having two cameras was proposed in

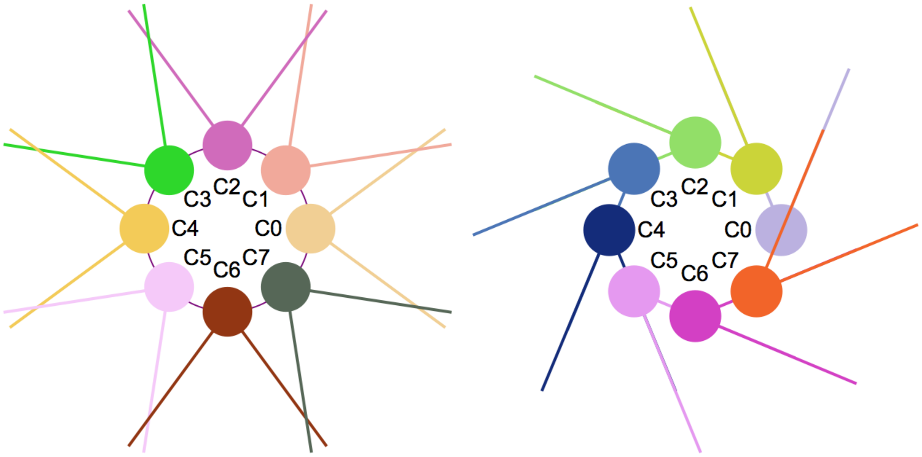


Figure 1.12 – Position of eight camera in two our design, left is Radial model and, right is Tangential model.

Huang and Hung [1998], Shum and Szeliski [1999] as shown in figure 1.10. A stereo head with two cameras is rotated and two panoramic mosaics are created from the two different cameras.

A Single rotating camera can also be sufficient under some conditions in Peleg and Ben-Ezra [1999], Ishiguro et al. [1992], Shum and Szeliski [1999]. In the case of a single moving camera, different sides of the same image are used to mosaic the two images for the different eyes. They assume the following conditions: the environment is static, a camera is mounted on rotation stage located in a fixed position in an indoor environment. Its motion is represented by three parameters: its location on the 2D floor and its azimuth. The camera is rotating around its vertical axis and its optical axis is kept horizontal, as shown in figure 1.11.

In this thesis, we do not consider depth estimation and scene reconstruction. Why ? This is because we only focus n the visual perception of the panorama , not for its photogrammetric use. The goal of this thesis is to propose new geometry of cameras which have less artifact problem caused by parallax. We compare two geometries, and in both models, cameras are setuped in a circle with the same radius. Our first model is a radial geometry that is common (see Jump Google Camera, Figure 1.8). Our second model is a tangential geometry where the optical axis of each camera is tangential to the camera circle, as shown in figure 1.12. Also, we use eight of the same normal cameras. We prefer to use normal cameras since they are easier to calibrate, cheaper, feature a higher resolution and image quality.

2 Camera Model

Since a single panoramic image is built from multiple images, it is essential to know precisely how the cameras are positioned relative to each other.This information is provided as a mathematical ”camera model” which can be computed with various methods, using a process called calibration.

This chapter explains the model we use in the panoramic camera setup and how to compute calibration.

In a typical panoramic setup, all cameras have the same internal parameters (fov, aspect ratio,...) and the positions are known since we place the cameras on a circle, with a specific orientation. However, there are variations from these ideal values so they should be estimated from sample points, to accurately reflect the real geometry of the setup.

2.1 Camera Model

A camera model is a function which maps our $3D$ world onto a $2D$ plane, called the image plane. Generally, this function is designed to closely model a real-world, physical camera. This section presents background material on camera models and camera parameters.

The classic model for a camera is a pinhole at a constant distance from an image plane. It this model light travels in a straight line, so, each point in the image defines a ray directed towards the scene. This gives us the standard perspective projection model which is only an approximation to the optical physics over the real camera. A pinhole camera has an infinitesimally small aperture, through which light accesses the camera and forms an image on the surface facing the aperture, and the camera operates a perspective transformation of $3D$ space to the two-dimensional image plane.

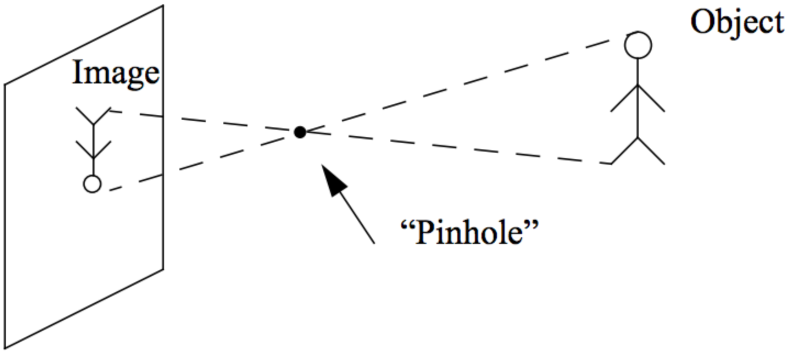


Figure 2.1 – Pinhole Camera Model

The basic assumption behind this model is that the relation between the world coordinate and the image coordinate is linear projective, straight lines project to straight lines. As shown in figure 2.1 the image in a pinhole model is inverted, thus, it is common to hold the geometric model of figure 2.2 equivalent to Figure 2.1 , in which the image is on the same side of pinhole as an object. As a result, the image is not inverted. It consists of a plane R called the image plane in which the image is formed into a perspective projection: a point C, the optical center, placed at a distance f, the focal length of the optical system, is applied to form the image m in the image plane of the $3D$ point M as the intersection of the line $<M,C>$with the image plane. The optical axis is the line passing through the optical center C and also perpendicular to the image plane, which it enters at a point c. Another plane of interest is the focal plane going through C and parallel to the image plane.

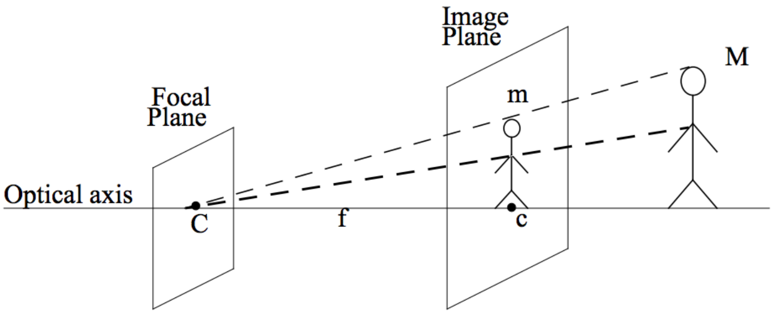


Figure 2.2 – Perspective Projection Model

It is useful, from a mathematical point of view, to think the world as embedded in a three-dimensional projective space, and the image plane is placed in a projective space in two-dimension. This facilitates the expression of the projective transformation. Points in the scene and image projective spaces are expressed as vectors in homogeneous coordinates. The projective transformation from the world points to the image points is given by:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} T \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} M \end{bmatrix} \begin{bmatrix} N \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where T is a matrix 3×4, which maps the scene point $p = [x, y, z, 1]^T$ to image point $w = [u, v]^t$ expressed in homogenous coordinates. M is a 3 by 4 matrix, N is a 4 by 4 matrix, the product of these two matrix equals to T , the Transformation Matrix.

2.2 Camera Parameters

The purpose of this section is to describe what the matrices should be when we relate $3D$ world coordinates to $2D$ image coordinates, which will allow us to define intrinsic and extrinsic parameters for the camera.

Extrinsic Parameters define the location and orientation of the camera with respect to a known world reference. These are the parameters that identify uniquely the transformation between the unknown camera reference and the known world reference. Typically, determining these parameters involves:

- Finding the translation vector between the relative positions of the origins of the two references.
- Finding the rotation matrix that brings the corresponding axes into the same origin.

The extrinsic parameters can be represented by N, a 4×4 displacement matrix which represents camera position and orientation. It expresses the displacement from the world coordinate system (X, Y, Z centered at O) to the camera-centered coordinate system ($\acute{X}$ $\acute{Y}$ $\acute{Z}$ centered at optical center C) by a translation and rotations around axes.

Internal Parameters represent the projection of a point in the scene, expressed in the camera frame of reference, to a point on the camera sensor. They are represented by M, which expresses what happens when we change the origin of the image coordinate system and the units on the u and v axes. The camera coordinate system is centered at the intersection $\acute{O}$ of the optical axis with the image plane and it has the same units on both axes. We go from the camera coordinate system to the image coordinates system, which has its origin at a point I in the image. If we denote the original of the camera coordinates system $\acute{O}$ by u_0 and v_0 , the scaling factor from the camera coordinate system to the image coordinate is k_u, k_v on both axes. As we decided that the Z axis of the camera centered coordinate system is pointing at the image plane, and the projection of X, Y axis on the image plane is $\acute{\acute{X}}$,$\acute{\acute{Y}}$, by the perspective transformation, an image point coordinates in the $\acute{\acute{X}}$,$\acute{\acute{Y}}$ coordinate system on the image plane are:

$$\acute{\acute{X}} = \frac{\acute{X} f}{\acute{Z}}$$
$$\acute{\acute{Y}} = \frac{\acute{Y} f}{\acute{Z}}$$

Now we convert the measurement units to some different units on both axes, scale by k_u in the $\acute{\acute{X}}$ axis and k_v in the $\acute{\acute{Y}}$ axis. Finally, we translate the origin to I. If the image center $\acute{O}$ has coordinates (u_0, v_0), the projection (u, v) in the image is:

$$u = u_0 + \frac{k_u \acute{X} f}{\acute{Z}}$$
$$v = v_0 + \frac{k_v \acute{Y} f}{\acute{Z}}$$

Using the homogeneous coordinate system this can be written as:

$$\begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} = \begin{bmatrix} M \end{bmatrix} \begin{bmatrix} \acute{X} \\ \acute{Y} \\ \acute{Z} \\ 1 \end{bmatrix}$$

where

$$\begin{bmatrix} M \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

f_x ,f_y ,c_x ,c_y are called camera intrinsic parameters, as they do not depend on the position and orientation of the camera in space. Knowledge of the intrinsic parameters allows us to perform metric measurements with a camera.

Now the transformation matrix T can be expressed by camera parameters:

$$\begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} M \end{bmatrix} \begin{bmatrix} N \end{bmatrix}$$

The intrincsic parameters matrix M is expressed in terms of 4 parameters f_x, f_y, c_x, c_y and The extinsic parameters matrix N is expressed in terms of 6 parameters R_x, R_y, R_z, T_x, T_y, T_z. Thus the whole image formation process can be expressed by:

$$\begin{aligned} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} &= \begin{bmatrix} T \end{bmatrix} \begin{bmatrix} \acute{X} \\ \acute{Y} \\ \acute{Z} \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 1 & r_1 2 & r_1 3 & 0 \\ r_2 1 & r_2 2 & r_2 3 & 0 \\ r_3 1 & r_3 2 & r_3 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \acute{X} \\ \acute{Y} \\ \acute{Z} \\ 1 \end{bmatrix} \end{aligned}$$

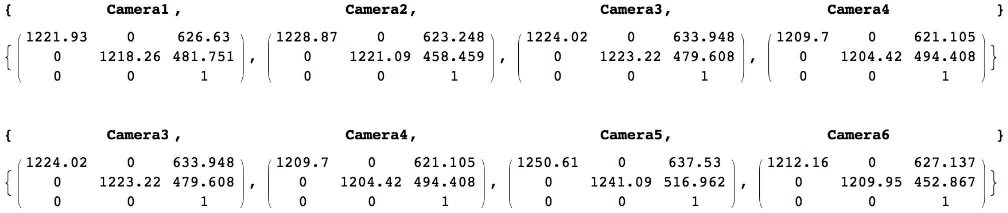
When projecting a world point onto the screen, the depth information is lost in the conversion ; only the x and y position on the screen remains. Where $(u, v, w)^T$ is a $2D$ projective representation of the projected point on the screen. The Euclidian sensor coordinate (x_s, y_s) is obtained from (u, v, w) as:

$$(x_s, y_s) = (\frac{u}{w}, \frac{v}{w}) \tag{2.1}$$

The camera center position (the camera origin) can be retrieved from the inverse of the matrix T by taking the last column $(Cam_x, Cam_y, Cam_z)^T$of T^{-1}.

$$T^{-1} = \begin{bmatrix} ... & ... & ... & Cam_x \\ ... & ... & ... & Cam_y \\ ... & ... & ... & Cam_z \\ ... & ... & ... & 1 \end{bmatrix}$$

In our configuration, we have 8 cameras that require external and internal parameters calibration. We present two models, tangential and radial, which differ only by their external parameters. The internal parameters are the same in both models as shown in figure 2.3, but external parameters are different, as shown in tangential model in figure 2.4 and radial model in figure 2.5.



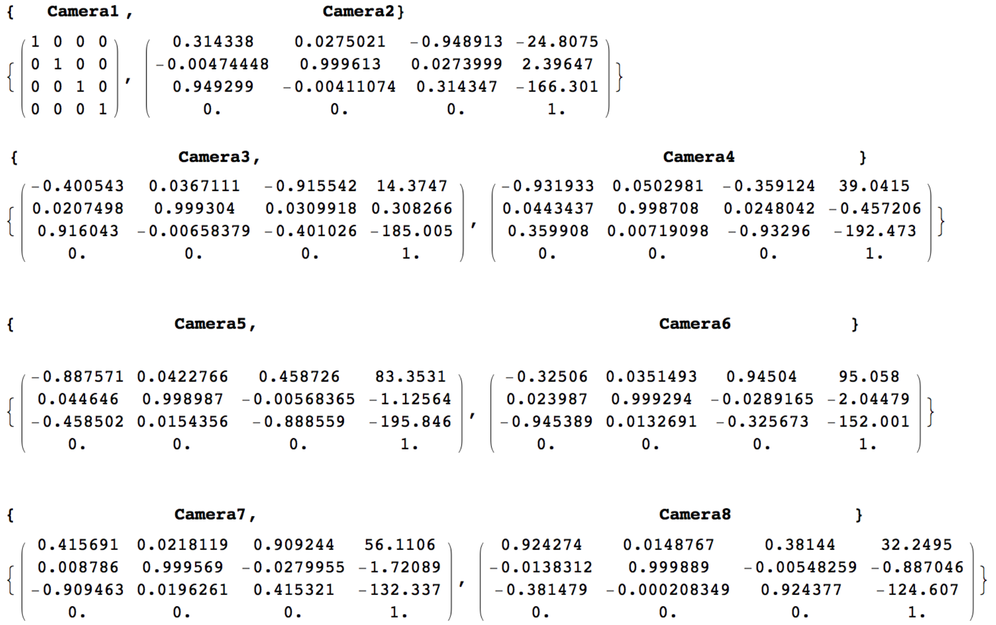
$$\{\ \textbf{Camera1},\quad \textbf{Camera2},\quad \textbf{Camera3},\quad \textbf{Camera4}\ \}$$

$$\left\{\begin{pmatrix} 1221.93 & 0 & 626.63 \\ 0 & 1218.26 & 481.751 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1228.87 & 0 & 623.248 \\ 0 & 1221.09 & 458.459 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1224.02 & 0 & 633.948 \\ 0 & 1223.22 & 479.608 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1209.7 & 0 & 621.105 \\ 0 & 1204.42 & 494.408 \\ 0 & 0 & 1 \end{pmatrix}\right\}$$

$$\{\ \textbf{Camera3},\quad \textbf{Camera4},\quad \textbf{Camera5},\quad \textbf{Camera6}\ \}$$

$$\left\{\begin{pmatrix} 1224.02 & 0 & 633.948 \\ 0 & 1223.22 & 479.608 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1209.7 & 0 & 621.105 \\ 0 & 1204.42 & 494.408 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1250.61 & 0 & 637.53 \\ 0 & 1241.09 & 516.962 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1212.16 & 0 & 627.137 \\ 0 & 1209.95 & 452.867 \\ 0 & 0 & 1 \end{pmatrix}\right\}$$

Figure 2.3 – Internal parameters for eight cameras in our model



$$\{\ \textbf{Camera1},\quad \textbf{Camera2}\}$$

$$\left\{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0.314338 & 0.0275021 & -0.948913 & -24.8075 \\ -0.00474448 & 0.999613 & 0.0273999 & 2.39647 \\ 0.949299 & -0.00411074 & 0.314347 & -166.301 \\ 0. & 0. & 0. & 1. \end{pmatrix}\right\}$$

$$\{\ \textbf{Camera3},\quad \textbf{Camera4}\ \}$$

$$\left\{\begin{pmatrix} -0.400543 & 0.0367111 & -0.915542 & 14.3747 \\ 0.0207498 & 0.999304 & 0.0309918 & 0.308266 \\ 0.916043 & -0.00658379 & -0.401026 & -185.005 \\ 0. & 0. & 0. & 1. \end{pmatrix}, \begin{pmatrix} -0.931933 & 0.0502981 & -0.359124 & 39.0415 \\ 0.0443437 & 0.998708 & 0.0248042 & -0.457206 \\ 0.359908 & 0.00719098 & -0.93296 & -192.473 \\ 0. & 0. & 0. & 1. \end{pmatrix}\right\}$$

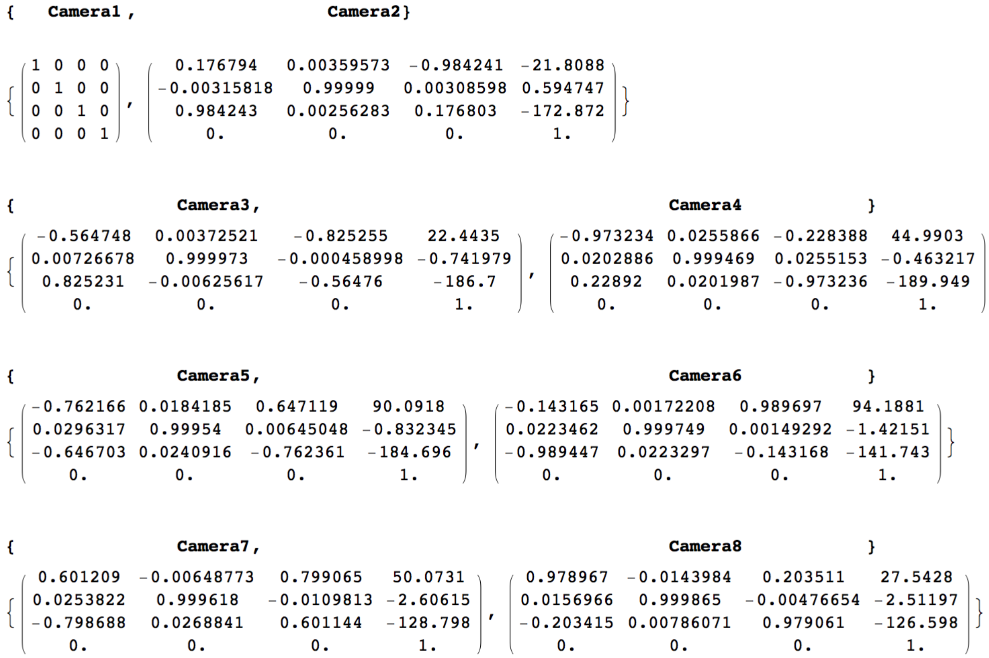
$$\{\ \textbf{Camera5},\quad \textbf{Camera6}\ \}$$

$$\left\{\begin{pmatrix} -0.887571 & 0.0422766 & 0.458726 & 83.3531 \\ 0.044646 & 0.998987 & -0.00568365 & -1.12564 \\ -0.458502 & 0.0154356 & -0.888559 & -195.846 \\ 0. & 0. & 0. & 1. \end{pmatrix}, \begin{pmatrix} -0.32506 & 0.0351493 & 0.94504 & 95.058 \\ 0.023987 & 0.999294 & -0.0289165 & -2.04479 \\ -0.945389 & 0.0132691 & -0.325673 & -152.001 \\ 0. & 0. & 0. & 1. \end{pmatrix}\right\}$$

$$\{\ \textbf{Camera7},\quad \textbf{Camera8}\ \}$$

$$\left\{\begin{pmatrix} 0.415691 & 0.0218119 & 0.909244 & 56.1106 \\ 0.008786 & 0.999569 & -0.0279955 & -1.72089 \\ -0.909463 & 0.0196261 & 0.415321 & -132.337 \\ 0. & 0. & 0. & 1. \end{pmatrix}, \begin{pmatrix} 0.924274 & 0.0148767 & 0.38144 & 32.2495 \\ -0.0138312 & 0.999889 & -0.00548259 & -0.887046 \\ -0.381479 & -0.000208349 & 0.924377 & -124.607 \\ 0. & 0. & 0. & 1. \end{pmatrix}\right\}$$

Figure 2.4 – External parameters for eight cameras in tangential model



```
{    Camera1 ,                    Camera2}

 (1 0 0 0)   (  0.176794    0.00359573  -0.984241  -21.8088 )
{(0 1 0 0) , ( -0.00315818  0.99999     0.00308598  0.594747) }
 (0 0 1 0)   (  0.984243    0.00256283  0.176803  -172.872  )
 (0 0 0 1)   (  0.          0.          0.           1.     )

{                 Camera3,                                            Camera4              }
 ( -0.564748   0.00372521   -0.825255      22.4435  )   ( -0.973234  0.0255866  -0.228388   44.9903  )
{(  0.00726678 0.999973    -0.000458998  -0.741979 ) , (  0.0202886 0.999469   0.0255153  -0.463217 ) }
 (  0.825231  -0.00625617   -0.56476     -186.7     )   (  0.22892   0.0201987 -0.973236  -189.949   )
 (  0.         0.            0.            1.       )   (  0.        0.         0.           1.      )

{                 Camera5,                                            Camera6              }
 ( -0.762166  0.0184185   0.647119     90.0918  )   ( -0.143165  0.00172208   0.989697    94.1881  )
{(  0.0296317 0.99954     0.00645048  -0.832345 ) , (  0.0223462 0.999749     0.00149292 -1.42151  ) }
 ( -0.646703  0.0240916  -0.762361   -184.696   )   ( -0.989447  0.0223297   -0.143168  -141.743   )
 (  0.        0.          0.           1.       )   (  0.        0.           0.           1.      )

{                 Camera7,                                            Camera8              }
 (  0.601209  -0.00648773   0.799065    50.0731  )   (  0.978967  -0.0143984    0.203511    27.5428  )
{(  0.0253822  0.999618    -0.0109813  -2.60615  ) , (  0.0156966  0.999865    -0.00476654 -2.51197  ) }
 ( -0.798688   0.0268841    0.601144  -128.798   )   ( -0.203415   0.00786071   0.979061  -126.598   )
 (  0.         0.           0.           1.      )   (  0.         0.           0.           1.      )
```

Figure 2.5 – External parameters for eight cameras in radial model

2.3 Camera Lens Correction

Lens corrections help offset imperfections present in nearly every camera image. These might include darkening near the corners of the frame, otherwise straight lines appearing curved, or color fringes near edge detail. Even though these often aren't obvious in the original photo, their removal almost always provides the benefits. However, lens corrections also have the potential to make images worse if not carefully computed, and depending on the subject, some imperfections can actually be beneficial. The three most common lens corrections aim to address one of the following:

1. Vignetting. This appears as a progressive darkening toward the edges of the image.
2. Distortion. This appears as otherwise straight lines bending inwards or outwards.
3. Chromatic Aberration. This appears as color fringing along high contrast edges.

However, lens correction software is typically only able to fix certain types of each imperfection above, so being able to identify them is key. The following sections describe the types and causes of each imperfection, when correction is possible, and how to minimize imperfections in the first place.

Vignetting This describes the gradual fall-off of light towards the corners of your image and is perhaps the most straightforward lens imperfection to observe and correct. Vignetting can be grouped into two general categories:

1. **Physical Vignetting** Often not correctable except by cropping or manual brightening/cloning. Appears as a strong, abrupt darkening usually only in the very corners of an image. Caused by stacked/large filters, lens hoods or other objects physically blocking light near an image's edges.
2. **Internal Vignetting** Usually easily correctable. Appears as a gradual and often subtle darkening away from the image's center. Caused by the inner workings of your particular lens and camera. It's typically most apparent at lower f-stops, with zoom and wide angle lenses, and when focusing on distant objects. Digital SLR cameras with cropped sensors are also less susceptible to vignetting because the darker edges get cropped out (when using full-frame lenses).

Distortion This can give otherwise straight lines the appearance of bending inward or outward and can influence depth perception. The most common categories of image distortion include:

1. **Pincushion Distortion** Appears when otherwise straight lines curve inward. Typically caused by telephoto lenses, or at the telephoto end of a zoom lens.
2. **Barrel Distortion** Appears when otherwise straight lines curve outward. Typically caused by wide angle lenses, or at the wide end of a zoom lens.
3. **Perspective Distortion** Appears when otherwise parallel lines converge. Caused by the camera not facing these parallel lines perpendicularly. With trees and architecture, this usually means that the camera isn't pointed at the horizon. We can not correct this type of distortion because it is part of the natural image formation process.

Chromatic Aberration appears as unsightly color fringes near high contrast edges. Unlike the other two lens imperfections, chromatic aberrations are typically only visible when viewing the image on-screen at full size, or in large prints. Chromatic aberration is perhaps the most varied and complicated lens imperfection, and its prevalence is highly dependent on the subject matter.

2.3.1 Limitations

Some limitations that are not addressed by this thesis are listed here:

- **Lack of texture** Occurs when some part of the image does not have any texture, making it difficult to match to another image to retrieve a valuable depth (ill-posed problem).
- **Non-Lambertian surfaces**
- **Translucency**
- **Image noise**
- **Imperfect calibration** When the calibration is not precise, unrelated pixels are matched together and will lead to imprecise results.
- **Differences in exposure and white balance** Differences in camera photometric parameters from one pose to another is not handled in this thesis. Some preliminary steps like histogram equalization and color corrections would be needed to remove these differences.

Even with all these restrictions, the method developed in this thesis gives good results and is usable in many contexts. Since the goal of this thesis is to compare different cameras geometries, it is more important to do accurate comparison than to achieve the ultimate image quality.

2.4 Homography

This part aims at explaining homography. A homography is a projective transformation from one plane to another. In the context of this thesis, we will use homographies mostly during calibration, because we rely on images of planar objects (checkerboards, LCD monitor).

1. **Scene plane to image pixels**
   Suppose we have a planar surface in the world (e.g. a wall, a ground plane) and we view it with a camera with projection matrix P. The planar surface

is $2D$ and so we can give it a coordinate system (s, t). Points on the plane are situated in the $3D$ world and their $3D$ positions can be expressed in XYZ camera coordinates. We can transform from (s, t) to camera coordinates XYZ by multiplying $(s, t, 1)$ by a 4×3 matrix, as follows:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} a_x & b_x & X_0 \\ a_y & b_y & Y_0 \\ a_z & b_z & Z_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}$$

The first two columns can be interpreted as direction vectors corresponding to the coordinate system's basis vectors, namely where $(1, 0, 0)$ and $(0, 1, 0)$ are mapped to. The third column is the $3D$ position of the origin of the plane. The image pixel (x, y) corresponding to a point (s, t) in the scene plane is obtained by:

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} P \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

These two mappings together define a 3×3 matrix H mapping $(s, t, 1)$ to (wx, wy, w),

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} H \end{bmatrix} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}$$

Since we are working in homogeneous coordinates, the relationship between two corresponding points x and $\acute{x}$ shown in equation(2.1).where w is any non-zero constant,$(x, y, 1)^T$ represents $\acute{x}$ and $(s, t, 1)^T$ represents x and

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

Such a matrix H is called a homography.

2. **Two Cameras, One Scene Plane**

Suppose the same scene plane is viewed by a second camera which would have a different XYZ coordinate system. We would now have two homographies H_1 and H_2, defined by the two cameras. This implies that the composite mapping $H_2 H_1^{-1}$ maps pixels in the first camera to pixels in the second camera. That is, each camera defines a homography of the form

$$H^{-1} \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} \tag{2.2}$$

but the right side is the same for both since it is independent of the camera, so we just equate the left sides for the two cameras. Since the product of two 3 × 3 invertible matrices is itself an invertible matrix, we see that the mapping from pixels of the first camera to pixels of the second camera is a homography. This the construction relies critically on the scene being a planar surface.

3. **Panorama**

Surprisingly, homographies can arise for general scenes as well (non-planar), particularly if it is seen by two cameras from the same center of projection. In practice, this occurs when you have one camera and you use it to take more than one image by rotating the camera around the center of projection. This is often done is modern digital cameras when you try to stitch images together to make panorama images. To see that two images taken under these conditions are related by a homography, let (X, Y, Z) be a scene point visible to the first position of the camera and written in the first position of camera coordinate system.

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = K \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}$$

If the position of the second camera's coordinate system is a rotation R

relative to first camera position, we have

$$\begin{bmatrix} \acute{w}x \\ \acute{w}y \\ \acute{w} \end{bmatrix} = K\acute{R} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}$$

then

$$\begin{bmatrix} \acute{w}x \\ \acute{w}y \\ \acute{w} \end{bmatrix} = K\acute{R}K^{-1} \begin{bmatrix} wx \\ wy \\ w \end{bmatrix}$$

Now $K.R.K^{-1}$ is an invertible 3×3 matrix and is a homography. Notice that the distance of the points from the camera plays no role here. Rotating the camera will not have any effect on the directions from which the scene points are seen. So its no really surprising that the distance to the points plays no role.

The Direct Linear Transform (DLT) algorithm is a simple algorithm used to solve for the homography matrix H given a sufficient set of point correspondences. Dividing the first row of equation (2.1) by the third row and the second row by the third row we get the following two equations:

$$-h_1x - h_2y - h_3 + (h_7x + h_8y + h_9)u = 0$$
$$-h_4x - h_5y - h_6 + (h_7x + h_8y + h_9)u = 0$$

Equations (2.2) and (2.3) can be written in matrix form as:

$$A_ih = 0$$

$$A_ih = \begin{pmatrix} -x & -y & -1 & 0 & 0 & 0 & ux & uy & u \\ 0 & 0 & 0 & -x & -y & -1 & vx & vy & v \end{pmatrix} \tag{2.3}$$

$$h = \begin{pmatrix} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 & \end{pmatrix} \tag{2.4}$$

Since each point correspondence provides 2 equations, 4 correspondences are sufficient to solve for the 8 degrees of freedom of H. The restriction is that no 3 points can be collinear. Four 2×9 A_i matrices (one per point correspondence) can be stacked on top of each other to get a single 8×9 matrix A. The $1D$ null space of A is the solution space for h. In many cases, we may be able to use more than 4 correspondences to ensure a more robust solution. However many point correspondences are used, if all of them are exact then A will still have rank 8 and there will be a single homogeneous solution.

2.5 Camera Calibration

Camera calibration is the process of determining the intrinsic and extrinsic parameters of the camera setup. The intrinsic parameters are those specific to the camera, such as the focal length, principal point, and lens distortion. Extrinsic parameters refer to the $3D$ position and orientation of the camera.

Geometric camera calibration also referred to as camera resectioning, estimates the parameters of a lens and image sensor of an image or video camera. You can use these parameters to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene. These tasks are used in applications such as machine vision to detect and measure objects. We describe two methods for camera calibration here, first with $2D$ objects, plane-based technique, and structured light . We use the structured light technique for obtaining intern and external parameters in our cameras.

Plane-based Technique

Without loss of generality, we assume all points lie in a plane, their Z component is 0 in world coordinates. Using a $2D$ checkerboard pattern, we set the origin of the world coordinate system to the corner of the checkerboard. Let's denote the i^{th} column of the rotation matrix R by r_i,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_11 & r_12 & r_13 & t_1 \\ r_21 & r_22 & r_23 & t_2 \\ r_31 & r_32 & r_33 & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

Thus, we can delete the 3rd column of the extrinsic parameter matrix.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 1 & r_1 2 & t_1 \\ r_2 1 & r_2 2 & t_2 \\ r_3 1 & r_3 2 & t_3 \end{bmatrix}}_{Homography} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{K} \quad \underbrace{\begin{bmatrix} r_1 1 & r_1 2 & t_1 \\ r_2 1 & r_2 2 & t_2 \\ r_3 1 & r_3 2 & t_3 \end{bmatrix}}_{r_1, r_2, t}$$

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}$$

So,

$$r_1 = k^{-1} h_1$$
$$r_2 = k^{-1} h_2$$

Note that (r_1, r_2, r_3) form an orthonormal basis, thus

$$r_1^T r_2 = 0 \rightarrow h_1^T k^{-T} k^{-1} h_2 = 0$$
$$\|r_1\| = \|r_2\| = 1 \rightarrow h_1^T k^{-T} k^{-1} h_1 - h_2^T k^{-T} k^{-1} h_2 = 0$$

$k^{-T} k^{-1}$ is symmetric and positive definite, K can be calculated using Cholesky factorization, since reordering of two equation above, leads to the system of the final equations. In computer vision, the calibration of the camera is common and often rely on OpenCV, using a function called ”calibrateCamera”, which estimates the intrinsic camera parameters and extrinsic parameters for each of the views. The algorithm is based on Zhang [2000]. The coordinates of $3D$ object points and their corresponding $2D$ projections in each view must be specified. That may be achieved by using an object with a known geometry and easily detectable feature points. Such an object is called a calibration rig or calibration pattern, and OpenCV has built-in support for a checkerboard as a calibration rig. Currently, initialization of intrinsic parameters is only implemented for planar calibration patterns, where

Z-coordinates of the object points must be all zeros. $3D$ calibration rigs can also be used as long as an initial camera matrix is provided. The algorithm performs the following steps:

1. Project $3D$ points to the image plane given intrinsic and extrinsic parameters.
2. Compute extrinsic parameters given intrinsic parameters, a few $3D$ points, and their projections.
3. Estimate intrinsic and extrinsic camera parameters from several views of a known calibration pattern, every view is described by several $3D-2D$ point correspondences.
4. Estimate the relative position and orientation of the stereo camera “heads” and compute the rectification transformation that makes the camera optical axes parallel.

To get good results, we must establish good correspondences. This is usually achieved using a checkboard, but it only provides few points. Also, this method requires a large overlap of the cameras field of views, which is hard to achieve in a panoramic setup with multiple cameras. To solve this problem of lack of overlap, we preferred to use structured light to get dense correspondences which are also more accurate.

2.6 Structured light

When thinking of $3D$ imaging, structured-light systems are probably the most familiar to everyone. The original Microsoft Kinect is an example of a structured light methodology. The basic principle is to illuminate the scene with a series of distinctive light patterns, usually featuring alternating black and white stripes. A standard $2D$ camera is used to acquire the scene. Structured light actually has two subclasses: fixed-pattern and multi-pattern. A fixed-pattern system uses a single pattern, while a multi-pattern employs a sequence of patterns as shown in figure 2.6. The sequence of reflected images is processed to generate the depth of the objects in the scene. The capture system consists of standard $2D$ camera modules, enabling the capture of high-resolution $2D$ images as well as low-cost camera implementations. With a fixed-pattern system, a simple filter is usually placed over the light source to provide the structured scene illumination. Again this can be a fairly low-cost implementation. In a multi-pattern system, one must either dynamically change the filter placed over the illumination system, or use a system that can modulate the light appropriately. Texas Instruments' DLP systems offer one method to create such a sequence of patterns. With the DLP illumination subsystem, it's possible to sequence a high number of patterns to maximize accuracy.

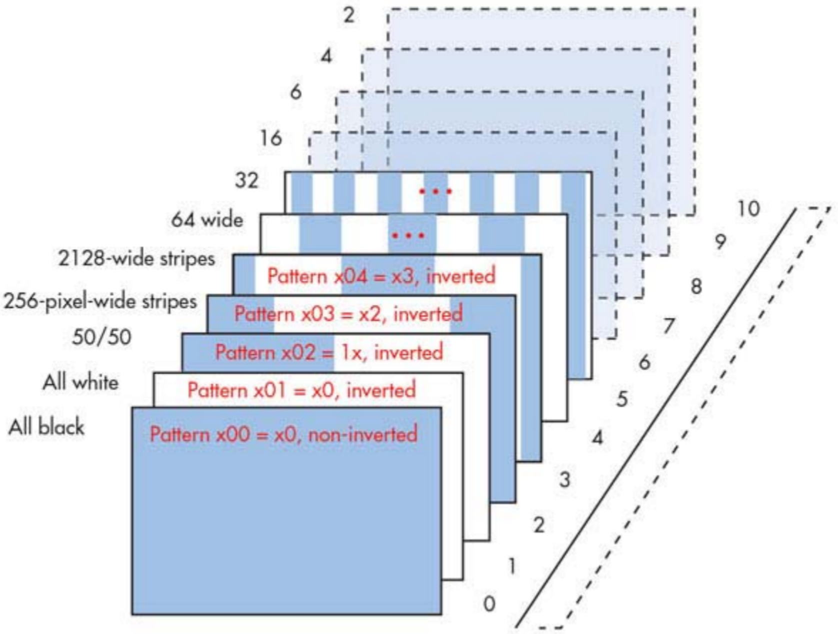

2
4
6
16
32
64 wide
2128-wide stripes
256-pixel-wide stripes
50/50
All white
All black
Pattern x04 = x3, inverted
Pattern x03 = x2, inverted
Pattern x02 = 1x, inverted
Pattern x01 = x0, inverted
Pattern x00 = x0, non-inverted
10
9
8
7
6
5
4
3
2
1
0

Figure 2.6 – Internal Stripe Pattern Example

Since multiple frames of data need to be analyzed to track how the pattern's reflections change over time, inherent latency exists within a structured light system. In addition, the camera system's native resolution is reduced because the image processing requires interpolation between pixels. Due to the need for the patterns of light to travel a minimum distance so that the image-processing system can accurately interpret the returned disruptive light patterns, it does not scale very well to close interaction applications. The final drawback is that fairly precise alignment between the illumination source and the camera system is required to properly analyze the light patterns and how they move in the scene.

2.7 Panoramic Images

Panoramic images create a photo-realistic, three-dimensional, navigable environment. A panoramic image captures the surroundings of a location in a 360° cylindrical or spherical view. This section introduces cylindrical panoramic images.

2.7.1 Cylindrical Coordinates

Assume that the camera is at its canonical position, where the optic axis is aligned with the z-axis and the y-axis is vertical. When a picture is taken, the light goes through the lens and falls on the image plane of the camera. The picture gets projected to the image plane. Consider a cylinder of unit radius centered at $O = (0, 0, 0)$. Each point A on this cylinder can be parameterized by an angle θ and a height h. θ is the angle that the projection on the xz plane of vector OA makes with the x-axis. h is the component of the vector in the direction of the y-axis. Consider a point $P = (x, y, z)$ in space. Let's say the projection of this point is specified by $(0, \theta)$ on the surface of the cylinder. By similar triangles we have:

$$(\sin(\theta), h, \cos(\theta)) \propto (x, y, f)$$

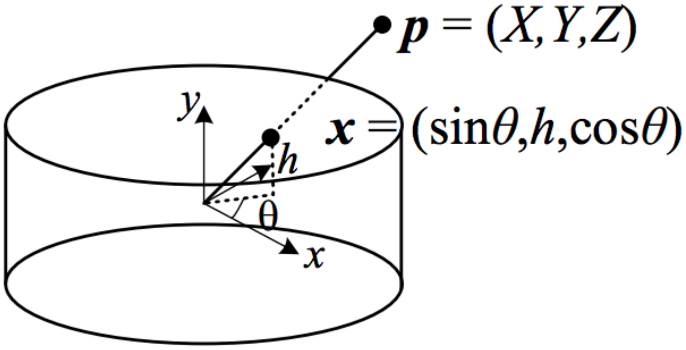

$\boldsymbol{p} = (X,Y,Z)$
y
$\boldsymbol{x} = (\sin\theta, h, \cos\theta)$
h
θ
x

Figure 2.7 – Projection from $3D$ to a cylindrical

From this relationship, a formula that maps a point to its projection on the cylindrical surface can be derived as (Szeliski and Schum 1997):

$$\begin{aligned}\acute{x} &= s\theta = s\tan^{-1}\frac{x}{s},\\ \acute{y} &= sh = s\frac{y}{\sqrt{x^2+f^2}},\end{aligned}$$

where s is an arbitrary scaling factor, sometimes called the radius of the cylinder. To reduce the distortion in the center of the image, s can be set to be f.

The image is wrapped around a cylinder with an appropriate radius. That is, the width of the panoramic image determines the radius of the cylinder.

$$R = \frac{imagewidth}{2\pi}$$

Warping an image directly might create holes in the warped picture, so we will use inversed warping by interpolation:

$$\begin{aligned}x &= f\tan(\theta) = f\tan(\frac{x}{s}),\\ y &= h\sqrt{x^2+f^2} = \frac{\acute{y}}{s}\sqrt{1+\tan^2(\frac{\acute{x}}{s})} = f\frac{\acute{y}}{s}\sec\frac{\acute{x}}{s}\end{aligned}$$

3 Comparison of Radial and Tangential Geometries for Cylindrical Panorama

This chapter presents the following paper, submitted to International Conference on 3D Vision:

Comparison of Radial and Tangential Geometries for Cylindrical Panorama, Faezeh Amjadi and Sebastien Roy submitted to the 4th International Conference on 3D Vision (3DV), 2016.

3.1 Abstract

This paper presents a new approach which build 360° cylindrical panoramic images from multiple cameras. In order to ensure a perceptually correct result, mosaicing typically requires either a planar or near-planar scene, parallax-free camera motion between source frames, or a dense sampling of the scene. When these conditions are not satisfied, various artifacts may appear. There are many algorithms to overcome this problem. The panoramic setup has cameras placed evenly around a circle. Instead of looking outward, which is the traditional ”radial” configuration, we propose to make the optical axes tangent to the camera circle, a ”tangential” configuration. We will demonstrate that this configuration is very insensitive to depth estimation, which reduces stitching artifacts. This property is only limited by the fact that tangential cameras usually occlude each other along the circle. Besides an analysis and comparison of radial and tangential geometries, we provide an experimental setup with real panoramas obtained in realistic conditions.

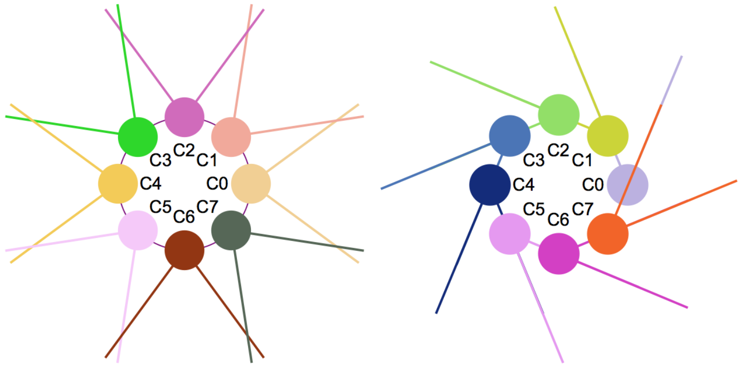


Figure 3.1 – Position of eight camera in two our design, left is Radial model and, right is Tangential model

3.2 Introduction

People have always been fascinated about capturing the entire view of the scenes. Panoramic images are a natural way of capturing a wide visual field of view. A panorama is a compact representation of the environment viewed from one 3D position. While a single photograph of a scene is just a static snapshot with a limited field of view captured from a single viewpoint, a panorama combines multiple input images, typically with some overlap, to produce an output with all wide field of views. Many techniques have been proposed to extend the ways in which a scene can be visualized by taking multiple photographs.

A number of techniques have been developed for capturing panoramic images of real-world scenes. Panoramas can be created on an extended planar image surface, on a cylinder Chen [1995], Szeliski [1996], Tsai and Huang [1981], or on a sphere Szeliski and Shum [1997], Gumustekin and Hall [1996]. Panorama can be generated with the setup a camera or multiple cameras or combined with mirrors and cameras. The simplest mosaics are created by a camera from a set of images whose mutual displacements are pure image-plane translations Chen [1995], Shum and Szeliski [1997], Sawhney et al. [1998], Szeliski [1996] , or single planar scene in which both center of projection and image sensor free to translate and rotate Huang and Hung [1998]. In this method a camera is mounted on a rotational robotic arm, so the optical center of the camera is offset from the vertical axis of rotation, as shown in figure (1-a), The camera is looking outward from the rotational center.

Panoramic images are generated by repeatedly shifting the rotational arm by an angle which corresponds to a single pixel column of the captured image. By assembling the center columns of these images, we get a mosaic panoramic image, These methods contain little or no parallax effects. Because there is no motion parallax, you can't see the relative depth of points in the scene as you rotate, so the images might as well be located on any plane. Another way to build panoramic images is by taking another column out of a captured image and mosaicing the columns. Such panoramic images are called multi-perspective panoramic imagesRademacher and Bishop [1998], Wood et al. [1997], because it is possible to change the selected column. The crucial property of two or more multi-perspective panoramic images is that they capture the information about the motion parallax effect since the columns forming the panoramic images are captured from different perspectives, from this panorama it is possible to recover depth using stereoscopic $3d$ reconstruction and dense sampling, which is time-consuming. Panoramic images can be created by using cameras with a wide field of view (fish-eye) lens that works for a dynamic scene but the images must undergo substantial distortions, and map an entire scene into the limited resolution of a video camera compromises image qualityWeng et al. [1992], Basu and Licardie [1995], Xiong and Turkowski [1997]. Catadioptric cameras, which relies on mirrors, can also be used to panoramic imagesNayar and Peri [2001], Nayar and Karmarkar [2000]. Such panoramic cameras are appropriate for low-resolution reconstruction of dynamic scenes and for motion estimation. It is also known that epipolar geometry can be simply generalized when the camera is single view point, which is the case for hyperbolic mirrors viewed for perspective cameras and parabolic mirrors viewed from orthographic cameras Svoboda and Pajdla [2002].

The another way to produce a panorama is to capture images simultaneously from multiple cameras, with some overlap. Creating a panorama without overlap require specialized algorithms. These techniques have to decide where to cut the images to remove the overlaps. Since the overlap depends on scene depth, this is a difficult problem. This is why we also need to rely on algorithms to seamlessly blend overlapping images, to make cutting error less visible, and also reduce the problem of scene motion and lens distortion. Early methods estimate a 2D transformation, a homography, between two input images and provide it to align them Szeliski and Shum [1997], Brown and Lowe [2007]. When the input images have little parallax, this homography work well, but when input images have large parallax, artifacts like ghosting happens. Since a homography cannot account for parallax, these methods require images that should be taken from the same viewpoint or the scene should be exactly planar.

Local warping Shum and Szeliski [1998] guided by motion estimation can be used to reduce overlap problems. Also, advanced image composition techniques, such as seam cuttingAgarwala et al. [2004], Kwatra et al. [2003] and blending Burt and Adelson [1983], Pérez et al. [2003], can reduce the ghosting artifacts. However, they cannot address significant misalignment. Recent image stitching methods use spatially-varying warping algorithms to align input imagesLin et al. [2011], Zaragoza et al. [2013]. While it can handle parallax better than homography, it still cannot work well on images with large parallax. The recent dual-homography warping method can stitch images with parallax, but it requires the scene content to be modeled by two planes Gao et al. [2011].

Another method is Parallax-tolerant Image Stitching which first aligns input images, then use a content-preserving warping and seam cutting algorithm to find a seam to piece aligned images together and explores both image content and geometric alignment and finally employ a multi-band blending algorithm to create the final stitching result, this method handles images with large parallax well, but this method be failed when input images have very large parallax or are full of salient structuresZhang and Liu [2014].

In this paper, we intend to work on the geometry of cameras instead of developing stitching algorithms. When capturing hand-held panoramas or using an array of cameras, parallax has to be accounted for. A common strategy, which is also the basis for our work, is to change the orientation of cameras to reduce these artifacts. Our goal is to compare various camera geometries and figure out in which conditions we can obtain the highest qualities panorama with the least algorithmic effort. The general design of our system is all cameras are placed on a common circle and are symmetric and have the same distance from the center of the circle. Each camera optical axis will varies from ”radial” configuration to ”tangential” configuration (see figure(1-b)).

In the radial configuration, optical axes are perpendicular to the camera’s circle, in the tangential configuration is that the optical axis of each camera is tangential to camera’s circle. In both configuration each camera has the same position, only the optical axis varies. To the best of our knowledge, these issues have not been explored in previous works.

This paper, we will show that these configurations respond very differently to ghosting, blind spot and other artifacts induced by parallax. Section 2 introduces the camera model. Section 3 will present the calibration of the camera setup. Section 4 explains the panorama generation and section 5 presents experimental result with real cameras.

3.3 Camera Model

In this section, we explain our panoramic camera model. The models are designed to generate panorama by a number of cameras that are placed on the circumference of a circle and all cameras are equidistant from each other and at the same distance from the center. In other words, the angle between two consecutive cameras is the same. The number of the camera must be sufficient to get overlap between successive cameras. In this model all cameras have the same optical axis orientation relative to the normal of the camera circle. We consider to cases. First, the optical axis is parallel to the normal, yielding a ”radial” configuration as shown in fig(1-Left). Second, the optical axis is perpendicular to the normal, yielding a ”tangential” configuration as shown in Figure(1-Right). Any optical axis orientation between radial to the tangential configuration is also possible.

All the camera configurations provide similar panorama but mostly differ in the sensitivity to the parallax. Our panorama geometry is cylindrical that commonly used because of their ease of construction. In Practice, a single camera can be rotated around the circle center to provide an image sequence. Also, it is possible to use multiple fixed cameras to provide a similar sequence. For static scene both approaches works fine, but if a dynamic scene multiple cameras must be used because they can capture simultaneously. Assuming each camera is calibrated, all perspective images can be warped into cylindrical coordinates, so can be stitched together. Since the parallax is present, knowledge of depth will have an effect on the stitching process.

Suppose we have two cameras c_1, c_2 as shown in figure 3.2, the position of camera is on the first circle with center c and radius r_1, and angle θ between two

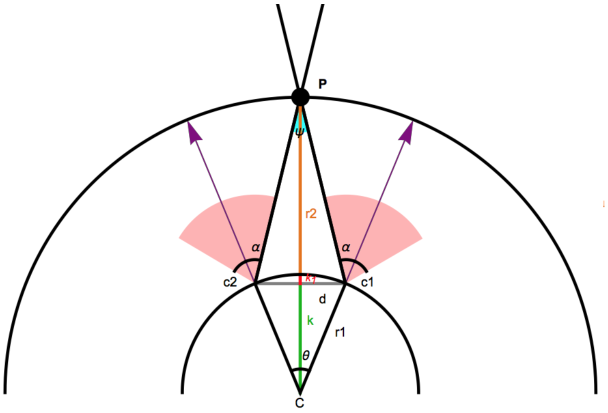


Figure 3.2 – Radial camera model. The FOV of each camera must be increased from the minimum (45°) to account for the position of P.

cameras rather to center is $\frac{2\pi}{n}$ where n is the number of cameras. These cameras are located in distance of $2d$ and radius of second circle is r_2 which represent the reference depth for stitching. The field of view for each camera is α and a reference point p is located between the cameras at distance $(r_1 + r_2)$. The purple arrows are optical axis which can vary from zero to 90° to obtain configurations from radial to tangential. For stitching we have to cut both images at the projection of point p. In practice, we use a vertical line on the cylinder which contains p to cut the images. Notice that in Figure 3.3 moving point P to a different depth along the red line does not have any effect on the stitching. This lack of sensitivity to depth is a very desirable property, which makes stitching free of artifact.

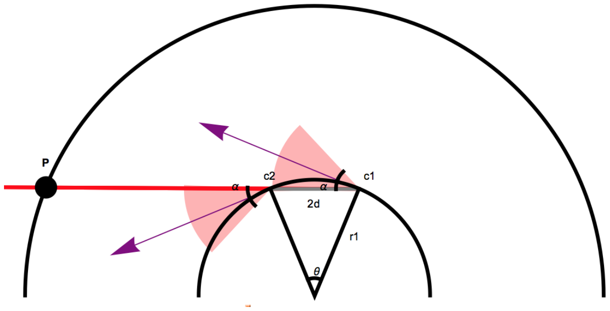


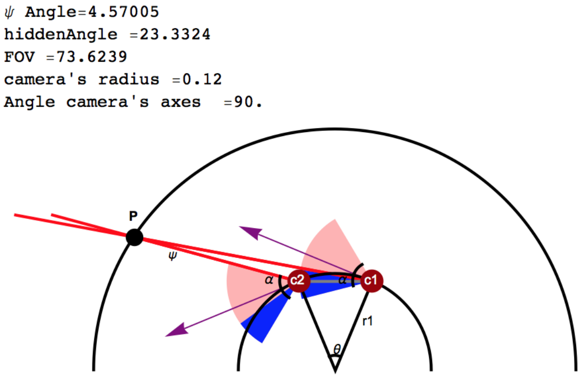
Figure 3.3 – Ideal tangential model (assuming negligible camera size). The FOV is set to the minimum achievable(45°).

3.3.1 Camera hiding Problem

In the ideal tangential model of figure 3.3 we assume that cameras are invisible or of negligible size. In practice, cameras will take up space and become visible at the cutting line by the next camera. See Figure 3.4 illustrate what happens when camera size is taken into account. To remediate this, the cutting points P must be moved until the camera is not visible anymore but the next camera. This makes the model less than ideal and will weaken the depth insensitivity property. To accomplish this, the fov of the camera must be increased to make point P visible on the right side of the image. At the left of the camera image, the other camera is visible but not needed (blue area in figure 3.4). Cutting at point p will remove it and allow reducing the field of view, as illustrated in figure 3.6 . This compensate the increase of fov on the right, so the resulting fov is only slightly increased.

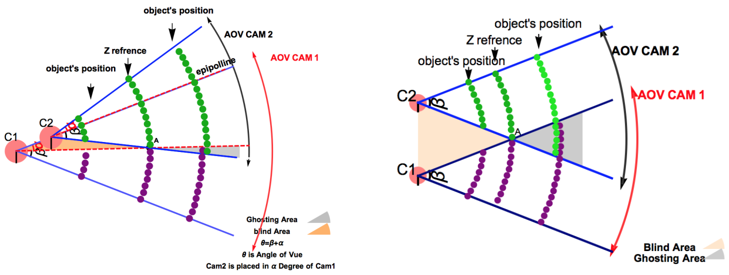
In radial model, it is necessary to have fov more than 45° to cut point P, in our example as show in figure 3.2, fov is 73°. In ideal tangential model, as shown in figure 3.3, cameras is rotated to 90° and fov is 45°, so in our implement of tangential model, cameras is rotated to 90° and fov is 73.6° as shown in figure 3.4, but because of camera radius, we have a hidden area that is defined by blue color and called hidden angle. Notice that the hidden angle is depended in camera radius,"r_c", and distance d between two cameras :

$$\sigma = \arcsin(\frac{r_c}{d}).$$



```
ψ Angle=4.57005
hiddenAngle =23.3324
FOV =73.6239
camera's radius =0.12
Angle camera's axes  =90.
```

Figure 3.4 – Non ideal tangential model, for non zero camera size. The fov has to be increased from the minimum (45°) to account for P. Note that all angles in top of image are in degree.



(a) Tangential Model **(b)** Radial Model

Figure 3.5 – Blind spot and Ghosting area in two modeles according to a depth reference. The gray colour is ghosting area that seen by both cameras and light orange parts are blind spot that not seen by these cameras.

So this blue part is seen by next camera but in different view, in fact we have a small part stereo in panorama image, so a rectangle part in the image is stereo, that each object in world in this part is seen by two camera in small difference angle. Our tangential model is in this situation. We can calculate the blue part in figure 3.4 from image, w is horizontal hidden pixel and f is focal length:

$$\psi = 2.\arctan(\frac{w}{2f}).$$

We have option to turn camera until remove hidden angle as shown in figure 3.6, then cameras is rotated to 77° and fov is 46.6° if we have the same camera size as figure 3.4.

3.3.2 Artifact Angle

The reference depth, where point p is located, is an important parameter in our configuration. If we have an object in a depth and reference depth is placed on it, an object in panorama image is uniform, but if reference depth is farther or closer than real depth, ghosting or blind spot occur in image respectively as shown in the gray and orange area in figure 3.5. The angle of this area is called ”artifact angle”. If this angle is zero then parallax is not on stitching, there is no parallax. If this angle is small, then the effect of parallax will be small, so the impact of bad depth estimate will be less visible. The major property of this angle is related to optical axis orientation. It is maximal for radial configuration and minimal for tangential as shown in figure 3.5.

The artifact angle can be computed from our camera model. First, we present calculation of this angles for the radial configuration as shown in figure 3.2. From model parameters, we calculate the artifact angle ψ as follow :

$$\begin{aligned} r_1 + r_2 = R. \qquad k_1 + k &= r_1. && (3.1)\\ \sin(\frac{\theta}{2}) &= \frac{d}{r_1}. \\ \tan(\frac{\omega}{2}) &= \frac{d}{r_2 + k_1}. && (3.2) \end{aligned}$$

From equations 1 and 3:

$$\begin{aligned} R &= \frac{d}{\sin(\frac{\theta}{2})} + \frac{d}{\tan(\frac{\psi}{2})} - k_1. \\ \psi &= 2.\arctan(\frac{d}{R + k_1 - \frac{d}{\sin(\frac{\theta}{2})}}). \end{aligned}$$

as θ and d are constant, then ψ is depend on r_2 and r_1.

For the tangential model as shown in figure 3.6, we can also compute ψ using additional parameters. We define length L as the distance between point p and first camera c_1, length K as the distance between point p and second camera c_2, angle δ between r_1 and R, and angle λ between L and the line connecting c_1 and c_2. Suppose we have a triangle $< p, C, c_1 >$, and we know angle of rotation of point P, then δ is known, so from law of cosines in triangle, it is possible to compute L and λ:

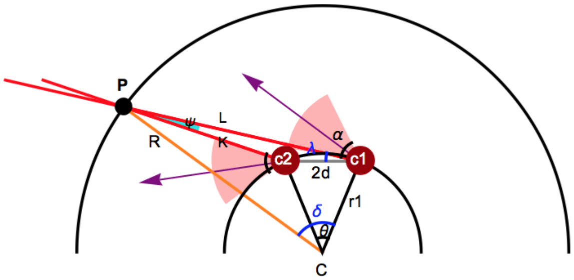


Figure 3.6 – Tangential model with minimal camera FOV (46.6°), with rotated cameras (77°). The artifact angle ψ is (4.57°)

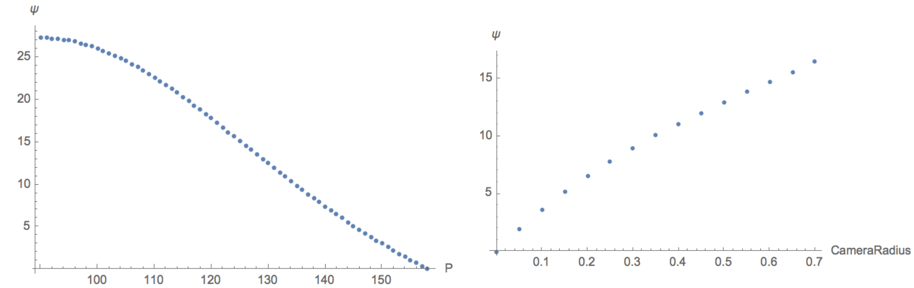
$$\cos(\delta) = \frac{r_1^2 + R^2 - L^2}{2Rr}. \tag{3.3}$$
$$\lambda = \arcsin(\frac{r_1}{2d}). \tag{3.4}$$
$$\cos(\lambda + (\frac{\pi}{2} - \frac{\theta}{2})) = \frac{r_1^2 + L^2 - R^2}{2Lr_1}. \tag{3.5}$$

now from equation 3.5 λ is known, then in triangle $< p, c_1, c_2 >$, from equation 3.6 k is known and from equation 3.7 calculate ψ. In our example in figure 3.2 artifact angle is 27.29°.

As because of rotation of camera, the blinding and ghosting areas are smaller than radial model, then:

$$\cos(\lambda) = \frac{(2d)^2 + L^2 - k^2}{2(2d)L}. \tag{3.6}$$
$$\cos(\psi) = \frac{k^2 + L^2 - (2d)^2}{2kL}. \tag{3.7}$$
$$\psi = \arccos(\frac{k^2 + L^2 - (2d)^2}{2kL}). \tag{3.8}$$

In fact the two derivation of ψ are related. In our example in figure 3.6 artifact angle is 4.52°. In the tangential derivation if point p is allowed to rotate around center C, then both derivation are equal when p is above c_1 and c_2(rotated zero degree) as in figure 3.2. When point p is rotated until it collinear with c_1 and



(a)

(b)

Figure 3.7 – Artifact angle ψ as a function of the angle (in degrees) of point P (a) and as a function of camera size relative to the inter-camera distance (b)

c_2, then ψ is zero. This illustrated in figure(3.7-a), where we observe that radial configuration is the most sensitive to the parallax and this sensitivity decrease as the geometry become more tangential. Notice that point p can be collinear to c_1 and c_2 (which happens near 160°in the examples of figure (3.7-a)). This situation does not occur in practice since near that position one camera will hide point p from the other camera. Also the impact of camera size on the artifact angle ψ can be seen in figure(3.7-b), so in practical on camera size, we observe that the angle ψ is very small.

3.4 Calibration

Proper calibration is an important part of using any multiple camera setups, where we estimate the intrinsic and extrinsic parameters of the cameras. Can be found through this processZhang [2000].

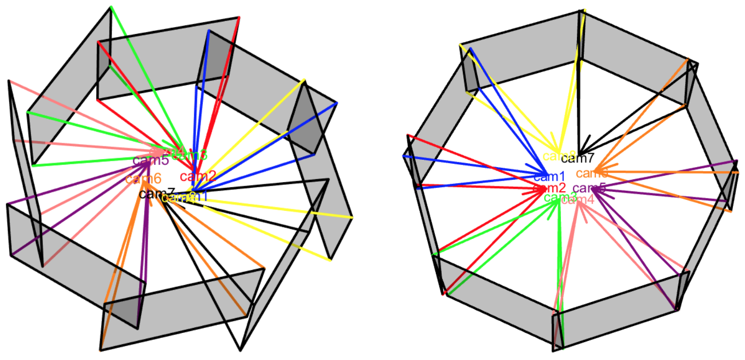
First, the intrinsic parameters are estimated using the classical checkerboard method Zhang [2000], applied to each camera individually.

Determining the extrinsic parameters is more difficult, mostly because of the small overlap between views, and because the cameras form a closed loop which can't be solved linearly in a single step.

In order to solve the small overlap problem, we rely on Structured-light on LCD monitor to replace the checkerboard and establish a dense correspondence between views Couture et al. [2011]. While the density is very good, the correspondences can only be established between two successive cameras.

To obtain a full calibration, we first estimate the extrinsic parameters between two successive cameras (i and $i+1$) which yield rotation R_i and translation T_i, with i from 1 to 8. Because we have a closed loop, R_8 and T_8 are the rotation and translation from camera 8 to camera 1.

The reference system is camera 1, so the final transformation M for each camera



(a) Tangential Model **(b)** Radial Model

Figure 3.8 – Position of all cameras after calibration.

is

$$\begin{aligned} M_1 &= K_1[I|0] \\ M_2 &= K_2[R_1|t_1] \\ M_3 &= K_3[R_2|t_2].[R_1|t_1] \\ &\dots \\ M_8 &= K_8[R_8|t_8].[R_7|t_7].\dots.[R_1|t_1] \end{aligned}$$

However theses transformations are inaccurate and must be optimized globally. For each camera, we define a correction transformation. Using the various correspondences obtained from structured light, we can estimate the re-projection error in pixels, and optimize this error globally over the correction transformations. This yields the optimal camera geometry, shown in figure 3.8, which we use for stitching. Note that we do not optimize the internal camera parameters, since they are obtained first and assumed of good precision.

3.5 Panorama generation

After estimating the relative camera poses, a panorama can be generated by projecting all images into a common surface. A traditional way to do this is to choose either a cylindrical or spherical map. Cylindrical panoramas are commonly used because of their ease of construction. When the cameras are calibrated, each perspective image can be warped into cylindrical coordinates and the color is associated with each pixel is computed by first deriving a 3D ray from the pixel position, and then mapping this ray into each input image through our known transformation. The cylinder parameters (center, axis, radius) are important. The obvious choice for the center is the average of all camera centers. The average Y axis of all cameras can be used as the cylinder axis. Since the radius represents depth, its selection is the most difficult aspect of stitching. Estimating depth by stereo is impossible because of the small overlap between cameras. However, at the overlap, a wrong depth estimate will induce a visible artifact which magnitude is related to the artifact angle (see section 2.2). The tangential geometry will be much more resilient to a bad estimate of depth, thereby making it possible to use a single depth for a whole panorama without much visible artifact. This not only simplifies stitching but can make it real-time.

Each pixel in a panorama can be classified as visible or not visible from the cameras, and when it is visible, it can be from a single camera or two cameras. Usually, overlaps between cameras are visible by two cameras, but in a tangential panorama part of the overlap is obstructed by a camera body, so it effectively becomes partially visible as a single camera. This is illustrated in figure 3.9 for tangential model and figure 3.10 for radial model.

Figure 3.11 illustrates the impact of depth estimation. Consider an object that is seen by two cameras in a tangential panorama (top of figure 3.11-a), and radial panorama (bottom of figure 3.11-a). The object is highlighted by a red rectangle. Starting from the correct depth of this object, we varied the depth of the panorama point from much closer to much further. The reprojections in both images, relative to the correct point match, are displayed superimposed together in figure 3.11-b for tangential, and figure 3.11-c for radial panoramas. Clearly, the wrong depth estimation induces much more separation between the reprojected points in the radial case than the tangential case. The pixel distance is also illustrated in Fi-

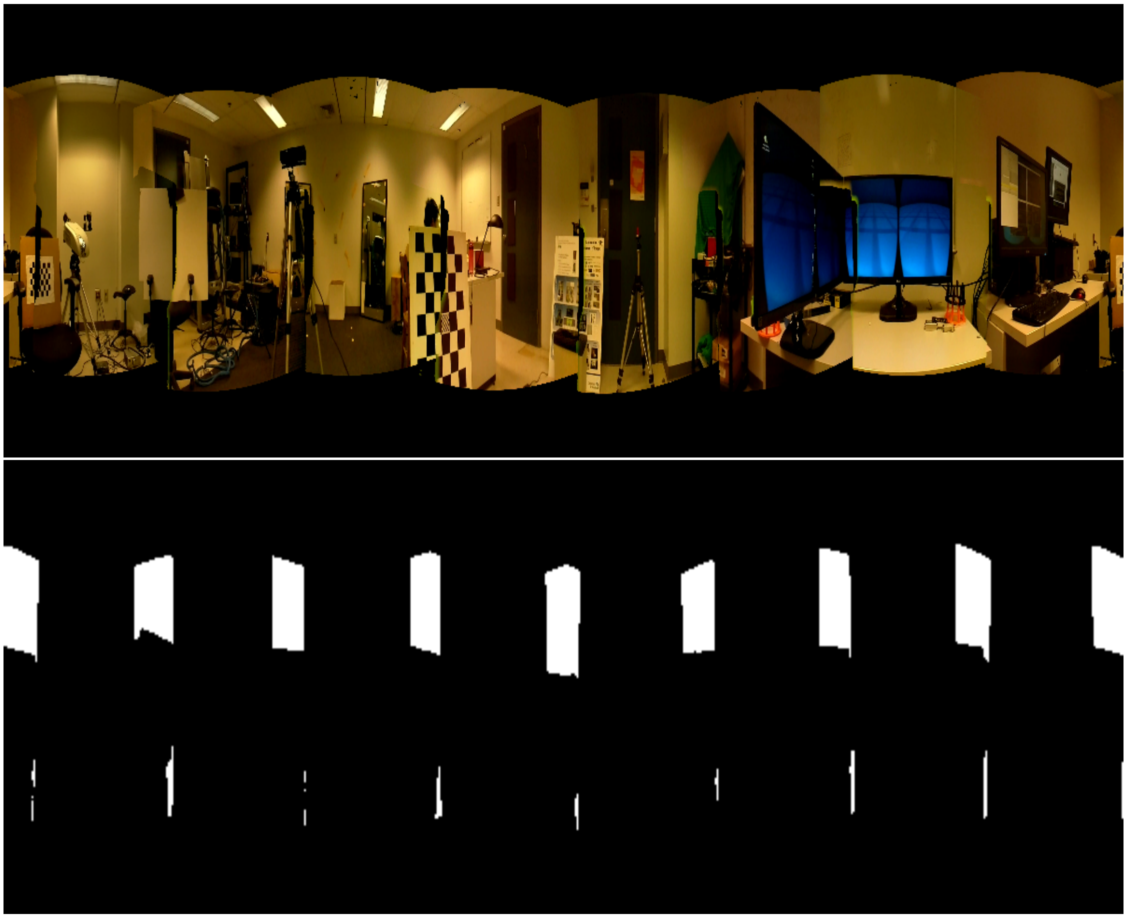


Figure 3.9 – Sample tangential panorama (top) with mask (bottom) illustrating overlapping pixels (seen by two cameras). Smaller overlap is caused by camera occlusion.

gure 3.11-d, where the pixel separation is clearly larger for the radial panorama. In this example, the camera image dimension is 1296 × 972.

3.6 Experiments

We implemented the proposed panorama system using 8 raspberry pi cameras (figure 3.12-a) setup in a tangential and a radial arrangement illustrated in figure 3.12-b and c, respectively. The cameras feature a focal length of 3.6 mm, a $1.4 \times 1.4 \mu m$ pixel size. The image resolution is 2592×1944 pixels with a horizontal field of view of about 53°, and vertical field of view of 40°. The camera itself has

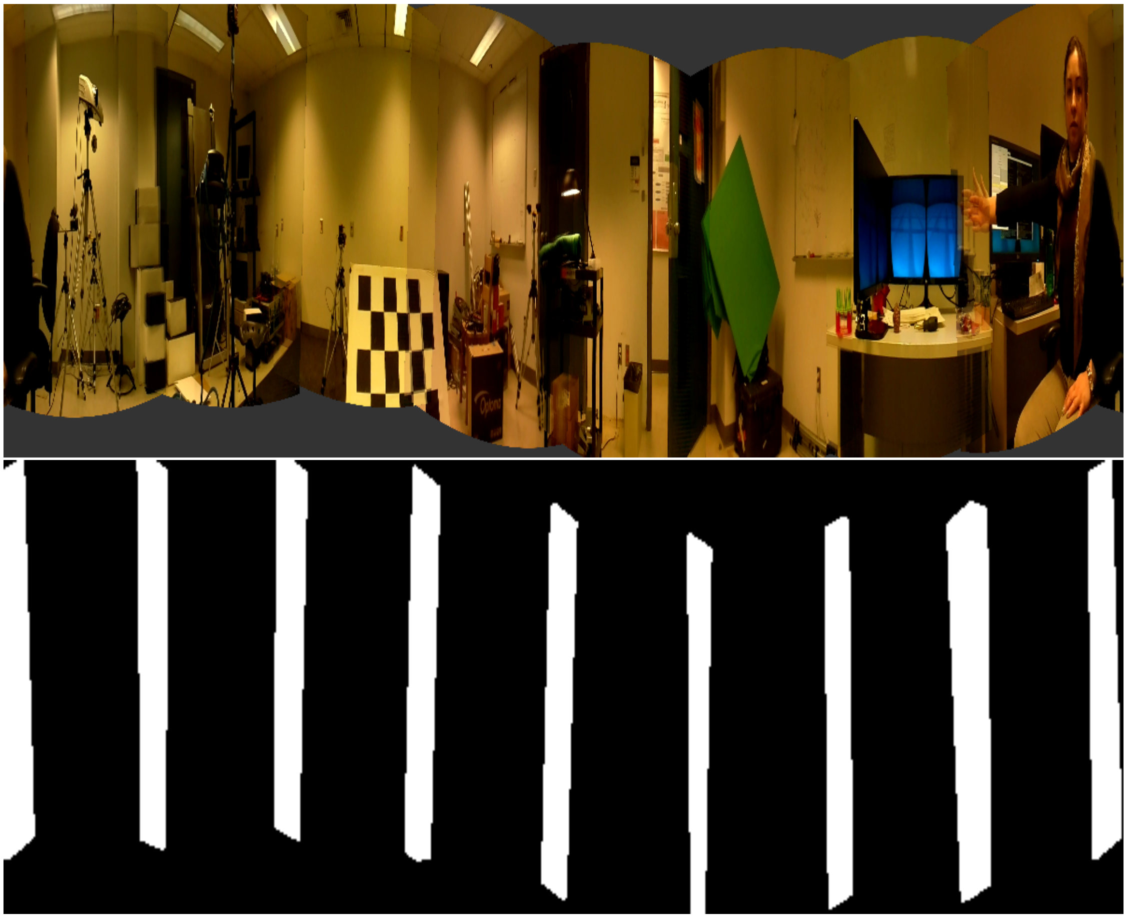


Figure 3.10 – Sample radial panorama (top) with mask (bottom) illustrating overlapping pixels (seen by two cameras).

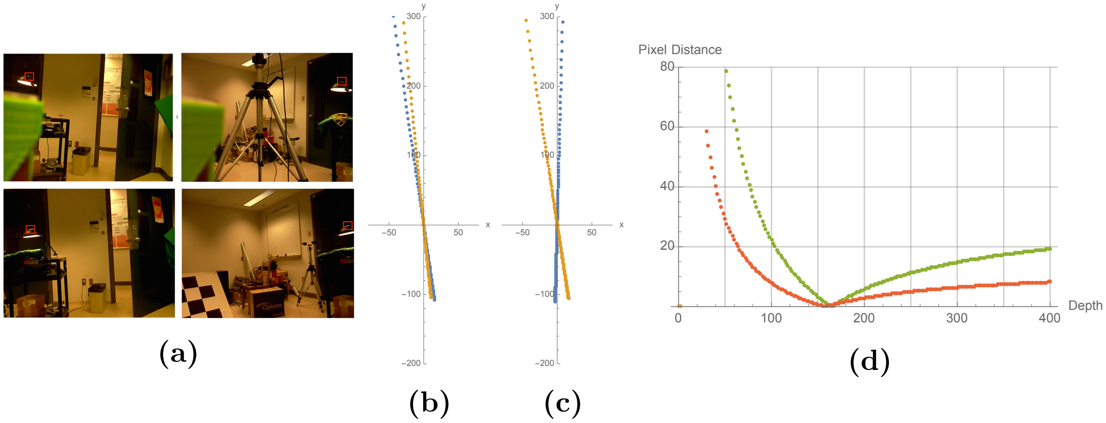
a size of 8 × 8 × 4 mm PI. The disposition of the cameras is a circle of 10*cm* diameter. In practice, the radial configuration could use a smaller diameter, but the tangential can't since reducing the inter-camera space increase the relative size of the camera occlusion. However, using a large diameter has potential applications for stereoscopic panoramas, so it is relevant Jum.

Sample panoramas are provided in figure 3.13. Both are built using a single depth for the whole scene, without any adjustments. Moreover, the stitching in overlap areas is a simple average of both images. This is done on purpose to make visual artifact standout. The tangential panorama (top) is generally cleaner, with less visual artifacts than the radial panorama (middle). Notice that some pixels in the tangential panorama are blacked out. This is caused by the field of view which was slightly smaller than required for this geometry. Since we used a perfect tangential geometry, it is possible to rotate the cameras slightly to reduce the field of view requirement and solve the issue (see section 2.2).

One closeup example of stitching error is provided at the bottom of figure 3.13. Clearly, the tangential geometry (left) provides a sharper result.

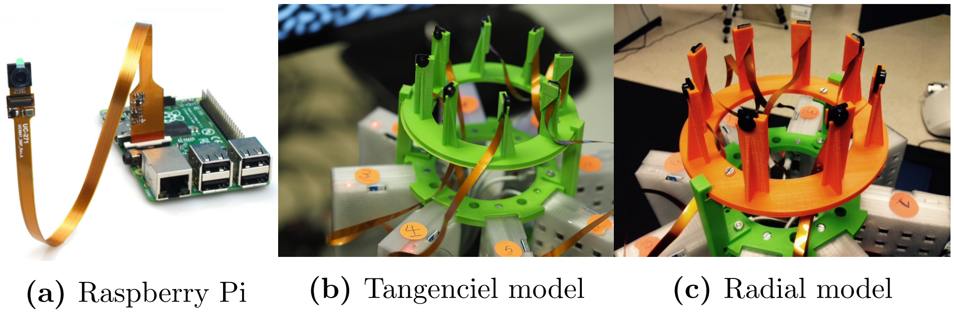
3.7 Conclusion

This paper presented a new approach to generating a multi-camera cylindrical panorama. The proposed geometry is called *tangential* because each camera has its optical axis tangent to the circle of cameras. Beside the fact that this configuration will induce occlusion from one camera to the next one along the camera circle, we demonstrate that this configuration is much less sensitive to the wrong estimate of scene depth. This allows the stitching process to be simplified enough to provide very good panoramas in real-time using a single depth estimate for a whole scene. Examples are provided for an experimental setup of 8 cameras, but we anticipate that stereoscopic panoramas will be easy to build with this configuration, and would provide real-time images.



(a) (b) (c) (d)

Figure 3.11 – A) Object seen from two cameras in a tangencial setup (top) and radial setup (bottom), highlighted by a red rectangle. B) Image reprojections for various depth, relative to the correct match, for tangencial setup and C) for radial setup. D) Pixel distances as a function of depth, for tangencial (red) and radial (green). Correct depth is 160.



(a) Raspberry Pi **(b)** Tangenciel model **(c)** Radial model

Figure 3.12 – Panoramic camera setup.

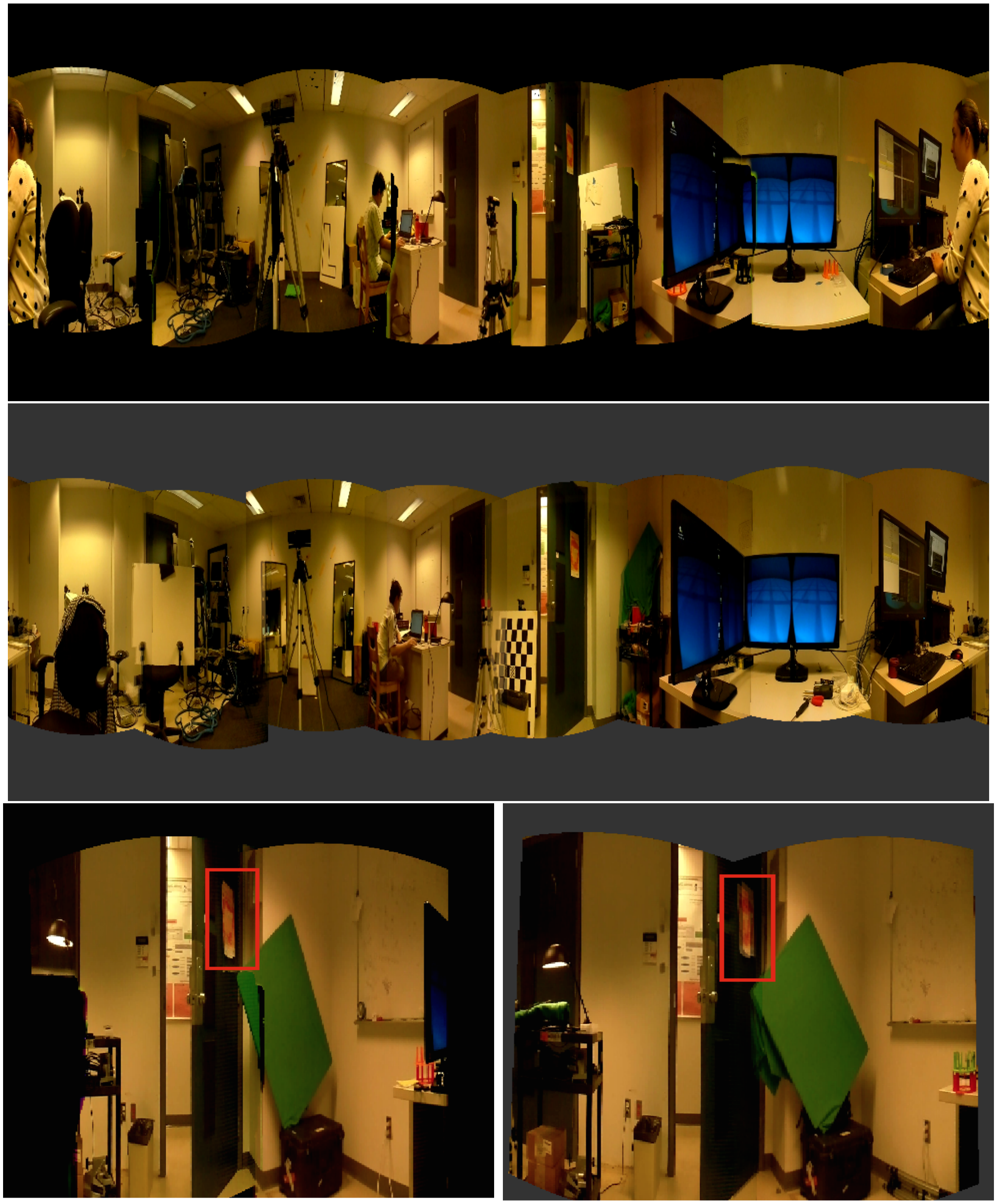


Figure 3.13 – Sample panoramas with tangential model (top) and radial model (middle). At bottom, a single object highlighted with a red rectangle illustrates the difference between tangential model (left) and radial model (right).

4 Supplementary material

In this chapter, we present additional material that was not added to the paper for space consideration. We show results on panorama image in two geometries and other possible position of cameras as indicated in Sec. 3.5 and Sec. 3.3.1 respectively in chapter 3.

4.1 Camera Hiding Problem

In the tangential model, the rotation of cameras is such that some part of one image is hidden by the next camera body, as shown in figure (4.1).

To handle this problem properly, three options are available . As a reference, consider a radial geometry in which there is no hidden camera problem. The field of view is 73.6°, the height of camera's frame is 6.8 mm, and the angle of cameras axis is *zero*° as shown in figure 4.2a.

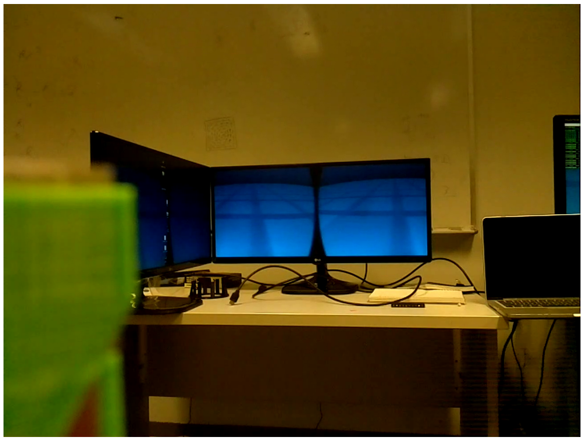
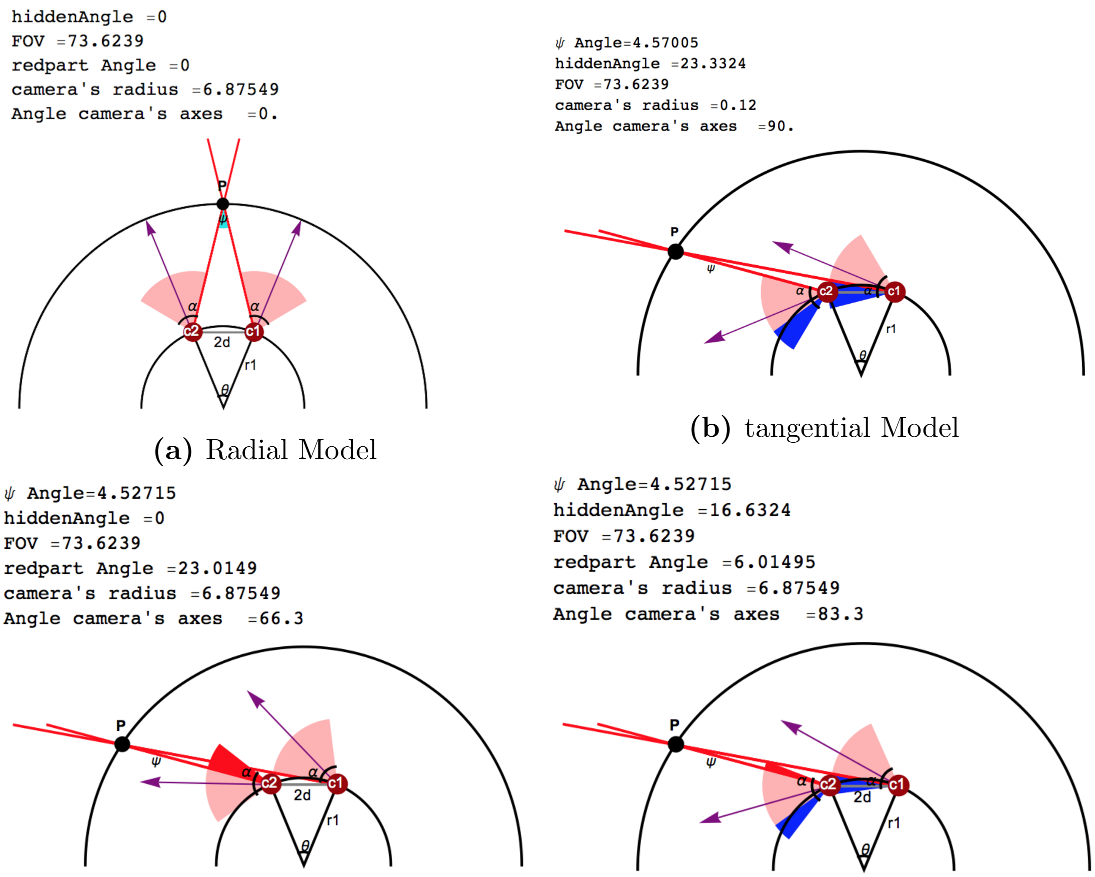


Figure 4.1 – A image is captured in tangential system and hidden part is shown in green color.



(a) Radial Model

(b) tangential Model

(c) nontangential Model

(d) nontangential Model

Figure 4.2 – Estimate relation between Field of View and rotation of cameras.

From this geometry, we rotate each camera by 90° to obtain a tangential geometry as shown in figure 4.2b, with the same camera radius and field of view. The blue part in this figure (4.2b) is a hidden angle which is unused (23.3°) so only 50.2° is used from the full field of view of 73.6°.

In this situation, we have two options. First, we can rotate the cameras from 90° to about 66.3°, as shown in 4.2c, which will remove the unused hidden part (in blue) but replace it with a useless overlap, displayed in red. This geometry is not purely tangential, but this is not a problem. However, there is no clear advantage for this setup.

The second option is to rotate the cameras from 90° to about 83.3°, as shown in 4.2d, in order to balance the unused hidden part (blue) and the unused overlap (red). The exact ratio between the parts is not 1:1 but depends on the actual camera body size which affects only the red part. In this example, we obtain 6° and 16° for the red and blue parts respectively. This is a useful configuration since there are unused parts on both side of the image, so the actual field of view of the camera can be reduced (by $2 \times 6°$ in this case). This is the ideal setup for tangential geometry in our opinion.

In our experimental setup, we used the tangential configuration of figure 4.2b instead of the ideal configuration of 4.2d. This is simply because we did not know in advance the camera body size and so we could not easily compute the angle beforehand. Our simpler configuration wastes some field of view but is simpler to put in place.

4.2 Panorama image

In this section, we show more examples of our two geometries from chapter 3.

Tangential geometry example

To illustrate further the tangential geometry, we provide an additional panorama in figure 4.4, and a closeup of that panorama in figure 4.3. As seen in these examples, we don't have obvious parallax problems on overlapping parts, which are highlighted in red rectangles.

Radial geometry example

To illustrate further the radial geometry, we provide an additional panorama in figure 4.6, and a closeup of that panorama in figure 4.5. As seen in these examples, they feature more parallax problems on overlapping parts, which are highlighted in red rectangles, than the tangential examples.

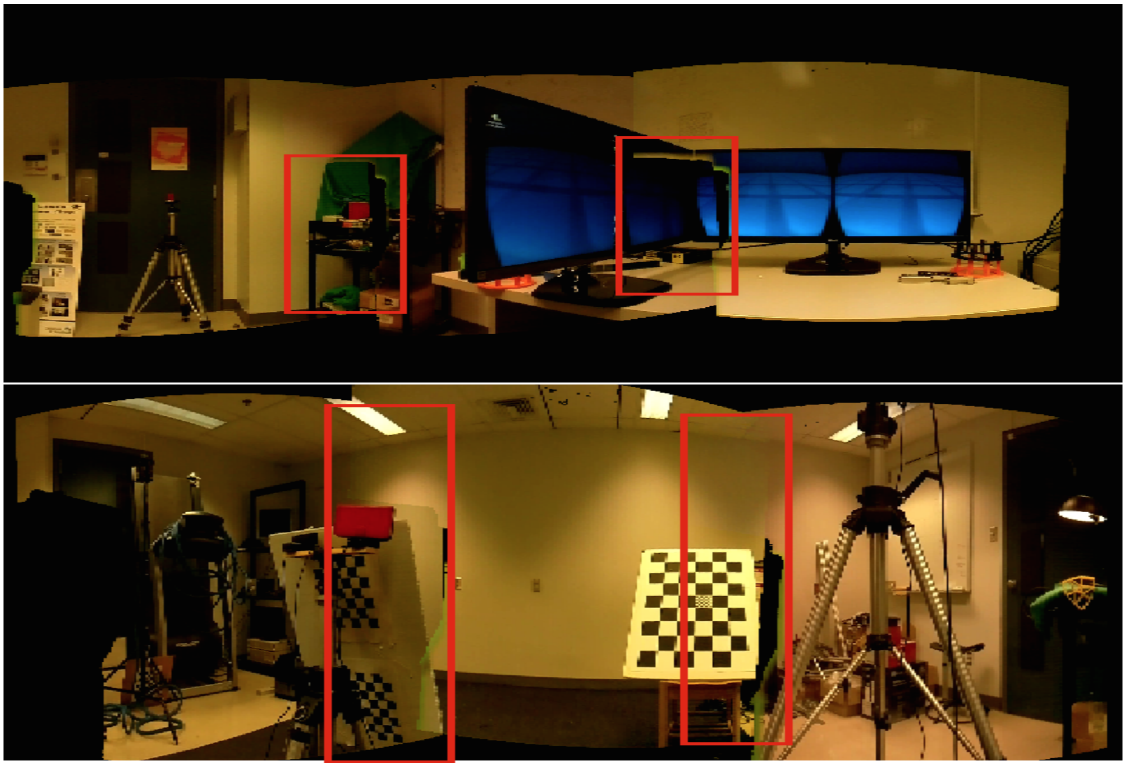


Figure 4.3 – Sample panoramas with tangential model by some cameras.

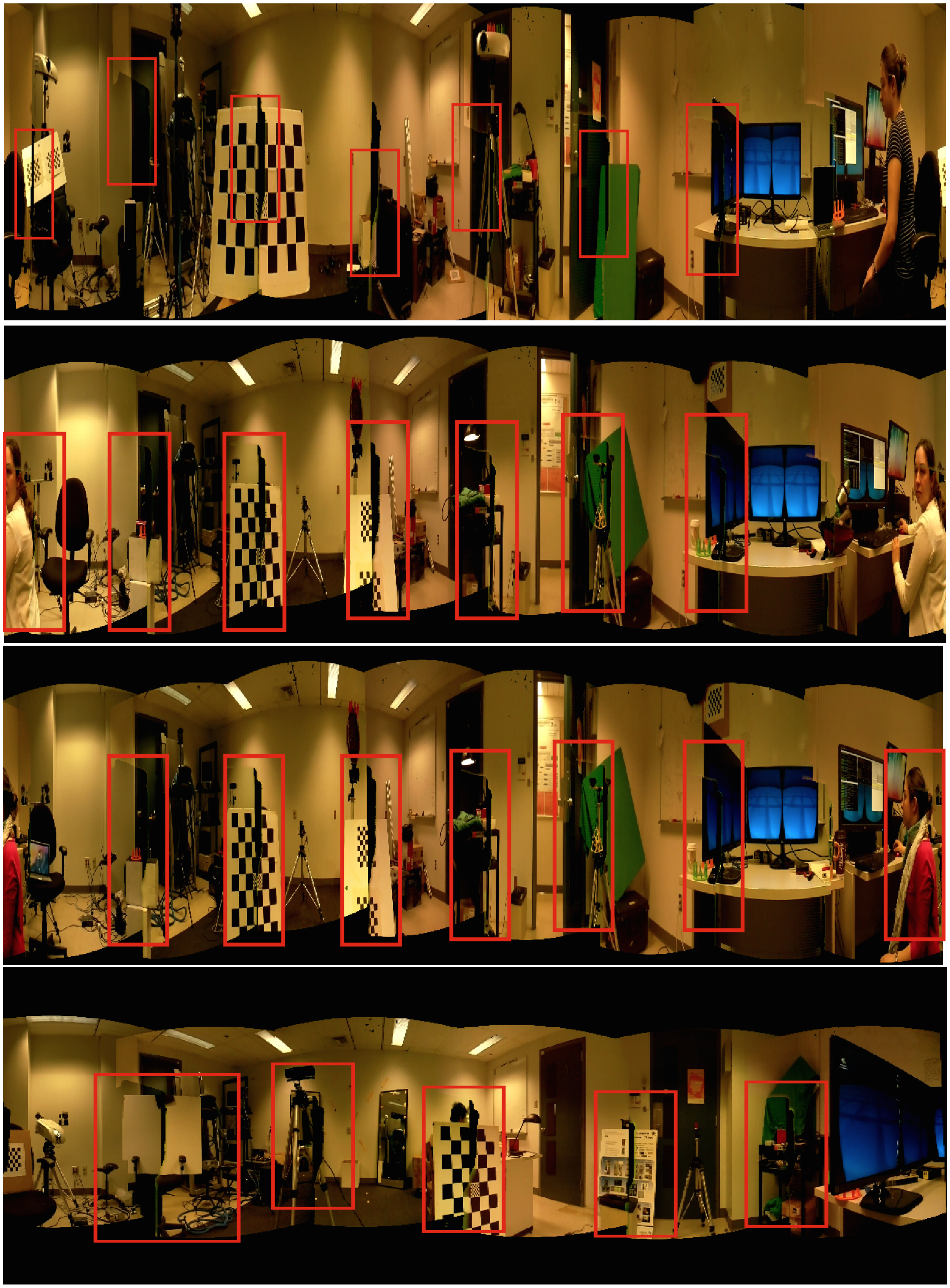


Figure 4.4 – Sample panoramas with tangential model.

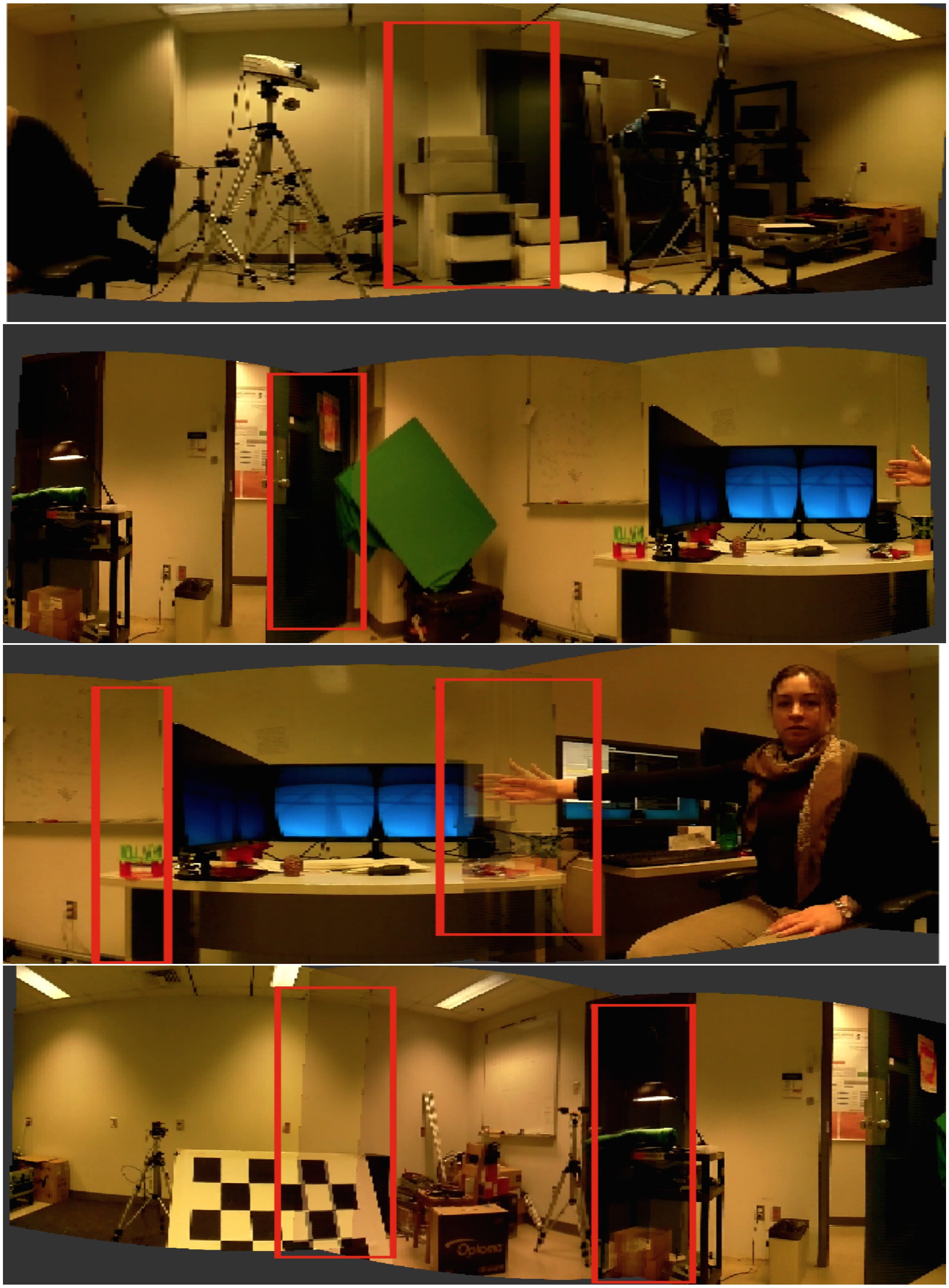


Figure 4.5 – Sample panoramas with radial model.

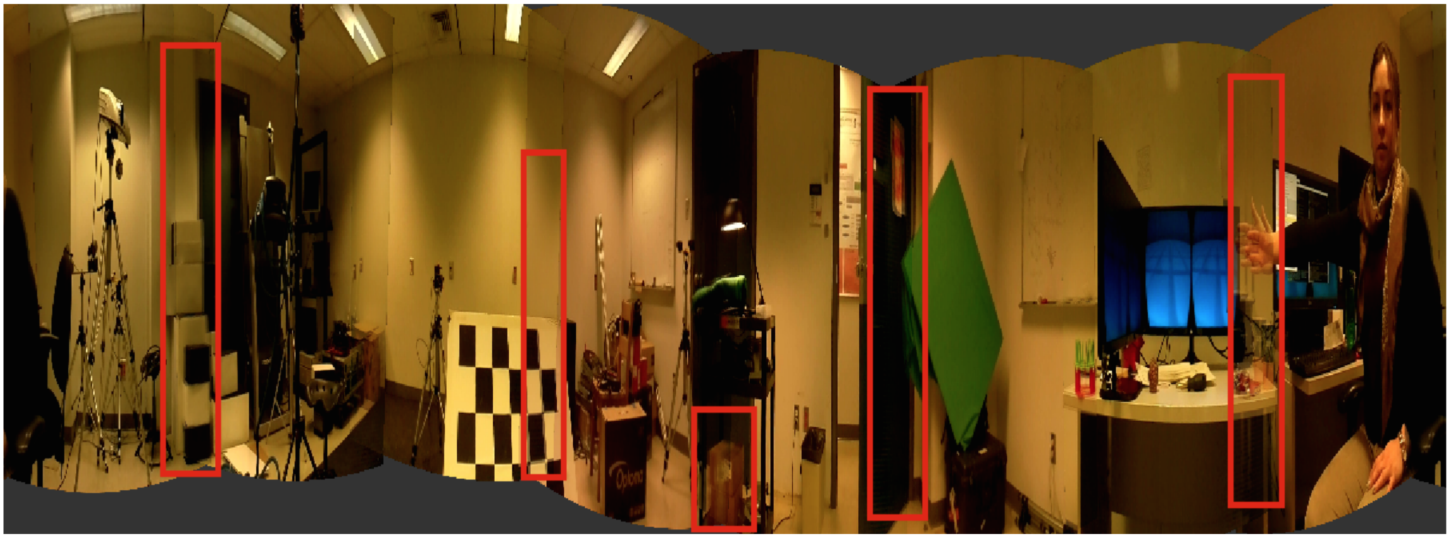


Figure 4.6 – Sample panoramas with radial model.

5 Conclusion

This master thesis explored the creation of panoramic images. The main goal was to compare various camera geometry models to achieve good image quality with reduced stitching artifacts.

We demonstrated that the tangential camera model we proposed is significantly reducing parallax and stitching artifacts, compare to the radial model. To the best of our knowledge, it is original and the best approach to panoramic imaging, especially for real-time stitching. We also built a fully functional prototype panoramic camera with 8 cameras using a low cost raspberry pi.

As a direction for future development of this work, it is possible to double the number of cameras in a ”back to back” tangential configuration to achieve true 3d stereoscopic cylindrical capture as shown in figure 5.1. We expect this extension to eventually make stereo-immersive capture much more accessible.

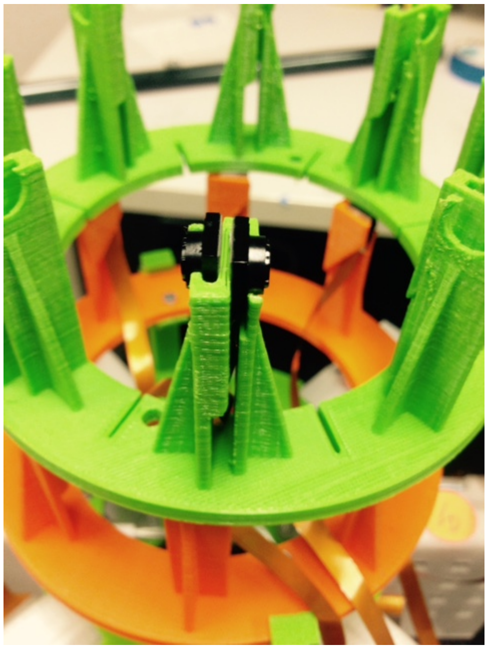


Figure 5.1 – Tangential model for stereo-immersive capture

Bibliographie

Jump camera rig. https://vr.google.com/jump/.

Raspberry Pi information and products for custom applications and engineering. http://www.truetex.com/raspberrypi.

Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 294–302. ACM, 2004.

Anup Basu and Sergio Licardie. Alternative models for fish-eye lenses. *Pattern Recognition Letters*, 16(4):433–441, 1995.

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

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

Shenchang Eric Chen. Quicktime vr: An image-based approach to virtual environment navigation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 29–38. ACM, 1995.

Vincent Couture, Nicolas Martin, and Sebastien Roy. Unstructured light scanning to overcome interreflections. In *2011 International Conference on Computer Vision*, pages 1895–1902. IEEE, 2011.

Junhong Gao, Seon Joo Kim, and Michael S Brown. Constructing image panoramas using dual-homography warping. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 49–56. IEEE, 2011.

Sevket Gumustekin and Richard W Hall. Mosaic image generation on a flattened gaussian sphere. In *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*, pages 50–55. IEEE, 1996.

Rajiv Gupta and Richard I Hartley. Linear pushbroom cameras. *IEEE Transactions on pattern analysis and machine intelligence*, 19(9):963–975, 1997.

Ho-Chao Huang and Yi-Ping Hung. Panoramic stereo imaging system with automatic disparity warping and seaming. *Graphical Models and Image Processing*, 60(3):196–208, 1998.

Hiroshi Ishiguro, Masashi Yamamoto, and Saburo Tsuji. Omni-directional stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):257–262, 1992.

David Jack, Rares Boian, Alma S Merians, Marilyn Tremaine, Grigore C Burdea, Sergei V Adamovich, Michael Recce, and Howard Poizner. Virtual reality-enhanced stroke rehabilitation. *IEEE transactions on neural systems and rehabilitation engineering*, 9(3):308–318, 2001.

Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003.

Wen-Yan Lin, Siying Liu, Yasuyuki Matsushita, Tian-Tsong Ng, and Loong-Fah Cheong. Smoothly varying affine stitching. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 345–352. IEEE, 2011.

Shree K Nayar and Amruta Karmarkar. 360×360 mosaics. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 388–395. IEEE, 2000.

Shree K Nayar and Venkata Peri. Folded catadioptric cameras. In *Panoramic vision*, pages 103–119. Springer, 2001.

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. IEEE, 1999.

Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003.

Paul Rademacher and Gary Bishop. Multiple-center-of-projection images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 199–206. ACM, 1998.

Harpreet S Sawhney, R Kumar, Gary Gendel, J Bergen, Douglas Dixon, and Vince Paragano. Videobrush tm: Experiences with consumer video mosaicing. In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pages 56–62. IEEE, 1998.

Heung-Yeung Shum and Richard Szeliski. Panoramic image mosaics. Technical report, Citeseer, 1997.

Heung-Yeung Shum and Richard Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Computer Vision, 1998. Sixth International Conference on*, pages 953–956. IEEE, 1998.

Heung-Yeung Shum and Richard Szeliski. Stereo reconstruction from multiperspective panoramas. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 14–21. IEEE, 1999.

Tomáš Svoboda and Tomáš Pajdla. Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, 49(1):23–37, 2002.

Richard Szeliski. Video mosaics for virtual environments. *IEEE computer Graphics and Applications*, 16(2):22–30, 1996.

Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258. ACM Press/Addison-Wesley Publishing Co., 1997.

Roger Tsai and T Huang. Estimating three-dimensional motion parameters of a rigid planar patch. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1147–1152, 1981.

Juyang Weng, Paul Cohen, Marc Herniou, et al. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on pattern analysis and machine intelligence*, 14(10):965–980, 1992.

Daniel N Wood, Adam Finkelstein, John F Hughes, Craig E Thayer, and David H Salesin. Multiperspective panoramas for cel animation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 243–250. ACM Press/Addison-Wesley Publishing Co., 1997.

Yalin Xiong and Kenneth Turkowski. Creating image-based vr using a self-calibrating fisheye lens. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 237–243. IEEE, 1997.

Julio Zaragoza, Tat-Jun Chin, Michael S Brown, and David Suter. As-projective-as-possible image stitching with moving dlt. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2339–2346, 2013.

Fan Zhang and Feng Liu. Parallax-tolerant image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3269, 2014.

Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.