Chapter 1.

Introduction

1.1 Motivation

As the field of view captured by a photographic device is usually small, how to enlarge the field of view by integrating multiple images taken with a photographic device is an important research topic which has received considerable attention in recent years. A wide viewing space can be shown in a single image by using an image-mosaic or stitching algorithm to align and composite these captured images. Traditionally, researches of building mosaics have been restricted to planar scenes or a sequence of images is taken by a camera mounted on a leveled tripod and rotated around its optical center with a carefully controlled camera motion during shooting [3][9][10][11][19][20]. Restricted to these two cases, previous related researches have presented good experimental results to build mosaics, but these methods have strong limitations on imaging conditions. Due to these disadvantages, it is desired to build mosaics under an uncontrolled camera motion, such as motion caused by shooting with a hand-held camera. Moreover, because building panoramic mosaics is usually time-consuming, it is also desired to develop a real-time image-acquisition and image-stitching system for constructing panoramic mosaics from video cameras. In the first part of this work, a real-time distance-scanning system is developed to achieve these purposes.

A wide-field drawing, such as wide-field paintings in libraries and museums, can not be acquired in a single shot with an acceptable resolution, and thus how to reconstruct a wide-field drawing becomes an important issue. Similarly, a solution to this problem in computer vision is combining multiple images to reconstruct a wide-field drawing. In previous related study [3], such a work may be inconvenient and inefficient due to the complication of image acquisition and the estimation of geometric relationships of multiple images. In this research, instead of stitching multiple images taken with perspective cameras, we use the developed real-time distance-scanning system to acquire several partial mosaic images of a wide-field drawing under some constraints by using video cameras, and a linear pushbroom camera model [6] is employed to model the mosaic images acquired in this way. Furthermore, a theoretical study on geometric relationships among these partial mosaic images is given, which is employed for developing a new approach for reconstructing wide-field drawings automatically.

Since the partial mosaics may be taken under various conditions of camera positioning and lighting, the problem of color blending is arisen when reconstructing wide-field drawings from several partial mosaics. In this thesis, a multiresolution color blending technique is employed and implemented to make the stitching boundary between a pair of partial mosaics invisible. First, the mosaic images are decomposed into a set of band-pass component images. Next, the component images in each spatial frequency band are assembled into corresponding band-pass images. Finally, these band-pass images are summed to reconstruct a seamless wide-field drawing.

In summary, in this thesis we establish a real-time distance-scanning system, conduct mathematical models for exploring the relationships between partial mosaic images and reconstructed wide-field drawings, and adopt a color blending technique to obtain the seamless drawings. Finally, our system can be used for several applications in different fields, such as visualization, video enhancement, video indexing, environmental surveillance, mosaic-based video compression, and so forth.

1.2 Survey of Related Works

As a standard camera has a very limited field of view, the environment is hardly to be captured by a single shot. In recent years, a number of techniques have been developed for representing the real-world scenes from panoramas. Generally speaking, image-based rendering is more popular than model-based rendering to simulate a visually rich tele-presence or virtual reality scene [19]. That is, a collection of images is used to render the environment rather than to build and render a complex 3-D model of the environment. For example, a single cylindrical image enables users to pan and zoom inside the environment created from images. Base on this idea, some methods have been proposed for constructing panoramas from multiple images in order to cover a larger or the whole viewing space. The major methods for building panoramic mosaics are surveyed as follows:

(1) Cylindrical and Spherical Panoramas: [5][11][19]

The most commonly used panorama is the cylindrical panorama because of its

ease of construction. To build a cylindrical panorama, the camera is rotating around the cylinder's axis with fixed focal length, and the world coordinate **p** = (*X*, *Y*, *Z*) can map into 2-D cylindrical screen coordinate (*q*, *v*) by using

$$q = \tan^{-1}(\frac{X}{Z}), \quad v = \frac{Y}{\sqrt{X^2 + Z^2}}, \qquad \text{(Eq. 1.1)}$$

where *q* is the panning angel and *v* is the scan-line.

If each input image is warped, it becomes a pure translation problem to construct the panoramic mosaics. Usually, only the unknown rotation angles have to be covered to build a cylindrical panorama from a horizontal panning sequence. But small vertical translations are needed for the compensation of vertical jitter and optical twist in practice. That is, both the horizontal translation t_x and vertical translation t_y are estimated for each input image when constructing a cylindrical panorama.

On the other hand, to build a spherical panorama, the camera is allowed to rotate around its optical center to an arbitrary direction with a fixed focal length, and the constructed panorama is projected onto a sphere whose center is the camera's optical center and whose radius is equal to the camera's focal length. Similar to the cylindrical panorama, the world coordinate **p** = (*X*, *Y*, *Z*) can map into 2-D spherical coordinate (*q*, *f*) by using

$$q = \tan^{-1}(\frac{X}{Z}), \quad f = \tan^{-1}(\frac{Y}{\sqrt{X^2 + Z^2}}) \qquad \text{(Eq. 1.2)}$$

The translational motion can be estimated by minimizing the SSD (Sum of Square Difference) error *E* of corresponding pixels over all overlapped region. *E* is defined as

$$E = \min \sum_i [I_1(\mathbf{x}_i^{'}) - I_0(\mathbf{x}_i)]^2, \qquad \text{(Eq. 1.3)}$$

where $\mathbf{x}_i = (x_i, y_i)$ and $\mathbf{x}_i^{'} = (x_i + t_x, y_i + t_y)$ are corresponding points in two images I_0 and I_1, respectively. Thus **t** = (t_x, t_y) is the estimated translational motion vector.

Creating a cylindrical or a spherical panorama has several restrictions. First, it requires knowing the effective focal length. Second, it requires a carefully

controlled camera motion of pure rotation.

(2) Perspective (8-parameter) Panoramas: [5][9][10][19][21]

In order to conquer limitations of cylindrical and spherical panoramas, several previous researches have recommended using full planar perspective motion model. The planar perspective transformation (or referred to as the *planar homography*) uses 8 parameters to warp an image as follows:

$$\mathbf{x}' \sim \mathbf{M}\mathbf{x} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \qquad \text{(Eq. 1.4)}$$

where $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$ are the homogeneous or projective coordinates, and ~ indicates equality up to scale. Eq. 1.4 can be rearranged as Eq. 1.5:

$$x' = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1}$$
$$y' = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1} \qquad \text{(Eq. 1.5)}$$

In the translational-motion case, only the two parameters m_2 and m_5 are used. Again, matrix **M** can be calculated by minimizing error E defined in Eq. 1.3. Although the 8-parameter perspective transformation works well, the motion model contains more free parameters. Therefore, some researches prefer to use 3-parameter rotational model described next.

(3) Rotational (3-parameter) Panoramas: [19][20]

Unlike previous panoramic stitching methods, this method does not require any controlled motion or constraint on how the images are taken. It directly recovers 3-D rotation parameters instead of general 8-parameter planar perspective transformation to make this system more fast and robust since it has fewer unknowns. In rotational panorama, mosaic is represented by a set of transformations and each transformation corresponds to one image of input image sequence and represents the mapping between image pixels and viewing directions in the world.

The relationships between a 3-D point $\mathbf{p} = (X, Y, Z)^T$ and its image coordinate $\mathbf{x} = (x, y, 1)^T$ with a camera centered at the origin can be represented as follows:

$$\mathbf{x} \sim \mathbf{TVRp}, \qquad \text{(Eq. 1.6)}$$

where

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ and } \mathbf{R} = [r_{ij}]$$

are the image plane translation, focal length scaling, and 3-D rotation matrix, respectively. In order to simplify the notation, we assume the origin is at the image center, and thus $c_x = c_y = 0$. This assumption allows us to eliminate **T**. From Eq. 1.6, the 3-D direction corresponding to an image pixel **x** is given by $\mathbf{p} \sim \mathbf{R}^{-1}\mathbf{V}^{-1}\mathbf{x}$.

As a camera rotates around its center of projection, the mapping between two images *k* and *l* is given by

$$\mathbf{M} \sim \mathbf{V}_k \mathbf{R}_k \mathbf{R}_l^{-1} \mathbf{V}_l^{-1}, \qquad \text{(Eq. 1.7)}$$

where each image can be represented by $\mathbf{V}_k\mathbf{R}_k$. That is, each image can be described by the focal length and 3-D rotation. Now, we assume that the focal length is known and is the same for all images, $\mathbf{V}_k = \mathbf{V}$, thus each image can be represented only by a 3-D rotation matrix **R** with 3 unknown parameters.

This method is robust and accurate than perspective panorama because of only 3 unknown parameters in the rotation matrix **R** are needed to be estimated rather than 8 unknown parameters. However, this parameterized method can only be used when the camera motion is pure rotation.

(4) Manifold Panoramas: [5][15][16][17][18]

There exists a variety of restrictions of camera motions and scene structures in the previous three panorama projection types. But these limitations do not exist in the manifold panorama proposed by Peleg and Herman in 1997 [16][17]. Unlike other panorama construction methods, a single column image in the center of each image of an input image sequence is combined to build a panorama. The alignments of all these 1-D array images are decided by motion vectors estimated from continuous images. Intuitively, this is equivalently to create a panorama with a line sensor. As a result, the manifold panorama is much more efficient and can also yield

good visual result. The manifold projection is adopted for building the distance-scanning system in this research.

1.3 Background Knowledge

The background knowledge of this thesis is described in this subsection. In subsection 1.3.1, the concept of the manifold projection model is reviewed. Another important background knowledge, linear pushbroom camera model, is surveyed in subsection 1.3.2, which plays a crucial role in our theoretical study for reconstructing wide-field drawings by using several real-time distance-scanning systems.

1.3.1 Manifold Projection Model [16][17]

Existing mosaicing methods for creating panoramic mosaics have several limitations and drawbacks. Unlike previous researches, manifold projection provides a very different viewpoint for building panoramic mosaics. This projection model enables the creation of panoramic mosaics from video sequence with uncontrolled camera motion (such as a hand-held camera motion).

The basic concept is that manifold projection imitates a brush to sweep the scene with a one-dimensional sensor array, as illustrated in Figure 1.1. The scene can be swept by such a 1-D sensor with any arbitrary camera motion. As a result, if we can estimate the camera motion, then we can figure out how to align the acquired 1-D array images, and thus the 2-D panoramic image of the scene can be constructed. In the case that the sensor array travels in a straight line with constant velocity and the orientation of the camera is fixed over the image acquisition duration, mosaics generated in this manner can be modeled by using a linear pushbroom camera as will be described in subsection 1.3.2.

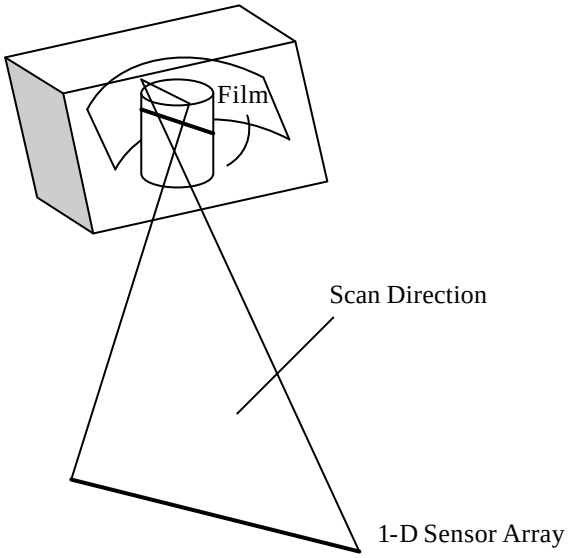

Figure 1.1 Scan system with 1-D sensor array.

In general, the camera motion may be arbitrary, and it seems impossible to correctly paste the 1-D array images coming from arbitrary camera motion to form a panorama. This problem becomes easier when the input is not a set of 1-D array images but a video sequence. Every frame in a video sequence can be considered as consisting of infinite stripes. A characteristic of a video sequence is that there exists overlapping regions between continuous frames. Thus the motion of the camera can be estimated from the overlapped regions. Then, the estimated motion vector can be used for alignment. The panoramic mosaic created by combining a set of aligned 1-D strips forms a new scene-to-image projection, which is referred to as the manifold projection.

Manifold projection has three major advantages than previous techniques in photo-mosaicing:

1 This projection can allow almost any arbitrary camera motion and scene structure.

2 The resolution and object size in the panoramic mosaics are the same as those in the original images. That is, there are no resolution distortions caused by manifold projection.

3 Computation is more economical as a result of only rotations and

translations are used for stripe alignment.

As to the selection of stripes, there are infinite stripes in a sequence of images, but only the stripes closest to the center of images are adopted for manifold projection. There are two major reasons for this selection:

1. As lens distortion is a radial effect, the stripe in the center is usually minimum than in the edge of the captured frame.

2. The alignment is ordinarily better in the center than any other places.

It is deserved to mention that two types of camera motion, forward motion and digital zoom, and the parallax of the scene failed the ordinary manifold projection. However, in [15] and [18], Peleg et al. proposed a new approach called pipe projection which is a new type of adaptive manifold projection that can be successfully applied on these conditions.

1.3.2 Linear Pushbroom Camera Model [6]

An optical system projecting an image onto a 1-D array sensor, typically a CCD array, is called a linear camera. Only the points which lie in the plane defined by the optical center and the sensor array are imaged. This plane is called the view plane as shown in Figure 1.2. A linear pushbroom sensor consists of a linear camera mounted on a moving platform. As the platform moves, the view plane sweeps out a region of space and 1-D images are acquired. Finally, the whole 1-D images constitute a 2-D image which lies on a plane called the image plane in 3-D space.

In order to simplify the camera model to facilitate computation and to provide a foundation of theoretical study and mathematical models, the linear pushbroom camera model has two assumptions as follows:

1. The sensor array is moving in a straight line with constant velocity.

2. The orientation of the camera, and hence the view plane, is constant over the image acquirement duration.

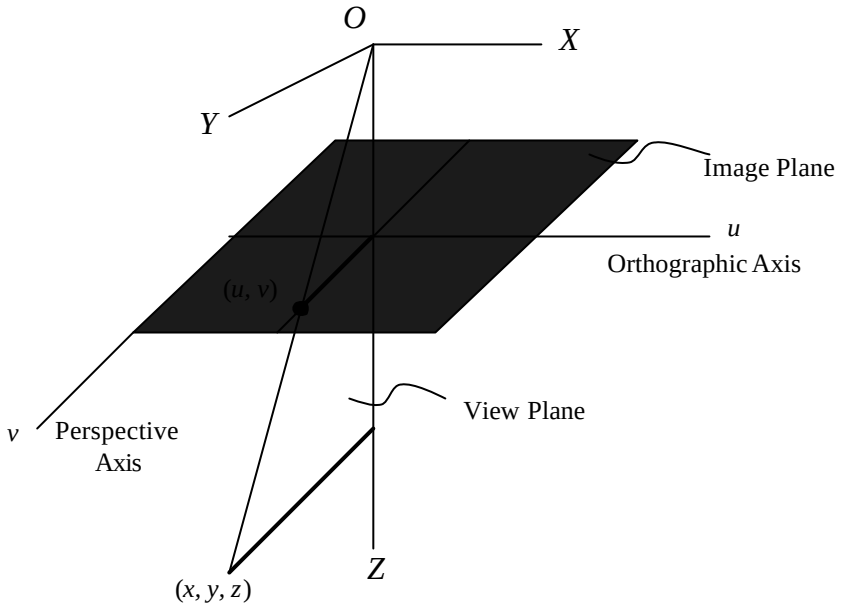

Figure 1.2 Projection under linear pushbroom camera.

This camera can be considered as a perspective camera moving in a linear orbit with a constant velocity and a fixed orientation as shown in Figure 1.2. Consequently, an arbitrary point **x** in space is imaged and represented by two coordinates u and v, respectively. The first coordinate, u, represents the time when the point **x** is imaged, and the second coordinate, v, represents the perspective projection on the sensor array. Besides, an orthogonal coordinate system affiliated to the camera frame is a right-handed coordinate system. The origin of this coordinate system is the center of projection and the z-axis is the orientation of the camera (such that the visible points have positive z coordinates). The x-axis is the camera's moving direction and the y-axis is defined as the cross product of the z and x axes.

First of all, if the coordinate of a point in space is $(0, y, z)$ with respect to the camera, then the 1-D projection coordinate of this point is $v = fy/z + p_v$, where f is the focal length and p_v is the principal point offset in the v direction. This equation can be written as Eq. 1.8:

$$\begin{bmatrix} wv \\ w \end{bmatrix} = \begin{bmatrix} f & p_v \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix}, \qquad \text{(Eq. 1.8)}$$

where w is a scale factor which is actually equal to z.

The question can be simplified by assuming that the camera is fixed and the world is moving, rather than considering a stationary world with a moving camera. Thus a point in space can be denoted as $\mathbf{x}(t) = (x(t), y(t), z(t))^T$, where t denotes time. The velocity vector of the points in the world with respect to the camera frame is denoted as $-\mathbf{V} = -(V_x, V_y, V_z)^T$ due to the velocity vector of the camera with respect to the world is $\mathbf{V}$. Consider that a moving point in space crosses the view plane at time t_{im} at position $(0, y_{im}, z_{im})^T$ will be imaged. In 2-D image plane, the coordinate of this imaged point will be at location (u, v), where $u = t_{im}$ and v can be expressed by Eq. 1.8. That is, location (u, v) can be represented as follows:

$$\begin{bmatrix} u \\ wv \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & f & p_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_{im} \\ y_{im} \\ z_{im} \end{bmatrix} \qquad \text{(Eq. 1.9)}$$

Because all points are moving with the same velocity, the coordinates of all points can be expressed as a function of time. Thus let $\mathbf{x}_0$ be the coordinate of a moving point $\mathbf{x}$ at time $t = 0$, the coordinates of all points are given by the following function of time:

$$\mathbf{x}(t) = \mathbf{x}_0 - t\mathbf{V} = (x_0, y_0, z_0)^T - t(V_x, V_y, V_z)^T \qquad \text{(Eq. 1.10)}$$

Since the view plane is the plane at $x = 0$, and in this moment the point $\mathbf{x}$ crosses the view plane at $t_{im} = x_0/V_x$. This shows that the point is at position

$$(0, y_{im}, z_{im})^T = (0, y_0 - x_0 V_y/V_x, z_0 - x_0 V_z/V_x)^T$$

This can be re-written as

$$\begin{bmatrix} t_{im} \\ y_{im} \\ z_{im} \end{bmatrix} = \begin{bmatrix} 1/V_x & 0 & 0 \\ -V_y/V_x & 1 & 0 \\ -V_z/V_x & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \qquad \text{(Eq. 1.11)}$$

Combining Eq. 1.9 and Eq. 1.11 gives the following equation:

$$\begin{bmatrix} u \\ wv \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & f & p_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/V_x & 0 & 0 \\ -V_y/V_x & 1 & 0 \\ -V_z/V_x & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \qquad \text{(Eq. 1.12)}$$

In Eq. 1.12, $(x_0, y_0, z_0)^T$ is the coordinate of the point **x** in terms of the camera-based coordinate system. Generally speaking, the coordinate system is in terms of some fixed external orthogonal coordinate systems. Therefore, let $(x, y, z)^T$ be the coordinate of the point in such a coordinate system. As these two coordinate systems are orthogonal, the coordinates are related to each other via a translation and rotation as follows:

$$\begin{aligned}(x_0, y_0, z_0)^T &= \mathbf{R}\left((x,y,z)^T - \left(T_x, T_y, T_z\right)^T\right) \\ &= (\mathbf{R} \mid -\mathbf{RT})(x, y, z, 1)^T,\end{aligned} \qquad \text{(Eq. 1.13)}$$

where **R** is a rotation matrix and $\mathbf{T} = (T_x, T_y, T_z)^T$ is the position of the camera when time $t = 0$.

Finally, combining Eq. 1.12 and Eq. 1.13, the linear pushbroom camera model is conducted as follows:

$$\begin{aligned}\begin{bmatrix} u \\ wv \\ w \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & f & p_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/V_x & 0 & 0 \\ -V_y/V_x & 1 & 0 \\ -V_z/V_x & 0 & 1 \end{bmatrix} (\mathbf{R} \mid -\mathbf{RT}) \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \\ &= \mathbf{M}(x, y, z, 1)^T,\end{aligned} \qquad \text{(Eq. 1.14)}$$

where **M** is a 3×4 matrix. The linear pushbroom camera model, $(u, wv, w)^T = \mathbf{M}(x, y, z, 1)^T$, should be compared with the basic pin-hole camera model, $(wu, wv, w)^T = \mathbf{M}(x, y, z, 1)^T$, where $(x, y, z)^T$ is the coordinate of a point in the world. It is clear that a linear pushbroom image may be considered as a perspective projection in the v direction and an orthographic projection in the other direction u; nevertheless, both the coordinates, u and v, of the pinhole image are perspective projections. Another obvious difference is that the matrix of the pin-hold camera model is homogeneous; however, the linear pushbroom camera matrix is not. That is, by multiplying linear pushbroom camera matrix **M** with an arbitrary factor k, the v coordinate is unchanged while the u coordinate is scaled by k.

1.4 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 introduces the developed real-time distance-scanning system. In Chapter 3, mathematical models for acquiring a planar scene with more than one linear pushbroom camera are proposed and used for reconstructing wide-field drawings. Experimental results are shown in Chapter 4. Finally, conclusions and future works are given in Chapter 5.