Computer Science Depatrment of The University of Auckland
CITR at Tamaki Campus (http://www.tcs.auckland.ac.nz)

CITR-TR-28 August 1998

Automatic Generation of Stereo Images from Multiple Monocular Object Views

Yu-Fei Huang [1], Shou-Kang Wei [1], and Reinhard Klette [1]

Abstract

Traditionally, to produce binocular stereo images, each single object view requires two shots with a proper camera displacement to generate a stereo pair. The total number of images required is twice as much as for monocular views. Special rigs, such as a slider, are also required for camera displacement if a normal single-eye camera is used. So, it is of advantage if the stereo views can be generated from multiple monocular images automatically. In this report, we present an image-based approach generating the stereo images automatically for multiple monocular object views. This approach does not include or require 3-D object reconstruction. The image reprojection technique is adopted and ensures the resulting images are physically valid. For fully calibrated images, the stereo pair can be generated directly using the image reprojection while for the uncalibrated case a preprocess is required to obtain the relative camera projection matrices. The concept of epipolar geometry and its role played in the perspective camera projection matrix recovery processes are then discussed in depth. If the viewing angle between two camera positions associated with two object reference images is sufficiently large, the closer in-between views are synthesized based on 3-D morphing.

[1] The University of Auckland, Computer Science Department, CITR, Tamaki Campus (Building 731), Glen Innes, Auckland, New Zealand

Automatic Generation of Stereo Images from Multiple Monocular Object Views

Yu-Fei Huang, Shou-Kang,Wei, and Reinhard Klette

CITR, Computer Science Department, The University of Auckland
Tamaki Campus, Auckland, New Zealand

Abstract

Traditionally, to produce binocular stereo images, each single object view requires two shots with a proper camera displacement to generate a stereo pair. The total number of images required is twice as much as for monocular views. Special rigs, such as a slider, are also required for camera displacement if a normal single-eye camera is used. So, it is of advantage if the stereo views can be generated from multiple monocular images automatically. In this report, we present an image-based approach generating the stereo images automatically for multiple monocular object views. This approach does not include or require 3-D object reconstruction. The image reprojection technique is adopted and ensures the resulting images are physically valid. For fully calibrated images, the stereo pair can be generated directly using the image reprojection while for the uncalibrated case a preprocess is required to obtain the relative camera projection matrices. The concept of epipolar geometry and its role played in the perspective camera projection matrix recovery processes are then discussed in depth. If the viewing angle between two camera positions associated with two object reference images is sufficiently large, the closer in-between views are synthesized based on 3-D morphing.

Keywords: Object visualization, stereo image, epipolar geometry, image reprojection, view synthesis, image morphing.

1 Introduction

Object visualizations have been demanded in many applications [Che95, KSK98, MD97, Vin95]. One of the common approaches is playing some key views of objects in sequence to convey the audience of the impression of an entire object. In this case the audience plays a passive role and prior knowledge or experience about the object are required. An alternative approach is that users are allowed to manipulate the object interactively according to their preferences. To accomplish the task, the shape and surface details of an object are requisite to be reconstructed. Many 3-D reconstruction techniques are proposed. However, they have not been robust enough yet to deal with complex object shape, nor for intricate

surface details presenting in uncertain illumination conditions. The visibility and surface reflectance property analysis are known to be difficult to measure and approximate accurately. The result of this approach may not satisfy the audience due to its loss of both shape and color information.

Beside of 3-D shape reconstruction, many researchers have devoted themselves in exploiting the correspondences between multiple reference views. The epipolar constraint, for example, is one of useful invariant properties for finding the corresponding points between two or many uncalibrated images [AS98, ZDFL94]. Utilizing the coherence between multiple views to generate a novel view, i.e. a new viewing direction to the object, is commonly known as *view synthesis* technique. The result is convinced if dense correspondence maps are possible to generate.

S.E.Chen proposed an image-based solution to the problem instantly [Che95]. The idea is to tile up all possible views onto a 2-D movie-map. So users are impressed as manually rotating the object horizontally and vertically to see the different views. His approach has been implemented in the Apple's QTVR system. One nice feature of it is that the quality of rendered views is independent to the complexity of object shape and surface properties. It works nicely as long as the amounts of views are sufficient to describe a certain motion, such as object rotation. To be more specific, here "sufficient" means the transition between any two views is smooth enough that audience cannot notice the gap in-between. Generally, the amount of images required for smooth motion transition is contents dependent. For instance, a simple regular object with uniform surface pattern, e.g. a green glass vase, may not require many images while a complex sculpture may do. Despite, one major difficulty for this application is the camera setup, especially for multiple-layers[1], where the equipment allows acquiring arbitrary angles of object views is not widely available. Hence generating the in-between views from the basic images is essential to the quality improvement for this application.

So far, many applications of object visualizations still stay in conveying single eye's depth cues to the audience. However, in this report the object visualizations that provide binocular parallax to the audience are considered. Normally, to produce binocular stereo images, each single view requires two shots with a proper camera displacement to generate a stereo pair [WHK98]. The total number of images required is twice as much as for the monocular views. Special rigs, such as a slider, are also required for camera displacement if a normal single-eye camera is used. Therefore, it is helpful if the stereo views can be generated from monocular images automatically.

Our task is to achieve such automation. The input data we are dealing with are sets of uncalibrated object images that basically describe an object from various angles of view toward the rotation center, which is depicted in Fig. 1. To demonstrate the main idea of our approach, we describe our method mainly based on a single-layer-input data first. Generalization of the method to more images, i.e. between multiple-layers, is straightforward and will be discussed later.

The task is divided into two possible cases, one for dense input images, such as 36 images around a single layer; another for a sparse case, e.g. eight images attempting to describe 360 degrees of object views. For the first case, we show how the physical-valid stereo image can be generated directly from the given images without generation of in-between images. In contrast, the generation of in-between views

[1] Each layer, camera orbits around the object rotation center with fixed slant angle.

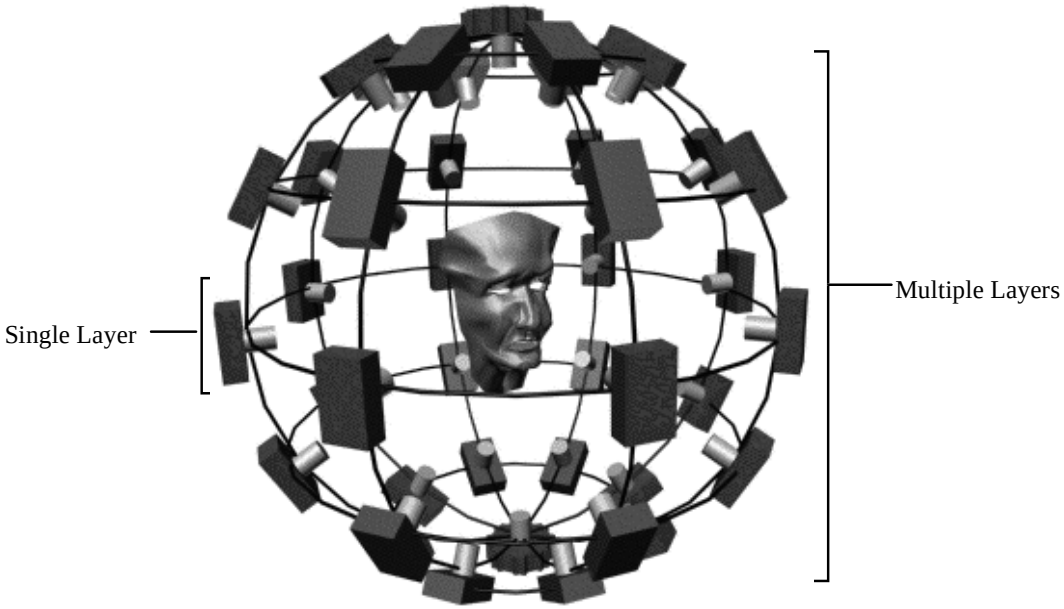

Figure 1: Multiple monocular object views as used in computer vision for aspect graph based object modelling.

are required in the second case to avoid ghost[2] area appears. Note that the angle between each rotation shown in the Fig. 1 may not need to be uniformly equal.

In the remainder of this report, we first, in Section 2, describe how to recover the camera projection matrix associated with each uncalibrated input image. Section 3 explains that the stereo views can be generated using image reprojection for both calibrated and uncalibrated images. To deal with the sparse input images case, in Section 4, we introduce the image morphing technique and show the fundamental problem that the invalidity of applying 2-D image morphing on 3-D case. A 3-D image morphing therefore is purposed and used for novel view synthesis. Finally future works and possible applications are concluded in Section 5.

2 Projection Matrices from Uncalibrated Images

The *camera projection matrix* specifies a transformation from 3-D to 2-D, and it is used to map a space-point to an image-point. It describes internal camera parameters, e.g. focal length, image unit length, and external pose with respect to the world coordinate system, e.g. orientation, translation. In practice, we are only given a set of reference images that describe the multiple views of an object, but leave each associated projection matrix unavailable. Our task is to recover *relative camera projection matrices*[3]. Each relative matrix is associated with one image and may be analyzed based on correspondences between reference images. The relative camera projection matrices are sufficient to compute the object stereo images, which will be explained in detail in Section 3.

[2]The object surfaces appear in one stereo image, but not in the other.

[3]They describe the relative geometrical relationship between multiple cameras' positons and orientations.

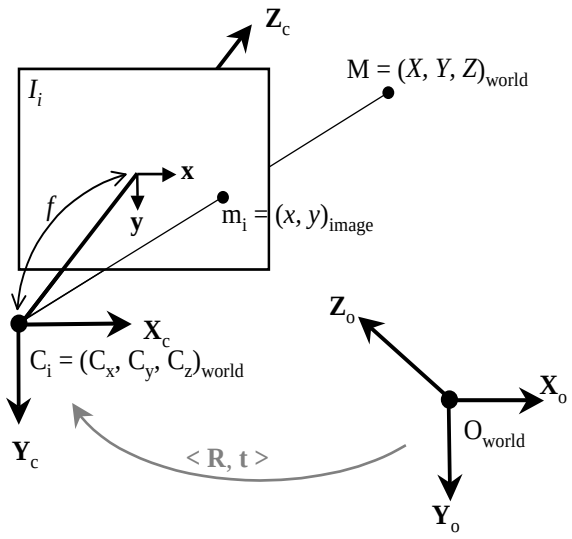

Figure 2: Perspective camera geometry in world coordinate system. The symbols are explained in the text.

In this section, the invariance of geometric relationships between space-points and two reference images, called *epipolar constraint*, is investigated in depth. We start by establishing notations used throughout this report. Then we give a definition of thecamera projection matrix and an introduction to the epipolar geometry. At last, we show how to use such a constraint to recover two relative camera projection matrices.

2.1 Notation

Our conventions follow G.Xu and Z.Zhang's book [XZ96]. A matrix is in uppercase bold-faced font. A vector is in lowercase bold-faced font. A point in 3-D space is in uppercase font. A point in 2-D space is in lowercase font. A point covered by a tilde "$\sim$" denotes a homogenous vector, i.e. padding 1 into the last element. A scalar or constant number is in italic font. The world coordinate system origin is denoted as O and the camera optical center is denoted as C_i, where index i stands for ith camera in the space. An arbitrary point in 3-D space is denoted as M with respect to the world coordinate system. The projection ray from C_i to M intersects the image plane, I_i, at the point m_i with respect to the ith image coordinate system. We depict these in Fig. 2. All the symbols are defined uniquely throughout the whole report.

2.2 Camera Projection Matrix

There are several camera models commonly used in the computer vision community, such as perspective, orthographic, weak projective and paraperspective projections. In this report we only present the perspective case.

Considering a canonical planar pinhole camera, a perspective projection matrix, denoted as $\mathbf{P}$, is a 3×4 matrix which projects a 3-D space-point onto the 2-D image plane. For instance, if we define a space-point $\mathrm{M} = (X, Y, Z)$ with respect to the world coordinate system and its projected image point $\mathrm{m} = (x, y)$ with respect to the image coordinate system, then

$$s\widetilde{\mathbf{m}} = s \begin{pmatrix} \mathrm{m} \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{P} \begin{pmatrix} \mathrm{M} \\ 1 \end{pmatrix} = \mathbf{P}\widetilde{\mathbf{M}}, \tag{1}$$

where s is an arbitrary nonzero scalar and f is the effective camera focal length. Let $\mathbf{E}$ be the 3×3 matrix

$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

from now on. The matrix $\mathbf{P}$ described above is a standard form of camera perspective projection matrix in which the camera coordinate system and the world coordinate system both coincide perfectly. But it is usually not applicable due to that the desire world coordinate system is different from the actual camera coordinate system, for instance, the case with multiple camera poses.

For the multiple views of an object, we have several camera poses situated in 3-D space. Each of them has a projection matrix associated with it to represent the mapping of 3-D space-points onto its image plane. So in the real application, the matrix $\mathbf{P}$ is defined by camera *extrinsic parameters* and *intrinsic parameters*. The extrinsic parameters of a camera define the orientation and position of the camera in 3-D space with respect to the world coordinate system. In the Fig. 2, camera's orientation can be represented by a 3×3 rotation matrix $\mathbf{R}$, which is the product of the standard rotation matrices in respect to each axis of the world coordinate system. The position of a camera optical center C can be expressed as a vector from O to C denoted as $\mathbf{t}$. Here we call $\mathbf{R}$ and $\mathbf{t}$ the extrinsic parameters of the camera. Hence the projection of a point M on the image plane (i.e. a 2-D point m with respect to the image coordinate system) can be described as

$$s\widetilde{\mathbf{m}} = s \left[\begin{array}{c|c} \mathbf{B} & -\mathbf{B}\mathbf{t} \end{array} \right] \widetilde{\mathbf{M}} = \mathbf{P}\widetilde{\mathbf{M}},$$

where $\mathbf{B} = \mathbf{E}\mathbf{R}$, is a 3×3 matrix, and s is an arbitrary nonzero scalar. We use the form $[\mathbf{H} \mid \mathbf{V}]$ to denote the decomposition of a 3×4 martix, where $\mathbf{H}$ is a 3×3 matrix and $\mathbf{V}$ is a 3-vector.

The intrinsic parameters of a camera define a mapping between the actual image plane and the ideal image coordinate system. It is illustrated in the Fig. 3, where the image-plane basis vectors $\mathbf{a}$ and $\mathbf{b}$ are not orthogonal and are separated by angle φ; and the pixel unit of the image along $\mathbf{a}$ and $\mathbf{b}$ are k_a and k_b respectively to the unit used in the ideal image coordinate system. Furthermore we have that the camera optical axis intersects the image plane at point $(\mathrm{c}_a, \mathrm{c}_b)$. All those parameters are called the intrinsic parameters of a camera. The relationship between an ideal image-coordinate-system point m_o

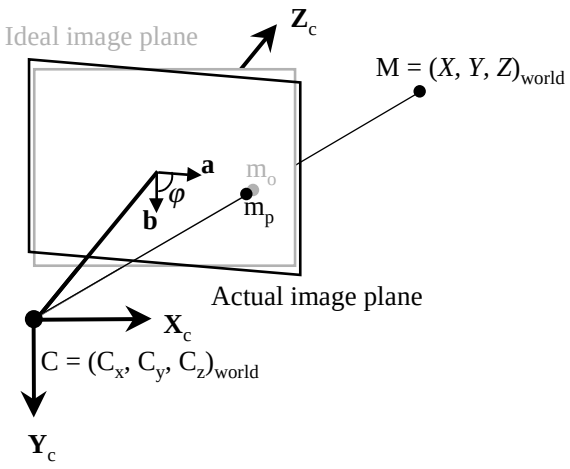

Figure 3: Camera intrinsic parameters. The symbols are explained in the text.

and the actual image-coordinate-system point m_p can be easily shown as

$$\widetilde{\mathbf{m}}_p = \begin{bmatrix} k_a & -k_a \cot\varphi & c_a \\ 0 & k_b/\sin\varphi & c_b \\ 0 & 0 & 1 \end{bmatrix} \widetilde{\mathbf{m}}_o = \mathbf{A}\widetilde{\mathbf{m}}_o .$$

Therefore the camera perspective projection matrix $\mathbf{P}$ for a canonical planar pinhole camera is defined as

$$\mathbf{P} = \mathbf{A} \begin{bmatrix} \mathbf{B} \mid -\mathbf{B}\mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{H} \mid -\mathbf{H}\mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{H} \mid \mathbf{h} \end{bmatrix}, \tag{2}$$

where $\mathbf{H} = \mathbf{A}\mathbf{B}$ and $\mathbf{h} = -\mathbf{H}\mathbf{t}$.

2.3 Epipolar Geometry

In Fig. 4(a), the optical centers C_1, C_2 of two cameras plus a point M in space define a plane called *epipolar plane* denoted by $\wp$. The projection of point C_1 on the image plane I_2 is called *epipole* denoted by e_2, and vise versa for e_1. The lines joining m_i and e_i are called *epipolar lines*. We may define infinite epipolar planes commonly intersecting the line joining two camera optical centers $\mathrm{C}_1\mathrm{C}_2$ as shown in Fig. 4(b). This implies that all points lying on the same epipolar plane will be projected onto the same epipolar line on each image plane. Thus, the corresponding points between two reference images are constrained on the epipolar lines. More importantly, later we show how to use such a geometric property to estimate the relative projection matrices for two uncalibrated cameras.

With two reference image planes I_1 and I_2 with respect to the camera optical centers C_1 and C_2, their relative perspective projection matrices are denoted as $\mathbf{P}_1$ and $\mathbf{P}_2$. Let $\mathbf{P}_i$ be decomposed as the concatenation of a 3×3 sub-matrix $\mathbf{H}_i$ and a 3-vector $\mathbf{h}_i$, i.e. $\mathbf{P}_1 = \begin{bmatrix} \mathbf{H}_1 \mid \mathbf{h}_1 \end{bmatrix}$ and $\mathbf{P}_2 = \begin{bmatrix} \mathbf{H}_2 \mid \mathbf{h}_2 \end{bmatrix}$. M is an arbitrary point in 3D space and m_1 and m_2 are its projection points on the images I_1 and I_2 respectively.

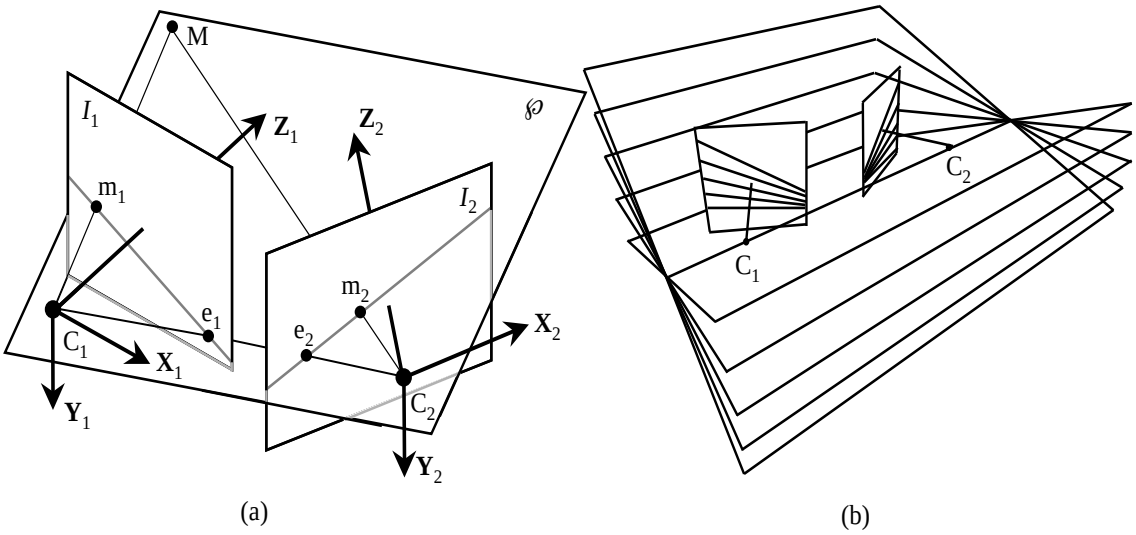

Figure 4: (a) Epipolar geometry. (b) Multiple epipolar planes and associated epipolar lines. The symbols are explained in the text.

From Section 2.2, we have

$$s_1\widetilde{\mathbf{m}}_1 = \left[\begin{array}{c|c} \mathbf{H}_1 & \mathbf{h}_1 \end{array}\right] \left[\begin{array}{c} \mathrm{M} \\ 1 \end{array}\right] \quad \text{and} \quad s_2\widetilde{\mathbf{m}}_2 = \left[\begin{array}{c|c} \mathbf{H}_2 & \mathbf{h}_2 \end{array}\right] \left[\begin{array}{c} \mathrm{M} \\ 1 \end{array}\right].$$

We expand these two equations and get

$$s_1\widetilde{\mathbf{m}}_1 = \mathbf{H}_1\mathrm{M} + \mathbf{h}_1 \quad \text{and} \quad s_2\widetilde{\mathbf{m}}_2 = \mathbf{H}_2\mathrm{M} + \mathbf{h}_2.$$

We assume matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ are both non-singular. Eliminating M from both equations we have

$$s_1\mathbf{H}_1^{-1}\widetilde{\mathbf{m}}_1 - \mathbf{H}_1^{-1}\mathbf{h}_1 = s_2\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2 - \mathbf{H}_2^{-1}\mathbf{h}_2.$$

Multiplying $\mathbf{H}_1$ to both sides gives

$$s_1\mathbf{H}_1\mathbf{H}_1^{-1}\widetilde{\mathbf{m}}_1 - \mathbf{H}_1\mathbf{H}_1^{-1}\mathbf{h}_1 = s_2\mathbf{H}_1\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2.$$

This implies

$$s_1\widetilde{\mathbf{m}}_1 = s_2\mathbf{H}_1\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2 + \mathbf{h}_1.$$

Perform cross product to both sides with vector $(\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2)$ results

$$s_1\left(\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right) \times \widetilde{\mathbf{m}}_1 = s_2\left(\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right) \times \mathbf{H}_1\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2.$$

By applying the left dot product with a row vector $\widetilde{\mathbf{m}}_1^T$ to both sides so that scalar s_1 and s_2 can be eliminated, we have

$$\widetilde{\mathbf{m}}_1^T \cdot \left(\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right) \times \widetilde{\mathbf{m}}_1 = \widetilde{\mathbf{m}}_1^T \cdot \left(\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right) \times \mathbf{H}_1\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2.$$

This implies

$$0 = \widetilde{\mathbf{m}}_1^T \cdot \left(\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right) \times \mathbf{H}_1\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2. \tag{3}$$

Here we introduce a notation $[\mathbf{v}]_\times$ to denote the *skew symmetric matrix* of vector $\mathbf{v}$, which is

$$[\mathbf{v}]_\times = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}_\times = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}.$$

So the Eq. 3 can be rewritten as

$$\widetilde{\mathbf{m}}_1^T \mathbf{F} \widetilde{\mathbf{m}}_2 = 0, \tag{4}$$

where $\mathbf{F} = \left[\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right]_\times \mathbf{H}_1\mathbf{H}_2^{-1}$.

The matrix $\mathbf{F}$ is called *fundamental matrix*, which describes the mapping between a pair of corresponding points on two images. Thus for any pair of corresponding points m_1 and m_2 on the image planes I_1 and I_2 respectively, they must satisfy the epipolar constraint of Eq. 4. The derivation of the fundamental matrix above has been described extensively in [XZ96].

As we know, an image line, say l, can be represented by the cross product of two image points, say m and n, which are incident to it, i.e. $l = \widetilde{\mathbf{m}} \times \widetilde{\mathbf{n}}$. Consider a point m_2 in the image plane I_2, its epipolar line on the image plane I_1 must pass through both the epipole and the m_2's corresponding point, say m_1. Thus, two points e_1 and m_1 can represent the epipolar line of m_2 in I_1, say l_{m_2}, i.e. $l_{m_2} = \widetilde{\mathbf{e}}_1 \times \widetilde{\mathbf{m}}_1$. In order to calculate l_{m_2} we need to compute both e_1 and m_1 in the world coordinate system first.

Image point e_1 is the projection of the camera optical center C_2 on the image plane I_1. Hence,

$$s_e \widetilde{\mathbf{e}}_1 = \mathbf{P}_1 \widetilde{\mathbf{C}}_2, \tag{5}$$

where s_e is an arbitrary nonzero scalar. The position of the camera C_2 with respect to the world coordinate system is defined as $\mathbf{t}_2$ that is a vector from origin O to C_2. So we have

$$\mathbf{P}_2 = \left[\ \mathbf{H}_2 \mid -\mathbf{H}_2\mathbf{t}_2\ \right] = \left[\ \mathbf{H}_2 \mid -\mathbf{H}_2\mathrm{C}_2\ \right] = \left[\ \mathbf{H}_2 \mid \mathbf{h}_2\ \right], \tag{6}$$

it implies that $-\mathbf{H}_2\mathrm{C}_2 = \mathbf{h}_2$, therefore $\mathrm{C}_2 = -\mathbf{H}_2^{-1}\mathbf{h}_2$. By Eq. 5, we get

$$s_e \widetilde{\mathbf{e}}_1 = \mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2 = \left[\ \mathbf{H}_1 \mid \mathbf{h}_1\ \right] \begin{pmatrix} \mathbf{H}_2^{-1}\mathbf{h}_2 \\ 1 \end{pmatrix} = \mathbf{P}_1 \widetilde{\mathbf{C}}_2.$$

As defined, image points m_1 and m_2 are the projection of the same 3-D point M onto the image planes I_1 and I_2. So the intersection of two rays, one starting from C_1 passing through m_1 and another from C_2 passing through m_2, defines the 3-D point M in the space. By given m_2, point M can be written as

$$\mathrm{M} = \mathrm{C}_2 + s\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2 = -\mathbf{H}_2^{-1}\mathbf{h}_2 + s\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2 = \mathbf{H}_2^{-1}\left(\ s\widetilde{\mathbf{m}}_2 - \mathbf{h}_2\ \right),$$

where s is a scalar. Project M onto the image I_1, we have

$$\begin{aligned} s_1\widetilde{\mathbf{m}}_1 &= \mathbf{P}_1\widetilde{\mathbf{M}}, \\ &= \left[\ \mathbf{H}_1 \mid \mathbf{h}_1\ \right] \begin{pmatrix} \mathrm{M} \\ 1 \end{pmatrix}, \\ &= \left[\ \mathbf{H}_1 \mid \mathbf{h}_1\ \right] \begin{pmatrix} \mathbf{H}_2^{-1}(s\widetilde{\mathbf{m}}_2 - \mathbf{h}_2) \\ 1 \end{pmatrix}, \\ &= \mathbf{H}_1\mathbf{H}_2^{-1}\left(\ s\widetilde{\mathbf{m}}_2 - \mathbf{h}_2\ \right) + \mathbf{h}_1, \\ &= s\mathbf{H}_1\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2 + \mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2, \end{aligned}$$

where s_1 is an arbitrary nonzero scalar.

Therefore, the epipolar line of m_2 in the image plane I_1 can be derived as follows,

$$\begin{aligned} l_{m_2} &= s_e\, s_1 \widetilde{\mathrm{e}}_1 \times \widetilde{\mathbf{m}}_1, \\ &= \left(\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right) \times \left(s\mathbf{H}_1\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2 + \mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right), \\ &= \left(\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right) \times \left(s\mathbf{H}_1\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2\right), \\ &= \left[\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right]_\times \mathbf{H}_1\mathbf{H}_2^{-1}\widetilde{\mathbf{m}}_2, \\ &= \mathbf{F}\widetilde{\mathbf{m}}_2. \end{aligned}$$

Since every epipolar line in the image plane I_1 must pass through the epipole e_1, the dot product of any epipolar line and epipole equals to zero. It means, $\mathrm{e}_1 \cdot l_{m_2} = \mathrm{e}_1^T\mathbf{F}\widetilde{\mathbf{m}}_2 = 0$, for any image point m_2 in I_2. This equality holds only if $\mathrm{e}_1^T\mathbf{F} = 0$. Therefore the dimension of row space of the matrix $\mathbf{F}$ is at most two. In general, the rank of $\mathbf{F}$ is equal to two; thus it defines a one-to-one mapping from a set of image points to a set of image lines. This sort of mapping is called *correlation*.

The fundamental matrix, if known, is very useful to assist the corresponding point search between two reference images. For example, if we have point m_1 in the image plane I_1 and looking for its corresponding point m_2 in the image plane I_2, then the matrix $\mathbf{F}$ is used to map m_1 to an epipolar line in the image plane I_2 where point m_2 lying on. Hence the 2-D search space over the image plane is reduced down to a 1D search, i.e. along the mapped epipolar line.

2.4 Estimating the Fundamental Matrix

In Section 2.3, the epipolar geometry can be discovered if we have the projection matrices for all the camera positions in the image acquisition setup. In other words, we must know the geometrical relationship between every pair of images. However, in our situation the complete geometrical relationship between any selected pair of images is unknown. Thus we cannot use Eq. 4 to compute the fundamental matrix. Instead, a partial geometrical relationship may be established by involving some human intervention in which a few matching points are identified. Therefore we may apply a parameter estimation algorithm to find an optimal solution for the matrix $\mathbf{F}$ numerically.

Assuming $\mathrm{m} = (m_x, m_y)$ and $\mathrm{n} = (n_x, n_y)$ are one pair of matched points in two reference images respectively. We know that

$$\widetilde{\mathbf{m}}^T\mathbf{F}\widetilde{\mathbf{n}} = 0,$$

which can be expanded as follows

$$(m_x, m_y, 1)\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}\begin{pmatrix} n_x \\ n_y \\ 1 \end{pmatrix} = 0,$$

where f_{ij} is an element of $\mathbf{F}$. This equation can be rewritten as a linear and homogenous function Q with four variables and nine unknown coefficients that are the elements of $\mathbf{F}$, i.e.

$$Q(m_x, m_y, n_x, n_y) \tag{7}$$

$$= f_{11}m_x n_x + f_{12}m_x n_y + f_{13}m_x + f_{21}m_y n_x + f_{22}m_y n_y + f_{23}m_y + f_{31}n_x + f_{32}n_y + f_{33}$$
$$= 0.$$

Hence, our task becomes that by given a set of k pairs of corresponding points $\{(\ m_{x_i},\ m_{y_i},\ n_{x_i},\ n_{y_i}) : i = 1, \ldots, k\}$, we want to find those nine coefficients that best fit to the Eq. 7. In a shorter form, let $\mathbf{w} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^T$, and $\mathbf{d}_i = (m_{x_i}n_{x_i},\ m_{x_i}n_{y_i},\ m_{x_i},\ m_{y_i}n_{x_i},\ m_{y_i}n_{y_i},\ m_{y_i},\ n_{x_i},\ n_{y_i},\ 1)^T$, so that $Q(m_{x_i},\ m_{y_i},\ n_{x_i},\ n_{y_i}) = \mathbf{d}_i^T\mathbf{w}$.

Because the fundamental matrix $\mathbf{F}$ is defined up to a scalar factor, even though it consists of nine elements, we may set one of nine elements of $\mathbf{F}$ to be 1. Now it is only left eight variables in its place need to be computed. Furthermore, we have shown that the rank of $\mathbf{F}$ is at most 2, thus the number of free parameter of $\mathbf{F}$ can then be further reduced down to seven [XZ96]. Ideally, to determine the fundamental matrix $\mathbf{F}$, we need at least seven matched points (i.e. seven pairs of corresponding points, $k = 7$) from two reference images. In practice, the factors of false matching and discontinuity of discrete data arisen from the image digitalization, an optimization scheme is requisite to minimize the potential errors it may occur. One way is to strengthen our solution by giving more matching points, i.e. more than seven, to constrain the estimation. There are many of numerical approximation methods available to estimate $\mathbf{F}$ with over-specified data. Here we demonstrate a simple approach - linear least-squares.

For more than seven matching points (i.e. $k > 7$), the problem of finding an optimal solution is equivalent to minimize the following function:

$$U(\mathbf{w}) = \sum_{i=1}^{n} Q^2(m_{x_i}, m_{y_i}, n_{x_i}, n_{y_i}).$$

Clearly, there exists a trivial solution $f_{ij} = 0$ for all $i, j = 1, \ldots, 3$, which is not what we want. In order to avoid it, we should impose some constraints to the coefficients of $Q(m_x, m_y, n_x, n_y)$. One way is to set one of the coefficients to 1. Without loss of generality, we assume that f_{33} is not equal to zero, and hence we can set f_{33} = -1. Let $\mathbf{w}' = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32})^T$, and $\mathbf{d}_i' = (m_{x_i}n_{x_i},\ m_{x_i}n_{y_i},\ m_{x_i},\ m_{y_i}n_{x_i},\ m_{y_i}n_{y_i},\ m_{y_i},\ n_{x_i},\ n_{y_i})^T$. Moreover, $\mathbf{D} = [\mathbf{d}_1', \mathbf{d}_2', \ldots, \mathbf{d}_k']^T$, and $\mathbf{v} = (v,\ v,\ \ldots,\ v)^T$, where $\mathbf{v}$ is a k-vector and its element v is the last element of $\mathbf{d}_i$. By given k points, our system equation can be rewritten as

$$Q(m_{x_i}, m_{y_i}, n_{x_i}, n_{y_i}) = \mathbf{d}_i'^T\mathbf{w}' - v = 0.$$

Thus the function to be minimized becomes

$$U(\mathbf{w}) = (\mathbf{D}\mathbf{w}' - \mathbf{v})^T(\mathbf{D}\mathbf{w}' - \mathbf{v}).$$

The solution can be obtained by setting its first derivative to be zero and yield

$$\frac{\partial U(\mathbf{w})}{\partial \mathbf{w}'} = 2\mathbf{D}^T(\mathbf{D}\mathbf{w}' - \mathbf{v}) = 2\mathbf{D}^T\mathbf{D}\mathbf{w}' - \mathbf{D}^T\mathbf{v} = 0.$$

Hence, function $U(\mathbf{w})$ is minimum when

$$\mathbf{w}' = \left(\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{D}^T\mathbf{v},$$

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}' \\ -1 \end{pmatrix},$$

if there exists an unique global minimum. This method will fail if the element we set to 1 is actually zero or much smaller than the other elements. Since we will not know which element is not zero priori, we could set each element of $\mathbf{F}$ to -1 for nine iterations and choose the one with minimum value of $U(\mathbf{w})$.

2.5 Projection Matrix from Fundamental Matrix

For given two uncalibrated reference images, our goal is to recover the relative perspective projection matrices associated with those two cameras. Z. Zhang has pointed out a nice property that, if the epipolar geometry of two uncalibrated images is known, i.e. the fundamental matrix, then the relative camera projection matrices can be determined [ZX97]. However, it is only up to a linear transformation depending on the projection model of the camera. For example, with perspective projection cameras the recovered camera projection matrices are defined up to a *projective transformation* in 3-D space while with orthographic, weak projective and paraperspective projections cases, their recovered camera projection matrices are defined up to an *affine transformation*. A projective transformation is represented by a non-singular 4×4 matrix acting on homogenous vectors. When we said "only up to a projective transformation", it means if $\mathbf{P}_1$ and $\mathbf{P}_2$ are two perspective camera projection matrices and their epipolar geometry, fundamental matrix $\mathbf{F}$, is established, for any arbitrary projective transformation $\mathbf{T}$ in 3-D space $\mathbf{P}'_1 = \mathbf{P}_1\mathbf{T}$ and $\mathbf{P}'_2 = \mathbf{P}_2\mathbf{T}$ remain constantly consistent with $\mathbf{F}$.

Let us recall the notations from the previous sections, that we will use below for the camera projection matrix recovery calculation. We have two uncalibrated images, and the associated fundamental matrix $\mathbf{F}$ which has been determined using the method described in the Section 2.4. The relative two camera perspective projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ with respect to image planes I_1 and I_2 are our target to be recovered.

The property described above has suggested that there is an infinite number of projective bases which all satisfy the epipolar geometry, hence there is no way to recover the absolute camera projective matrices associated with those images. Nevertheless, any pair of recovered camera projective matrices satisfying the epipolar geometry has fulfilled our primary goal in reprojecting those two images, which will be explained in detail in Section 3.

One way to represent the relative camera projective matrices recovered from the fundamental matrix $\mathbf{F}$ is to use a *canonical representation* described in [BZM94, LT94, ZX97]. It can be expressed as follows,

$$\mathbf{P}_1 = [\mathbf{H}_1 \mid \mathbf{h}_1] \quad \text{and} \quad \mathbf{P}_2 = [\mathbf{I} \mid 0]\,. \tag{8}$$

Here $\mathbf{P}_1$ and $\mathbf{P}_2$ are defined with respect to the world coordinate system which is assumed to coincide with the second camera coordinate system. This representation is equivalent to the strongly calibrated case in which the camera intrinsic parameters are known. In this case, $\mathbf{F}$ is called the *essential matrix*. Matrix $\mathbf{H}_1$ describes the orientation of the first camera with respect to the second camera coordinate system, i.e. the world coordinate system. Vector $\mathbf{h}_1$ is equal to $-\mathbf{H}_1\mathrm{C}_1$, inferable from Eq. 6, where C_1 is the position of first camera optical center with respect to the second camera coordinate system.

From Eq. 4 the fundamental matrix $\mathbf{F}$ can be calculated as follows:

$$\mathbf{F} = \left[\mathbf{h}_1 - \mathbf{H}_1\mathbf{H}_2^{-1}\mathbf{h}_2\right]_\times \mathbf{H}_1\mathbf{H}_2^{-1}.$$

Consider Eq. 8 above, we have $\mathbf{H}_2 = \mathbf{I}$, and $\mathbf{h}_2 = 0$. Substitute to Eq. 4, we have

$$\mathbf{F} = [\mathbf{h}_1]_\times \mathbf{H}_1,$$

where $[\mathbf{h}_1]_\times$ is a skew matrix of $\mathbf{h}_1$, and $\mathbf{H}_1$ is a 3×3 rotation matrix.

Since epipole e_1 in image plane I_1 is the projection of the second camera optical center onto the image plane I_1, so we can write

$$\widetilde{\mathrm{e}}_1 \simeq \mathbf{P}_1 \widetilde{\mathrm{C}}_2 = [\mathbf{H}_1 \mid \mathbf{h}_1]\, \widetilde{\mathrm{C}}_2 = [\mathbf{H}_1 \mid -\mathbf{H}_1 \mathrm{C}_1]\, \widetilde{\mathrm{C}}_2 = -\mathbf{H}_1 \mathrm{C}_1,$$

where $\simeq$ means "equal" up to a scale factor and we have $\mathrm{C}_2 = (0, 0, 0)$ as world coordinate system origin. And we know $\mathbf{h}_1$ is equal to -$\mathbf{H}_1\mathrm{C}_1$, so $\widetilde{\mathrm{e}}_1 = \mathbf{h}_1$. The epipole e_1 in the image I_1 has the property $\widetilde{\mathrm{e}}_1^T \mathbf{F} = 0$, it is equivalent to $\mathbf{F}^T \widetilde{\mathrm{e}}_1 = 0$. Since $\mathbf{F}$ is singular so multiply both sides by $\mathbf{F}$ and then we have $\mathbf{F}\mathbf{F}^T \widetilde{\mathrm{e}}_1 = 0$. Matrix $\mathbf{F}\mathbf{F}^T$ is symmetric, hence e_1 is the eigenvector of matrix $\mathbf{F}\mathbf{F}^T$ associated to the smallest eigenvalue.

In order to calculate $\mathbf{P}_1$ and $\mathbf{P}_2$ from matrix $\mathbf{F}$ we can first factorize $\mathbf{F}$ into a product of this form $[\widetilde{\mathrm{e}}_1]_\times \mathbf{H}_1$. The factorization is not unique in general, since once we find a $\mathbf{H}_1$ satisfying it, any matrix in this form, $\mathbf{H}_1 + \widetilde{\mathrm{e}}_1 \mathbf{v}^T$, will also be a solution for any 3-vector $\mathbf{v}$. It can be verified easily that we always have $[\widetilde{\mathrm{e}}_1]_\times \widetilde{\mathrm{e}}_1 \mathbf{v}^T = 0$. In particular, $\mathbf{H}_1$ can be obtained by the following equations. Starting from

$$\mathbf{F} = [\widetilde{\mathrm{e}}_1]_\times \mathbf{H}_1,$$

we may multiply the $[\widetilde{\mathrm{e}}_1]_\times$ to both sides of the equation, and then we have

$$[\widetilde{\mathrm{e}}_1]_\times \mathbf{F} = [\widetilde{\mathrm{e}}_1]_\times{}^2 \mathbf{H}_1.$$

Since $\mathbf{v}\mathbf{v}^T = [\mathbf{v}]_\times{}^2 + \| \mathbf{v} \|^2 \mathbf{I}_3$ for any 3-vector $\mathbf{v}$,

$$[\widetilde{\mathrm{e}}_1]_\times \mathbf{F} = (\widetilde{\mathrm{e}}_1 \widetilde{\mathrm{e}}_1^T - \| \widetilde{\mathrm{e}}_1 \|^2 \mathbf{I}_3) \mathbf{H}_1.$$

The right-hand-side of the equation can be expanded, so we have

$$[\widetilde{\mathrm{e}}_1]_\times \mathbf{F} = \widetilde{\mathrm{e}}_1 \widetilde{\mathrm{e}}_1^T \mathbf{H}_1 - \| \widetilde{\mathrm{e}}_1 \|^2 \mathbf{H}_1.$$

Since $[\widetilde{\mathrm{e}}_1]_\times \widetilde{\mathrm{e}}_1 \mathbf{v}^T = 0$ for any 3-vector $\mathbf{v}$,

$$[\widetilde{\mathrm{e}}_1]_\times \mathbf{F} = -\| \widetilde{\mathrm{e}}_1 \|^2 \mathbf{H}_1.$$

It implies

$$\mathbf{H}_1 = (-1/\| \widetilde{\mathrm{e}}_1 \|^2)[\widetilde{\mathrm{e}}_1]_\times \mathbf{F}.$$

Therefore,

$$\mathbf{P}_1 = \left[(-1/\| \widetilde{\mathrm{e}}_1 \|^2)[\widetilde{\mathrm{e}}_1]_\times \mathbf{F} \mid \widetilde{\mathrm{e}}_1\right] \quad \text{and} \quad \mathbf{P}_2 = [\mathbf{I} \mid 0],$$

the relative perspective camera projection matrices, are recovered using the fundamatal matrix F derived from two uncalibrated reference images.

3 Stereo Image Generation by Image Reprojection

Stereoscopic visualization techniques are known to be very active topics in computer vision, virtual reality and augmented reality [DM96, HH97, Vin95, WHK98]. To deliver a stereo form of images to human visual perception, two elements are essential. The first is a source that provides visual stimulation to our visual perception, called stereo images, usually one for each eye, let us call left/right image. The other is a filter, normally a 3-D eyeglass or a specialized screen/monitor, that ensures we perceive the correct stereo image for each eye synchronously with the predefined display frame rate. Our task is to produce such a source so audience can perceive of 3-D binocular depth cues out from a 2-D display media.

In order to give the audience binocular depth cues, normally, each single object view requires two shots with a proper camera displacement to generate a stereo pair [Vin95, WHK98]. This camera model can be referred to as standard binocular stereo model, i.e. with horizontal disparity only. Our goal is to automatically generate such a binocular stereo pair from the object images that basically describe an object from various angles of view toward the rotation center. In this section, we first study the geometry of image reprojection and derive its mapping formula. The automatic stereo generation processes are explained separately for the situations of calibrated input images and uncalibrated input images in Section 3.2 and 3.3.

3.1 Image Reprojection Geometry

Image reprojection has been widely used in many applications, such as reprojection on a cylinder for panoramic visualization [Che95, MB95, WHK98]; on a coarse 3-D architecture model for details recovery processing [DTM95]; on a sphere as environment map in computer graphic rendering [FvDFH90]; or on a quarter sphere acting as windscreen for the flight simulator [Vin95]. The reprojection model we are concerned with is different from those described above. We reproject the image of an actual camera, i.e. one of our reference image, onto another virtual camera image plane, and restrict both cameras sharing the same optical center. Figure 5 shows the geometry of this reprojection model. This section will study the geometric relationship between those two images planes sharing the same camera optical center.

Let us sketch this approach first. We assume the basic element used to describe the 3-D world is the object surface point. Consider one camera's projection as a bundle of rays starting from the camera optical center toward each surface point in the scene. Each ray is associated with some color information of a particular surface point. The image acquired can be described as the intersections of those rays with the camera's image plane, and colors associated with those rays will be assigned onto the image. It is not difficult to think of that the projection rays associated with one camera will not change, in terms of its position, orientation and the colors it associates, from the projection rays of another camera as long as both camera share the same camera optical center and objects in the scene are static.

In the Section 2, we have shown that a different position or orientation of the image plane with respect to the same camera optical center defines a different camera projection matrix. Now assuming image planes I and I' share the common camera optical center at optical center C with respect to the world coordinate

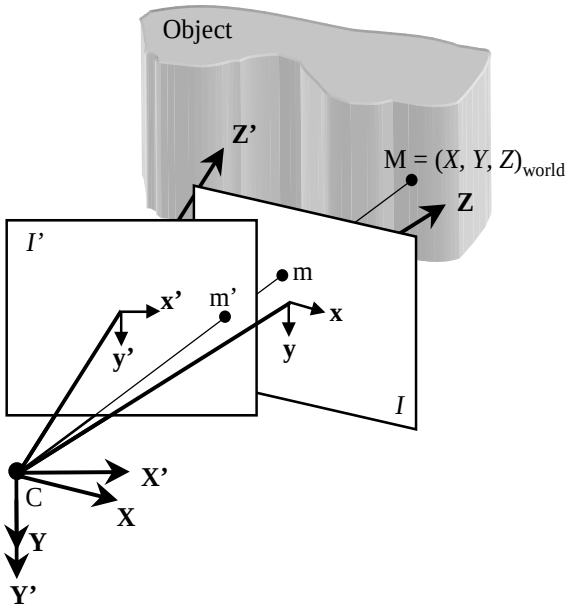

Figure 5: Geometry of perspective planar reprojection model with identical optical centers. The symbols are explained in the text.

system, and $\mathbf{P} = [\mathbf{H} \mid -\mathbf{H}\mathrm{C}]$ and $\mathbf{P}' = [\mathbf{H}' \mid -\mathbf{H}'\mathrm{C}]$ are their perspective projection matrices respectively. The camera projection matrix allows us to calculate the exact intersection position of each projection ray on the image plane. Let us consider a particular projection ray projecting from the camera optical center to a surface point M in the scene, see the Fig. 5, the two intersecting points on image planes I and I', denoted as m and m$'$, can be calculated using $\mathbf{P}$ and $\mathbf{P}'$. Since the image planes I and I' may not be only related by a simple rotation transformation about the common camera optical center. We should be able to establish the mapping between point m in I and point m$'$ in I' directly from the information provided by $\mathbf{P}$ and $\mathbf{P}'$. The formula maps the points on image I to the points on image I' can be derived as following. We know

$$\widetilde{\mathbf{m}} \simeq \mathbf{P}\mathrm{M} = [\mathbf{H} \mid -\mathbf{H}\mathrm{C}]\,\mathrm{M} \quad \text{and} \quad \widetilde{\mathrm{m}}' \simeq \mathbf{P}'\mathrm{M} = \left[\mathbf{H}' \mid -\mathbf{H}'\mathrm{C}\right]\mathrm{M},$$

where $\simeq$ means "equal" up to a scale factor. So,

$$\begin{aligned}
\widetilde{\mathbf{m}} &\simeq \mathbf{P}\mathrm{M} \\
&= [\mathbf{H} \mid -\mathbf{H}\mathrm{C}]\,\mathrm{M} \\
&= \mathbf{H}\,[\mathbf{I}_{3\times 3} \mid -\mathrm{C}]\,\mathrm{M} \\
&= \mathbf{H}\left[\mathbf{H}'^{-1}\mathbf{H}' \mid -\mathbf{H}'^{-1}\mathbf{H}'\mathrm{C}\right]\mathrm{M} \\
&= \mathbf{H}\mathbf{H}'^{-1}\left[\mathbf{H}' \mid -\mathbf{H}'\mathrm{C}\right]\mathrm{M} \\
&= \mathbf{H}\mathbf{H}'^{-1}\mathbf{P}'\mathrm{M} \\
&\simeq \mathbf{H}\mathbf{H}'^{-1}\widetilde{\mathrm{m}}'.
\end{aligned}$$

Therefore the relationship between two corresponding points m in I and m$'$ in I' can be described by a 3×3 linear transformation matrix $\mathbf{H}\mathbf{H}'^{-1}$. Note that it is in fact a one-to-one mapping between any

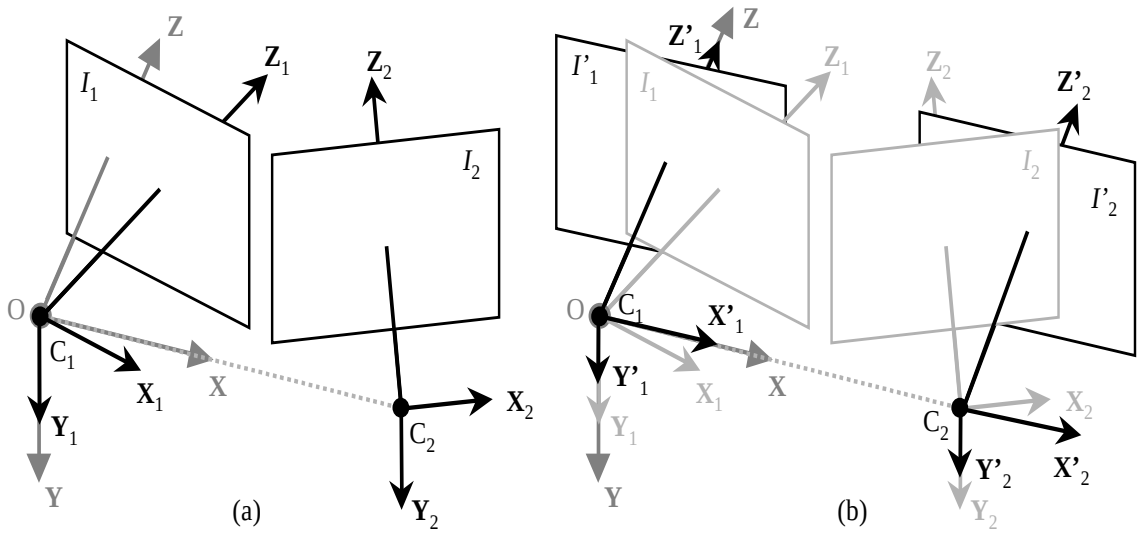

Figure 6: (a) Two reference image planes with respect to the world coordinate system. (b) Two reprojected image planes (i.e. coplanar) with respect to their original image planes. The symbols are explained in the text.

pairs of corresponding points in I and I'. It is quite useful, especially in the sense of simplification of the implementation, to use this simple linear transformation formula to map the corresponding points in different image coordinate systems. We summarize it as follows:

$$\widetilde{\mathbf{m}} = \mathbf{H}\mathbf{H}'^{-1}\widetilde{\mathrm{m}}', \tag{9}$$

where m and m$'$ are any pair of corresponding points in image planes I and I' respectively, and $\mathbf{H}\mathbf{H}'^{-1}$ is a 3×3 linear transformation matrix.

While an image plane is transformed with respect to the same camera optical center, it is equivalent to changing our viewing direction to the scene in which both shape structure and color information of every surface point remain unchanged. Therefore such image reprojection method guarantees the generated views are physically correct, i.e. the 3-D structure and surface reflectance property of the scene are preserved.

3.2 Reprojection with Fully Calibrated Images

We have shown that the formula in Eq. 9 allows us to modify the camera viewing direction to the scene. This result is useful to our stereo generation process, because we can apply it to change the camera viewing directions to the object without loss of validity. There are two possible cases. One is for the fully calibrated input images in which all the intrinsic and extrinsic parameters of the camera projection matrices are known. The other is for the uncalibrated input images in which the method discussed in the Section 2 to recover their relative camera projection matrices is committing prior. In this section we focus on the first case, the second case will be dealt with in the next section.

To generate a stereo pair from two reference images, first we normalize two input images by the inverse intrinsic matrix $(\mathbf{EA})^{-1}$ in which all the camera intrinsic parameters are assumed to be well calibrated. Then we calculate the camera projection matrices that only contain extrinsic parameters for both images under the assumption that the world coordinate system origin coincides with the I_1 image's optical center, and its Z-axis is perpendicular to the line joining two camera's optical centers, as well as its Y-axis is parallel to both image planes. Figure 6(a) shows the idea. So we have

$$\mathbf{P}_1 = [\mathbf{H}_1 \mid 0] \quad \text{and,} \quad \mathbf{P}_2 = [\mathbf{H}_2 \mid -\mathbf{H}_2 \mathrm{C}_2] \,,$$

where C_2 is the camera optical center for the image plane I_2.

Now we can use the formula in Eq. 9 to reproject images I_1 onto I_1' and I_2 onto I_2'. The projection matrices for the image planes I_1' and I_2' are $\mathbf{P}_1' = [\mathbf{I} \mid 0]$ and $\mathbf{P}_2' = [\mathbf{I} \mid -\mathrm{C}_2]$ respectively, so the image planes I_1' and I_2' are coplanar as shown in Fig. 6(b). It is clear that the image planes I_1' and I_2' are the standard binocular stereo image pair in which all the epipolar lines on both images coincide with image rows, thus corresponding points always lie on the same row in the image coordinate system.

This procedure can be repeated for every pair of adjacent object views to generate a sequence of stereo images. For the multiple layers case, each layer can be processed independently. Assuming the camera parameters are constant for each single layer, i.e. the object is rotated in uniform angle with fixed camera pose and settings, then calculating the camera projection matrices only needs to be done once for each layer. If the viewing angle θ between two adjacent object views is too wide, i.e. the overlapping area of the object in both images I_1' and I_2' is small and does not allow that the stereo view can be fused [Vin95, WHK98], then a closer in-between view is desired. We will introduce an image-based approach to generate a novel in-between view without 3-D reconstruction in Section 4.

3.3 Reprojection with Uncalibrated Images

If the input images are uncalibrated, we do not have any information about both intrinsic and extrinsic parameters of the camera. But in order to apply the image reprojection equation, which was shown in the previous subsection, the camera projection matrices of both the reference and desired images must be known. One way to recover the relative camera projection matrices for the uncalibrated input images is to use the method described in the Section 2. Once we have found the relative camera projection matrices for any pair of adjacent object views, the image reprojection method is again used to generate the stereo view. Since the recovered projection matrices for the uncalibrated image case are in different form from the calibrated case shown in the Section 3.2, in this section we show how to use them to produce its associated stereo images.

The recovered projection matrices for two reference images I_1 and I_2 should be in the form $\mathbf{P}_1 = [\mathbf{H}_1 \mid \mathbf{h}_1]$ and $\mathbf{P}_2 = [\mathbf{I} \mid 0]$, where $\mathbf{P}_1$ and $\mathbf{P}_2$ are defined with respect to the world coordinate system which is assumed to coincide with the camera coordinate system of the image plane I_2. In the discussion in the Section 2, we already mentioned that matrix $\mathbf{H}_1$ here is equivalent to a rotation matrix. The situation is shown in Fig. 7(a). A rotation matrix with respect to Y-axis of the world coordinate system is in this

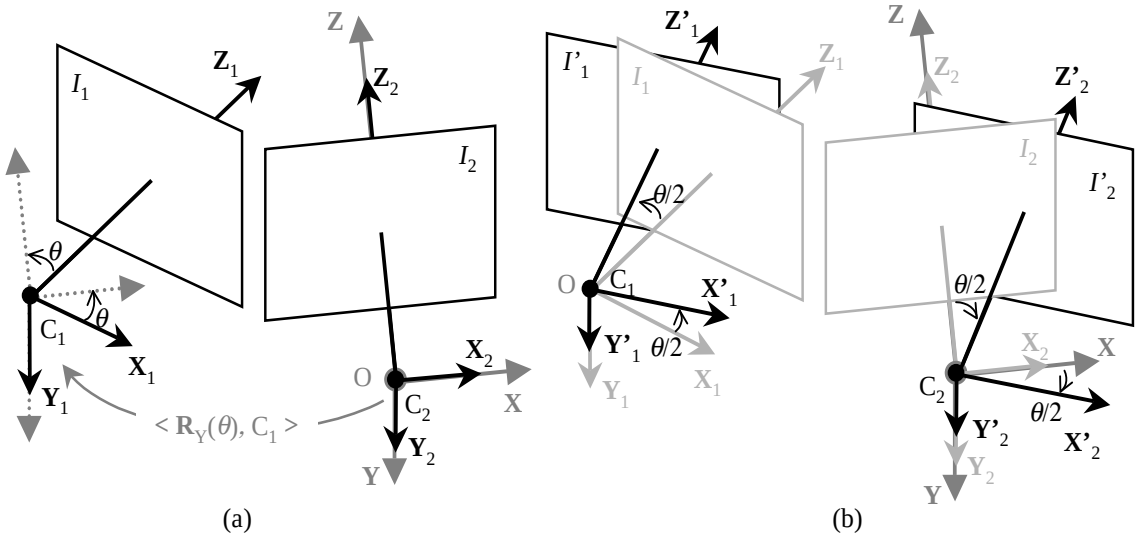

Figure 7: (a) Camera coordinate system associated with image plane I_2 coincides with the world coordinate system. (b) Two reprojected image planes (i.e. coplanar) with respect to their original image planes. The symbols are explained in the text.

form

$$\mathbf{R}_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix},$$

where θ is the rotation angle. So we have

$$\mathbf{H}_1 = \begin{bmatrix} h_{11} & 0 & h_{13} \\ 0 & 1 & 0 \\ h_{31} & 0 & h_{33} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} = \mathbf{R}_Y(\theta),$$

where h_{ij} is an element of the matrix $\mathbf{H}_1$. We can estimate the rotation angle θ by

$$\theta = \left(\cos^{-1} h_{11} + \sin^{-1}(-h_{13}) + \cos^{-1} h_{31} + \sin^{-1} h_{33}\right)/4.$$

Because we know the rotation angle θ, the projection matrices for two image planes I'_1 and I'_2 can be determined as $\mathbf{P}'_1 = [\mathbf{R}_Y(-\theta/2) \mid -\mathbf{R}_Y(\theta/2)\mathrm{C}_1]$ and $\mathbf{P}'_2 = [\mathbf{R}_Y(\theta/2) \mid 0]$ respectively, where C_1 is the camera optical center position of image plane I'_1, as shown in Fig 7(b). Now we can use the formula in Eq. 9 to reproject the images I_1 onto I'_1 and I_2 onto I'_2.

Here we demonstrate an example which can be extended to operate on large object image databases converting them to binocular stereo form automatically. We have input data from an existing QTVR object movie available over the Internet[4], called "Green Horse's Head[5]", from Asian Art Museum of San Francisco[6]. It has a total of 36 views and covers 360 degree of the horse head in the single layer fashion.

[4]http://sfasian.apple.com/Mongolia/Views/Views.htm

[5]GREEN HORSE'S HEAD FOR MAITREYA'S CART, Dulamin Damdinsuren (1868-1938), early 20th century, wood, velvet, metal fittings, paint, and horsehair. During the Maitreya Festival, Maitreya's horse-headed cart (pictured in no. 42) was pushed around the ceremonial circle by monks, who stopped at the cardinal points to chant prayers.

[6]http://www.asianart.org/

Figure 8: Multiple stereo green horse head views. The top row shows the successive left images and the bottom row shows the corresponding right images.

We have no idea about what camera they used and how they setup the image acquisition whatsoever. It is a totally uncalibtated case for those images. Figure 8 shows the result of selected four stereo pairs automatically generated from their corresponding input images.

4 View Synthesis

In the Section 3, we have shown how binocular stereo images can be reprojected from the set of multiple monocular object images. In fact, the given images may not necessarily fully describe the object, such as the angle between two adjacent viewing axes is too wide with respect to the shape complexity of the object. Figure 9 illustrates this situation. The ghost problem may occur if the object in two stereo images has large non-overlapped area. To accommodate this situation, generation of a novel view in between is necessary.

In this section, we first introduce the image morphing technique which guarantees a smooth transition between two source images. We also show the fundamental problem that the invalidity of applying 2-D image morphing on 3-D case. A 3-D image morphing therefore is purposed and used for our in-between view synthesis. To morph two images, the correspondence between two images needs to be established first. We introduce a semi-automatic scheme for the correspondence reconstruction. The result of our approach is presented at last.

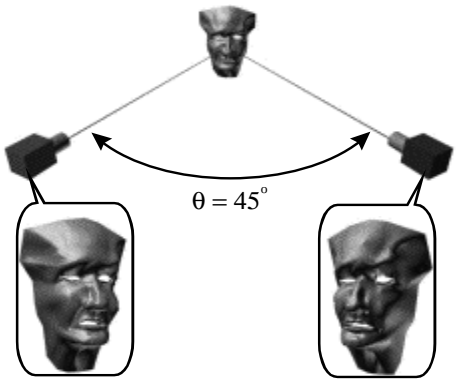

Figure 9: A 45 degree separation of two cameras causes a smaller overlapping region of the facial mask in comparison to a less degree separation. It is difficult to fuse the generated binocular stereo pairs with such separation by human eyes.

4.1 Image Morphing

Image morphing has been adopted to generate in-between images for many 2-D applications, such as cartoons, movies, games, etc. Human animators manually specify the corresponding points and outlines for two source images. Then the in-between images are generated by software based on position interpolation and color dissolving, normally warping linearly. The transition between two source images is allowed to be as smooth as possible with this approach. It works nicely as long as the correspondences between two source images are identified accurately. However, heavy human labor-work and large time-consuming are bottleneck to have accurate correspondence map, especially for complex contents in two source images.

When image morphing apply to the 3-D case, the considerations of computation become even more complicate. Figure 10 shows the problem where the shape of a table is distorted during the morphing procedures although the full corresponding points and outlines are identified correctly. S.M.Seitz has first pointed out this problem in [MD96] . The problem is because two corresponding points move toward each other linearly in 2-D image space, without obeying the 3-D geometric constraint, i.e. not moving along the epipolar line. For instance, Fig 11(a) shows two image planes, a 3-D point M projected onto two image planes , I_1 and I_2, and their intersections, m_1 and m_2, with the epipolar plane. Figure 11(b) depicts those

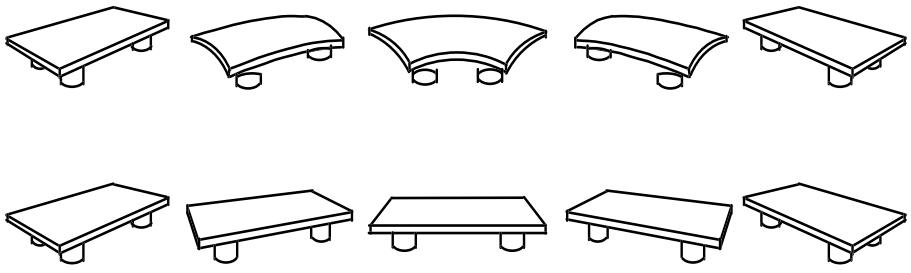

Figure 10: The top row shows the distortion of a table shape during the 2-D image morphing processes. The bottom row shows the expected shape of the table (i.e. visual vaild) with respect to the same camera motion as for the top row.

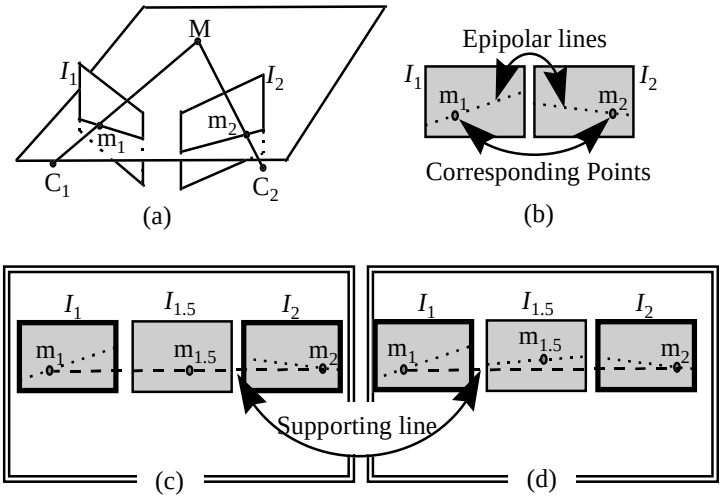

Figure 11: The fundamental problem of the invalidity by applying 2-D image morphing on 3-D geometry case. (a) shows two camera poses, a 3-D point M projected onto two image planes, I_1 and I_2, and their intersections, m_1 and m_2, with the epipolar plane. (b) two images are placed side by side as in 2-D morphing software with the corresponding points and dashed epipolar lines indicated. (c) shows its 2-D morphing result with morphing ratio 0.5. A supporting line there indicates the path connecting two corresponding points in the image space and the 2-D interpolated point $m_{1.5}$ will be laid on. (d) shows 3-D morphing result with ratio 0.5. It illustrates the physically correct position with respect to its 3-D geometric information, i.e. the interpolated point $m_{1.5}$ must lies on the associated epipolar line.

two images placed side by side as in 2-D morphing software with the corresponding points and epipolar lines (i.e. dashed lines) indicated. Figure 11(c) shows its 2-D morphing result with morphing ratio 0.5 while Fig. 11(d) illustrates the its physically correct position in the 2-D image plane with respect to its 3-D geometric information. In general, the 3-D image morphing can be interpreted geometrically, shown

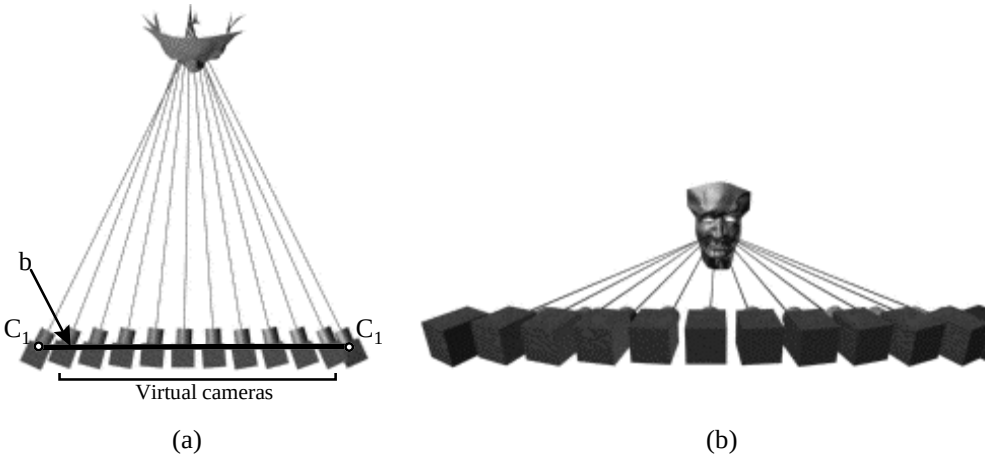

Figure 12: Virtual cameras along the path from one camera position C_1 in 3-D space to the other C_2 along the line b connecting C_1 and C_2. The orientation of the virtual camera is turning from orientation of one camera to the other steadily with respect to the pace of movement. (a) Top view. (b) Front view.

in Fig 12(a), as moving a virtual camera from one camera position C_1 in 3-D space to the other C_2 along the line b connecting C_1 and C_2. And the orientation of the virtual camera is turning from orientation of one camera to the other steadily with respect to the pace of movement. Figure 12(b) shows the front view.

Although in the Section 2 we have pointed out that it is sufficient to search corresponding points only along with mapped epipolar line based on the associated fundamental matrix. It is much more convenient, as we can see, to have all the epipolar lines coincide with the image row precisely. Because the searching operation is then required only on the same row of two reference images as in scan-line fashion. This is not only simplify the implementation complexity, but also take the advantage of less memory cache missing. In the Section 3, we have shown that the image after reprojection is equivalent to the standard binocular stereo image in which the corresponding points lying on the same image row. Therefore, our preprocessing for synthesizing a novel view is to re-project those two reference images as the standard binocular stereo image. Then the corresponding searching along the same image row is operated in the same way as the standard method used in correspondence analysis for binocular stereo.

4.2 Correspondence Reconstruction

Here we present a semi-automated approach for constructing the correspondence between two re-projected images. Instead of fully depending on human intervention as with 2-D morphing software, our program automatically computes the corresponding point for the two re-projected images and comes out with estimated-correspondence highlighted. We also provide a set of parameters in which animators are allowed to fine-tune the coarse estimation without having to reallocate each missing point one by one. Eventually the in-between images can be generated according to the morphing ratio specified.

There are some literature [HS93, KSK98, RK93] discussing about the correspondence analysis for binocular stereo. We propose a variant of cross-validation correlation algorithm with respect to the object images, which is listed in Fig. 13. As described above, animators specify the points and outlines manually to construct the correspondence between two images. The homogeneous regions are left to software to perform the interpolation. The same scenario can be applied here, only edge parts of object are looking for the correspondence. The threshold can be adjusted manually to ensure the edges detected are sufficient to describe the structure of object surface, i.e. outlines of object.

In the algorithm, we treat the silhouette points in the different way from the rest of object's outline points. To search the corresponding silhouette points, the special design of local window for its similarity testing is supplied. The idea is to exclude the background information while comparing the similarities between the two windows. Fig. 14 illustrates this idea in comparison with the standard local window. Notice that the dark gray pixels define the shape of local window associated with current silhouette point, and the correlation coefficient is computed only based on intensities of those dark gray pixels. The shape may vary depending on the shape of object silhouette. Nevertheless once it is defined, all its corresponding local windows along the searching interval in the other reference image is fixed with that defined shape, as F in the Fig. 13.

```
                                        {Searching corresponding points of IMAGEA in IMAGEB}
for each edge pixel p in IMAGEA do
    if  p is silhouette point of object on IMAGEA then
        F(p) = specialized local window shape associated with p in IMAGEA.
    else                {specialized/standard local window are explained in the text}
        F(p) = standard local window shape associated with p in IMAGEA.
    end if
    S = search interval over IMAGEB;
    maxCorr = -1;     { highest correlation coefficient so far}
    matchedP = 0;     { the position of corresponding point for p in IMAGEB}
    for each pixel q within S in IMAGEB do
        corr = Correlation_Similarity_Measure(F(p), F(q));
        if corr > maxCorr then
            maxCorr = corr;
            matchedP = q;
        end if
    end for
                                                    { cross-validation for found matchedP}
    S' = search interval over IMAGEA;
    maxCorr = -1;
    matchedP' = 0;
    for each pixel p' within S' in IMAGEA do
        corr = Correlation_Similarity_Measure(F(matchedP),F(p'));
        if corr > maxCorr then
            maxCorr = corr;
            matchedP' = p';
        end if
    end for
    if matchedP == matchedP' then
        Accept matchedP as the corresponding point of p;
    end if
end for
/*****************************************************************************/
```

$$\text{Correlation\_Similarity\_Measure}\left(E_m, E'_{m'}\right) = \frac{\sum_{i=0}^{n-1}(E_m(i)-\bar{E})(E'_{m'}(i)-\bar{E'})}{\sqrt{\sum_{i=0}^{n-1}(E_m(i)-\bar{E})^2 \sum_{i=0}^{n-1}(E'_{m'}(i)-\bar{E'})^2}},$$

where $E_m(i)$ and $E'_{m'}(i)$ are the intensities of ith pixel in two local windows with respect to the reference point m and m'. The $\bar{E}$ and $\bar{E'}$ are the average intensities for each local window.

Figure 13: A variant of cross-validation correlation algorithm with respect to the object images.

In practice, corresponding edge points found are rather sparse, which will influentially degrade the morphing result. One of the possible improvements is to grow up the found corresponding points along the edges connected. This approach heavily depends on the correctness of corresponding points found by similarity testing, the result may not be very stable. So the adopted cross-validation scheme in Fig. 13

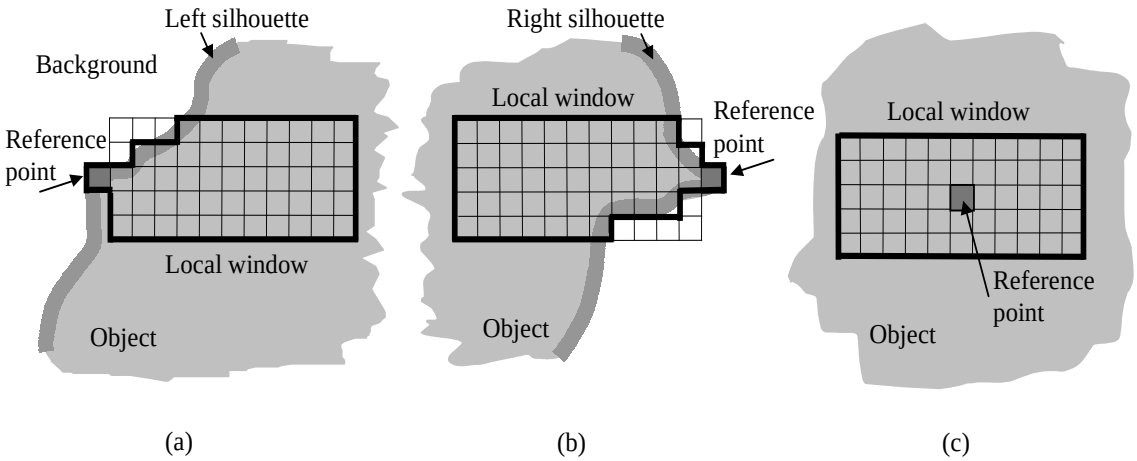

Figure 14: (a) Specialized local window shape for object's left silhouette. (b) Specialized local window shape for object's right silhouette. (c) A standard local window.

to enhance the possibility of the higher correctness is importane. To grow up the found points along the connected edges, two criteria are followed. First, the sequence of corresponding points along the image row are preserved in the same order for both reference images, i.e. correspondence under monotonicity. Second, a edge in one reference image should appear as edge in another image (i.e. a big assumption but practical). The results of correspondence between two reference images before and after the edge grow-up operation are shown in Fig. 15(a) and Fig. 15(b) where the real image data - a smiling Buddha is used. Without loss of validity, the 3-D morphing can also be benefited by operating in the scan-line fashion after two reference images are reprojected. The novel stereo views generated is illustrated in Fig. 16.

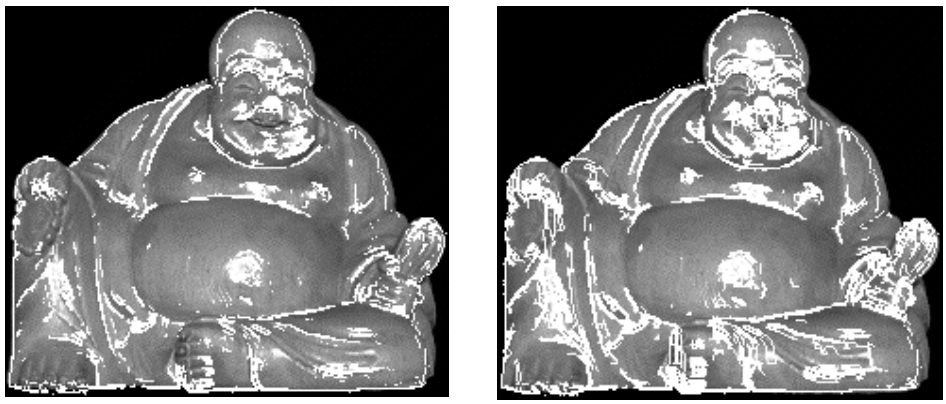

Figure 15: (a) The corresponding edge points found before applying edge growth operation for a smiling Buddha front view. (b) The corresponding edge points found after applying edge growth operation.

Figure 16: The result of synthetic views, $I_{0.4}$ and $I_{0.6}$, are generated from two original Buddha views, $I_{0.0}$ and $I_{1.0}$, with rotation angle 20 degrees.

5 Conclusions

There are few advantages with this approach. First, the object need not be 3-D reconstructed to generate the binocular stereo pair. Second, the generated binocular stereo pair are physically correct with photorealistic quality in comparison with the approch of textured 3-D model reconstruction which heavily depends on the number of meshes. Third, less number of images are required to produce the stereo images, save from the images acquisition time. Four, the camera setup need not be well-posed and the uncalibrated images are allowed, save money from purchasing the special rigs and time from the camera calibration. The last, the stereo imaging generation can be processing automatically in which the large object image databases are potentially able to be converted to the stereo form as long as more than one closely referenced image of the object are available. The future work from this direction can extend this simple canonical planar camera model to other planar camera models or non-planar sorts, such as orthographic, weak projective, paraperspective or fish-eyes, panoramic camera models.

Acknowledgments

Many thanks go to Georgy Gimel'farb for various helpful discussions at the early stages of this work and valuable remarks.

References

[AS98] Shai Avidan and Amnon Shashua.
Threading fundamental matices.
In *ECCV'98*, Frieburg, Germany, June 1998.

[BZM94] P. Beardsley, A. Zisserman, and D. Murray.
Navigation using affine structure from motion.
In *Proc. 3rd European Conf. on Computer Vision*, volume 2, pages 85–96, Stockholm, Sweden, May 1994.

[Che95] Shenchang Eric Chen.
QuickTimeVR - an image-based approach to virtual environment navigation.
In *Proc. SIGGRAPH'95*, pages 29–38, 1995.

[DM96] David Drascic and Paul Milgram.
Perceptual issues in augmented reality.
In *Stereoscopic Displays and Virtual Reality Systems III*, volume 2653, pages 123–134, San Jose, California, USA, Jan. 1996. SPIE.

[DTM95] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik.
Modeling and rendering architecture from phtographs: A hybrid geometry- and image-based approach.
In *Proc. SIGGRAPH'96*, pages 11–20, 1995.

[FvDFH90] James Foley, Andries van Dam, Steven Feiner, and John Hughes.
Computer Graphics Principles and Practice Second Edition.
Addison-Wesley, Reading, Massachusetts, 1990.

[HH97] Ho-Chao Huang and Yi-Ping Hung.
Disparity morphing and automatic generation of stereo panoramas for photo-realistic virtual reality systems.
Technical Report 002, Academia Sinica, Taipei, Taiwan, 1997.

[HS93] Robert M. Haralick and Linda G. Shapiro.
Computer and Robot Vision, volume II.
Addison-Wesley, Reading, Massachusetts, 1993.

[KSK98] Reinhard Klette, Karsten Schlüns, and Andreas Koschan.
Computer Vision - Three-Dimensional Data from Images.
Springer, Singapore, 1998.

[LT94] Q.-T. Luong and T.Vieville.
Canonic representations for the geometries of multiple projective views.
In *Proc. 3rd European Conf. on Computer Vision*, volume 1, pages 589–599, Stockholm, Sweden, May 1994.

[MB95] Leonard McMillan and Gary Bishop.
Plenoptic modeling: An image-based rendering system.
In *Proc. SIGGRAPH'95*, pages 39–46, 1995.

[MD96] Steven M.Seitz and Charles R. Dyer.
View morphing.
In *Proc. SIGGRAPH'96*, pages 21–30, New Orieans, Louisiana, USA, August 1996.

[MD97] Paul Milgram and David Drascic.
Perceptual effects in aligning virtual and real objects in augmented reality displays.
In *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting*, Albuquerque, USA, 1997.

[RK93] Azriel Rosenfeld and Avinash C. Kak.
Digital Picture Processing, volume 2.
Academic Press, London, England, second edition, 1993.

[Vin95] John Vince.
Virtual Reality Systems.
Addison-Wesley, Wokingham, England, 1995.

[WHK98] Shou Kang Wei, Yu Fei Huang, and Reinhard Klette.
Color anaglyphs for panorama visualizations.
Technical Report 19, CITR, Auckland University, New Zealand, Feb. 1998.

[XZ96] Gang Xu and Zhengyou Zhang.
Epipolar Geometry in Stereo, Motion and Object Recognition.
Kluwer, Netherlands, 1996.

[ZDFL94] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong.
A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry.
Technical Report 2273, INRIA, Lucioles, France, May 1994.

[ZX97] Zhengyou Zhang and Gang Xu.
A general expression of the fundamental matrix for both perspective and affine cameras.
In *IJCAI'97*, pages 23–29, Nagoya, Japan, August 1997.