This is to certify that the
dissertation entitled

MULTI-VIEW IMAGE CODING IN 3-D SPACE BASED ON AUTOMATIC 3-D SCENE RECONSTRUCTION

presented by

YONGYING GAO

has been accepted towards fulfillment
of the requirements for the

Ph.D. degree in Department of Electrical and Computer Engineering

Major Professor's Signature

3/29/2005

Date

MSU is an Affirmative Action/Equal Opportunity Institution

MULTI-VIEW IMAGE CODING IN 3-D SPACE BASED ON AUTOMATIC 3-D SCENE RECONSTRUCTION

By

Yongying Gao

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

2005

ABSTRACT

MULTI-VIEW IMAGE CODING IN 3-D SPACE BASED ON AUTOMATIC 3-D SCENE RECONSTRUCTION

By

Yongying Gao

With the rapid development of modern computers and networks, a great deal of research has been focused on the transformation from 2-D visual applications to the 3-D visual world. The need for image coding is naturally magnified when dealing with 3-D applications that employ a large number of highly correlated 2-D images. Recent studies have shown that in multi-view image coding, if the 3-D scene geometry is known, the coding efficiency, decoding speed and rendering visual quality can be dramatically improved. Motivated by this exciting observation and related research problems in existing 3-D geometry based multi-view coding schemes, this dissertation proposes and develops a multi-view coding system that operates directly in the 3-D space based on automatic 3-D scene reconstruction.

Furthermore, this dissertation makes two contributions in the field of automatic 3-D scene reconstruction. First, a new multistage self-calibration algorithm is proposed. We derive a polynomial optimization function of the intrinsic parameters that makes the optimization simple and insensitive to the initialization. Then, based on a stability analysis of the intrinsic parameters, a multistage procedure to refine the self-calibration is proposed. Second, we present a new proof that there are only four possible solutions in

recovering the camera relative motion from the essential matrix. The new proof concentrates on the geometry among the essential-matrix, the camera rotation, and the camera translation. In addition, we provide a generalized SVD-based proof for the four possible solutions in decomposing the essential matrix.

We propose a multi-view image coding system in 3-D space based on automatic 3-D scene reconstruction. We establish a unifying 3-D scene voxel model for all the available image views and then encode the 3-D scene voxel model and the residual data (optional) for compression. There are several advantages of the 3-D voxel model over the mesh model as well as the texture data, which are applied in many existing multi-view image coding systems. First, the 3-D voxel model is much simpler than the mesh model in structure. Second, reconstruction of the original images or generation of synthetic images from the 3-D voxel model is straightforward. It can be achieved by re-projecting the 3-D model back to the image planes; meanwhile image reconstruction from the mesh model requires mapping the texture data to the mesh model. Third, since the 3-D voxel model is an extension from 2-D data to 3-D data, many existing techniques for the image/video coding can be applied for the coding of the 3-D voxel model. Recent examples of these coding techniques include the H.264 video coding standard and 3-D SPIHT coding scheme, both of which we employed in the proposed multi-view image coding system. Experimental results show the potential of our proposed multi-view image coding system in terms of coding efficiency and flexibility for various multimedia applications.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Hayder Radha, for his constant encouragement, advice, support and patience. I thank the other members of my degree committee, Dr. Selin Aviyente, Dr. George C. Stockman, and Dr. Lalita Udpa, for their time and suggestions.

I am also indebted to my colleagues in the Wireless and Video Communications (WAVES) lab. I would like to especially thank Ramin Eslami, Shirish Karande, Ali Khayam, and Kiran Misra for their suggestions and assistance.

My sincere thanks to Margaret Conner, Pauline Van Dyke, Sheryl Hulet, and Roxanne Peacock for their friendly help in departmental matters.

Last but not the least, I would like to thank my family members for their love and support that words cannot describe.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Motivation of the Work

Humans perceive the world through various forms and sources of information, of which approximately 70% is visual. With the rapid development of computers and networks, nowadays people can enjoy not only conventional two-dimensional (2-D) images/videos, but also three-dimensional (3-D) visual scenes. These 3-D visual scenes include both real-world scenes and computer-generated objects.

For many decades, capturing and display of visual data have been confined to 2-D techniques and methods. Moving from 2-D to 3-D visual representation is naturally a challenging task, and hence, current and emerging 3-D visual applications rely (in a significant way) on well-established 2-D visual sources and related methods. In general, there are two major areas of 3-D visual research. Under the first area, research is focused on depicting the 3-D scenes using 2-D representations. For instance, image-based rendering (IBR) techniques can be used to create photo-realistic representations of real-world or computer-generated scenes [1][2][3][4]. Since the visual quality depends on the number of available images, typically hundreds to thousands of images are required to achieve convincingly realistic rendering results. Another application in this area is medical volumetric data generated by computer tomography (CT) or magnetic resonance (MR). Medical volumetric data typically contains many image slices that represent cross sections of a part of human anatomy. In both applications, efficient multi-view image

coding techniques have become a key process in storing or transmitting multiple images over a network. During the past few years, many schemes for multi-view image coding have been proposed specifically for the applications of IBR, such as vector quantization [2], discrete cosine transform (DCT) coding [5], wavelet coding [6][7], predictive image coding [8][9], and approaches that are based on video coding standards [10]. Many techniques have also been developed to compress the medical volumetric data, such as transform coding [11], predictive coding [12] and 3-D wavelet-based compression [13][14][15][16]. In the second 3-D visual research area, research is focused on reconstructing the 3-D scene from the available multiple images. The recovered 3-D scene information can be used in many applications, such as virtual reality. There are two crucial issues in automatic 3-D scene reconstruction. The first aspect is how to obtain the camera parameters, including the camera calibration information [17][18][19] and camera relative motion [20][21][22]. The second aspect is how to establish the 3-D model based on obtained 3-D scene information [23][24].

It is important to emphasize that the need for image coding is naturally magnified when dealing with 3-D applications that employ a large number of highly correlated 2-D images. Furthermore, the two research areas (outlined above) are closely related to each other, especially in the context of multi-view image coding. Recent studies show that in multi-view image coding, if 3-D scene geometry information is given, the coding efficiency, decoding speed and rendering visual quality can be dramatically increased [25][26]. M. Magnor et al. [27] described two different multi-view coding schemes in

which 3-D scene geometry information is employed: texture-based coding and model-aided coding. In texture-based coding [28][25], scene geometry information is used to convert images to view-dependent texture maps prior to compression, while in model-based coding [29], images are successively estimated by using scene geometry information to predict new views from previously encoded images. The prediction residual between the estimated image and the original image is additionally coded.

The existing multi-view coding schemes using 3-D scene information [29][28][25][26][27] do greatly improve the compression ratio as well as the rendering quality, compared with the conventional coding schemes employing only simple extensions of 2-D compression techniques. However, there are still some aspects of these coding schemes that can be improved. First, although some of the existing multi-view coding schemes allow some form of progressive coding, which makes them (to some degree) comparable to scalable coding and networking solutions [30][31][32][33], this progressive coding aspect can be further improved. In particular, in the 3-D geometry-based multi-view coding, the scene geometry information and the image (texture) data must be encoded separately. This character limits the flexibility of the coding scheme, since the decoding of the 3-D geometry information must be completed prior to the decoding of the image (texture) data. Furthermore, it is rather challenging to optimize the rate-distortion (R-D) performance of the coding scheme if we vary the bit rate of the encoded 3-D geometry information as well as the bit rate of the encoded image (texture) data simultaneously. Hence, developing a (truly) scalable scheme that does not

rely on R-D optimization of geometry coding will be desirable. Second, the 3-D geometry coding is computationally complex. Eisert et al. [34] proposed a multi-hypothesis volumetric reconstruction to obtain the 3-D scene geometry. According to this algorithm, the 3-D space containing the considered object is discretized into volume elements (voxels) and then a volumetric model is constructed. To reduce the data volume of the 3-D geometry model, a 3-D mesh model is derived to triangulate the volumetric model surface [35]. A number of progressive mesh-coding algorithms have been proposed focusing on the trade-off between the geometry reconstruction accuracy and the geometry coding bit rate [36][37][38]. From the above discussion we can see that the whole procedure of obtaining 3-D geometry information is complex in computation. Third, the generally used 3-D geometry representations, such as the mesh model used in the existing 3-D geometry-based multi-view coding schemes, are suitable to represent 3-D objects of simple surface but difficult to represent objects of complicated surface, which are often shown in natural scenes.

The existing problems stated above have motivated our studies in further combining the automatic 3-D scene reconstruction into multi-view image coding. We propose a multi-view coding framework that operates directly in 3-D space and is based on automatic 3-D scene reconstruction.

1.2 Main Contributions

This section provides an overview of the main contributions of this dissertation in the field of automatic 3-D scene reconstruction and in the field of multi-view image coding.

1.2.1 A Multistage Camera Self-calibration Algorithm

As mentioned in Section 1.1, determining the camera calibration information is a key step in recovering the 3-D scene information from multiple images. Unlike the classical calibration problem [17] where a calibration jig is used to establish some well-defined 3-D points, a self-calibration algorithm attempts to find the camera intrinsic parameters from a set of images without the ground truth [18][19]. Camera self-calibration makes it possible to realize online automatic 3-D reconstruction.

We have proposed and developed a new camera self-calibration algorithm that uses a low-complexity multistage optimization approach [39]. We derive a polynomial optimization function with respect to the camera intrinsic parameters, based on the equal singular value property of the essential matrix. In terms of the stability analysis of the intrinsic parameters, we propose a multistage procedure to refine the estimation. Experimental results with both synthetic and real images show the accuracy and robustness of our method while maintaining low-complexity.

1.2.2 A New Proof for the Four Solutions in Recovering the Camera Relative Motion from the Essential Matrix

Determining the camera relative motion is another important point in automatic 3-D scene reconstruction. There exist various approaches for recovering camera relative motion (rotation and translation) from the essential matrix, as well as disagreement on the number of possible solutions. We present a new proof that there are only four possible solutions in recovering the camera relative motion in the non-degenerate case (e.g., when the translation vector has a non-zero norm) [40]. Differing from the Singular Value Decomposition (SVD)-based proof, we concentrate our proof on the geometry among the essential matrix, the camera rotation, and the camera translation. Based on our proof, we provide some further insight into the existing methods. In particular, we provide a generalized SVD-based proof for the four possible solutions in decomposing the essential matrix.

1.2.3 Multi-view Image Coding in 3-D Space

Based on existing research achievements in automatic 3-D scene reconstruction, we propose a new multi-view coding scheme that performs the coding in 3-D space. At the encoder, a 3-D scene model is first derived from the available camera calibration information and relative motion information [41] (Chapter 13-4 and 13-8) to represent the object in 3-D space as a volumetric model. The color information of each 3-D space element (in a volumetric model, it is a voxel) on the object surface is extracted from the

available multiple images and then mapped to the corresponding 3-D space voxel. Then a complete 3-D *voxel model* of the object surface is constructed. The next step is to encode the 3-D voxel model in 3-D space. The encoded 3-D data is transmitted to the decoder as well as the camera parameters (including camera calibration and relative motion information, which is trivial in size compared to the encoded 3-D data). At the decoder, the compressed 3-D data is first decoded and then the recovered 3-D model is re-projected back to image planes in terms of the camera parameters to reconstruct the multiple images. To increase the reconstruction quality, the residual between the original images and the reconstructed images from the re-projection of the 3-D model can be obtained and compressed at the encoder and can then be transmitted to the decoder together with the encoded data of 3-D voxel model and the camera parameters. At the decoder, the decoded residual is used to compensate for the reconstructed images.

Reconstructing the 3-D scene/object from the available multiple images is a key step in our proposed multi-view coding scheme. Our approach for the volumetric 3-D reconstruction is similar to Eisert's approach [34]. However, we made two modifications to improve its performance.

The step of encoding the 3-D voxel model is another key issue in our proposed coding scheme and distinguishes our approach from the existing 3-D geometry-based multi-view coding schemes [29][28][25][26][27]. In those coding schemes, 3-D scene geometry information is used only to "support" the coding of the multiple views and is encoded independently. The encoding of the image (texture) data still belongs to the

category of 2-D data compression. On the contrary, in our proposed coding scheme, the 3-D voxel model is used as a complete representation of the available multiple views in 3-D space and we directly encode the 3-D data. In addition to avoiding complex geometry-based 3-D model extraction, the potential advantages of our proposed approach are as follows. First, since the 3-D voxels provide a unifying model that captures all of the available (highly correlated) multiple images, the redundancy among all the images can be dramatically reduced and high compression ratio can be achieved. Second, the techniques employed in image/video coding can be extended to 3-D data coding. For instance, scalable coding can be realized if we employ the 3-D discrete wavelet transform, which was first proposed by Karlsson and Vetterli [42] for video compression, to encode the 3-D model. Furthermore, techniques based on rate-distortion optimization [43] can be integrated into the coding scheme to improve the coding performance. Third, since we encode the obtained 3-D voxels, which form the representation of the 3-D scene, we can generate arbitrary new views after we decode the transmitted 3-D data. This feature is potentially advantageous in the application of IBR, virtual reality, video conferencing system, etc.

The residual coding is an optional procedure in our multi-view coding system. However, it will be required for high-quality reconstruction of original images in many applications, in which the quality of reconstructed images from re-projection of the 3-D scene model does not meet the specific requirement. The possibility of compressing the residual data is that there are still some correlations among the residual images, since one

voxel that contains incorrect color information will lead to correlative errors among all the considered images.

1.3 Outline of the Dissertation

The remainder of this dissertation is organized as follows. In Chapter 2, background and related work are provided. We first introduce theories and methods in the field of automatic 3-D reconstruction. Then research work on multi-view image coding and volumetric data compression is discussed. Finally a brief introduction to video coding techniques is provided. In Chapter 3, details about one of our contributions in automatic 3-D scene reconstruction — a new multistage camera self-calibration algorithm — are provided. Chapter 4 discusses another contribution in automatic 3-D scene reconstruction — study on the exact number of possible solutions in recovering camera relative motion from the essential matrix. In Chapter 5 we first propose a multi-view image coding framework in 3-D space based on automatic 3-D scene reconstruction. Then we discuss in detail the crucial functional blocks of the proposed multi-view coding framework: volumetric 3-D reconstruction, 3-D data coding and residual coding. Chapter 6 summarizes this dissertation and discusses future work.

2 Background and Related Work

2.1 Automatic 3-D Scene Reconstruction

This chapter reviews theories and related implementations in the field of 3-D scene reconstruction, including the camera perspective projection model, camera self-calibration, camera motion recovery, and 3-D modeling.

2.1.1 Camera Perspective Projection Model

A widely used camera model is the pinhole model. In this model, the camera performs a perspective projection of a 3-D point $\boldsymbol{M} = [x, y, z]^T$ in a world coordinate system onto a pixel $\boldsymbol{m} = [u, v]^T$ in the retinal image coordinate system. By denoting the homogeneous coordinates of a vector $\boldsymbol{x} = [x1, x2, ...]^T$ as $\tilde{\boldsymbol{x}} = [x1, x2, ..., 1]^T$, the relationship between the 3D point $\boldsymbol{M}$ and its image point $\boldsymbol{m}$ is [44]:

$$s\tilde{\boldsymbol{m}} = \mathbf{P}\tilde{\boldsymbol{M}}, \tag{2-1}$$

where s is an arbitrary scale factor, $\mathbf{P}$ is a 3×4 *perspective projection matrix*, and $\tilde{\boldsymbol{m}}$ is the homogeneous vector representation of the point vector $\boldsymbol{m}$. (The homogeneous coordinates are used so the matrix $\mathbf{P}$ can capture both rotation and translation as discussed further below.) The target of 3-D reconstruction is accomplished by recovering the 3-D coordinates $\boldsymbol{M}$ from a set of 2-D coordinates $\boldsymbol{m}$. Theoretically, according to Equation (2-1), the recovery of $\mathbf{P}$ leads to the recovery of $\boldsymbol{M}$ up to a scale factor.

The matrix $\mathbf{P}$ can be decomposed into two parts: intrinsic and extrinsic parameters shown below:

$$\mathbf{P} = \boldsymbol{K}[\boldsymbol{R} \quad \boldsymbol{t}], \tag{2-2}$$

where $\boldsymbol{K}$ is a 3×3 matrix consisting of the camera *intrinsic parameters*, $\boldsymbol{R}$ is a 3×3 matrix representing the camera *relative rotation*, and $\boldsymbol{t}$ is a 3×1 vector representing the camera *relative translation* (The camera relative rotation and translation represent the camera orientation and position relative to some reference camera.). $\boldsymbol{R}$ and $\boldsymbol{t}$ together are called the camera *extrinsic parameters*.

The most general camera *calibration matrix* $\boldsymbol{K}$ can be expressed as:

$$\boldsymbol{K} = \begin{bmatrix} fk_u & -fk_u\cot(\theta) & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2-3}$$

where

- f is the focal length of the camera in millimeters,
- k_u and k_v are the vertical and horizontal scale factors respectively, whose values are the number of pixels per millimeter,
- u_0 and v_0 are the coordinates of the principal point of the camera, i.e., the intersection between the optical axis and the image plane, and
- θ is the angle between the retinal axes. In practice, it is very close to $\pi/2$.

Letting α_u and α_v be fk_u and fk_v respectively and assuming θ to be $\pi/2$ (this assumption is quite reasonable because of the current state-of-the-art), we can rewrite the calibration matrix in a much simpler form as:

$$K = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2-4}$$

2.1.2 Epipolar Geometry and Fundamental Matrix

In the case of two cameras looking at the same scene, the epipolar geometry is the basic constraint that arises from the existence of two viewpoints. We consider two images taken by perspective projection from two different locations, as shown in Figure 2-1.

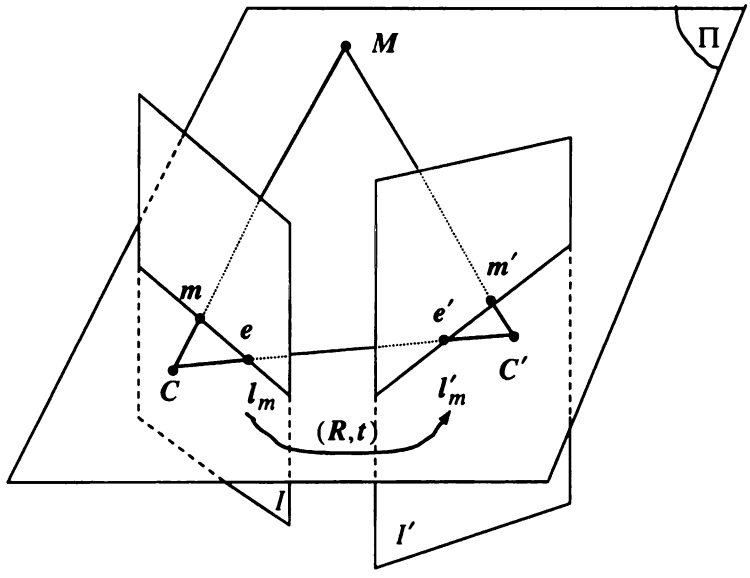


Figure 2-1 The epipolar geometry. C and C' represent the optical centers of the first and the second cameras, respectively. Given a point m in the first image, its corresponding point in the second image is constrained to lie on a line called epipolar line of m, denoted by l'_m. The line l'_m is the intersection of the plane Π, defined by m, C and C', with the second image plane I'. Furthermore, all the epipolar lines of the points in the first image pass through a common point e', which is the intersection of the line CC' with the image plane I'. Point e' is called an epipole.

We show the mathematical expression of the epipolar geometry — the fundamental matrix. The point $\boldsymbol{m}$ in the first image and its epipolar line $\boldsymbol{l}'_m$ in the second image is related through the fundamental matrix $\boldsymbol{F}$:

$$\boldsymbol{l}'_m = \boldsymbol{F}\tilde{\boldsymbol{m}}. \tag{2-5}$$

Since by definition the point $\boldsymbol{m}'$ (in image I') corresponding to point $\boldsymbol{m}$ (in image I) belongs to the epipolar line $\boldsymbol{l}'_m$, it holds that:

$$\tilde{\boldsymbol{m}}'^{T}\boldsymbol{F}\tilde{\boldsymbol{m}} = 0. \tag{2-6}$$

Two important properties of the fundamental matrix $\boldsymbol{F}$ are listed below:

- The fundamental matrix is of rank 2;
- Let $\boldsymbol{e}$ and $\boldsymbol{e}'$ denote the epipoles in the first and second image, respectively, then the following equations are true:

$$\boldsymbol{F}\boldsymbol{e} = \boldsymbol{F}^{T}\boldsymbol{e}' = 0. \tag{2-7}$$

More detailed discussion about the epipolar geometry can be found in [44][45][46].

2.1.3 Camera Self-calibration

Obtaining the camera intrinsic parameters introduced in Section 2.1.1 is crucial in 3-D scene reconstruction. The camera intrinsic parameters are determined only by the camera's specifications and have nothing to do with the camera motion. The term *camera calibration* refers to the recovery of the vertical and horizontal focal length α_u and α_v, and the principal point $[u_0, v_0]^T$ in the image plane. A widely used camera calibration method was proposed by Tsai [17]. The calibration procedure requires relating the

locations of pixels in a set of images taken by a given camera to the locations of the points in the 3-D scene being imaged. A calibration jig is usually required and the calibration procedure has to be done each time the camera intrinsic parameters are changed. Hence, the idea of camera self-calibration was motivated by the fact that in many applications either on-line calibration is needed or the true locations of the points in the 3-D scene (*ground truth*) are not available. Consequently, a great deal of work has been done on camera self-calibration [18][47][19][48][49][39].

Camera Self-calibration Based on Kruppa's Equations. In the case of two or more cameras looking at the same scene, the rigidity constraint exists along with the camera displacement. Kruppa's equations link the epipolar transformation to the image ω of the absolute conic Ω (which is a particular conic at the plane of infinity expressed by $x^2 + y^2 + z^2 = 0$). The absolute conic Ω is invariant under rigid motions and under uniform changes of scale. The invariance of Ω under rigid motions ensures that ω is independent of the pose and position of the camera. In another word, the conic ω depends only on the camera intrinsic parameters. The inverse is also true in that ω determines the intrinsic parameters. Therefore, in theory, Kruppa's equations make camera self-calibration possible. Faugeras et al. [18] proposed a theory of self-calibration expressed by the Kruppa's equations and a numerical method based on the Kruppa's equations. However, approaches based on Kruppa's equations for self-calibration are computationally difficult and not robust to noise. (In the case of two cameras, the absolute conic Ω corresponds to the real world point M in Figure 2-1, while its image

ω^1 (on the first image plane) and ω^2 (on the second image plane) correspond to the image point $\boldsymbol{m}$ and $\boldsymbol{m}'$ in Figure 2-1, respectively.)

Camera Self-calibration Based on Modulus Constraints. Another approach for camera self-calibration is based on the so-called modulus constraints, first proposed by Pollefeys [19]. The modulus constraints are closely associated with the plane at infinity. The homography of the plane at infinity, called infinity homography, can be written as functions of projective entities and the position of the plane at infinity in the specific projective reference frame. The modulus constraint shows that the three eigenvalues of the infinity homography must have the same moduli. Because the constraint that the plane at infinity be the same over the entire image sequence is enforced, this method is superior to the one that is based on Kruppa's equations. On the other hand, requirement for a consistent projective frame over all the views and the computational complexity of modulus constraints make this method difficult in practice.

Camera Self-calibration Based on the Equal Singular Value (ESV) Property of the Essential Matrix. Recently some researchers show interest in applying the ESV property of the so-called essential matrix in camera self-calibration [47][48][49] (details about essential matrix will be given in Section 2.1.4.). The ESV property establishes a link between the motion and the intrinsic parameters of the associated pair of cameras, in the same sense that the Kruppa's equations do. In practice, camera self-calibration can be formulated as a problem of optimization with the objective function derived from the

ESV property of the essential matrix. This optimization based approach represents one of the contributions of this dissertation as discussed further below.

2.1.4 Camera Relative Motion Estimation

Recovering the camera relative motion is another crucial aspect of 3-D scene reconstruction. As stated in Section 2.1.1, theoretically the 3-D structure can be recovered up to a scale factor once we obtain both the camera intrinsic and extrinsic parameters. In fact, once the camera intrinsic parameters are estimated, the problem of estimating camera motion can be solved using a well-established classical approach [20][21][22].

In [20], the camera motion estimation is based on the *essential matrix*, which has been mentioned in Section 2.1.3. Given the camera intrinsic parameters, the essential matrix is a 3×3 matrix representing the epipolar geometry between two images taken from two different viewpoints. The essential matrix $\boldsymbol{E}$ is related to the fundamental matrix $\boldsymbol{F}$ by:

$$\boldsymbol{E} = \boldsymbol{K}'^{T}\boldsymbol{F}\boldsymbol{K}, \tag{2-8}$$

where $\boldsymbol{K}$ and $\boldsymbol{K}'$ represent the calibration matrix of the first and second camera, respectively.

On the other hand, assuming that the camera undergoes a 3-D movement represented by the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$, both of which are of the same meaning as those in Equation (2-2), we can directly relate the essential matrix to $\boldsymbol{R}$ and $\boldsymbol{t}$:

$$\boldsymbol{E} = \boldsymbol{TR}, \tag{2-9}$$

where $\boldsymbol{T}$ is a 3×3 matrix defined by $\boldsymbol{t}$ such that $\boldsymbol{Tx} = \boldsymbol{t} \wedge \boldsymbol{x}$ for any 3-D vector $\boldsymbol{x}$ ($\wedge$ denotes the cross product). Given $\boldsymbol{t} = [t_x, t_y, t_z]^T$, $\boldsymbol{T}$ can be represented as follows:

$$\boldsymbol{T} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}. \tag{2-10}$$

According to Tsai and Huang [21] 's method, the camera extrinsic parameters, $\boldsymbol{R}$ and $\boldsymbol{t}$, can be determined by the singular value decomposition of the essential matrix, up to a scale factor for the translation parameters.

2.1.5 3-D Modeling

In this section, we discuss the problem of 3-D modeling, given multiple calibrated images (Here "calibrated images" refer to images with known camera intrinsic and extrinsic parameters). Theoretically, once image pixel correspondences are set up over at least two images and the camera calibration information is available, the coordinates of their corresponding 3-D point can be recovered up to a scale factor, according to Equation (2-1). However, this method cannot be employed directly in practice. The main reason is that by the method of matrix inversion, we can recover only the position of discrete points in 3-D space, but we need to recover a smooth 3-D surface model of the considered 3-D object/scene in most applications. In other words, the concept "3-D modeling" refers to the derivation of one or more 3-D surface models of the object(s) in

the scene. Hence, it is a rather involved process that is more complex than a simple matrix inversion. In this section, we introduce two different 3-D modeling approaches.

3-D Mesh Model. The 3-D mesh model is mainly used in the field of computer graphics for visualization purposes. In this approach, the 3-D object surface model is established as follows:

1) *Dense depth map* computation based on linking two or more image views to obtain the set of depth information (in 3-D) of the points on the object's surface in the 3-D scene.

2) Mesh triangulation to obtain a triangular wire-frame mesh to reduce geometry complexity and to tailor the model to the requirements of computer graphics visualization systems.

3) Texture mapping onto the wire-frame model to enhance the realism of the model.

Multi-Hypothesis Volumetric 3-D Reconstruction. Differing from the 3-D modeling approach based on mesh models, Eisert [34] presented a volumetric 3-D reconstruction method that is based on a tessellation of the 3-D space by voxels. Neither image point correspondences nor explicit 3-D surface description are needed. The calibrated images are directly used as input to the 3-D reconstruction algorithm and a volumetric 3-D model is obtained. This approach proceeds in four steps:

1) Volume initialization;

2) Color hypothesis generation for all voxels from all available camera views;

3) Consistency check and hypothesis elimination considering all the views;

4) Determination of the best color hypothesis for the remaining surface voxels.

2.1.6 Automatic 3-D Reconstruction from Multiple Uncalibrated Images

This section discusses the general case of 3-D scene reconstruction: the available images are uncalibrated. There exist a number of methods for 3-D reconstruction from multiple uncalibrated images.

Eisert et al. [24] proposed a system for the automatic reconstruction of real-world objects from multiple uncalibrated views. The system proceeds in five steps.

1) *Camera Calibration*: A reference 3-D object is used to calibrate the imaging geometry of the camera.

2) *Camera Motion Estimation from Two Views*: Using the first two views of the scene, the relative motion of the camera can be estimated under the assumption of rigid body motion.

3) *Structure from Stereo*: Given the camera relative motion from the first view to the second one, a dense map of depth values can be computed.

4) *Model-based Shape and Camera Motion Estimation*: The depth map resulting from the previous step is used to adapt a generic 3-D shape model to the object.

5) *Volumetric Reconstruction*: Since all the camera parameters are available now, we discard the approximate geometry obtained from the previous step and perform a

volumetric reconstruction (Readers are referred to Section 2.1.5 for details.) that leads to a set of object voxels with associated color information.

Another system of automatic 3-D reconstruction was proposed by M. Pollefeys, et al. in [23]. The system consists of four successive steps:

1) *Projective Reconstruction* to construct the projective 3-D model;

2) *Self-calibration* to upgrade the projective 3-D model to metric 3-D model;

3) *Dense Matching* to obtain the dense depth maps;

4) *3-D Modeling* to construct the textured metric 3-D surface model.

2.1.7 A Proposed Generic Framework for Automatic 3-D Scene Reconstruction

Here we present a general framework of automatic 3-D reconstruction as shown in Figure 2-2. This framework is based on the camera perspective projection model and has been adopted as a part of our proposed framework of multi-view coding, which will be discussed later.

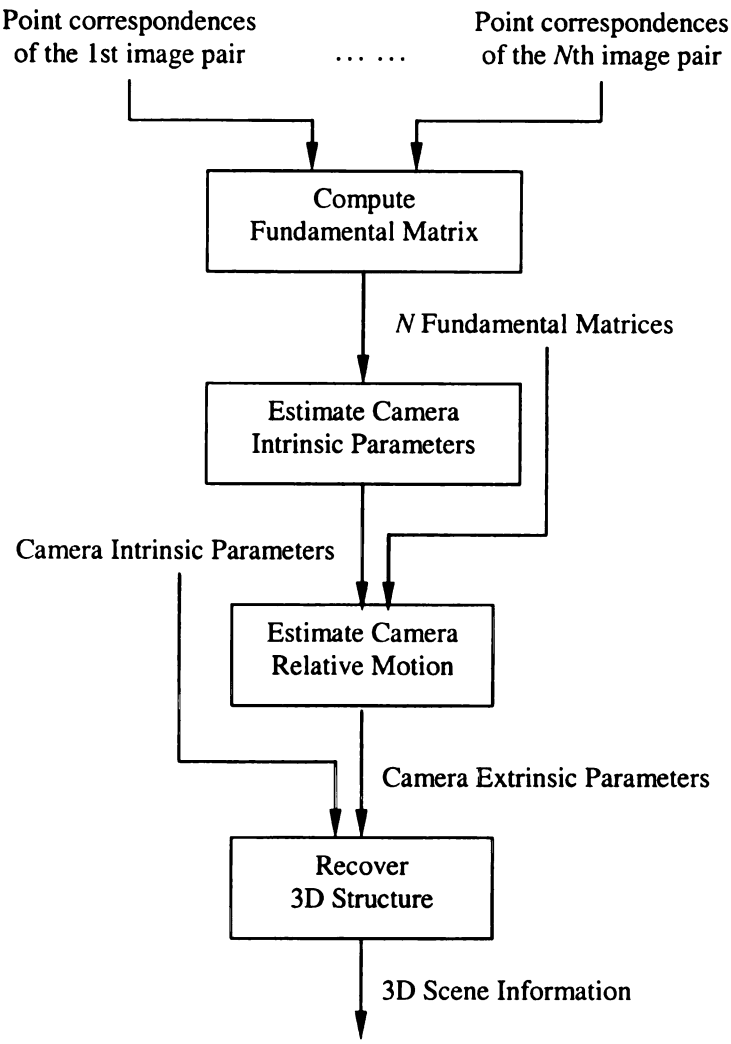


Figure 2-2 A generic framework of automatic 3-D scene reconstruction based on the camera perspective projection model.

In Figure 2-2, the fundamental matrix is first computed for each camera displacement. The input is point correspondences of N image pairs. The output is N fundamental matrices, which are fed into a camera self-calibration function block to estimate the camera intrinsic parameters. Once the fundamental matrices and the calibration matrix are available, we can form the essential matrix for each camera displacement. Then

camera motion can be obtained by decomposing the essential matrix into rotation and translation part, with the translation up to a scale factor. The final step of the system is the recovery of the 3D structure. It is based on the camera perspective projective matrix. The final output is recovered 3D scene information.

2.2 Multi-view Image Coding Using 3-D Scene Geometry

This section reviews related work in the field of multi-view image coding using 3-D scene geometry. Recent studies in the multi-view image coding area show that, if 3-D scene geometry information is given, the coding efficiency, decoding speed and rendering visual quality can be dramatically increased [25][26]. M. Magnor et al. [27] described two different multi-view coding schemes in which 3-D scene geometry information is employed: *texture-based coding* and *model-aided coding*.

2.2.1 3-D Geometry Reconstruction and Geometry Coding

In the existing geometry-based multi-view coding schemes, scene geometry must be coded in addition to image data. The *multi-hypothesis volumetric reconstruction* [34] (Readers are referred to Section 2.1.5 for details.) is first employed to obtain a voxel model for all the available image views. Then the m*arching cubes* algorithm [35] is used to triangulate the voxel model surface, for the purpose of reducing the large number of voxels in the volumetric model. Furthermore, the *progressive meshes* algorithm [36] is used to reduce the number of triangles until the maximum distortion of the resulting mesh

corresponds to half the size of a voxel. While high geometry accuracy can increase the coding efficiency of the texture data, it inevitably increases the geometry coding bit rate. To determine the point of best overall coding performance, the geometry model must be enclosable at different levels of accuracy and with correspondingly different bit rates. A number of progressive mesh-coding algorithms [37][38] aim at trading off the geometry reconstruction accuracy versus the geometry coding bit rate. In particular, differing from most algorithms which encode only mesh connectivity in a progressive fashion, the *embedded mesh coding* (EMC) [38] algorithm progressively encodes mesh connectivity as well as vertex coordinates simultaneously. By using EMC in conjunction with multi-view coding schemes, geometry coding bit rate can be continuously varied.

2.2.2 Texture-based Coding

Texture-based coding is inspired by the view-dependent texture mapping techniques [25] developed in IBR research. The advantage of view-dependent texture mapping is that by transforming images of a 3-D object into texture maps, disparity-induced differences among the images can be eliminated. In *progressive texture-based coding* (PTC), reconstructed 3-D scene/object geometry is used to convert images to view-dependent texture maps. After undefined regions in the texture maps are suitably interpolated, a wavelet coding scheme is employed to encode the texture information while simultaneously exploiting texture correlation within as well as between texture maps. The progressive coding technique continuously increases attainable reconstruction

quality with available bit rate. Several important aspects concerning the PTC are discussed below.

Geometry Model Generation for Texture Mapping. To obtain a planar (2-D) texture map from the closed surface of a volumetric object model, the object surface must be cut once or several times. For objects of *genus* 0, which are topologically equivalent to a sphere, a simple rectangular surface parameterization can be obtained by starting from the shape of an octahedron.

Texture Map Optimization. In the texture plane, each triangle corresponds to one geometry triangle. However, the texture-map triangles are identical while the geometry triangles differ in size and shape. To minimize the coinciding pixel mappings, relative texture-map triangle size is matched to their corresponding geometry triangle area by iteratively shifting texture-map vertex positions [50].

Sparse Texture Map Interpolation to Fit the 4-D Wavelet-based Coding. To circumvent aliasing artifacts due to different resolutions in the image and texture domain and to guarantee exact reconstruction, the texture domain is chosen significantly larger than the pixel area covered by the object in the images. Since many more texels are available than object pixels in the images, each texture map is only sparsely filled. The obtained texture maps exhibit statistical properties very similar to conventional images. In addition, texels at the same coordinates in different texture maps display high correlations as they correspond to the same object surface point. Therefore, the 4-D structure of texture-maps allows exploiting intra-map as well as inter-map similarities by

applying the 1-D wavelet kernel separately along all four dimensions. The resulting hierarchical octave subbands constitute a joint multiresolution representation of all texture maps. To compress the wavelet coefficients array, the *set partitioning in hierarchical tree* (SPIHT) codec [51] is modified to be applicable to the 4-D coefficient field [52]. The undefined texel values in texture-maps represent a large number of degrees of freedom that can be exploited to keep the bitstream size to a minimum by matching the texture interpolation to the 4-D wavelet-based SPIHT codec that follows. Because the 4-D wavelet-based SPIHT coding method performs best if high-frequency coefficients are small relative to low-frequency coefficients, undefined texels must be interpolated subject to the constraint that the applied wavelet transform results in minimal high-frequency coefficient values and maximal low-frequency coefficients.

2.2.3 Model-aided Coding

Differing from the texture-based coding, the model-aided coding successively predicts image appearance by disparity compensation and occlusion detection on a pixel basis. The images are hierarchically ordered for encoding, yielding a multiresolution representation of the multi-view set during decoding and rendering.

Model-aided Prediction. In model-aided coding, an image is predicted by warping multiple reference images [53]. First, the geometry model is rendered for the image viewpoint that is to be predicted. The geometry model is then rendered for all reference images. For each pixel in the predicted image, the corresponding pixels in the reference

images are sought using the pixel's triangle index and its barycentric coordinates. (For any point K inside a triangle ΔABC, there exists three masses w_A, w_B and w_C such that, if placed at the corresponding vertices of the triangle, their center of gravity (*barycenter*) will coincide with the point K. w_A, w_B and w_C are defined as the *barycentric* coordinates of K.) Because multiple reference images are used for prediction, the number of completely invisible regions is small. These regions are filled by interpolation using a multiresolution pyramid of the predicted images estimate.

2.2.4 Comparison between texture-based and model-aided coding

Magnor et al. [27] also provided a comparison between the texture-based and model-aided coding, from both the experimental results and theoretical analysis. In brief, the texture-based coding scheme is observed to yield superior compression results if exact 3-D scene geometry is available. For approximate 3-D geometry, however, the model-aided predictive coding achieves better compression performance.

2.3 3-D Wavelet-based Compression of Volumetric Data

Many techniques have been developed to compress the volumetric data, such as transform coding [11], predictive coding [12] and 3-D wavelet-based compression [13], 3-D *embedded zerotree wavelet* (EZW) coding [14][15], and 3-D integer wavelet-based compression [16]. The approaches in [14][15][16] show better compression performance than [11][12][13]. One main reason is that these approaches exploit the correlation

between neighboring image slices. Furthermore, by applying an integer wavelet transform in the 3-D based coding scheme, we can generate a single embedded bitstream that allows progressive lossy-to-lossless compression. This section reviews related techniques in 3-D wavelet-based compression of medical volumetric data.

2.3.1 3-D Wavelet Transform

Karlsson and Vetterli [42] first proposed a 3-D sub-band coding scheme for video. In their method, the video signal is filtered and sub-sampled in all three dimensions (temporal, horizontal and vertical) to yield the sub-bands. The sub-bands can be much more efficiently encoded than the input original signal in terms of compression ratio and reconstruction quality. This technique was later developed to spatial-temporal wavelet transform, which is usually called 3-D wavelet transform. (Strictly speaking, it might be more appropriate to refer to this technique as "2-D + T".) In the case of medical volumetric data compression, 2-D wavelet transform is first applied on the image slices and then an appropriate 1-D wavelet transform is applied on the third dimension.

We consider the wavelet transform that is directly extended from the 2-D plane (x-direction and y-direction) to the 3-D space (x-direction, y-direction and z-direction). Separable wavelet kernel is used in decomposing the 3-D signal. At each level of decomposition, eight sub-bands are generated for the considered cube, as shown in Figure 2-3.

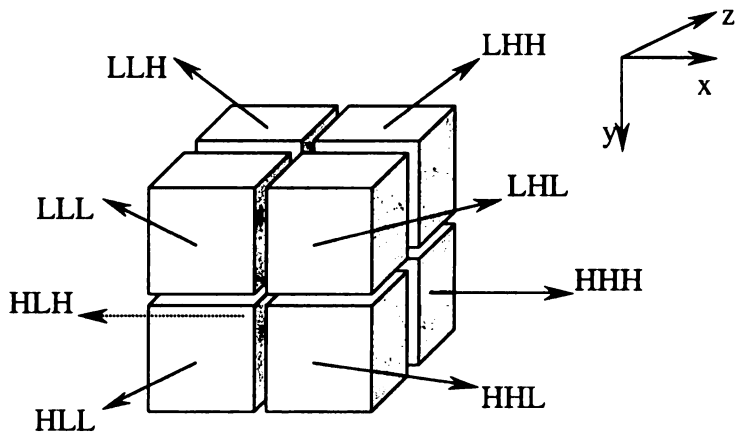


Figure 2-3 Wavelet transform in 3-D space. At each level of decomposition, the considered cube is decomposed into eight sub-bands (for the convenience of display, the eight sub-bands are drawn separated to each other), which are generated by low-pass or high pass filtering along the three dimensions.

2.3.2 Integer Wavelet Transform

Invertible wavelet transforms that map integers to integers have important applications in lossless coding. Traditionally, integer wavelet transforms are not easy to construct. However, the construction becomes very simple with lifting [54]. The lifting scheme can be described in general from the transform point of views, as shown in Figure 2-4 and Figure 2-5 [54].

Since every wavelet transform can be realized using lifting, building an integer version of every wavelet transform is simply straightforward: In each lifting or dual lifting step, the result of the filter is rounded off before the adding or subtracting. Below, we provide a brief outline of the mathematical expression of the integer wavelet transform using lifting shown in Figure 2-4 and Figure 2-5.

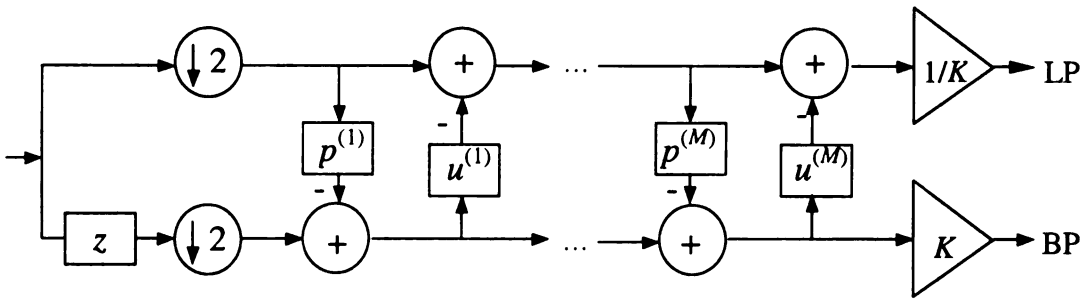


Figure 2-4. The forward wavelet transform using lifting. First the Lazy wavelet transform, then alternating dual lifting steps, and finally a scaling.

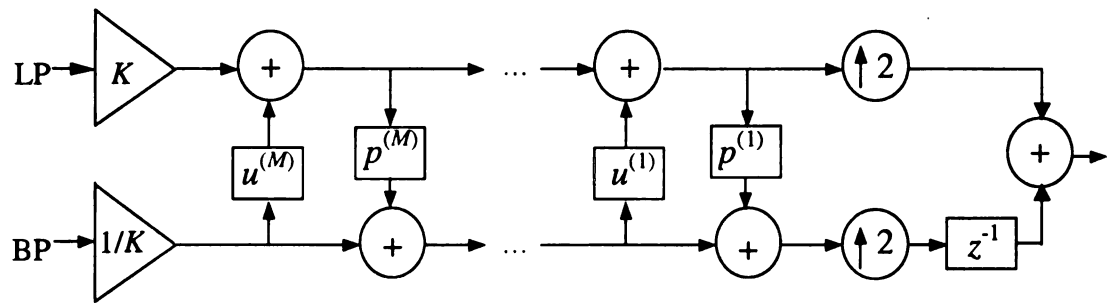


Figure 2-5. The inverse wavelet transform using lifting. First a scaling, then alternating lifting and dual lifting steps, and finally the inverse Lazy wavelet transform. The inverse wavelet transform can be immediately derived from the forward by running the scheme backwards and flipping the signs.

In the forward integer wavelet transform using lifting, the *Lazy* wavelet transform is first computed, which simply splits the signal into its even and odd indexed samples:

$$s_{1,l}^{(0)} = s_{0,2l} \text{ and } d_{1,l}^{(0)} = s_{0,2l+1}, \tag{2-11}$$

where s_0 represents the original signal, and $s_1^{(0)}$ and $d_1^{(0)}$ represent the even and odd indexed samples of s_0, respectively.

A *dual lifting* step consists of applying a filter to the even samples and subtracting the result from the odd ones:

$$d_{1,l}^{(i)} = d_{1,l}^{(i-1)} - \left\lfloor \sum_k p_k^{(i)} s_{1,l-k}^{(i-1)} + 1/2 \right\rfloor, \tag{2-12}$$

where $p_k^{(i)}, k = 1,2,\cdots$ represent the filter coefficients of the i-th step of filtering the even samples.

A *primal lifting* step does the opposite: applying a filter to the odd samples and subtracting the result from the even samples:

$$s_{1,l}^{(i)} = s_{1,l}^{(i-1)} - \left\lfloor \sum_k u_k^{(i)} d_{1,l-k}^{(i-1)} + 1/2 \right\rfloor, \tag{2-13}$$

where $u_k^{(i)}, k = 1,2,\cdots$ represent the filter coefficients of the i-th step of filtering the odd samples.

After M pairs of dual and primal lifting steps, the even samples become the low-pass coefficients while the odd ones become the high-pass coefficients, up to a scaling factor K:

$$s_{1,l} = s_{1,l}^{(M)} / K \quad and \quad d_{1,l} = K d_{1,l}^{(M)}. \tag{2-14}$$

The inverse transform is obtained by reversing the operations and flipping the signs in the forward transform:

$$s_{1,l}^{(M)} = K s_{1,l} \quad and \quad d_{1,l}^{(M)} = d_{1,l} / K. \tag{2-15}$$

$$s_{1,l}^{(i-1)} = s_{1,l}^{(i)} + \left\lfloor \sum_k u_k^{(i)} d_{1,l-k}^{(i-1)} + 1/2 \right\rfloor. \tag{2-16}$$

$$d_{1,l}^{(i-1)} = d_{1,l}^{(i)} + \left\lfloor \sum_k p_k^{(i)} s_{1,l-k}^{(i-1)} + 1/2 \right\rfloor. \tag{2-17}$$

$$s_{0,2l} = s_{1,l}^{(0)} \text{ and } s_{0,2l+1} = d_{1,l}^{(0)}. \tag{2-18}$$

2.3.3 3-D Set Partitioning in Hierarchical Tree (3-D SPIHT)

The well-known SPIHT image coding algorithm was first proposed by Said and Pearlman [51]. The SPIHT algorithm, which builds on the Embedded Zero-tree Wavelets (EZW) algorithm [55], utilizes three basic concepts: 1) searching for sets in spatial-orientation trees in a wavelet transform; 2) partitioning the wavelet transform coefficients in these trees into sets defined by the level of the highest significant bit in a bit-plane representation of their magnitudes; and 3) coding and transmitting bits associated with the highest remaining bit planes first.

Kim et al. [56] applied a 3-D extension of the SPIHT for a low bit rate embedded video coding scheme. The 3-D SPIHT has the following three similar characteristics: 1) partial ordering by magnitude of the 3-D wavelet transformed video with a 3-D *set partitioning* algorithm; 2) ordered bit plane transmission of refinement bits; and 3) exploitation of self-similarity across *spatio-temporal* orientation trees. A typical spatio-temporal orientation tree is shown in Figure 2-6:

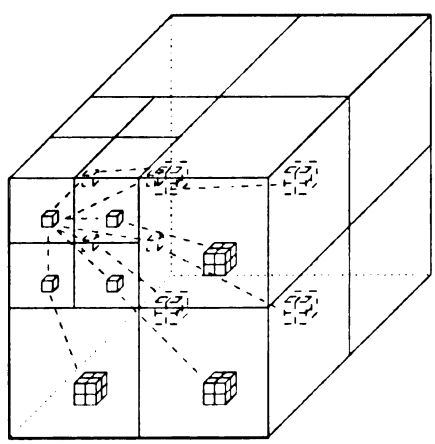


Figure 2-6 3-D spatio-temporal orientation tree. *This figure is adopted from [16]. Courtesy to Xiong, Wu, Cheng and Hua.

2.3.4 Lossy-to-Lossless Compression of Medical Volumetric Data Using 3-D Integer Wavelet Transforms

Xiong et al. [16] proposed a lossy-to-lossless compression scheme for medical volumetric data by 3-D integer wavelet transforms. To achieve good lossy coding performance, it is important to have transforms that are unitary. In [16], a general 3-D integer wavelet packet transform structure was proposed that allows implicit bit shifting of the wavelet coefficients to approximate a 3-D unitary transform. In detail, the same wavelet filters are used along all three dimensions to perform a separable wavelet decomposition. The 2-D spatial transform is accomplished by performing a 2-D dyadic wavelet decomposition on each image slice, while the temporal transform is done by performing a 1-D wavelet packet decomposition along the resulting image slices. Two state-of-the-art wavelet-based video coding techniques, 3-D SPIHT (Readers are referred to Section 2.3.3 for details) and 3-D embedded subband coding with optimal truncation (EBCOT) [57], are modified and applied to compress the obtained wavelet coefficients.

2.4 Digital Video Coding Techniques

In the past two decades, digital image and video coding techniques have developed from an academic research area to a highly commercial business [58]. Modern data compression techniques offer the possibility to store or transmit the vast amount of data necessary to represent digital images and video in an efficient and robust way. New audio-visual applications in the field of communication, multimedia and broadcasting

became possible based on digital video coding technology. This section provides an overview of today's generalized video coding algorithms and video coding standards based on these video coding algorithms.

2.4.1 Video Coding Algorithms

Generally speaking, video sequences contain a significant amount of statistical and subjective redundancy within and between frames. The ultimate goal of video source coding is bit-rate reduction by exploring both statistical and subjective redundancies, and to encode a "minimum set" of information using entropy coding techniques. This usually results in a compression of the coded video data compared to the original source data. The performance of video compression techniques depends on the amount of redundancy contained in the image data as well as on the actual compression techniques used for coding. In practical coding schemes, a trade-off between coding performance (high compression with sufficient quality) and implementation complexity is sought.

Source Model. Video sequences usually contain statistical redundancies in both temporal and spatial directions. The basic statistical property upon which image compression techniques rely is inter-element correlation, including the assumption of simple correlated motion between consecutive frames. Thus, the magnitude of a particular image pixel can be predicted from nearby pixels within the same frame (using Intra-frame coding techniques) or from pixels of a nearby frame (using Inter-frame techniques and motion estimation).

Subsampling and Interpolation. The basic concept of subsampling is to reduce the dimension of the input video (horizontal dimension and/or vertical dimension) and thus the number of pixels to be coded prior to the encoding process. At the receiver the decoded images are interpolated for display. This technique also makes use of specific physiological characteristics of the human eye to remove subjective redundancy contained in the video data. This concept is also used to explore subjective redundancies contained in chrominance data, i.e., the human eye is more sensitive to changes in brightness than to chromaticity changes.

Entropy Coding. The pixel color values of digital video frames are usually pre-quantized to fixed-length words with typically 8 bits or 10 bits accuracy per color component. We can reduce the average number of bits per word if color values having lower probability are assigned longer code words, whereas values having higher probability are assigned shorter code words. This method is entropy coding and forms one of the most basic elements of today's video coding standards, especially in combination with transform domain or predictive coding techniques. If the resulting code words are concatenated to form a stream of binary digits (bits), then correct decoding by a receiver is possible if the code words are uniquely decipherable. Conceptions and principles related to entropy coding are systematically described in information theory [59], which has become one of the kernel theories of modern communications technology.

Predictive Coding. With predictive coding the redundancy in video is determined from the neighboring pixels within frames or between frames. In basic predictive coding systems, an approximate prediction of the pixel to be coded is made from previously coded information that has been transmitted. The difference between the actual pixel and the prediction is usually quantized and entropy coded. This is the well-known *differential pulse code modulation* (DPCM) technique.

Motion Compensation. The concept of motion compensation is based on the estimation of motion between video frames, i.e., a limited number of estimated motion vectors. Since the spatial correlation between motion vectors is often high, it is sometimes assumed that one motion vector is representative for the motion of a "block" of adjacent pixels. To this aim images are usually separated into disjoint blocks of pixels and only one motion vector is estimated and coded for each of these blocks. The motion compensated DPCM technique (used in combination with Transform coding, see next paragraph) has proven to be highly efficient and robust for video data compression and has become a key element for the success of "state-of-the-art" coding standards.

Transform Domain Coding. The purpose of Transform coding is to de-correlate the picture content and to encode transform coefficients rather than the original pixels of the images. The *Discrete Cosine Transform* (DCT) [60]and DCT-based implementations are used in most image and video coding standards due to their high decorrelation performance and the availability of fast DCT algorithms suitable for real time implementations. In recent years, the *Discrete Wavelet Transform* (DWT) [61], which is

well localized in both the space domain and the frequency domain and then leads to multiresolution analysis, has attracted more and more attentions in the field of digital image compression. A major objective of transform coding is to make as many transform coefficients as possible small enough so that they are insignificant (in terms of statistical and subjective measures) and need not be coded for transmission. At the same time it is desirable to minimize statistical dependencies between coefficients with the aim to reduce the amount of bits needed to encode the remaining coefficients.

2.4.2 International Video Coding Standards

We provide a brief introduction to some proposed and commercially successful video coding standards, including H.261, H.263, MPEG-1, MPEG-2, MPEG-4 and on-going H.264/JVT. Generally speaking, all these video coding standards propose a motion-compensated hybrid coding scheme.

H.261: A Video Coding Standard for ISDN Picture Phones and for Video Conferencing Systems. The CCITT (reformed to ITU-T in 1992) "Specialist Group on Coding for Visual Telephony" recommended H.261 video coding algorithm in 1990, which was designed and optimized for low target bit rate applications suitable for transmission of color video over ISDN at multiple of 64 kbits/s. Typically the picture quality is acceptable with limited motion in the scene at 128 kbits/s [62]. The H.261 standard specifies a Hybrid DCT/DPCM coding algorithm with motion compensation. For an H.261 video coder the input video source consists of non-interlaced color frames

of either CIF (which specifies frames with 352×288 active luminance pixels (Y) and 176×144 pixels for each chrominance band (U or V), each pixel represented with 8 bits.) or quarter CIF (QCIF) format and a frame rate from 7.5 to 30 frames per second.

MPEG-1: A Generic Standard for Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s. The MPEG-1 standard was recommended by ISO/IEC to cover many applications from interactive systems on CD-ROM to the delivery of video over telecommunications networks [63]. The MPEG-1 video coding standard is thought to be generic: a diversity of input parameters, including flexible picture size and frame rate, can be specified by the user to support the wide range of applications profiles. The MPEG-1 video algorithm was sought to retain a large degree of commonality with the ITU-T H.261 standard so that implementations supporting both standards were plausible. However, MPEG-1 was primarily targeted for multimedia CD-ROM applications, requiring additional functionality supported by both encoder and decoder. Important features provided by MPEG-1 include frame based random access of video, fast forward/fast reverse (FF/FR) searches through compressed bit streams, reverse playback of video and editability of the compressed bit stream.

H.262/MPEG-2: Standards for Generic Coding of Moving Pictures and Associated Audio. In 1994 the MPEG-2 draft international standard was released as a joint IUT-T/MPEG standard [64]. Basically, MPEG-2 can be seen as a superset of the MPEG-1 coding standard. Specifically, MPEG-2 aimed at providing video quality not lower than NTSC/PAL and up to CCIR 601 quality. New coding features were added by

MPEG-2 to achieve sufficient functionality and quality, i.e., the availability of efficiently compress interlaced digital video at broadcast quality, improved coding efficiency by different quantization, and various scalability modes.

H.263: An International Standard for Picture Phones over Analog Subscriber Lines. Compared with H.261, the H.263 standard [65] supports arbitrary bit rate, typically 20 kbits/s for PSTN. The picture quality is as good as H.261, while at half rate of H.261 and with more options. Different from H.261, H.263 applies the *overlapped block motion compensation* (OBMC), which provides half-pixel accuracy. H.263 supports more image formats with frame rate usually below 10 frames per second. The H.263 is also the compression core of MPEG-4 standard.

MPEG-4: An International Standard for Coding of Video at Very Low Bit Rate. The ISO MPEG-4 started its standardization activities in July 1993 targeting developing a generic video coding algorithm for a wide range of low bit rate multimedia applications [66]. The MPEG-4 provides advanced functionalities, i.e., interactivity, scalability and error resilience. It also enables the multiplexing of audiovisual objects and composition in a scene. The core features of MEPG-4 include object-based video coding, DWT-based still texture coding, and face and body animation. As mentioned before, H.263 is the compression core of MPEG-4.

H.264/AVC: the Newest International Video Coding Standard. The new joint ITU-T/MPEG standard H.264 [67] is designed for various application areas as broadcast over cable, satellite, cable modem, DSL, etc.; interactive or serial storage on optical and

magnetic devices; conversational services over ISDN, Ethernet, LAN, DSL, wireless and mobile networks, modems, etc.; multimedia messaging services over ISDN, DSL, Ethernet, LAN, wireless and mobile networks, etc. A feature highlight of H.264 for different application considerations include [68]: (1) variable block-size motion compensation with small block size; (2) quarter-pixel accuracy in motion compensation; (3) multiple reference picture motion compensation; (4) directional spatial prediction for intra coding; (5) hierarchical block transform; (6) exact-match inverse transform; (7) *context-adaptive binary arithmetic coding* (CABAC); (8) parameter set structure; (9) flexible slice size; (9) flexible macroblock ordering; (11) flexible slice ordering; (12) redundant pictures; (13) data partitioning, etc. Feature (1)-(7) aim at enhancing the coding efficiency, while feature (8)-(12) aim at improving the robustness to data errors/losses and flexibility for operation over a variety of network environments.

2.5 Summary

In this chapter, we provided a brief description of a broad range of related topics in the areas of 3-D scene reconstruction, multi-view 3-D image coding, and digital video coding techniques. These techniques provide a basis for key contributions of this dissertation. In particular, digital video coding techniques are related to multi-view coding techniques in the sense that video streams can be considered as a sequence of images. We are expecting the benefit from these developed video coding techniques in our proposed multi-view image coding in 3-D space.

3 A Multistage Camera Self-calibration Algorithm

This chapter provides details on one of our contributions in the field of automatic 3-D scene reconstruction — a new multistage camera self-calibration algorithm.

3.1 Introduction

The idea of camera self-calibration was motivated by the fact that in many applications either on-line calibration is expected or the locations of the points in the 3-D scene (ground truth) are not available. In fact, camera self-calibration has attracted a great deal of attention [18][47][19][48][49][39] in the field of computer vision because of its role in automatic 3D reconstruction. For instance, in the general framework of automatic 3-D scene reconstruction based on camera perspective projection model, as shown in Figure 2-2, the camera self-calibration is applied to estimate the camera intrinsic parameters.

We propose a new multistage self-calibration algorithm based on the *equal singular value* (ESV) property of the essential matrix. Differing from previous approaches [47][48][49], where the optimization function is not explicit with respect to the camera intrinsic parameters, we derive a polynomial optimization function of the intrinsic parameters, and then follow a multistage procedure to refine the results. We applied our method to both synthetic and real images. The experimental results show the accuracy

and robustness of our approach when compared with other leading approaches such as the ones proposed in [49][69].

3.2 Polynomial Optimization Function Based on the ESV Property

Two assumptions are made in our algorithm. First, we assume that the camera intrinsic parameters keep unchanged while taking images for the same scene. Hence, Equation (2-8) can be rewritten as:

$$\boldsymbol{E} = \boldsymbol{K}^T \boldsymbol{F} \boldsymbol{K}\,, \tag{3-1}$$

where $\boldsymbol{F}$ and $\boldsymbol{E}$ represent the fundamental matrix and essential matrix from the first image to the second image, respectively, and $\boldsymbol{K}$ is the identical calibration matrix of both cameras. Second, the camera calibration matrix takes the form represented in Equation (2-4). Under this assumption, we have four intrinsic parameters to estimate.

According to Equation (2-9) and (2-10), the essential matrix $\boldsymbol{E}$ can be written as a skew-symmetrical matrix $\boldsymbol{T}$ postmultiplied by a rotation matrix $\boldsymbol{R}$. It is proven in [70] that the necessary and sufficient condition for a 3×3 matrix to be so decomposable is that one of its singular values is zero and the other two are equal. The zero singular value condition is automatically satisfied since the fundamental matrix $\boldsymbol{F}$ is of rank2 while the calibration matrix $\boldsymbol{K}$ is of full rank. Hence the constraint of two equal singular values becomes the basis of our self-calibration algorithm. We show that this constraint can be expressed as a polynomial with respect to the entries of $\boldsymbol{K}$.

The singular values of an arbitrary matrix $\boldsymbol{A}$ are nothing but the square roots of the nonzero eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$. Therefore, the property of the singular values of $\boldsymbol{E}$ corresponds to the property of the eigenvalues of $\boldsymbol{E}^T\boldsymbol{E}$: one of the three eigenvalues of $\boldsymbol{E}^T\boldsymbol{E}$ must be zero while the other two must be equal.

Let

$$\boldsymbol{A} = \boldsymbol{E}^T\boldsymbol{E} = (\boldsymbol{K}^T\boldsymbol{F}\boldsymbol{K})^T(\boldsymbol{K}^T\boldsymbol{F}\boldsymbol{K}) = \begin{bmatrix} a_1 & a_4 & a_5 \\ a_4 & a_2 & a_6 \\ a_5 & a_6 & a_3 \end{bmatrix}, \tag{3-2}$$

then the characteristic equation $\det(\lambda\mathbf{I} - \boldsymbol{A}) = 0$ can be expressed as:

$$\lambda^3 + l_2\lambda^2 + l_1\lambda + l_0 = 0, \tag{3-3}$$

where

$$l_2 = -(a_1 + a_2 + a_3), \tag{3-4}$$

$$l_1 = a_1a_2 + a_1a_3 + a_2a_3 - a_4^2 - a_5^2 - a_6^2, \tag{3-5}$$

$$l_0 = -a_1a_2a_3 - 2a_4a_5a_6 + a_1a_6^2 + a_2a_5^2 + a_3a_4^2. \tag{3-6}$$

The three eigenvalues λ_1, λ_2 and λ_3 of $\boldsymbol{A}$ should satisfy $\lambda_1 = \lambda_2$ and $\lambda_3 = 0$, according to the ESV property of $\boldsymbol{E}$. Substituting $\lambda_3 = 0$ in Equation (3-3) leads to an order-reduced equation

$$\lambda^2 + l_2\lambda + l_1 = 0. \tag{3-7}$$

Furthermore, $\lambda_1 = \lambda_2$ leads to:

$$l_2^2 - 4l_1 = 0. \tag{3-8}$$

Equation (3-8) is the mathematical expression of the ESV constraint in our self-calibration algorithm. Substitute for l_1 and l_2 by Equation (3-4) and (3-5), then we rewrite Equation (3-8) as follows:

$$(a_1 + a_2 + a_3)^2 - 4(a_1a_2 + a_1a_3 + a_2a_3 - a_4^2 - a_5^2 - a_6^2) = 0, \tag{3-9}$$

Given

$\boldsymbol{F} = \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix}$, the entries of $\boldsymbol{A}$ can be expressed in terms of the entries of $\boldsymbol{K}$ and $\boldsymbol{F}$, according to Equation (3-2):

$$\begin{aligned}
a_1 &= F_{11}^2\alpha_u^4 + F_{21}^2\alpha_u^2\alpha_v^2 + p_3^2\alpha_u^2, \\
a_2 &= F_{22}^2\alpha_v^4 + F_{12}^2\alpha_u^2\alpha_v^2 + p_4^2\alpha_v^2, \\
a_3 &= p_1^2\alpha_u^2 + p_2^2\alpha_v^2 + s^2, \\
a_4 &= F_{11}F_{12}\alpha_u^3\alpha_v + F_{21}F_{22}\alpha_u\alpha_v^3 + p_3p_4\alpha_u\alpha_v, \\
a_5 &= F_{11}p_1\alpha_u^3 + F_{21}p_2\alpha_u\alpha_v^2 + p_3s\alpha_u, \\
a_6 &= F_{22}p_2\alpha_v^3 + F_{12}p_1\alpha_u^2\alpha_v + p_4s\alpha_v,
\end{aligned} \tag{3-10}$$

where

$$p_1 = F_{11}u_0 + F_{12}v_0 + F_{13},$$

$$p_2 = F_{21}u_0 + F_{22}v_0 + F_{23},$$

$$p_3 = F_{11}u_0 + F_{21}v_0 + F_{31},$$

$$p_4 = F_{12}u_0 + F_{22}v_0 + F_{32},$$

$$s = F_{11}u_0^2 + (F_{12} + F_{21})u_0v_0 + F_{22}v_0^2 + (F_{13} + F_{31})u_0 + (F_{23} + F_{32})v_0 + F_{33}.$$

We make it clear that in Equation (3-10), all the equations are written explicitly in terms of α_u and α_v. This is for the convenience of further description of our self-calibration approach. In fact, there are four variables in Equation (3-10): α_u, α_v, u_0 and v_0.

Substituting Equation (3-10) for the variables $a_i, i = 1, \cdots, 6$ in Equation (3-9) results in the following equation explicitly with respect to α_u and α_v:

$$\begin{aligned} & c_1\alpha_u^8 + c_2\alpha_v^8 + c_3\alpha_u^6\alpha_v^2 + c_4\alpha_u^4\alpha_v^4 + c_5\alpha_u^2\alpha_v^6 \\ & + c_6\alpha_u^6 + c_7\alpha_v^6 + c_8\alpha_u^4\alpha_v^2 + c_9\alpha_u^2\alpha_v^4 \\ & + c_{10}\alpha_u^4 + c_{11}\alpha_v^4 + c_{12}\alpha_u^2\alpha_v^2 \\ & + c_{13}\alpha_u^2 + c_{14}\alpha_v^2 + c_{15} = 0 \end{aligned}, \tag{3-11}$$

where

$c_1 = F_{11}^4$,

$c_2 = F_{22}^4$,

$c_3 = 2F_{11}^2(F_{12}^2 + F_{21}^2)$,

$c_4 = (F_{12}^2 - F_{21}^2)^2 - 2F_{11}^2F_{22}^2 + 8F_{11}F_{12}F_{21}F_{22}$,

$c_5 = 2F_{22}^2(F_{12}^2 + F_{21}^2)$,

$c_6 = 2F_{11}^2(p_1^2 + p_3^2)$,

$c_7 = 2F_{22}^2(p_2^2 + p_4^2)$,

$c_8 = 2(F_{21}^2 - F_{12}^2)(p_3^2 - p_1^2) - 2F_{11}^2(p_2^2 + p_4^2) + 8F_{11}F_{21}p_1p_2 + 8F_{11}F_{12}p_3p_4$,

$c_9 = 2(F_{21}^2 - F_{12}^2)(p_4^2 - p_2^2) - 2F_{22}^2(p_1^2 + p_3^2) + 8F_{12}F_{22}p_1p_2 + 8F_{21}F_{22}p_3p_4$,

$$c_{10} = (p_3^2 - p_1^2)^2 + 8F_{11}p_1p_3s - 2F_{11}^2s^2,$$

$$c_{11} = (p_4^2 - p_1^2)^2 + 8F_{22}p_2p_4s - 2F_{22}^2s^2,$$

$$c_{12} = 2(p_3^2 - p_1^2)(p_4^2 - p_2^2) - 2(F_{12}^2 + F_{21}^2)s^2 + 8F_{12}p_1p_4s + 8F_{21}p_2p_3s,$$

$$c_{13} = 2(p_1^2 + p_3^2)s^2,$$

$$c_{14} = 2(p_2^2 + p_4^2)s^2,$$

$$c_{15} = s^4. \tag{3-12}$$

An advantage of Equation (3-11) is that it contains only even-order items with respect to α_u and α_v. Letting $x = \alpha_u^2$ and $y = \alpha_v^2$, we can reduce the order of Equation (3-11):

$$\begin{aligned} & f(x, y, c_i, i = 1, \cdots, 15) \\ & = c_1x^4 + c_2y^4 + c_3x^3y + c_4x^2y^2 + c_5xy^3 + c_6x^3 + c_7y^3 + c_8x^2y + c_9xy^2 \\ & + c_{10}x^2 + c_{11}y^2 + c_{12}xy + c_{13}x + c_{14}y + c_{15} \\ & = 0 \end{aligned}, \tag{3-13}$$

where $c_i, i = 1, \cdots, 15$ is the coefficient of each item x^4, y^4, x^3y, and so on, and is expressed as a function of u_0, v_0 and the entries of $\boldsymbol{F}$, as expressed in Equation (3-12).

Equation (3-13) has two advantages. First, $f(x, y, c_i, i = 1, \cdots, 15)$ is a bivariate polynomial with respect to $x = \alpha_u^2$ and $y = \alpha_v^2$, based on the assumption that (u_0, v_0) is fixed (e.g., at the center of the image). This property enables the computation of exact derivatives, which simplifies the optimization. Second, since the coefficients of x^4 and y^4 are positive numbers, this function monotonously increases as x and y approach

infinity. Therefore, the initialization is less critical than it is in a usual optimization problem.

From the perspective of numerical analysis, we may achieve better performance if Equation (3-13) is weighted. We use a normalized version of Equation (3-8) as follows:

$$\frac{l_2^2 - 4l_1}{l_2^2} = 0. \tag{3-14}$$

Comparing Equation (3-14) with Equation (3-8), we take $1/l_2^2$ as the weight.

In practice, we have a set of N images so that we can obtain at most $N(N-1)/2$ fundamental matrices. The advantage of using all of the $N(N-1)/2$ fundamental matrices is twofold: first, the redundancy reinforces the numerical robustness; second, it avoids bias towards any given image. Hence we employ the following *weighted global optimization function*

$$C(x,y) = \sum_{i=1}^{N(N-1)/2} w_i f_i(x, y, c_j^i, j = 1, \cdots, 15), \tag{3-15}$$

where $f_i(x, y, c_j^i, j = 1, \cdots, 15)$ is the optimization function of the i-th image pair, and the weight $w_i = (1/l_2^2)_i$ is a function of x, y, u_0, v_0 and the entries of $\boldsymbol{F_i}$:

$1/(F_{i,11}^2 x^2 + F_{i,22}^2 y^2 + (F_{i,12}^2 + F_{i,21}^2)xy + (p_{i,1}^2 + p_{i,3}^2)x + (p_{i,2}^2 + p_{i,4}^2)y + s_i^2)^2$.

3.3 Multistage Approach to Camera Self-calibration

In practice, we do not directly minimize Equation (3-15) with respect to all of the four intrinsic parameters because it is computationally extensive and unstable. In fact, these parameters impact the final 3-D reconstruction quite differently. Zhang et al. [71]

stated that shifting the principal point from its true position does not cause large distortion of the reconstructed 3-D points, based on the assumption that the values of α_u and α_v are correct. In the case that none of the four parameters are known, the offset of the principal point impacts the estimation of α_u and α_v. However, experiments show that the estimated aspect ratio (α_v/α_u) remains close to its true value while suffering from the offset of the principal point. In [48] it is stated that the estimation of α_v/α_u is very robust to noise. This observation is extended here: the estimation of the aspect ratio is robust to both the noise of the coordinates of the image points and the noise caused by the incorrect location of the principal point.

Based on the above observation, we formulate our multistage algorithm of self-calibration as follows:

Step 1. Estimate α_u and α_v, assuming that (u_0, v_0) is located at the center of the image. The outcomes are denoted by $\tilde{\alpha}_u^{(1)}$ and $\tilde{\alpha}_v^{(1)}$.

Step 2. Refine the estimation of α_u, u_0 and v_0, assuming $\alpha_v/\alpha_u = \tilde{\alpha}_v^{(1)}/\tilde{\alpha}_u^{(1)}$. The outcomes are $\tilde{\alpha}_u^{(2)}$, $\tilde{u}_0^{(2)}$ and $\tilde{v}_0^{(2)}$.

Step 3. Refine the estimation of α_u and α_v, assuming $(u_0, v_0) = (\tilde{u}_0^{(2)}, \tilde{v}_0^{(2)})$. The outcomes are $\tilde{\alpha}_u^{(3)}$ and $\tilde{\alpha}_v^{(3)}$.

Step 4. Refine α_u, α_v, u_0 and v_0, with the initial conditions $\tilde{\alpha}_u^{(3)}$, $\tilde{\alpha}_v^{(3)}$, $\tilde{u}_0^{(2)}$ and $\tilde{v}_0^{(2)}$. The final outcomes are $\tilde{\alpha}_u$, $\tilde{\alpha}_v$, $\tilde{u}_0$ and $\tilde{v}_0$.

Step 1 and step 3 are accomplished by optimizing the objective function expressed in

Equation (3-15). In step 2, we need to estimate u_0 and v_0. However, Equation (3-15) is not a function of u_0 and v_0. Hence, instead of using Equation (3-15), we use the following optimization function [49], which directly computes the singular values of the essential matrix by singular value decomposition:

$$C(x,y) = \sum_{i=1}^{N(N-1)/2} (1 - \frac{\lambda_2^i}{\lambda_1^i}), \tag{3-16}$$

where λ_1^i and λ_2^i represent the nonzero singular values of $\boldsymbol{E}_i$, in descending order. Equation (3-16) is also used in step 4.

3.4 Experimental Results

We first provide experimental results with synthetic data. In this experiment, 20 synthetic images (512×512) were generated with 200 points randomly scattered in a cube of edge size 800 centered at (0,0,2000). The intrinsic parameters are chosen as $\alpha_u = 957.8$, $\alpha_v = 891.2$ and $(u_0, v_0) = (279,241)$, to simulate the standard settings of a real camera. We added (to the pixel locations) two types of noise: uniformly distributed noise in [-0.5 pixel, 0.5 pixel], which simulates the quantization error, and Gaussian noise with standard deviation of 1 pixel, which simulates the noise caused from pixel correspondence match. Fundamental matrices were computed from various numbers (4 to 20) of images using the normalized linear criterion [46].

To evaluate the performance of our method, we compared our algorithm (GR method) with the one presented in [49], which is equivalent to using only step 4 of our method.

This approach, referred to as the RW method, represents an example of an ESV-based state-of-the-art algorithm. Each estimation task based on a certain number of images was repeated $N = 100$ times. We measured the average value of the relative error $\varepsilon_\alpha = \frac{1}{N\alpha}\sum_{i=1}^{N}|\tilde{\alpha}_i - \alpha|$. Here $\alpha \in \{\alpha_u, \alpha_v\}$ is the true value and $\tilde{\alpha}_i \in \{\alpha_{u,i}, \alpha_{v,i}\}$ is the estimated value from the i-th round of the experiment. In Figure 3-1 and Figure 3-2, we present only the results for α_u. Similar results were obtained for α_v.

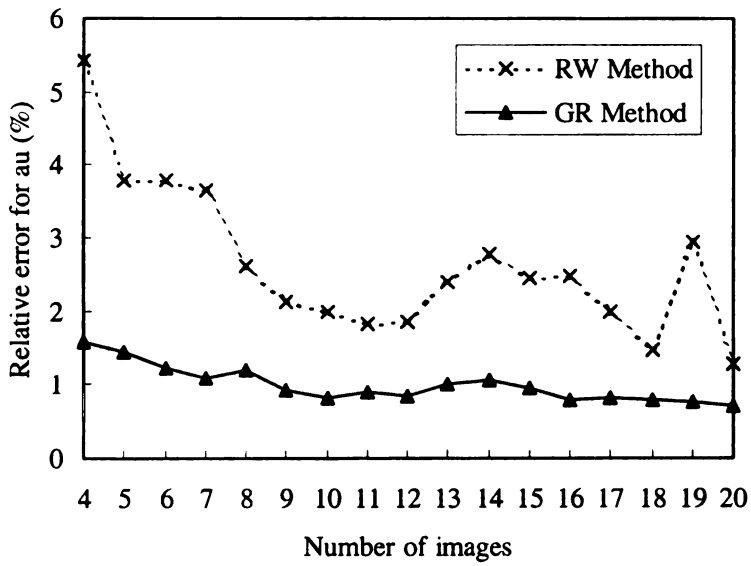


Figure 3-1 The comparison of the average relative error for α_u when uniformly distributed noise [-0.5 pixel, 0.5 pixel] is added to the pixel coordinates.

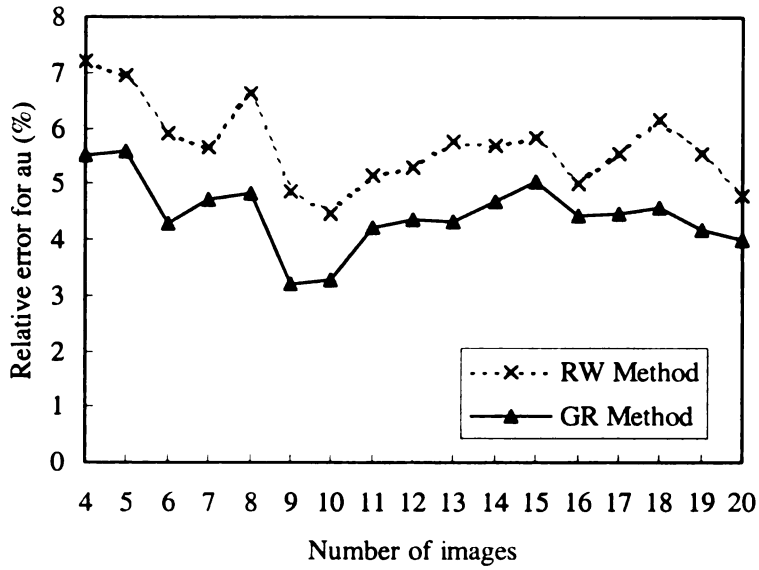


Figure 3-2 The comparison of the average relative error for α_u when the Gaussian noise with standard deviation of 1 pixel is added to the pixel coordinates.

We also measured the average relative error resulting from estimating the coordinates of the principal point:

$$\varepsilon_{pp} = \frac{1}{N\sqrt{u_0^2 + v_0^2}} \sum_{i=1}^{N} \sqrt{(\tilde{u}_{0,i} - u_0)^2 + (\tilde{v}_{0,i} - v_0)^2} .$$

Here (u_0, v_0) are the true coordinates of the principal point and $(u_{0,i}, v_{0,i})$ is the estimated value from the i-th round of experiment. The experimental results are shown in Figure 3-3 and Figure 3-4.

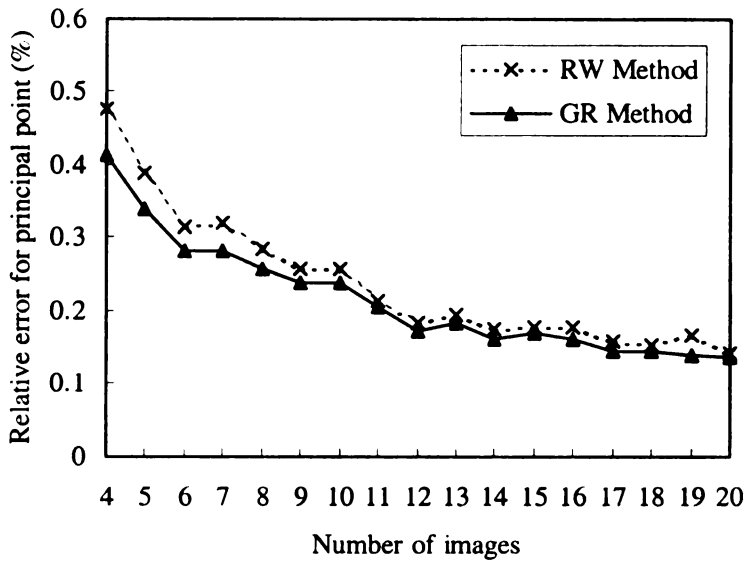


Figure 3-3 The comparison of the average relative error for the principal point when uniformly distributed noise [-0.5 pixel, 0.5 pixel] is added to the pixel coordinates.

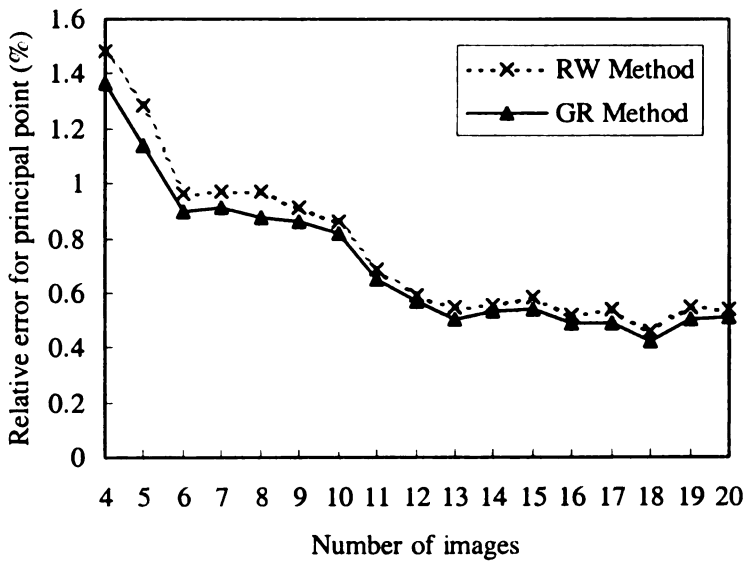


Figure 3-4 The comparison of the average relative error for the principal point when Gaussian noise with standard deviation of 1 pixel is added to the pixel coordinates.

From the experimental results, we can make the following conclusions:

• Our method outperformed the RW method for the estimation of both α_u and the principal point under the two noise conditions. The initial values of α_u and α_v are set to 1000 and 1000 respectively in the RW method while they are set to 2000 and 2000 respectively in our method. Hence, although we have selected worse initial values, our estimation results are still better than the results from the RW method. This observation is consistent with the statement in Section 3.3 that our optimization method is insensitive to the initialization.

• Our performance improvement over the RW method is greater with respect to the estimation of α_u than the estimation of the principal point. This result is not unsatisfactory because as mentioned in Section 3.1, the scaling factors α_u and α_v have more impact on the 3D reconstruction than the principal point does.

• The number of used images influences the estimation. Generally speaking, using more images may improve the estimation. In the case of real images, we may select well-estimated fundamental matrices for the self-calibration.

We then show the results of self-calibration for a set of images named "Valbonne Church" of size 768×512. We downloaded these images from the INRIA ftp site. We selected six images of this set for our experiment. The point correspondences were picked up manually. We computed all of the fifteen fundamental matrices and then selected "well-estimated" fundamental matrices in terms of the error of epipolar distance [45]. In

Table 3-1, we compare our results with those stated in [49][69].

Table 3-1 Estimation results of our method and other methods on real images

	α_u	α_v	(u_0, v_0)
Kruppa	679.285	681.345	(383.188, 258.802)
RW	605.5	—	—
GR	658.5	661.6	(406, 238)

In Table 3-1, the first row labeled "Kruppa" represents the estimation from [69], which is regarded as a precise estimation. The second row labeled "RW" represents the results from [49]. This method estimated only the focal length. The last row labeled "GR" shows our results. Compared with the results of the RW method, our estimated α_u and α_v are much closer to those obtained by the Kruppa method. But our estimation of the principal point is different from that by Kruppa method. The above estimation results illustrate that our proposed approach may, at minimum, provide a very close performance to other well-established approaches for self-calibration. More importantly, the proposed method provides new advantages such as stability and simplicity due to the polynomial form of our optimization function.

3.5 Summary

We proposed a multistage camera self-calibration algorithm based on the ESV property of the essential matrix. Unlike previous ESV-based approaches [47][48][49], we derived a polynomial optimization function, which is an explicit expression of the unknown intrinsic parameters. This makes the optimization simple and insensitive to the initialization.

We also performed a stability analysis of the intrinsic parameters and then proposed a multistage procedure to refine the self-calibration. We compared our method with the one presented in [49] on synthetic image data. The statistical results show that our method performed better than the method in [49]. We also compared our estimation results with the results from [49] and [69] on real image data. In this case, we obtained, at minimum, comparable performance to these well-established methods.

4 A New Proof for the Four Solutions in Recovering the Camera Relative Motion from the Essential Matrix

This chapter discusses our other contributions in the field of automatic 3-D scene reconstruction. In particular, the focus of this chapter is on investigating the exact number of possible solutions in recovering the camera relative motion from the essential matrix.

4.1 Introduction

Recovering camera relative motion (rotation and translation) from image point correspondences between two perspective views has been studied for two decades. As mentioned in Section 2.1.4, it plays an important role in obtaining 3D information from multiple images. Furthermore, 3D reconstruction has attracted increasing attention in the field of multi-view image coding and video coding. Higgins addressed the problem of scene reconstruction in [20], where the concept of the *essential matrix* ($\boldsymbol{E}$) was first introduced. Tsai and Huang [21] proposed a Singular Value Decomposition (SVD) method for estimating camera motion from $\boldsymbol{E}$. Weng et al. [22] proposed another approach for recovering the camera motion based on the camera projection geometry. Horn [72] presented a closed-form solution to camera motion recovery.

The number of possible solutions in recovering the camera relative rotation has also been studied over the same period [21][72][73][74]. Both Tsai [21] and Horn [72]

claimed that there are four possible solutions. Hartley and Zisserman [73] presented a proof that there are only four possible solutions by SVD of the essential matrix. However, Wang [74] stated that eight solutions can be derived from the SVD-based method.

We present a new proof that there are only four possible solutions in decomposing $\boldsymbol{E}$ in the non-degenerate configuration of camera motion. (The non-degenerate configuration corresponds to camera relative motions that can be recovered up to a limited number of solutions. For instance, it is a non-degenerate case when the translation vector has a non-zero norm.) Different from Tsai and Huang's method [21] and Hartley and Zisserman's method [73], our proof concentrates on the geometry among the essential-matrix, the camera rotation, and the camera translation. Based on our proof, we discuss the methods in [21][72][73][74] and argue that the methods presented in [21][72][73] are consistent with our conclusion. In particular, we propose a generalized SVD-based proof for the four possible solutions in decomposing $\boldsymbol{E}$. We also provide some insight into degenerate configurations of camera motion.

4.2 Determining the Number of Solutions in Recovering Camera Relative Motion From the Essential Matrix

As stated in Section 2.1.4, given a set of camera intrinsic parameters (which means that the image coordinate system is calibrated), $\boldsymbol{E}$ is a 3×3 matrix representing the epipolar geometry between two images taken from two different viewpoints. Let

$[u_1, v_1]^T$ and $[u_2, v_2]^T$ represent the calibrated image pixel coordinates of the same 3D point, taken by the first and second camera respectively. Thus, the epipolar geometry between the two image pixels is expressed as follows:

$$[u_2, v_2, 1]\boldsymbol{E}[u_1, v_1, 1]^T = 0. \tag{4-1}$$

Equation (4-1) is a homogeneous linear equation with respect to the nine entries of $\boldsymbol{E}$. In practice, given at least eight image pixel correspondences, $\boldsymbol{E}$ can be obtained by some numerical method.

On the other hand, $\boldsymbol{E}$ is associated with the camera relative rotation and translation from the first camera to the second one by

$$\boldsymbol{E} = \boldsymbol{TR}, \tag{4-2}$$

where $\boldsymbol{T}$ is a 3×3 matrix that satisfies $\boldsymbol{T}\mathbf{v} = \mathbf{t}\times\mathbf{v}$ for any vector $\mathbf{v}$, with $\mathbf{t}$ representing the translation vector from the first camera to the second one, and $\boldsymbol{R}$ represents the rotation matrix from the first camera to the second one.

Denoting $\boldsymbol{R} = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3]$, where $\mathbf{r}_i$ represents the i-th column of $\boldsymbol{R}$, Equation (4-1) can be rewritten as

$$\boldsymbol{E} = [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] = [\mathbf{t}\times\mathbf{r}_1 \quad \mathbf{t}\times\mathbf{r}_2 \quad \mathbf{t}\times\mathbf{r}_3]. \tag{4-3}$$

Equation (4-3) shows that each column of $\boldsymbol{E}$ is orthogonal to $\mathbf{t}$ (In other words, the three columns of $\boldsymbol{E}$ are coplanar.). Hence, $\mathbf{t}$ is parallel to the cross product of any two columns of $\boldsymbol{E}$. Apparently there are two possible solutions to $\mathbf{t}$ with opposite directions. According to Horn's method [72], the two solutions are

$$\mathbf{t} = \pm \frac{\mathbf{e}_i \times \mathbf{e}_j}{\|\mathbf{e}_i \times \mathbf{e}_j\|} \sqrt{\frac{1}{2} Trace(\boldsymbol{E}\boldsymbol{E}^T)}, \tag{4-4}$$

where $\mathbf{e}_i \times \mathbf{e}_j$ is the largest of the three possible pairwise cross-products $\mathbf{e}_1 \times \mathbf{e}_2$, $\mathbf{e}_2 \times \mathbf{e}_3$ and $\mathbf{e}_3 \times \mathbf{e}_1$, for the sake of numerical accuracy.

After obtaining $\mathbf{t}$ from Equation (4-4), we need to recover $\boldsymbol{R}$ that satisfies Equation (4-3). The ambiguity arises from the multiplicity of solutions to $\mathbf{e}_i = \mathbf{t} \times \mathbf{r}_i$. The following proposition can be easily proven to be true.

Proposition.1 Given $\mathbf{e}$ and $\mathbf{t}$, there are generally two unit vectors satisfying $\mathbf{e} = \mathbf{t} \times \mathbf{r}$, denoted as $\mathbf{r}^1$ and $\mathbf{r}^2$. The relationship between $\mathbf{r}^1$ and $\mathbf{r}^2$ can be shown to satisfy the following:

$$\mathbf{r}^2 = \mathbf{r}^1 - 2\frac{\mathbf{r}^1 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}} \mathbf{t}. \tag{4-5}$$

Proof: In terms of the definition of cross products, two conclusions can be made from $\mathbf{e} = \mathbf{t} \times \mathbf{r}$.

First, both $\mathbf{t}$ and $\mathbf{r}$ are located within the plane to which $\mathbf{e}$ is perpendicular.

Second, $\|\mathbf{e}\| = \|\mathbf{t}\|\|\mathbf{r}\|\sin(\theta)$, where θ is the angle between $\mathbf{t}$ and $\mathbf{r}$. Since $\mathbf{e}$ and $\mathbf{t}$ are given and $\mathbf{r}$ is a unit vector, the value of $\sin(\theta)$ is uniquely determined. However, two possible values of θ can be found for the same value of $\sin(\theta)$.

The above two conclusions show that there are two possible solutions to $\mathbf{r}$, denoted as $\mathbf{r}^1$ and $\mathbf{r}^2$. They are located within the plane to which $\mathbf{e}$ is perpendicular. If the

angle between $\mathbf{t}$ and $\mathbf{r}^1$ is denoted as θ, the angle between $\mathbf{t}$ and $\mathbf{r}^2$ is $(\pi - \theta)$.

The geometry among $\mathbf{e}$, $\mathbf{t}$ and $\mathbf{r}^{1,2}$ can be illustrated in Figure 4-1. By using the rule of vector summation and the definition of dot product, Equation (4-5) can be directly derived.

This completes the proof.

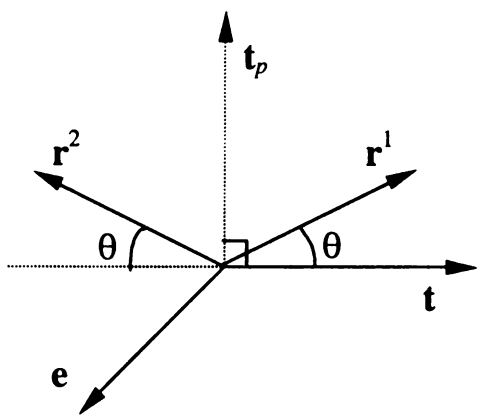


Figure 4-1 The geometry among $\mathbf{e}$, $\mathbf{t}$ and $\mathbf{r}^{1,2}$: $\mathbf{t}$ and $\mathbf{t}_p$ are orthogonal to each other and determine a plane $\mathbf{t}-\mathbf{t}_p$ to which $\mathbf{e}$ is perpendicular. $\mathbf{r}^1$ and $\mathbf{r}^2$ are in the same plane $\mathbf{t}-\mathbf{t}_p$ and are symmetric with respect to $\mathbf{t}_p$. $\mathbf{r}^1$ and $\mathbf{r}^2$ become identical when $\mathbf{r}^1$ or $\mathbf{r}^2$ is perpendicular to $\mathbf{t}$.

We know now that there are two solutions to $\mathbf{r}_i$ satisfying $\mathbf{e}_i = \mathbf{t} \times \mathbf{r}_i$, $i = 1,2,3$, with the constraint $\|\mathbf{r}_i\| = 1$. (This constraint arises from the property of $\boldsymbol{R}$ as a rotation matrix.) However, we prove that this ambiguity can be eliminated by other properties of $\boldsymbol{R}$ as a rotation matrix.

Proposition.2 In the non-degenerate configuration of camera motion, given $\boldsymbol{E}$ and $\mathbf{t}$, there are two possible orthonormal matrices $\boldsymbol{R}^1$ and $\boldsymbol{R}^2$ that satisfy Equation (4-3),

$$\boldsymbol{E} = [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] = [\mathbf{t}\times\mathbf{r}_1 \quad \mathbf{t}\times\mathbf{r}_2 \quad \mathbf{t}\times\mathbf{r}_3]$$

with one of these matrices being "proper" (corresponding to a pure rotation) and the other one being "improper".

Proof: We first give the definition of "proper" and "improper" orthonormal matrix. An orthonormal matrix $\boldsymbol{R}$ (which means $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix) is "proper" if $\det(\boldsymbol{R}) = 1$, while $\boldsymbol{R}$ is "improper" if $\det(\boldsymbol{R}) = -1$. The geometrical interpretation of the proper orthonormal matrix is that it represents pure rotation in a 3-D coordinate system and is referred to as a "rotation matrix" in computer vision. In a proper orthonormal matrix $\boldsymbol{R} = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3]$, the three unit vectors $\mathbf{r}_i$ satisfy:

$$\mathbf{r}_i \cdot \mathbf{r}_j = \begin{cases} 1, & if \quad i = j \\ 0, & if \quad i \neq j \end{cases} \quad \text{and} \quad \mathbf{r}_1 \times \mathbf{r}_2 = \mathbf{r}_3 . \tag{4-6}$$

In other words, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ determine a right-handed coordinate system. This property is equivalent to $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$ with $\det(\boldsymbol{R}) = 1$ and will be used later. On the other hand, the three unit vectors in an improper orthonormal matrix determine a left-handed coordinate system and do not represent a pure rotation in commonly used right-handed coordinate systems.

Let $[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3]$ be a proper orthonormal matrix satisfying Equation (4-3). Given

E and $\mathbf{t}$, we obtain the following counterparts according to Equation (4-5).

$$\mathbf{r}_{ic} = \mathbf{r}_i - 2\frac{\mathbf{r}_i \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t}, \quad i = 1,2,3. \tag{4-7}$$

We have:

$$\mathbf{r}_i \cdot \mathbf{r}_{jc} = \mathbf{r}_i \cdot (\mathbf{r}_j - 2\frac{\mathbf{r}_j \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t}) = -2\frac{\mathbf{r}_j \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{r}_i \cdot \mathbf{t}, \quad i \neq j.$$

This result shows that the orthogonality does not hold between $\mathbf{r}_i$ and $\mathbf{r}_{jc}$ if $i \neq j$, because in general $\mathbf{r}_i \cdot \mathbf{t} \neq 0$. Hence, no triple vectors coming from $\mathbf{r}_i$ and $\mathbf{r}_{jc}$ can construct an orthogonal matrix. At the same time we have:

$$\begin{aligned} & \mathbf{r}_{ic} \cdot \mathbf{r}_{jc} \\ & = (\mathbf{r}_i - 2\frac{\mathbf{r}_i \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t}) \cdot (\mathbf{r}_j - 2\frac{\mathbf{r}_j \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t}) \\ & = \mathbf{r}_i \cdot \mathbf{r}_j - 2\frac{\mathbf{r}_j \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{r}_i\mathbf{t} - 2\frac{\mathbf{r}_i \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t} \cdot \mathbf{r}_j + 4\frac{(\mathbf{r}_i \cdot \mathbf{t}) \cdot (\mathbf{r}_j \cdot \mathbf{t})}{(\mathbf{t} \cdot \mathbf{t})^2}\mathbf{t} \cdot \mathbf{t} \\ & = \mathbf{r}_i \cdot \mathbf{r}_j \end{aligned} \tag{4-8}$$

Equation (4-8) shows that $[\mathbf{r}_{1c} \quad \mathbf{r}_{2c} \quad \mathbf{r}_{3c}]$ is an orthonormal matrix. Hence, based on Equation (4-7)-(4-8), we proved that $[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3]$ and $[\mathbf{r}_{1c} \quad \mathbf{r}_{2c} \quad \mathbf{r}_{3c}]$ are the only two orthonormal matrices satisfying Equation (4-3). However, we now prove that the latter matrix $[\mathbf{r}_{1c} \quad \mathbf{r}_{2c} \quad \mathbf{r}_{3c}]$ is improper. Recall that the orthonormal matrix $[\mathbf{r}_{1c} \quad \mathbf{r}_{2c} \quad \mathbf{r}_{3c}]$ would be proper if $\mathbf{r}_{1c} \times \mathbf{r}_{2c} = \mathbf{r}_{3c}$. We show below that this condition is not satisfied.

$$
\begin{aligned}
&\mathbf{r}_{1c} \times \mathbf{r}_{2c} \\
&= (\mathbf{r}_1 - 2\frac{\mathbf{r}_1 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t}) \times (\mathbf{r}_2 - 2\frac{\mathbf{r}_2 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t}) \\
&= \mathbf{r}_1 \times \mathbf{r}_2 - 2\frac{\mathbf{r}_2 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{r}_1 \times \mathbf{t} - 2\frac{\mathbf{r}_1 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t} \times \mathbf{r}_2 + 4\frac{(\mathbf{r}_1 \cdot \mathbf{t})(\mathbf{r}_2 \cdot \mathbf{t})}{(\mathbf{t} \cdot \mathbf{t})^2}\mathbf{t} \times \mathbf{t} \cdot \\
&= \mathbf{r}_3 - 2\frac{\mathbf{r}_2 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{r}_1 \times \mathbf{t} - 2\frac{\mathbf{r}_1 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t} \times \mathbf{r}_2
\end{aligned}
\tag{4-9}
$$

Using the relationship $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}$, we rewrite the third term in Equation (4-9) as

$$
\begin{aligned}
&-2\frac{\mathbf{r}_1 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t} \times \mathbf{r}_2 \\
&= -\frac{2}{\mathbf{t} \cdot \mathbf{t}}(\mathbf{r}_1 \times ((\mathbf{t} \times \mathbf{r}_2) \times \mathbf{t}) + (\mathbf{r}_1 \cdot (\mathbf{t} \times \mathbf{r}_2))\mathbf{t}) \\
&= -\frac{2}{\mathbf{t} \cdot \mathbf{t}}(\mathbf{r}_1 \times ((\mathbf{t} \cdot \mathbf{t})\mathbf{r}_2 - (\mathbf{r}_2 \cdot \mathbf{t})\mathbf{t}) + (\mathbf{t} \cdot (\mathbf{r}_2 \times \mathbf{r}_1))\mathbf{t}) . \\
&= -\frac{2}{\mathbf{t} \cdot \mathbf{t}}((\mathbf{t} \cdot \mathbf{t})\mathbf{r}_3 - (\mathbf{r}_2 \cdot \mathbf{t})(\mathbf{r}_1 \times \mathbf{t}) - (\mathbf{r}_3 \cdot \mathbf{t})\mathbf{t}) \\
&= -2\mathbf{r}_3 + 2\frac{\mathbf{r}_2 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{r}_1 \times \mathbf{t} + 2\frac{\mathbf{r}_3 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t}
\end{aligned}
\tag{4-10}
$$

Substituting Equation (4-10) for the third item in Equation (4-9), we obtain

$$
\begin{aligned}
&\mathbf{r}_{1c} \times \mathbf{r}_{2c} \\
&= \mathbf{r}_3 - 2\frac{\mathbf{r}_2 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{r}_1 \times \mathbf{t} - 2\mathbf{r}_3 + 2\frac{\mathbf{r}_2 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{r}_1 \times \mathbf{t} + 2\frac{\mathbf{r}_3 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t} . \\
&= -(\mathbf{r}_3 - 2\frac{\mathbf{r}_3 \cdot \mathbf{t}}{\mathbf{t} \cdot \mathbf{t}}\mathbf{t}) = -\mathbf{r}_{3c}
\end{aligned}
\tag{4-11}
$$

Equation (4-11) shows that $\mathbf{r}_{1c}$, $\mathbf{r}_{2c}$ and $\mathbf{r}_{3c}$ determine a left-handed coordinate system. Hence, $[\mathbf{r}_{1c} \quad \mathbf{r}_{2c} \quad \mathbf{r}_{3c}]$ is an improper orthonormal matrix that satisfies Equation (4-3).

This completes the proof.

From proposition 2 we conclude that in the non-degenerate case, there is only one solution to $\boldsymbol{R}$ that satisfies Equation (4-3), given $\boldsymbol{E}$ and $\mathbf{t}$. Because there are two possible solutions to $\mathbf{t}$ and the sign of $\boldsymbol{E}$ is uncertain according to Equation (4-1), we can obtain four possible solutions to the camera motion, denoted as $\{\boldsymbol{R}^+,\mathbf{t}\} \leftarrow \boldsymbol{E}$, $\{\boldsymbol{R}^-,-\mathbf{t}\} \leftarrow \boldsymbol{E}$, $\{\boldsymbol{R}^-,\mathbf{t}\} \leftarrow -\boldsymbol{E}$ and $\{\boldsymbol{R}^+,-\mathbf{t}\} \leftarrow -\boldsymbol{E}$. In summary, we have two different solutions to $\boldsymbol{R}$ and two different solutions to $\mathbf{t}$. This result is consistent with the conclusion about the number of solutions in [21][72][73]. Furthermore, the relationship between the first and the forth solution is that the translation vector from the first to the second camera is reversed. The relationship between the first and the second solution is shown in Figure 4-2.

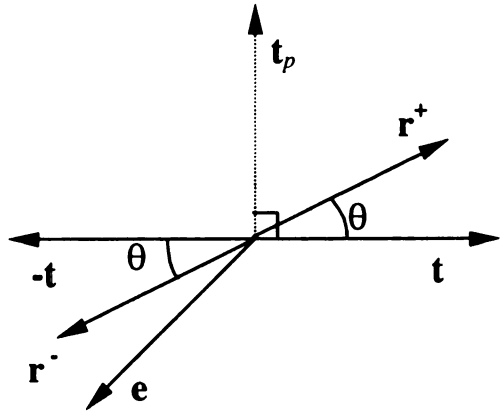


Figure 4-2 The relationship between $\{\boldsymbol{R}^+,\mathbf{t}\} \leftarrow \boldsymbol{E}$ and $\{\boldsymbol{R}^-,-\mathbf{t}\} \leftarrow \boldsymbol{E}$: $\mathbf{r}^-$ is rotated by 180° from $\mathbf{r}^+$ in the plane $\mathbf{t}-\mathbf{t}_p$.

Our statements about the relationship between different solutions are also consistent with those in [73]. However, [73] derived the conclusion from purely the perspective of

mathematics, while we reached our conclusion with a clear geometrical interpretation (Readers are referred to [73] for more details on the camera relative location for different solutions.). In practice, the final solution can be uniquely determined based on the fact that the 3D object must be in front of both cameras.

4.3 Discussion on Different Methods for Decomposing the Essential Matrix

In Section 4.2, we proved the existence of four possible solutions in decomposing $\boldsymbol{E}$. Tsai [21], Horn [72], and Hartley [73] proposed different methods to find the four solutions.

According to Horn's method, $\boldsymbol{R}$ can be obtained from $\boldsymbol{E}$ and $\mathbf{t}$ in terms of the following equations:

$$\begin{aligned} \mathbf{r}_1 &= (\mathbf{e}_2 \times \mathbf{e}_3 + \mathbf{e}_1 \times \mathbf{t})/(\mathbf{t} \cdot \mathbf{t}) \\ \mathbf{r}_2 &= (\mathbf{e}_3 \times \mathbf{e}_1 + \mathbf{e}_2 \times \mathbf{t})/(\mathbf{t} \cdot \mathbf{t}) . \\ \mathbf{r}_3 &= (\mathbf{e}_1 \times \mathbf{e}_2 + \mathbf{e}_3 \times \mathbf{t})/(\mathbf{t} \cdot \mathbf{t}) \end{aligned} \tag{4-12}$$

It can be easily proven that Equation (4-12) produces an orthonormal matrix that satisfies Equation (4-3), in the case that $\mathbf{t}$ is not equal to zero. However, Horn did not prove that the obtained matrix is exactly "proper". We give the proof in the following by using Equation (4-3).

$$
\begin{aligned}
&\mathbf{r}_3 \\
&= \frac{1}{\mathbf{t}\cdot\mathbf{t}}((\mathbf{t}\times\mathbf{r}_1)\times(\mathbf{t}\times\mathbf{r}_2)+(\mathbf{t}\times\mathbf{r}_3)\times\mathbf{t}) \\
&= \frac{1}{\mathbf{t}\cdot\mathbf{t}}((\mathbf{t}\cdot(\mathbf{t}\times\mathbf{r}_2))\mathbf{r}_1-(\mathbf{r}_1\cdot(\mathbf{t}\times\mathbf{r}_2))\mathbf{t}+(\mathbf{t}\cdot\mathbf{t})\mathbf{r}_3-(\mathbf{r}_3\cdot\mathbf{t})\mathbf{t}) \\
&= \frac{1}{\mathbf{t}\cdot\mathbf{t}}((\mathbf{t}\cdot(\mathbf{r}_1\times\mathbf{r}_2))\mathbf{t}+(\mathbf{t}\cdot\mathbf{t})\mathbf{r}_3-(\mathbf{r}_3\cdot\mathbf{t})\mathbf{t}) \\
&= \mathbf{r}_3+\frac{1}{\mathbf{t}\cdot\mathbf{t}}(\mathbf{t}\cdot(\mathbf{r}_1\times\mathbf{r}_2-\mathbf{r}_3))\mathbf{t}
\end{aligned}
$$

From above, we could conclude that:

$$\frac{1}{\mathbf{t}\cdot\mathbf{t}}(\mathbf{t}\cdot(\mathbf{r}_1\times\mathbf{r}_2-\mathbf{r}_3))\mathbf{t}=0.$$

This equation holds for any configuration of $\boldsymbol{R}$ and $\mathbf{t}$. Hence, we obtain $\mathbf{r}_1\times\mathbf{r}_2=\mathbf{r}_3$. Thus, Horn's method produces a proper orthonormal matrix $\boldsymbol{R}$ that satisfies Equation (4-3).

In Tsai's method, $\boldsymbol{R}$ and $\mathbf{t}$ are obtained simultaneously by performing the SVD on $\boldsymbol{E}$. The constraint $\det(\boldsymbol{R})=1$ is taken into account in this method. Readers are referred to [21] and [73] for details.

However, Wang [74] stated that they have found four possible choices of $\boldsymbol{R}$ resulting from the feasibility of SVD for $\boldsymbol{E}$, which is not considered in the existing SVD-based proof for the four possible solutions [73]. In the next section, we provide a generalized SVD-based proof to show that although the SVD for $\boldsymbol{E}$ is not unique, the different SVDs lead to only two possible solutions to $\boldsymbol{R}$.

Finally, we briefly discuss the degeneracy of the camera motion. As mentioned before, when $\|\mathbf{t}\|=0$, $\boldsymbol{R}$ cannot be recovered from Equation (4-12). In fact, $\|\mathbf{t}\|=0$

refers to pure rotation, which is a well-known degenerate configuration of camera motion. Another degenerate case occurs when $\mathbf{t}$ happens to be parallel to any $\mathbf{r}_i, i = 1,2,3$. However, in practice $\boldsymbol{E}$ is always obtained from some estimation procedure and offsets from the true value. Hence, the characteristics of degenerate case (such as $\|\mathbf{t}\| = 0$ in pure rotation) easily annihilates in noise. Numerically degenerate camera motion can produce large estimation errors.

4.4 A Generalized SVD-based Proof for the Four Possible Solutions in Decomposing the Essential Matrix

Hartley [73] presented a proof for the four possible solutions to the camera relative motion by SVD for $\boldsymbol{E}$. However, the feasibility in decomposing $\boldsymbol{E}$ was not considered in the proof. This problem was discussed in [74] and it was claimed in [74] that four additional solutions can be found resulting from the feasibility of SVD for $\boldsymbol{E}$. We here discuss in detail the problem of feasible SVDs for $\boldsymbol{E}$.

It is well-known that a 3×3 matrix is an essential matrix if and only if two of its singular values are equal and the third is zero [73]. Hence, $\boldsymbol{E}$ can be decomposed as $\boldsymbol{E} = \boldsymbol{U}\,\mathrm{diag}(1,1,0)\boldsymbol{V}^T$, where both $\boldsymbol{U}$ and $\boldsymbol{V}$ are unit orthogonal matrices. However, this decomposition is not unique for $\boldsymbol{E}$, since $\boldsymbol{E}$ has five degrees of freedom. In fact, there is a one-parameter family of SVDs for $\boldsymbol{E}$ [73]. We provide the following proposition.

Proposition. 3 The family of SVDs for $\boldsymbol{E}$ can be expressed as

$$\boldsymbol{E} = (\boldsymbol{UA})\,\mathrm{diag}(1,1,0)(\boldsymbol{VB})^T, \tag{4-13}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \cos\theta & \mp\sin\theta & 0 \\ \sin\theta & \pm\cos\theta & 0 \\ 0 & 0 & s_a \end{bmatrix} \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} \cos\theta & \mp\sin\theta & 0 \\ \sin\theta & \pm\cos\theta & 0 \\ 0 & 0 & s_b \end{bmatrix}, \quad \text{with} \quad \theta \in [0,2\pi)$$

representing the parameter for this SVD family, $|s_a| = 1$ and $|s_b| = 1$.

Proof: Suppose another SVD for $\boldsymbol{E}$ is expressed as $\boldsymbol{E} = (\boldsymbol{UA})\,\mathrm{diag}(1,1,0)(\boldsymbol{VB})^T$, and then it is easily deduced from the definition of SVD that $\boldsymbol{AA}^T = \boldsymbol{I}$ and $\boldsymbol{BB}^T = \boldsymbol{I}$. In addition, compare $\boldsymbol{E} = (\boldsymbol{UA})\,\mathrm{diag}(1,1,0)(\boldsymbol{VB})^T$ with the original SVD $\boldsymbol{E} = \boldsymbol{U}\,\mathrm{diag}(1,1,0)\boldsymbol{V}^T$, and then we find that the equation $\boldsymbol{A}\,\mathrm{diag}(1,1,0)\boldsymbol{B}^T = \mathrm{diag}(1,1,0)$ holds.

Denoting $\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ and $\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$, we can obtain the following equivalence after several matrix manipulations:

$$\begin{bmatrix} a_{11}b_{11}+a_{12}b_{12} & a_{11}b_{21}+a_{12}b_{22} & a_{11}b_{31}+a_{12}b_{32} \\ a_{21}b_{11}+a_{22}b_{12} & a_{21}b_{21}+a_{22}b_{22} & a_{21}b_{31}+a_{22}b_{32} \\ a_{31}b_{11}+a_{32}b_{12} & a_{31}b_{21}+a_{32}b_{22} & a_{31}b_{31}+a_{32}b_{32} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4-14}$$

The three homogeneous linear equations deduced from the last column of both sides lead to $b_{31} = 0$ and $b_{32} = 0$. Similarly, we obtain $a_{31} = 0$ and $a_{32} = 0$. Applying the fact that both $\boldsymbol{A}$ and $\boldsymbol{B}$ are unit orthogonal matrices, we can easily deduce that

$b_{33} = \pm 1$, $a_{33} = \pm 1$, $b_{13} = 0$ and $b_{23} = 0$, and $a_{13} = 0$ and $a_{23} = 0$. Hence, we obtain an order-reduced form of Equation (4-14):

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}^T = \boldsymbol{I}, \tag{4-15}$$

where $\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}$ and $\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & 0 \\ b_{21} & b_{22} & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}$.

Applying again the fact that both $\boldsymbol{A}$ and $\boldsymbol{B}$ are unit orthogonal matrices, we conclude that both $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and $\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$ are unit orthogonal matrices, and furthermore, $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$, according to Equation (4-15).

Based on above statements, we conclude that the only possible form for $\boldsymbol{A}$ and $\boldsymbol{B}$ is $\begin{bmatrix} \cos\theta & \mp\sin\theta & 0 \\ \sin\theta & \pm\cos\theta & 0 \\ 0 & 0 & s \end{bmatrix}$, where $\theta = [0, 2\pi)$ representing the parameter for the family of SVDs for $\boldsymbol{E}$, and $|s| = 1$.

This completes the proof.

Obviously the eight feasible SVDs for $\boldsymbol{E}$ that are described in [74] are nothing but special cases of Equation (4-13).

Our next task is to prove that although there exist multiple SVDs for $\boldsymbol{E}$, we can obtain only four possible solutions to the camera relative motion, which have already been discussed in [21] and [73].

For the original SVD $\boldsymbol{E} = \boldsymbol{U}\,\mathrm{diag}(1,1,0)\boldsymbol{V}^T$, the two solutions to $\boldsymbol{R}$ are

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^T \text{ or } \boldsymbol{R} = \boldsymbol{U}\boldsymbol{W}^T\boldsymbol{V}^T, \tag{4-16}$$

where $\boldsymbol{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & \det(\boldsymbol{U})\det(\boldsymbol{V}) \end{bmatrix}$.

The two solutions to $\mathbf{t}$ are

$$\mathbf{t} = \pm\mathbf{u}_3, \tag{4-17}$$

where $\mathbf{u}_3$ is the last column of $\boldsymbol{U}$.

If we consider the generalized SVD for $\boldsymbol{E}$, which is expressed as Equation (4-13), Equation (4-16) is changed to

$$\boldsymbol{R} = (\boldsymbol{U}\boldsymbol{A})\boldsymbol{W}(\boldsymbol{V}\boldsymbol{B})^T \text{ or } \boldsymbol{R} = (\boldsymbol{U}\boldsymbol{A})\boldsymbol{W}^T(\boldsymbol{V}\boldsymbol{B})^T, \tag{4-18}$$

where $\boldsymbol{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & \det(\boldsymbol{U})\det(\boldsymbol{V})\det(\boldsymbol{A})\det(\boldsymbol{B}) \end{bmatrix}$.

Equation (4-17) keeps unchanged.

After several matrix manipulations, we find that the two solutions to $\boldsymbol{R}$ calculated from Equation (4-18) are identical to those calculated from Equation (4-16). Therefore, although the SVD for $\boldsymbol{E}$ is feasible, the derived camera relative motion is limited to four possible solutions.

4.5 Summary

We addressed the problem of recovering camera relative motion from the *essential matrix* ($\boldsymbol{E}$). We presented a new proof that there are only four possible solutions to the

camera relative motion, based on the analysis of the geometry existing among $\boldsymbol{E}$, $\boldsymbol{R}$ and $\mathbf{t}$. Our conclusion is consistent with Tsai's, Horn's and Hartley's statements. In practice, the final solution can be uniquely determined based on the fact that the 3-D object must be in front of both cameras that are viewing that object.

Furthermore, Horn's method provides a closed-form solution to the camera motion. But this method cannot be directly used in practice because it is very sensitive to either the noise in image point coordinates or the offset in camera intrinsic parameters. On the other side, Tsai's method is robust because it is based on the SVD for $\boldsymbol{E}$. But the accuracy of the final result is heavily affected by the accuracy of the SVD procedure.

In addition, we provided a generalized SVD-based proof for the four possible solutions to the camera relative motion. Differing from Hartley's proof, we discussed the feasibility of SVDs for $\boldsymbol{E}$ and provided the only possible form of the feasible SVDs. Then we verified that the multiple SVDs for $\boldsymbol{E}$ lead to only four possible solutions to the camera relative motion.

5 Multi-view Image Coding in 3-D Space

Multi-view image coding has been increasingly attracting attention for its crucial role in various applications, i.e., image-based rendering, medical volumetric data compression, and virtual reality. These applications have the common goal of handling a large number of highly correlated 2-D images. This common goal also highlights the importance of multi-view image coding that is based on the employment of 3-D scene information. Consequently, existing multi-view coding schemes using 3-D scene information have shown significant improvement of the compression ratio as well as the rendering quality compared with the conventional coding schemes employing only simple extension of 2-D compression techniques.

However, there are still some aspects of these coding schemes that can be improved, which have motivated our studies in combining the automatic 3-D scene reconstruction with multi-view image coding. We propose a multi-view coding framework that operates directly in 3-D space and is based on automatic 3-D scene reconstruction. In this chapter, we first describe the framework of our proposed multi-view coding system and then provide details regarding technical solutions of the proposed scheme.

5.1 Framework for Multi-view Image Coding in 3-D Space

Studies in [29][28][25][26][27] show that multi-view image coding schemes using 3-D scene geometry information greatly improve the encoding efficiency, decoding speed

and the rendering quality, compared with the conventional coding schemes employing only simple extension of 2-D compression. However, in the 3-D geometry-based multi-view coding, the scene geometry information and the image data must be encoded separately. This requirement limits the flexibility of the coding scheme, since the decoding of the 3-D geometry information must be completed prior to the decoding of the image (texture) data. Furthermore, it is rather challenging to optimize the rate-distortion (RD) performance of the coding scheme if we vary the bit rate of the encoded 3-D geometry information as well as the bit rate of the encoded image (texture) data simultaneously.

The above constraint of existing multi-view coding schemes motivates the idea of directly encoding the constructed 3-D scene model that represents all the available multiple images in 3-D space. The framework of our proposed 3-D space multi-view coding is shown in Figure 5-1.

Similar to traditional source coding systems, our proposed multi-view coding system includes two basic parts: encoding and decoding. Generally speaking, the decoding part is an inverse procedure of the encoding part, except for the re-projection instead of the volumetric 3-D reconstruction. The scenarios linked by dotted lines represent optional residual coding.

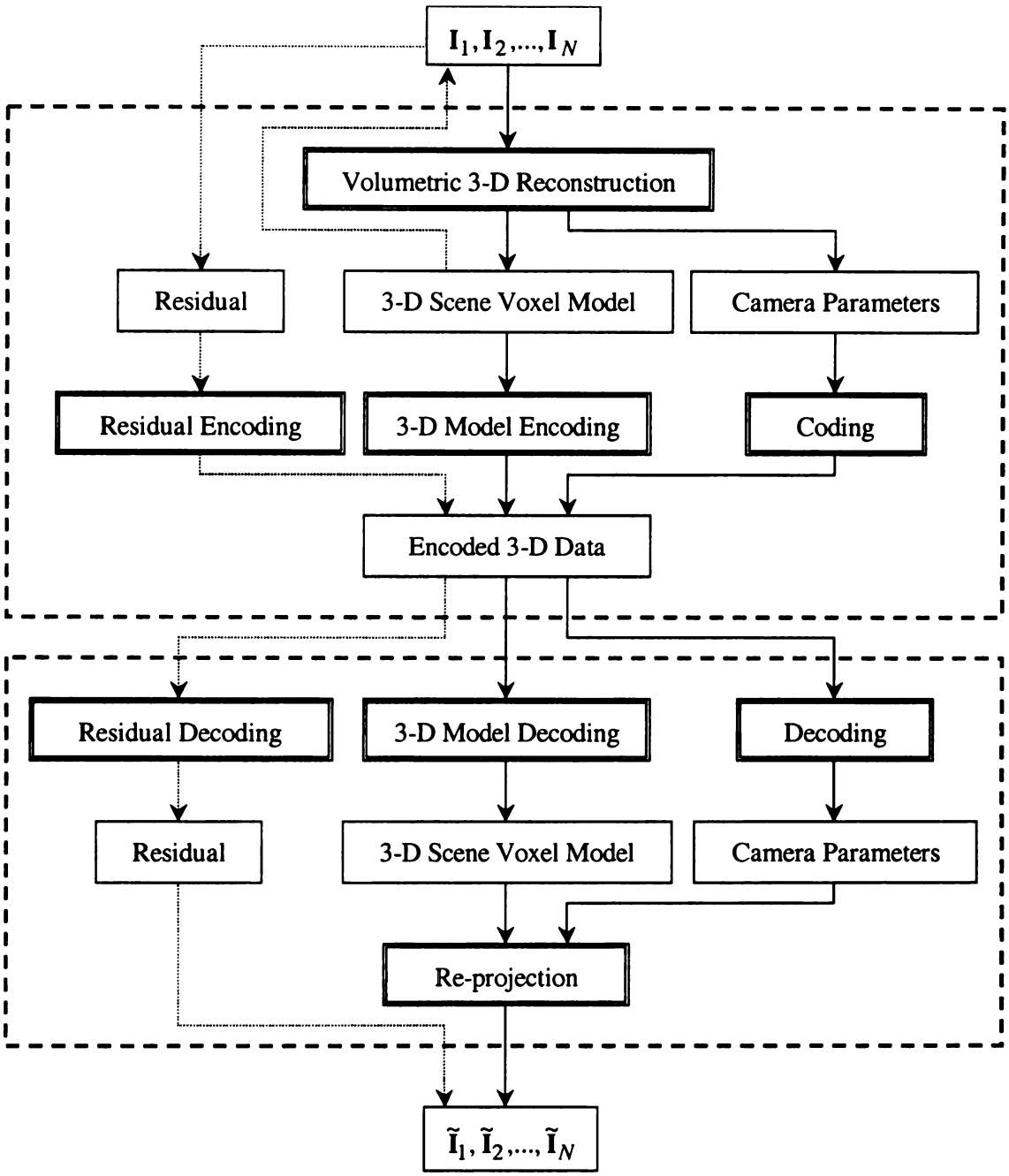


Figure 5-1 The proposed framework for multi-view image coding in 3-D space. The double-lined box represents a system function block, while the single-lined box represents: input to the system, output of the system and intermediate output. The boxes connected by dotted arrowed-line represent optional procedures.

The encoding part consists of three necessary blocks and one optional block:

1) *Volumetric 3-D reconstruction*: The input to the encoding part is a set of images, denoted by $\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_N$. The N images are then fed into the block of volumetric 3-D reconstruction to obtain the 3-D scene model. We state here that in the general case, the available multiple images are uncalibrated and hence the camera parameters are derived from a 3-D reconstruction process, as shown in Figure 5-1 (readers are referred to Section 2.1.6 for details about the 3-D scene reconstruction from multiple uncalibrated images). When the multiple images are calibrated, both the camera intrinsic and extrinsic parameters are known and thus the procedure for 3-D reconstruction can be greatly simplified. This step is one of the crucial steps in our multi-view coding system, since the quality of the reconstructed 3-D voxel model impacts the coding efficiency of the 3-D data coding and the optional residual coding. Details of the volumetric 3-D reconstruction are presented in Section 5.2.

2) *3-D Model Encoding*: This step is another crucial step in our proposed coding system. Generally speaking, we aim at encoding the 3-D scene model that represents the available multiple images in 3-D space. We propose two possible approaches: (1) employing the H.264 video coding standard for compressing the 3-D voxel model as a sequence of highly correlated 2-D images; and (2) 3-D wavelet-based SPIHT coding scheme. Currently, we have obtained experimental results using the H.264 and 3-D SPIHT. Details of the 3-D model coding are presented in Section 5.3.

3) *Coding of Camera Parameters*: The obtained camera intrinsic and extrinsic parameters are quantized for encoding purposes. High-precision quantization and

lossless coding are expected here since the accuracy of camera parameters, which are the link between the real-world and the 2-D images, crucially impacts the accuracy of the 3-D scene model and the quality of the recovered and rendered images. This step is straightforward and less important than 1), 2) and 4), because the encoded data size of the camera parameters is trivial compared with the encoded data size of the 3-D coding and the residual coding.

4) *Residual coding*: This is an optional procedure in our system. However, we anticipate that the residual coding will be required for high-quality applications. The reason for this requirement is as follows. The recovery of the input images is achieved by re-projecting the 3-D scene model to the image planes in terms of the camera parameters. Ideally the reconstructed images are identical to the original images; in practice, such perfect reconstruction will never happen. First, in many applications the provided images are uncalibrated so that the reconstructed 3-D scene model contains errors. Second, even in the case of calibrated images, the discretization of the 3-D space in reconstructing the 3-D scene inevitably introduces quantization errors in the obtained 3-D voxel model. Therefore, in the case that the quality of reconstructed images from re-projection of the 3-D scene model does not meet the requirement of the specific application, the residuals between the original images and re-projected images are computed. Then *quantization and entropy encoding*, which are commonly used steps in image/video coding techniques, are employed to encode the residual data. An important aspect of the residual coding is

that the residual images may also have some degree of correlation; hence, it might be more efficient to code them jointly. This aspect is discussed in detail in Section 5.4.

The final encoded 3-D data includes the encoded 3-D data, the encoded camera parameters and the (optional) encoded residual data.

The decoding part is basically an inverse procedure of the encoding part, except that the corresponding block of the 3-D reconstruction in the encoding part is the re-projection process. The target of re-projection is to recover the images from the decoded 3-D scene voxel model and the camera parameters. The procedure of re-projection is exactly the same as the one for obtaining the residual data, which we described in the residual coding. In fact, it is a relatively straightforward procedure using Equation (2-1).

5.2 Volumetric 3-D Reconstruction

As stated in Section 5.1, the volumetric 3-D reconstruction is a key step in the multi-view image coding scheme. The goal of volumetric 3-D reconstruction is to obtain a 3-D scene voxel model from the multiple images to be coded. In general, the images are uncalibrated and the whole procedure is called automatic 3-D scene reconstruction, of which a possible framework is shown in Figure 2-2 (readers are referred to Section 2.1.6 for more details). In this section, we concentrate on the problem of establishing the 3-D volumetric model from multiple calibrated images. We provide an algorithm for

volumetric 3-D reconstruction, which is a modified version of Eisert's approach [34]. In this approach, all operations are performed on voxels, which are the basic elements of the 3-D object, and not on image pixels, where multiple of them are representatives of the same (one) 3-D voxel. Therefore, we avoid the search for corresponding points and the fusion of incomplete depth estimates.

A set of assumptions associated to the 3-D scene voxel model is made before we describe the detailed algorithm for volumetric 3-D reconstruction. First, we assume that the considered images were captured under the perfect perspective projection of a pinhole camera, with the mapping of the 3-D world scene from the 3-D world coordinate system to the 2-D retinal coordinate system expressed by Equation (2-1). This assumption indicates that the considered images contain no aberrations caused by optical effects, such as radial distortion, spherical aberration or chromatic aberration. Second, we assume that the light condition is the same around the considered 3-D scene/object. Third, we assume that the considered 3-D scene/object is made from materials of constant refractive index and isotropic reflection property. These two assumptions indicate that the luminance and chrominance of the same part of the 3-D scene displayed in different images were not impacted by the different camera viewing positions. Fourth, we are interested in representing and constructing only the surface of the considered 3-D scene/object. The voxels inside the considered 3-D scene/object, which are invisible to us, are not considered in our application. This assumption makes the proposed 3-D scene model different from the medical volumetric data, of which the 3-D scene/object is

regarded as transparent. Fifth, we do not take on consideration the physical shape of the voxels. Hence, we can assume that each voxel is associated with a single piece of color information. Last, we make no assumptions for the topology of the considered 3-D scene/object. The surface of the 3-D scene/object can be either continuous or incontinuous.

Similar to Eisert's approach, our approach proceeds in four successive steps:

1) volume initialization;

2) color hypothesis generation for all voxels from all available camera views;

3) consistency check and hypothesis elimination considering all the views;

4) determination of the best color hypothesis for the remaining surface voxels.

The remainder of this section is organized as follows. Sections 5.2.1 to 5.2.4 depict the basic algorithms in our approach, which are similar to those of Eisert's approach. In Section 5.2.5, we present our first 3-D voxel model that is obtained following the approach depicted in Sections 5.2.1-5.2.4 (we refer to this 3-D voxel model as VM3a). Section 5.2.6 and 5.2.7 describe our modifications to step 2) and 3) to further remove the voxels outside the considered object/scene and the voxels inside the considered object/scene. The result of these modifications is a second 3-D voxel model (we refer to this 3-D voxel model as VM5b). Section 5.2.8 describes a proposed modification to a key formula that is used in the color hypothesis and consistency check in terms of physiological characteristics of the human visual system. Consequently, this results in a third 3-D voxel model that is presented in Section 5.2.9 (we refer to this 3-D voxel model

as VM5c), which is the best among the three 3-D scene models that we have developed in terms of achieving better coding efficiency due to improved 3-D voxel models. In section 5.2.10, we provide some experimental results for synthetic image generation from the obtained 3-D scene voxel model. Finally, we discuss the influence of the errors of camera intrinsic parameters on volumetric 3-D reconstruction.

5.2.1 Volume Initialization

The first step is to define a volume in the reference coordinate system that encloses the 3-D object to be reconstructed. The volume extensions are determined from the camera calibration information and its surface represents a conservative bounding box of the object. In practice, we search for the largest values of the extensions where the camera projects the eight vertices of the bounding box onto the range of the image plane.

The obtained volume is discretized along all the three dimensions to form an array of voxels with associated color, where the position of each voxel in the 3-D space is denoted by its indices (l,m,n). As stated in Section 5.1, the discretization of the considered/bounded 3-D space in reconstructing the 3-D voxel model is a source of errors in re-projected images. In general, the lower the resolution of the discretization is, the larger the errors in re-projected images may occur. To reduce this kind of error, we may want to select a high resolution level in discretizing the considered/bounded 3-D space. However, even a small increase of the resolution will result in a large increase of the data

size of the 3-D voxel model. Hence, in practice a trade-off between the resolution of the 3-D voxel model and the data size of it is required.

The projection from a 3-D point $[x, y, z]^T$ to a pixel $[u_i, v_i]^T$ in the i–th view ($i = 1,2,\cdots,N$, with N the number of all available images) is expressed as

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} [\boldsymbol{R}_i \; \boldsymbol{t}_i] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{5-1}$$

according to Equation (2-1), (2-2) and (2-4). In Equation (5-1), α_u and α_v represent the focal lengths in pixels along the vertical and horizontal direction respectively, (u_0, v_0) are the coordinates of the principal point, and $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$ represent the rotation matrix and translation vector of the i–th view, respectively. Thus, we obtain

$$\begin{cases} u_i = \alpha_u \dfrac{r_{11}x + r_{12}y + r_{13}z + t_x}{r_{31}x + r_{32}y + r_{33}z + t_z} + u_0 \\ v_i = \alpha_v \dfrac{r_{21}x + r_{22}y + r_{23}z + t_y}{r_{31}x + r_{32}y + r_{33}z + t_z} + v_0 \end{cases}, \tag{5-2}$$

where $\boldsymbol{R}_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ and $\boldsymbol{t}_i = [t_x, t_y, t_z]^T$.

5.2.2 Hypothesis Generation

In this step, a set of color hypotheses is assigned to each voxel of the initial volume. A hypothesis H_{lmn}^k for a voxel V_{lmn} is

$$H_{lmn}^k = [r(u_i, v_i), g(u_i, v_i), b(u_i, v_i)], \tag{5-3}$$

where $[u_i, v_i]^T$ represents the pixel position of the perspective projection of the voxel center $[x_{l0}, y_{m0}, z_{n0}]^T$ into the i-th image view ($i \in [1,2,\cdots,N]$), and $r(\bullet)$, $g(\bullet)$ and $b(\bullet)$ represent the red, green and blue information. The relationship between $[u_i, v_i]^T$ and $[x_{l0}, y_{m0}, z_{n0}]^T$ is expressed in Equation (5-1) and (5-2).

The hypothesis H_{lmn}^k is associated with voxel V_{lmn} if the projection of V_{lmn} into at least one other camera view $j \neq i, j = 1,2,\cdots,N$ leads to an absolute color difference that is less than a predefined threshold Θ

$$\left|r(u_j, v_j) - r(u_i, v_i)\right| + \left|g(u_j, v_j) - g(u_i, v_i)\right| + \left|b(u_j, v_j) - b(u_i, v_i)\right| < \Theta. \qquad (5\text{-}4)$$

For each view i, $i = 1,2,\cdots,N$, the hypothesis H_{lmn}^k that satisfies Equation (5-4) is stored with the color taken from view i according to Equation (5-3).

At this stage, we have no knowledge of the 3-D object geometry and cannot determine whether a voxel is visible or not. We therefore need to remove those hypotheses that do not correspond to the correct color of the considered object surface.

5.2.3 Consistency Check and Hypothesis Elimination

This step is performed iteratively over all the camera views. The basic idea of this step is to remove those hypotheses that are extracted from two or more consistent views but lead to contradictions with other views where the considered voxel is also visible. We start from the surface of the predefined volume and remove voxels until the correct shape of the 3-D object is recovered.

For each camera view, we determine the currently visible voxels and compare all associated hypotheses with the corresponding pixel color at the position given by Equation (5-1), according to the inequality constraint in Equation (5-4). If all hypotheses for one voxel are removed, this voxel is regarded as transparent. Consequently, we iterate several times over all available views until no more hypotheses are removed and the number of transparent voxels converges. The remaining nontransparent voxels constitute the volumetric description of the 3-D object.

A key problem in the hypothesis test and elimination is to determine the visible surface voxel from the current camera view. In practice, processing the voxels in order of their visibility for a certain view can be achieved by volume index permutation in combination with a decision of whether to index the voxels in increasing or decreasing order for each dimension. This leads to a total of 48 cases of volume traversal.

The algorithm for identifying the volume traversal order for a particular view works as follows. We first determine the dot product of the camera optical axis and the rotated six surface normals of the bounding box in the view *i*:

$$p_k = O \bullet (\boldsymbol{R}_i \mathbf{n}_k), \tag{5-5}$$

where $O = [0,0,-1]^T$ represents the camera optical axis, $\boldsymbol{R}_i$ is the rotation matrix of the *i*-th view, and $\mathbf{n}_k, k = 1,2,\cdots,6$ represents one of the six surface normal of the bounding box: $[0,0,1]^T$, $[0,0,-1]^T$, $[0,1,0]^T$, $[0,-1,0]^T$, $[1,0,0]^T$ and $[-1,0,0]^T$.

In fact, the outcome of the dot product in Equation (5-5) is nothing but the cosine of the angle between the optical axis and one of the six surface normals of the bounding box.

The largest three values of the dot product in combination with their order identify one of the 48 volume traversal cases.

5.2.4 Determination of the Voxel Color

The final step is to determine the best color for the obtained 3-D surface voxels. For each voxel V_{lmn}, we first determine all views where this voxel is visible (each view I_s where this voxel is visible must be included in the updated hypothesis set of V_{lmn}—H^k_{lmn}). Then we select the hypothesis H^{opt}_{lmn}, which leads to:

$$H^{opt}_{lmn} = \min_{\forall H^k_{lmn}} \left\{ \underset{\forall I_s}{\text{median}} \left\| H^k_{lmn} - (r(u_s, v_s), g(u_s, v_s), b(u_s, v_s)) \right\|_1 \right\}, \qquad (5\text{-}6)$$

yielding the projection into the original views with the smallest median color error according to the l_1-norm.

We now have established a 3-D volumetric model from multiple calibrated images, which consists of the indices of the voxels in three dimensions as well as the associated color information.

In addition, we discuss in brief the complexity of the basic approach for the volumetric 3-D reconstruction. We denote the resolution of the 3-D voxel model in three dimensions as D_x, D_y, and D_z, The number of all the available images is denoted as N. The algorithm complexity of step 2, 3 and 4 can be expressed as follows (the computational duty of step1 can be neglected.).

Hypothesis generation: $O_2(D_x, D_y, D_z, N) \propto D_x D_y D_z N(N-1)/2$;

Consistency check and hypothesis elimination: $O_3(D_x, D_y, D_z, N) \propto D_x D_y D_z N$;

Determination of the voxel color: $O_4(D_x, D_y, D_z, N) \propto D_x D_y D_z$.

The final algorithm complexity is the summation of the above three partial algorithm complexity: $O(D_x, D_y, D_z, N) \propto D_x D_y D_z (N(N+1)/2+1)$.

We have observed that the heaviest computational duty occurs in step2 (hypothesis generation). Furthermore, the computational duty of step 2 increases while the number of all the available images increases. Another observation of the algorithm complexity is that the image size does not explicitly impact the computational duty. However, it impacts the algorithm complexity in the aspect that it influences the resolution of the 3-D voxel model.

5.2.5 The First 3-D Voxel Model—VM3a

Below, we provide experimental results for the volumetric 3-D reconstruction described above. This corresponds to our first variation of the 3-D voxel models that we have developed, i.e., VM3a. The test image sequence, known as the *cup* sequence, was downloaded from [75]. Four sample images are shown in [34]. This image sequence consists of a total of 14 images (288×352) with known camera calibration information (camera intrinsic parameters, camera relative rotation matrix and camera relative translation vector). The original images 3, 6, 7, 9, 11, 14 are shown in Figure 5-2. (Images in this dissertation are presented in color.)

Original Image 3

Original Image 6

Original Image 7

Original Image 9

Original Image 11

Original Image 14

Figure 5-2 The original images 3, 6, 7, 9, 11, and 14 of the *cup* sequence.

The volume was initialized using the algorithm depicted in Section 5.2.1. The voxel resolution for the initial volume is chosen as $160\times160\times160$. The obtained 3-D voxel model, named "VM3a", was obtained using the algorithms depicted in Section 5.2.2 to 5.2.4. VM3a contains 146,005 voxels. We re-projected the VM3a back to image planes for the same camera viewing positions of all the original images. The average *Peak Signal-to-Noise-Ratio* (PSNR) of the reconstructed 14 images is 16.73 dB. We show in Figure 5-3 the reconstructed images corresponding to the images in Figure 5-2.

The calculation of the PSNR (dB) for each reconstructed image is shown as below:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right), \tag{5-7}$$

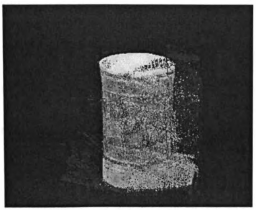
where

$$MSE = \frac{\sum_{u=1}^{H}\sum_{v=1}^{W}((\left|\hat{r}(u,v)-r(u,v)\right|+\left|\hat{g}(u,v)-g(u,v)\right|+\left|\hat{b}(u,v)-b(u,v)\right|)/3)^2}{H\times W}. \tag{5-8}$$

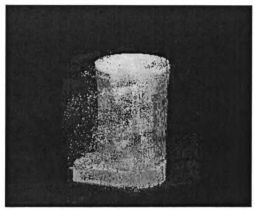
In Equation (5-8), H and W represent the height and width of the bounding frame of the considered image respectively (H and W are different from the vertical and horizontal resolution of the image), while $(r(\bullet),g(\bullet),b(\bullet))$ and $(\hat{r}(\bullet),\hat{g}(\bullet),\hat{b}(\bullet))$ represent the color information of a given pixel within the original image and reconstructed image respectively.

Rec. Image 3 (PSNR = 18.09 dB)

Rec. Image 6 (PSNR = 17.28 dB)



Rec. Image 7 (PSNR = 14.95 dB)



Rec. Image 9 (PSNR = 14.65 dB)

Rec. Image 11 (PSNR = 17.75 dB)

Rec. Image 14 (PSNR = 17.96 dB)

Figure 5-3 The reconstructed images resulting from re-projecting the VM3a back to image planes for the same camera viewing positions as in Figure 5-2, as well as the value of PSNR for each reconstructed image.

The re-projection of the 3-D voxel model is accomplished as follows. First we re-project each voxel to all the available images. Then for each pixel in each image that is a result of the re-projection of one or more voxels, we assign it the color value of the associated voxel that has the smallest depth to the considered image plane.

The experimental results for volumetric 3-D reconstruction in Figure 5-3 support the key idea behind our proposed multi-view coding in 3-D space — the obtained 3-D voxel model as well as the known camera calibration information can represent and reconstruct the original image sequence (at least coarsely). We also observed that the reconstruction results for image 3, 6, 11, and 14 are better than those for image 7 and 9. This observation is not surprising because image 3, 6, 11 and 14 belong to a group of "dense" camera viewing positions, which refers to a large number of camera viewing positions gathering within certain spatial volume, while image 7 and 9 belong to a group of "sparse" camera viewing positions. Hence, increasing the number of camera viewing positions and distributing the cameras equally within given space will improve the 3-D voxel model.

However, from both the reconstructed images and their calculated PSNR we can see that the current reconstruction results based on VM3a are not good enough for many coding applications. As stated in Section 5.1, the quality of the reconstructed 3-D voxel model impacts the coding efficiency of the 3-D data coding and the optional residual coding. In the following four subsections, we describe in detail two improvements of the basic approach for the volumetric 3-D reconstruction and the corresponding two new 3-D voxel models.

5.2.6 Enhanced Hypothesis Generation and Consistency Check

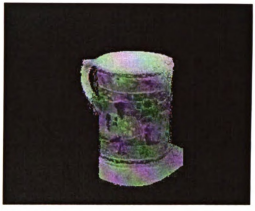
In the second step of our approach for volumetric 3-D reconstruction, hypotheses for each voxel within the bounded 3-D space are generated according to Equation (5-4). In the current algorithm, a voxel is determined to be "valid" for further consistency check if its associated hypotheses contain the color values from at least two different images. However, many voxels that are not on the considered object surface are determined to be "valid" according to such a rule, since it is considerably possible that two projected pixels on two different image planes for the same voxel that is not at all on the considered object surface satisfy Equation (5-4). Unfortunately, experiments show that not all these voxels can be detected and eliminated in the consistency-check step, especially those that are outside the object surface. In fact, the "noisy" pixels in the reconstructed images in Figure 5-3 result from the "pseudo-valid" voxels. To reduce the possibility of the "pseudo-valid" voxels, we increase the number of hypotheses of a "valid" voxel from 2 to K with K is an integer larger than 2 but no larger than the maximum number of all available image pairs. In particular, in the hypothesis-generation step, we claim a voxel to be "valid" for further consistency check only when its hypothesis set contains at least K hypotheses that come from K different images. Factors that may impact the selection of the value of K include the total number of the considered images, the camera parameters, the histogram of the considered images, and the resolution of the bounded 3-D space. In our approach, we selected $K = 3$. By using this method, we can remove a significant number of the voxels that are outside the considered object.

Another problem with the current approach for volumetric 3-D reconstruction is that there is no special processing in the consistency check for the occluded voxels; that is, voxels that are inside the considered object. This problem does not impact the quality of the reconstructed images but makes the 3-D voxel model contain a considerable number of useless voxels. This impact is undesirable for our final goal of data compression. To solve this problem, in the re-projection of the 3-D voxel model (readers are referred to Section 5.2.5 for more details), we remove those voxels in the 3-D model that are never assigned to image pixels. By using this method, we can remove a large number of occluded voxels from the 3-D model and hence improve the coding efficiency of our multi-view image coding system.

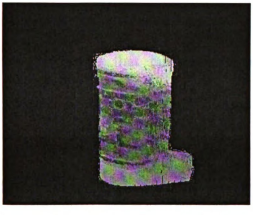
5.2.7 The Second 3-D Voxel Model—VM5b

By combining the modification described in Section 5.2.6 to our four-step approach for volumetric 3-D reconstruction, we obtained another 3-D voxel model for the same image sequence depicted in Section 5.2.5, named "VM5b". The VM5b contains 86,064 voxels, which is much more efficient than the VM3a (146,005 voxels). The average PSNR of the reconstructed 14 images is 19.48 dB with the gain of around 3dB over the average PSNR of the reconstructed images based on the VM3a. We show in Figure 5-4 the reconstructed images corresponding to the images in Figure 5-2.

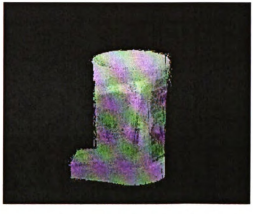
Rec. Image 3 (PSNR = 20.71 dB)



Rec. Image 6 (PSNR = 19.14 dB)



Rec. Image 7 (PSNR = 17.25 dB)



Rec. Image 9 (PSNR = 16.93 dB)

Rec. Image 11 (PSNR = 19.40 dB)

Rec. Image 14 (PSNR = 21.62 dB)

Figure 5-4 The reconstructed images resulting from re-projecting the VM5b back to image planes for the same camera viewing positions as in Figure 5-2, as well as the value of PSNR for each reconstructed image.

It is clear that the quality of the reconstructed images is much higher than that of the reconstructed images shown in Figure 5-3. Many "noisy" pixels outside the true surface of the cup in Figure 5-3 were removed and the surface of the cup looks smoother than that in Figure 5-3. In addition, the size of the VM5b is only around 59 percent of that of the VM3a. This fact indicates that many voxels that were inside the cup have been eliminated from the 3-D voxel model. In brief, both statistical data on the 3-D model VM5b and the reconstructed images, and the observations of the reconstructed images in Figure 5-4 show that our modification depicted in Section 5.2.6 improves the performance of the volumetric 3-D reconstruction.

5.2.8 A New Measurement for Pixel Color Information Difference

Equation (5-4) plays an important role in hypothesis generation and consistency check. This equation provides a mechanism for measuring the difference of the color information between two pixels by summing up the absolute difference of red value, green value, and blue value between the considered two pixels. However, the objective result of pixel color information difference using this measurement may not match the subjective result based on observations of human beings, since the RGB color system does not match physiological characteristics of the human visual system, i.e., the human eye is more sensitive to changes in brightness than to chromaticity changes. This character of the human visual system has also led to the YUV (one luminance and two

chrominance components) color image format in many of the standardized video coding schemes.

Therefore, we have modified the measurement for pixel color information difference based on the Y component (luminance). The conversion from RGB to Y is given as below:

$$Y = 0.299 \times r + 0.587 \times g + 0.114 \times b . \tag{5-9}$$

Equation (5-9) shows that the three components r, g, and b contribute quite different to the luminance, i.e., the green component impacts the luminance the most (it is why we say that the human eyes are more sensitive to green color than others.). Hence, we modified Equation (5-4) to a weighted summation of the absolute difference of the r, g, and b between two pixels, as shown below:

$$\begin{aligned} & 0.299 \times \left| r(u_j, v_j) - r(u_i, v_i) \right| + 0.587 \times \left| g(u_j, v_j) - g(u_i, v_i) \right| \\ & + 0.114 \times \left| b(u_j, v_j) - b(u_i, v_i) \right| < \Theta \end{aligned}, \tag{5-10}$$

where (u_i, v_i) and (u_j, v_j) represent the coordinates of two pixels and Θ is a pre-determined value.

5.2.9 The Third 3-D Voxel Model—VM5c

By combining the modification described in Section 5.2.6 and Section 5.2.8 to our four-step approach for volumetric 3-D reconstruction, we obtained the third 3-D voxel model for the same image sequence depicted in Section 5.2.5, named "VM5c". The VM5c contains 82,622 voxels. The average PSNR of the reconstructed 14 images is

20.26 dB. We show in Table 5-1 the comparison of data size and the average PSNR of reconstructed images among the obtained three 3-D voxel models.

Table 5-1 Comparison of data size and the average PSNR of reconstructed images among the obtained three 3-D voxel models—VM3a, VM5b and VM5c.

	VM3a	VM5b	VM5c
Voxel Number	146,005	86,064	82,622
Average PSNR of Rec. Images (dB)	16.73	19.48	20.26

Table 5-1 shows that the VM5c performs the best among the three 3-D voxel models according to both the model size and the quality of reconstructed images.

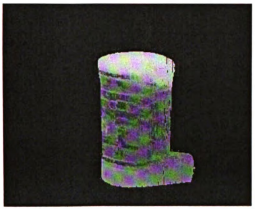
We also show in Figure 5-5 the reconstructed images corresponding to the images in Figure 5-2.

5.2.10Synthetic Image Generation from VM5c

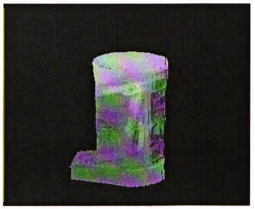
As shown in Section 5.2.5, 5.2.7 and 5.2.9, the original images can be reconstructed from the developed 3-D scene voxel model according to the corresponding camera calibration parameters. Moreover, synthetic images can even be generated from the obtained 3-D voxel model for new camera viewing positions that are different from those for the original images. Figure 5-6 shows some synthetic images generated from VM5c.

Rec. Image 3 (PSNR = 21.78 dB)

Rec. Image 6 (PSNR = 19.84 dB)



Rec. Image 7 (PSNR = 17.79 dB)



Rec. Image 9 (PSNR = 17.51 dB)

Rec. Image 11 (PSNR = 20.25 dB)

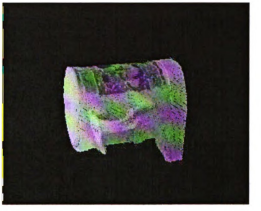
Rec. Image 14 (22.58 dB)

Figure 5-5 The reconstructed images resulting from re-projecting the VM5c back to image planes for the same camera viewing positions as in Figure 5-2, as well as the value of PSNR for each reconstructed image.

Synthetic Image 5

Synthetic Image 8

Synthetic Image 11



Synthetic Image 15

Synthetic Image 16

Synthetic Image 17

Figure 5-6 The generated synthetic images resulting from re-projecting the VM5c back to image planes for new camera viewing positions that are different from those for the original images.

In Figure 5-6, the camera rotation parameters for the synthetic image 5 are very close to those for the original image 5, while the camera translation parameters are quite different. The camera viewing position for the synthetic image 8 can be regarded as "between" that for the original images 8 and 9. Similarly, the camera viewing position for the synthetic image 8 can be regarded as "between" that for the original images 11 and 12. The synthetic images 15, 16 and 17 are generated for randomly selected camera viewing positions. In terms of the subjective observation for the synthetic images in Figure 5-6, the synthetic image 5 and 11 are of good quality while the others are not. The reason for the good quality of synthetic images 5 and 11 is that the camera viewing positions for them are close or related to the camera viewing positions (for the original images) that correspond to the well-performing part of the obtained 3-D voxel model VM5c. The quality of the synthetic images 8 is relatively poor since its camera viewing position corresponds to the poor quality of the obtained 3-D voxel model VM5c. The quality of the synthetic image 15, 16 and 17 is not as good as that of the synthetic image 5 and 8, since their camera viewing positions are relatively far from those of the original images. In brief, the quality of synthetic images is mainly determined by the performance of the developed 3-D voxel model. In addition, the selected new camera viewing position also impacts the quality of the corresponding synthetic image.

The capability of the developed 3-D voxel model to generate synthetic images enables the proposed multi-view image coding system to be employed in many nowadays

multimedia applications, e.g., virtual reality, video conferencing system and distance education.

5.2.11 Influence of the Camera Intrinsic Parameters on Volumetric 3-D Reconstruction

Section 5.2.1 to 5.2.9 discussed our approach for volumetric 3-D reconstruction from calibrated images (with known intrinsic and extrinsic parameters). This section discusses the influence of inaccurate camera intrinsic parameters on the 3-D voxel model and image reconstruction.

We suppose that the camera calibration matrix $\begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ in Equation (5-1) is taken place by its inaccurate version $\begin{bmatrix} \alpha_u + \Delta\alpha_u & 0 & u_0 + \Delta u_0 \\ 0 & \alpha_v + \Delta\alpha_v & v_0 + \Delta v_0 \\ 0 & 0 & 1 \end{bmatrix}$, then for the same 3-D point $[x, y, z]^T$, the coordinates of the projected image pixel become

$$\begin{cases} u_i' = (\alpha_u + \Delta\alpha_u)\dfrac{r_{11}x + r_{12}y + r_{13}z + t_x}{r_{31}x + r_{32}y + r_{33}z + t_z} + (u_0 + \Delta u_0) \\ v_i' = (\alpha_v + \Delta\alpha_v)\dfrac{r_{21}x + r_{22}y + r_{23}z + t_y}{r_{31}x + r_{32}y + r_{33}z + t_z} + (v_0 + \Delta v_0) \end{cases} . \tag{5-11}$$

The difference between the correct and inaccurate coordinated of the projected image pixel is shown as below:

$$\begin{cases} \Delta u_i = u_i' - u_i = \Delta\alpha_u \dfrac{r_{11}x + r_{12}y + r_{13}z + t_x}{r_{31}x + r_{32}y + r_{33}z + t_z} + \Delta u_0 \\ \Delta v_i = v_i' - v_i = \Delta\alpha_v \dfrac{r_{21}x + r_{22}y + r_{23}z + t_y}{r_{31}x + r_{32}y + r_{33}z + t_z} + \Delta v_0 \end{cases} . \tag{5-12}$$

We make two conclusions from Equation (5-12): first, the shifting of the principal point from its true position causes a uniform shifting of the projected image pixels, no matter what the coordinates of the projecting 3-D point are. Hence, theoretically, the errors in the position of the principal point will not introduce any distortion of the obtained 3-D voxel model. Consequently, the reconstructed images will be the same as those from the 3-D voxel model that is obtained from the true intrinsic parameters; second, the errors of the coordinates of the projected image pixel caused by the inaccurate focal lengths are determined by not only the errors in the focal lengths, but the camera relative motion as well as the coordinates of the projecting 3-D point. Therefore, the errors in the focal lengths will significantly degenerate the volumetric 3-D reconstruction and result in a poor 3-D voxel model.

5.3 3-D Voxel Model Coding

Having obtained the 3-D voxel model using the method given in Section 5.2, we target encoding the 3-D scene model that represents in 3-D space the available multiple images. We propose two possible approaches: (1) 3-D wavelet-based SPIHT coding scheme; and (2) employing H.264 video coding standard with regarding the 3-D voxel model as a sequence of highly correlated 2-D images. The idea behind the proposed

approaches (1) and (2) is that a video stream sequence, volumetric data (e.g., a set of medical images), and our 3-D voxel model are common in (a) they all can be regarded as three dimensional data, and (b) correlations exist along all the three dimensions. We will discuss in detail the 3-D data coding using 3-D SPIHT and H.264.

5.3.1 Label Coding for 3-D Voxel Model

Before we start the coding of the 3-D voxel model, it is necessary for us to consider the characteristics of the 3-D voxel model for any needed modification of the algorithm that we will apply. For example, our 3-D voxel model is different from the common volumetric data. The volumetric data can be regarded as a "solid" volume and every element contained in the volume is useful for the purpose of representation. On the contrary, in our 3-D volumetric model, a vast majority of the voxels within the predefined volume is not on the 3-D object surface and can be marked as "useless" (or "do not care") in representing the considered object. Two problems emerge from this character of our 3-D voxel model. First, we must find a way to identify the "useful" and "useless" voxels in encoding the 3-D voxel model. Second, if there is any possibility that we can use this character to improve the coding efficiency.

One possible solution is to assign all the "useless" voxels a special value to distinguish them from the "useful" ones and then encode the processed 3-D data. However, this method can lead to very poor image reconstruction. First, to well discriminate the "useless" voxels from the "useful" ones, we must select the assigned

value as far as possible from the peaks of the histogram of the "useful" voxels. If the distribution of the values of the "useful" voxels is even (uniform or almost uniform) among the valid range, the identification of the "useless" voxels will be difficult on the decoded 3-D data because of the inevitable distortion. The wrongly identified "useless" voxels will result in poorly reconstructed images by re-projection of the 3-D model. Second, even if we can find a special value for the "useless" voxels that is far from the peaks of the histogram of the "useful" voxels, there will be large distortions at the edges of "useless" and "useful" voxels, where high-frequency energy concentrates, after encoding and decoding processes. However, the edges of "useless" and "useful" voxels are nothing but where the considered object surface is located. Hence, the reconstructed image quality may be impacted badly.

To overcome the drawbacks of the above method, we label all of the "useful" voxels and the label set is stored and transmitted along with the 3-D voxel model as side information. With the label data, we can now assign the "useless" voxels any values of color information that are valid. It is well known that DCT, WT and many other transformations used in image coding perform best if high-frequency coefficients are small relative to low-frequency coefficients. Thus, we assign all the "useless" voxels the average color value of all the "useful" voxels to reduce the high-frequency energy. The pre-processed 3-D voxel model is then applied to the 3-D coding scheme. For the label

set, we applied a lossless coding scheme by first generating a set of run-length codes of the labels and then employing arithmetic coding[1] on the run-length code.

Therefore, the encoded 3-D data for the 3-D voxel model include both the encoded label data and the encoded pre-processed 3-D data. Related experimental results will be shown in Section 5.3.4.

5.3.2 3-D Wavelet-based SPIHT Coding Scheme

The well-known SPIHT image coding algorithm [51] is among state-of-the-art image coding techniques. The SPIHT algorithm utilizes three basic concepts: 1) searching for sets in spatial-orientation trees in a wavelet transform; 2) partitioning the wavelet transform coefficients in these trees into sets defined by the level of the highest significant bit in a bit-plane representation of their magnitudes; and 3) coding and transmitting bits associated with the highest remaining bit planes first.

Kim et al. [56] applied a 3-D extension of the SPIHT for a low bit rate embedded video coding scheme. The 3-D SPIHT has the following three similar characteristics: 1) partial ordering by magnitude of the 3-D wavelet transformed video with a 3-D *set partitioning* algorithm; 2) ordered bit plane transmission of refinement bits; and 3) exploitation of self-similarity across *spatio-temporal* orientation trees. The 3-D SPIHT was also successfully employed in medical volumetric data compression [16]. The

[1] The kernel of the software for the arithmetic coding is copyrighted in 2004 by Amir Said (said@ieee.org) & William A. Pearlman (pearlw@ecse.rpi.edu).

advantage of the 3-D SPIHT is that it does not only exploit the correlation within one image frame/slice but also exploits the correlation that exists among neighboring image frames/slices.

We applied the 3-D wavelet-based SPIHT coding scheme[2] for our obtained three 3-D voxel models—VM3a, VM5b and VM5c. Experimental results will be presented in Section 5.3.4.

5.3.3 H.264-based 3-D Data Coding Scheme

The 3-D voxel models generated by our multi-view image coding system can be considered as a set of highly correlated "video frames". Hence, algorithms that were originally designed for video coding schemes can be applied in volumetric data compression [16][56], and vice versa. In this section, we consider applying a video coding scheme onto our 3-D voxel model. By splitting the 3-D voxel model along certain dimension, i.e., z-dimension, the 3-D voxel model can be regarded as a sequence of image frames and input into a video encoder. Therefore, the motion compensation that is commonly applied in video coding schemes helps to exploit the correlations along the z-dimenstion of our 3-D voxel model.

In our simulations, we applied the H.264 video coding standard[3] to our obtained 3-D voxel models. H.264 is the newest international video coding standard [67]. It combines

---

[2] The software [76] we used is copyrighted in 1995, 1996, 1997, 1998, 1999 by Amir Said and William A. Pearlman or Beong-Jo Kim, Zixiang Xiong, William A. Pearlman, and Amir Said.

many features that are not included or specified in former international video coding standard such as H.261/263 or MPEG-1/2/4, aiming at improving the coding efficiency, robustness to data errors/losses and flexibility to a variety of application environments. In particular, the following features of H.264 certainly help in improving the coding efficiency of our 3-D voxel model: (1) variable block-size motion compensation with small block size; (2) quarter-pixel accuracy in motion compensation; (3) multiple reference picture motion compensation; (4) directional spatial prediction for intra coding; (5) hierarchical block transform; (6) exact-match inverse transform; and (7) *context-adaptive binary arithmetic coding* (CABAC).

Experimental results will be provided in Section 5.3.4.

5.3.4 Experimental Results of 3-D Voxel Model Coding Using H.264 and 3-D SPIHT

In this section we provide a set of experimental results of 3-D voxel model coding covering the contents and algorithms described in section 5.3.1 to 5.3.3. To simplify the problem, we consider only the luminance information of the original image sequence "the *cup*" and the obtained three 3-D voxel models—VM3a, VM5b and VM5c. Focusing on the luminance performance is both reasonable and widely acceptable. The performance of

---

[3] The software that we used is copyrighted in 2001 by International Telecommunications Union (ITU), Geneva.

luminance compression represents the key criterion used in evaluating image/video coding schemes because of the sensitivity of human eyes to changes in luminance.

As we discussed in Section 5.3.1, our encoded 3-D data for the 3-D voxel model include two parts: the encoded label data and the encoded pre-processed 3-D data. Figure 5-7 and 5-8 depict the coding performance for the three 3-D voxel models using 3-D wavelet-based SPIHT coding scheme and H.264-based coding scheme, respectively. In the SPIHT coding scheme, the number of frames in one segment is 16. The shown bit rate includes both the encoded label data and the encoded pre-processed 3-D data. For a certain 3-D scene model, the size of encoded label data is fixed and does not impact the shape of the rate-distortion (R-D) curve of the used coding scheme.

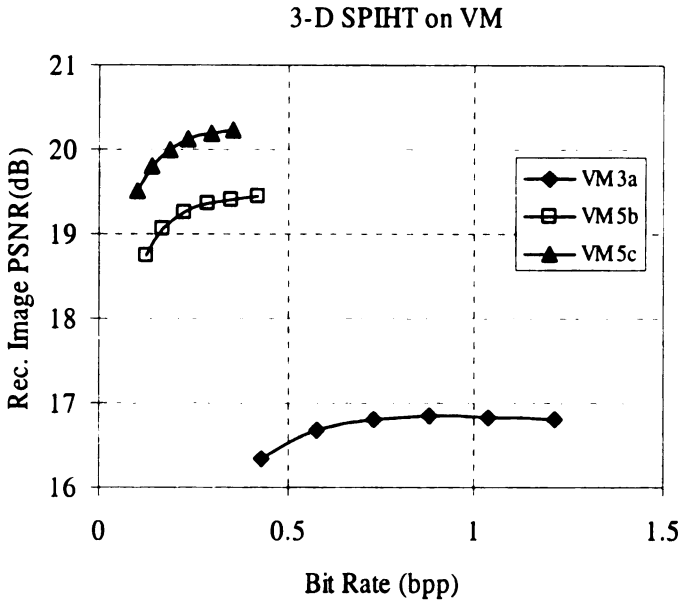


Figure 5-7 Rate-PSNR curves of the 3-D wavelet-based SPIHT coding for the three 3-D voxel models. The x-axis represents the bit rate of the encoded 3-D voxel model; the y-axis represents the average image quality over the available reconstructed images from re-projection of the decoded 3-D voxel model.

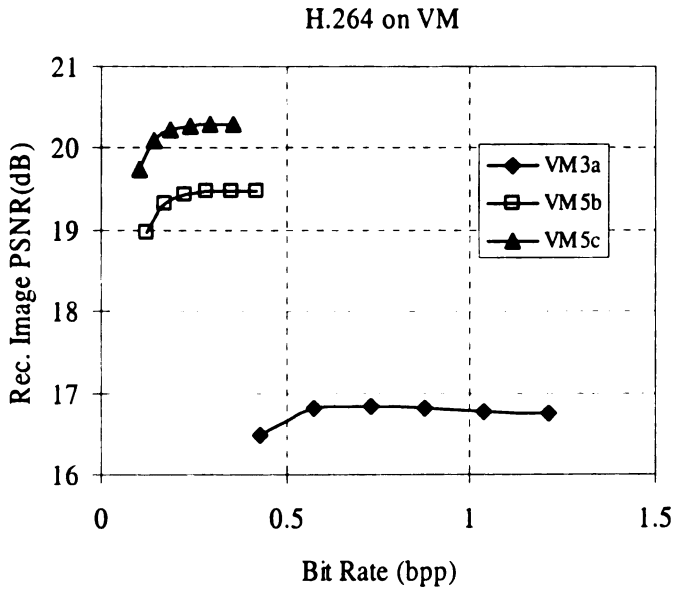


Figure 5-8 Rate-PSNR curves of the H.264-based coding for the three 3-D voxel models. The x-axis represents the bit rate of the encoded 3-D voxel model; the y-axis represents the average image quality over the available reconstructed images from re-projection of the decoded 3-D voxel model.

We can conclude from Figure 5-7 and 5-8 that the coding performance for the VM5c is the best among the three models, regardless if the 3-D SPIHT coding scheme or the H.264-based coding scheme is employed. This observation is consistent with the conclusion we made in Section 5.2.9 that the VM5c performs the best among the three 3-D voxel models according to both the model size and the quality of reconstructed images directly obtained from the re-projection process. In particular, the coding performance for the VM3a is significantly worse than that for the VM5b and the VM5c. This observation is reasonable since the performance for the VM5b and VM5c is close to each other while the performance for the VM3a is far from them (readers are referred to Figure 5-3, 5-4, 5-5 and Table 5-1 for details).

Next, we compared the coding performance for the same 3-D voxel model using the two proposed coding schemes, as shown in Figure 5-9.

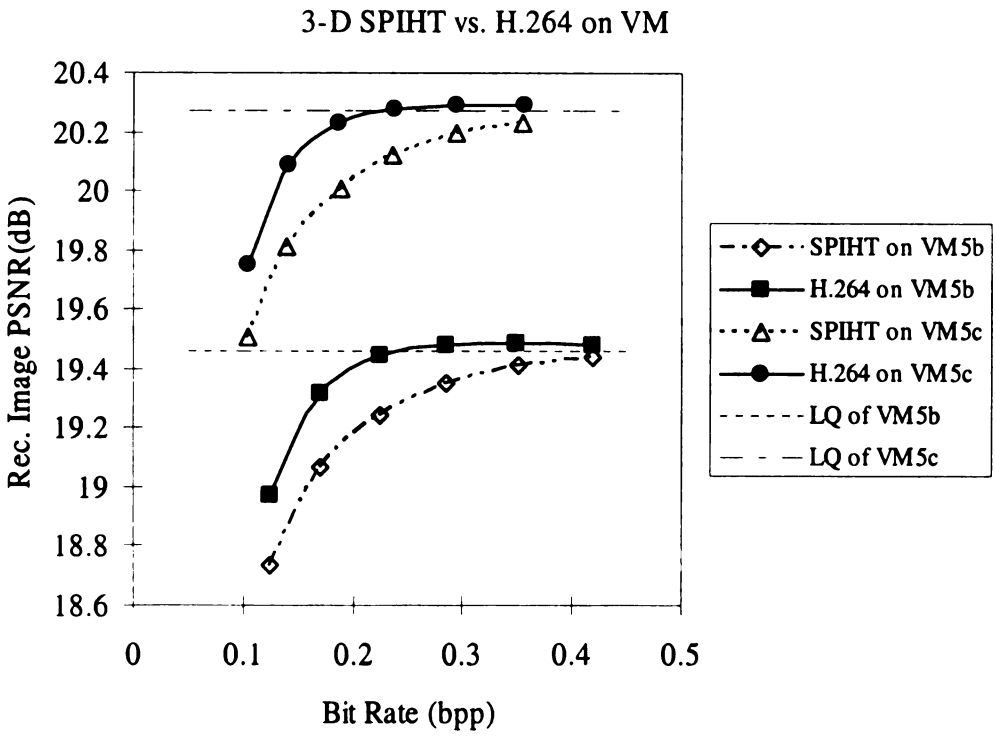


Figure 5-9 Comparison of the rate-PSNR curves between the 3-D SPIHT and the H.264-based coding scheme for VM5b and VM5c. The x-axis represents the bit rate of the encoded 3-D voxel model; the y-axis represents the average image quality over the available reconstructed images from re-projection of the decoded 3-D voxel model. The "LQ" is the abbreviation of "Lossless Quality", which represents the quality of the reconstructed images directly from re-projection of the corresponding 3-D voxel model without encoding and decoding processes.

There are two observations from Figure 5-9. First, the H.264-based coding scheme outperforms the 3-D SPIHT coding scheme for both of the 3-D voxel models. Second, for the same 3-D voxel model, both of the two rate-PSNR curves approach the Lossless Quality (19.46 dB for VM5b and 20.27 dB for VM5c, shown in Figure 5-9). However, the rate-PSNR curve for the H.264-based coding scheme converges to the Lossless Quality more quickly than that for the 3-D SPIHT coding scheme does.

The rate-PSNR curves of H.264-based coding schemes for both of the two 3-D voxel models are unusual compared with common rate-PSNR curves in image/video compression in two aspects. First, they are of non-monotony. Second, the peak value of the PSNR (for both of the two 3-D models) is greater than the goal PSNR (LQ), which is supposed to be the upper bound of the rate-PSNR curve. The main reason for these two unusual characters is that we measured the quality of the reconstructed images from re-projecting the 3-D voxel model, not the quality of the decoded 3-D voxel model. Moreover, since our 3-D voxel models are far from a perfect construction of the true 3-D scene, it is possible that the distortions occurring during the H.264-based coding procedure of the 3-D voxel model improve the quality of the reconstructed images. However, these unusual phenomena vary with the considered image sequence, the quality of the obtained 3-D voxel model, etc. With the improvement of the quality of the 3-D voxel model, these phenomena will be weakened. This expectation is consistent with the experimental results in Figure 5-9.

At the end of this section, we compared our label coding with another method of identifying the "useless" voxel by assigning all the "useless" voxels a special value. For the label coding, we recorded the size of encoded label data, the bit rate of encoded 3-D data (including the encoded label data and the encoded pre-processed 3-D data by 3-D SPIHT), as well as the quality of reconstructed images from re-projection of the decoded 3-D voxel model. For the compared method, we recorded the bit rate of encoded

pre-processed 3-D data as well as the quality of reconstructed images from re-projection of the decoded 3-D voxel model. A set of experimental results is listed in Table 5-2.

Table 5-2 Comparison between the 3-D data coding using label coding and the compared method of encoding the pre-processed 3-D data with special assignment of all the "useless" voxels.

	VM3a		VM5b		VM5c	
	Label Coding	Comp. Method	Label Coding	Comp. Method	Label Coding	Comp. Method
Label Data (bytes)	103,516	—	36,583	—	29,488	—
Bit Rate (bpp)	0.7292	0.7301	0.2250	0.2255	0.1867	0.1867
Rec. Image PSNR (dB)	16.80	9.14	19.25	10.88	20.01	11.35

For each 3-D voxel model, the overhead for label coding is fixed regardless of the bit rate of the overall encoded 3-D data. As shown in Table 5-2, the better the quality of the 3-D voxel model, the less the size of encoded label data. It makes sense, since a 3-D voxel model of better quality means fewer voxel number and fewer noisy voxels, which no doubt improves the coding efficiency of the label data. For the convenience of comparison, we chose close bit rate of the encoded 3-D data for both the label coding and the compared method for the same 3-D voxel model. The results of the quality of the reconstructed images in Table 5-2 show that under approximately same bit rate of the encoded 3-D data, the reconstruction quality of our proposed label coding is much higher than that of the compared method. In fact, the reconstruction quality of the compared

method is so poor that this method cannot be used in practice. Possible reasons leading to the poor reconstruction quality has been discussed in Section 5.3.1.

5.4 Residual Coding

As we discussed in Section 5.1, the residual coding is an optional procedure in our multi-view coding system. However, the residual coding will be required for high-quality reconstruction of original images in many applications, since the image reconstruction directly from re-projection of the 3-D scene model is far from perfect. For instance, in our experiments on the *cup* sequence, since the constructed 3-D voxel model is not perfect (experimental results shown in Section 5.2.5, 5.2.7 and 5.2.9), the quality of reconstructed images directly from re-projection of the 3-D model (experimental results shown in Section 5.3.4) is not comparable with that commonly required in image compression and cannot be used in many applications. Therefore, the residual coding is an important option that is needed to improve the reconstruction quality.

However, in our multi-view coding system, the residual between the original images and re-projected images is quite different from that in video coding schemes in two aspects. These two aspects require special considerations in coding the residual data and will be discussed in details in the remainder of this section.

5.4.1 Residual De-correlation

In many video coding schemes, the residual data is obtained after motion compensation-based prediction, and this data shows little correlation among neighboring frames. In our case, the origin of the residual is the difference between the true 3-D scene structure and the estimated 3-D voxel model. One voxel that contains incorrect color information will lead to correlated errors among all the considered images. Hence, the residual images in our multi-view coding scheme show correlations with each other.

To de-correlate the residual images, and similar to our 3-D voxel model coding, we propose to employ the H.264 video coding standard or 3-D SPIHT coding scheme to the residual images. Hence, we did experiments using the two approaches for our multi-view image coding. Experimental results of each approach and the comparison between them will be presented in Section 5.4.3.

5.4.2 Residual Regulation

Another character of the residual data in our case is that it can be distributed in a larger range of values, unlike the residual data in many video coding schemes, which usually has a smaller variation. For instance, in the widely used 8-bit representation of basic color component (either YUV or RGB color system), the valid value is between 0 and 255. However, the residual data between the original images and the re-projected images can be as least as –255 or as great as 255. In other words, the residual data require 9-bit representation instead of 8-bit, which may make the residual incompatible to many

existing image/coding techniques or many applications. To resolve this issue when using coding standards (e.g., H.264) that operates on 8-bit pixels, we consider two methods to regulate the residual data.

Residual Splitting. In the first way, which we refer to as "residual splitting", we split each residual image into two images: one contains all the residual pixels of positive values while the other one contains all the residual pixels of negative values. By this scheme, either the positive residual images or negative residuals are located in the range of [0, 255]. Then the coding schemes we discussed in Section 5.4.1 can be employed to both of the positive and negative residual data. The final reconstructed image can be calculated by:

$$\hat{I} = I_{rep} + R_p - R_n, \tag{5-13}$$

where $\hat{I}$ represents the final reconstructed image, I_{rep} represents the re-projected images from the 3-D voxel model, and R_p and R_n represent the positive and negative residual image respectively.

There is no error introduced in the residual splitting. However, it results in double-sized residual, which is undesirable in our goal of data compression.

Residual Rescaling. In the second way, which we refer to as "residual rescaling", we regulate the residual data by shifting and rounding-off:

$$R_r = \lfloor (R + 255)/2 \rfloor, \tag{5-14}$$

where R and R_r represent the original residual and the rescaled residual, respectively. Now the rescaled residual is located in [0, 255] and can be represented by 8 bits.

The final reconstructed image is calculated by:

$$\hat{I} = I_{rep} + (2R_r - 255), \tag{5-15}$$

where $\hat{I}$ represents the final reconstructed image, I_{rep} represents the re-projected images from the 3-D voxel model, and R_r represents the rescaled residual image.

We should notice that unlike the residual splitting, the residual rescaling does not increase the size of residual data at the expense of ignoring the least significant bit to rescale the residual from 9-bit representation to 8-bit representation and resulting in a "lossy" residual data. However, if we consider the possible distortions caused in the residual coding, we expect that the residual rescaling outperforms the residual splitting. Comparisons between these two methods will be made in Section 5.4.3.

5.4.3 Experimental Results of Residual Coding

This section provides experimental results for the coding performance of the residual de-correlation and regulation. We consider only the latter two 3-D scene models—VM5b and VM5c. For each 3-D model, we accomplished our multi-view image coding by two approaches: (1) H.264 based coding scheme for both the 3-D data coding and the residual coding; (2) 3-D SPIHT based coding scheme for both the 3-D data coding and the residual coding. For each 3-D model and each coding approach, the 3-D data coding was fixed (which means that we chose a fixed bit rate for the 3-D data coding). In practice, we tried to select close bit rate for different approach for a certain 3-D model so that the performance of the whole coding scheme is "almost" consistent with the performance of

the residual coding, if not "completely" the same. The bit rate of the encoded 3-D data and the associated image quality from re-projection of the decoded 3-D voxel model are listed in Table 5-3:

Table 5-3 The selected bit rate of the encoded 3-D data and the associated image quality from re-projection of the decoded 3-D voxel model for each 3-D voxel model and each approach for the multi-view image coding.

	VM5b		VM5c	
	3-D SPIHT	H.264	3-D SPIHT	H.264
Bit Rate (bpp)	0.2250	0.2242	0.1867	0.1854
Rec. Image PSNR (dB)	19.25	19.45	20.01	20.23

All the values in Table 5-3 were chosen from Figure 5-9.

We first compared the coding performance of the residual splitting with that of the residual rescaling for VM5c, using the H.264-based coding scheme for both the 3-D data coding and the residual coding.

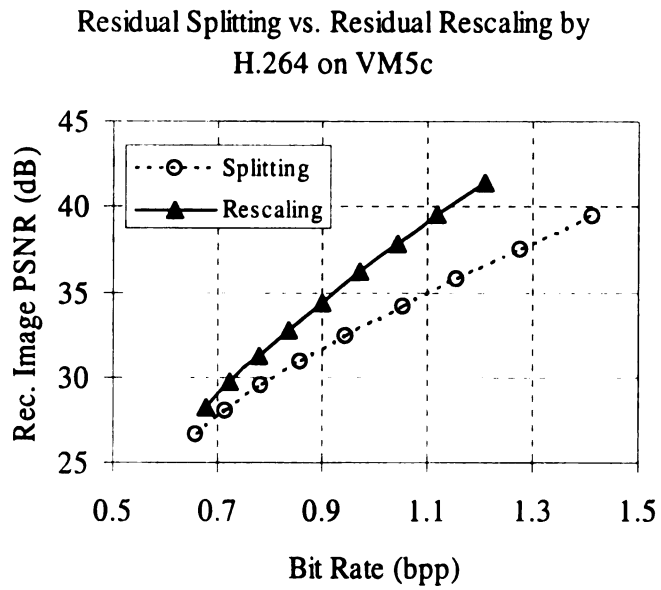


Figure 5-10 Comparison of the rate-PSNR curves between the residual splitting and the residual rescaling for VM5c by using the H.264-based coding scheme. The x-axis represents the bit rate of the encoded 3-D data and the encoded residual data; the y-axis represents the average image quality over the available final reconstructed images from the re-projected images (from the decoded 3-D voxel model) plus the compensation (from the decoded residual data).

The two rate-PSNR curves displayed in Figure 5-10 show that the residual rescaling outperforms the residual splitting. This observation is consistent with the hypothesis we made at the end of Section 5.4.2. Moreover, in Figure 5-10, the coding efficiency of the residual rescaling over the residual splitting becomes greater and greater with the increase of the bit rate. As stated in Section 5.4.2, since the residual rescaling generates a lossy residual, we cannot obtain perfect reconstruction of the original images with the rescaled residual data. However, this problem does not impact the use of the residual rescaling in practice, since the whole procedure of our multi-view image coding in nature contains

data distortion and we care about the coding performance, not necessarily if perfect reconstruction can be achieved.

Because of the advantage of the residual rescaling over the residual splitting, in the next experiment, we employed only the residual rescaling technique in the two different residual coding approaches—3-D SPIHT coding scheme and H.264-based coding scheme—to compare their coding performance. VM5b and VM5c were involved in this experiment and, again, the values listed in Table 5-2 were chosen for the 3-D data coding. The coding result is shown in Figure 5-11.

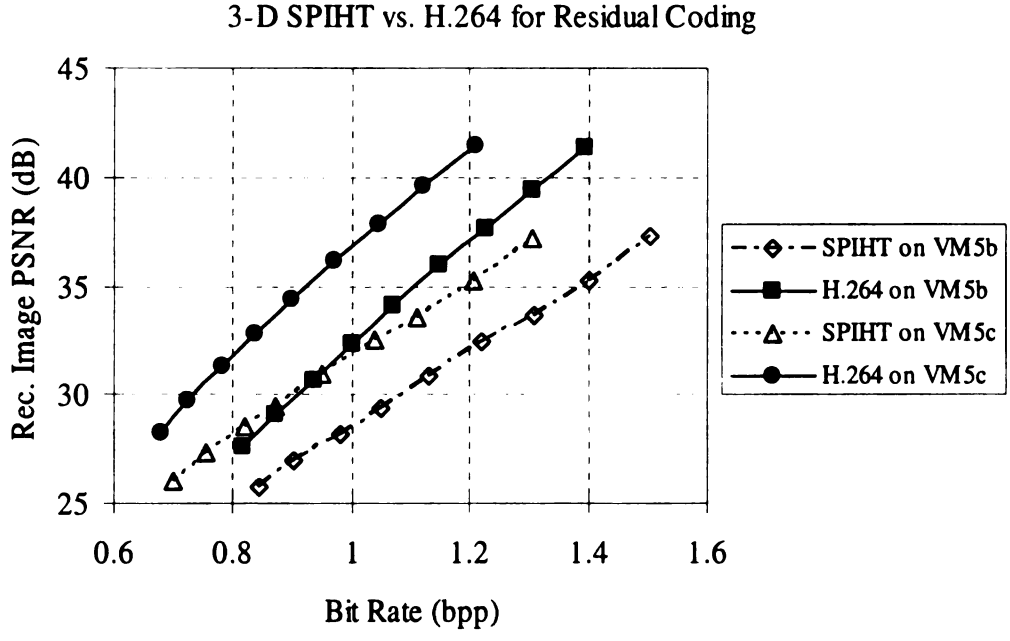


Figure 5-11 Comparison of the rate-PSNR curves between the 3-D SPIHT residual coding and the H.264-based residual coding for VM5b and VM5c, using the residual rescaling technique. The x-axis represents the bit rate of the encoded 3-D data and the encoded residual data; the y-axis represents the average image quality over the available final reconstructed images from the re-projected images (from the decoded 3-D voxel model) plus the compensation (from the decoded residual data).

Figure 5-11 shows that the performance of the H.264-based residual coding is better than that of the 3-D SPIHT residual coding for the considered two 3-D voxel models. Moreover, for the same 3-D voxel model, the improved coding efficiency of the H.264-based coding scheme over the 3-D SPIHT coding scheme becomes greater and greater with the increase of the bit rate.

Finally, we provide some examples of the final reconstructed images from the re-projected images (from the decoded 3-D voxel model) plus the compensation (from the decoded residual data) at some values of bit rate, with respect to different 3-D voxel models and different coding schemes. The values of bit rate, the used 3-D voxel models and the applied coding schemes were selected from Figure 5-11.

First, Figure 5-12 compares the final reconstructed image 6[4] using the 3-D SPIHT-based coding scheme and that using the H.264-based coding scheme, given the same 3-D voxel model VM5c. The bit rates for both coding schemes are very close to each other. Figure 5-13 displays another set of reconstructed image 6. All of the experimental conditions are the same as those in Figure 5-12, except for the bit rates for both coding schemes.

[4] The reason for choosing Image 6 is that the camera viewing position for Image 6 is typical in the original image sequence and the PSNR for reconstructed Image 6 is close to the average PSNR for all the available 14 reconstructed images.

Rec. Image 6 using 3-D SPIHT (PSNR = 28.34 dB, bit rate = 0.8176 bpp)

Rec. Image 6 using H.264 (PSNR = 30.92 dB, bit rate = 0.7800 bpp)

Figure 5-12 The reconstructed image 6 resulting from the re-projected image (from the decoded 3-D voxel model) plus the compensation (from the decoded residual data), as well as the value of PSNR and the overall bit rate. The used 3-D voxel model: VM5c.

Rec. Image 6 using 3-D SPIHT (PSNR = 32.29 dB, bit rate = 1.037 bpp)

Rec. Image 6 using H.264 (PSNR = 35.96 dB, bit rate = 0.9715 bpp)

Figure 5-13 The reconstructed image 6 resulting from the re-projected image (from the decoded 3-D voxel model) plus the compensation (from the decoded residual data), as well as the value of PSNR and the overall bit rate. The used 3-D voxel model: VM5c.

Figure 5-12 and 5-13 clearly illustrate that the quality of reconstructed images using the H.264-based multi-view coding scheme is significantly better than that using the 3-D SPIHT-based multi-view coding scheme at very close bit rate (if not completely the same), no matter which 3-D voxel model was used.

Second, we compare in Figure 5-14 the final reconstructed image 6 using VM5b and that using VM5c, with the same H.264-based coding scheme was applied. The bit rates for both 3-D voxel models are very close to each other. Figure 5-15 shows another set of reconstructed image 6. All of the experimental conditions are the same as those in Figure 5-14, except for the bit rates for both 3-D models.

Rec. Image 6 using VM5b (PSNR = 27.42 dB, bit rate = 0.8163 bpp)

Rec. Image 6 using VM5c (PSNR = 30.92 dB, bit rate = 0.7800 bpp)

Figure 5-14 The reconstructed image 6 resulting from the re-projected image (from the decoded 3-D voxel model) plus the compensation (from the decoded residual data), as well as the value of PSNR and the overall bit rate. The applied coding scheme: H.264-based coding scheme.

Rec. Image 6 using VM5b (PSNR = 32.10 dB, bit rate = 0.9991 bpp)

Rec. Image 6 using VM5c (PSNR = 35.96 dB, bit rate = 0.9715 bpp)

Figure 5-15 The reconstructed image 6 resulting from the re-projected image (from the decoded 3-D voxel model) plus the compensation (from the decoded residual data), as well as the value of PSNR and the overall bit rate. The applied coding scheme: H.264-based coding scheme.

Figure 5-14 and 5-15 clearly illustrate that the quality of reconstructed images using VM5c is significantly better than that using VM5b at very close bit rate (if not completely the same), with the same H.264-based coding scheme applied. Once again, it shows that the quality of the 3-D voxel model significantly impacts the coding efficiency of our multi-view coding system.

5.5 Summary

In this chapter, we proposed a multi-view image coding system in 3-D space and discussed in detail the crucial functional blocks of the proposed multi-view coding framework: volumetric 3-D reconstruction, 3-D data coding and residual coding.

Our approach for the volumetric 3-D reconstruction is similar to Eisert's approach [34]. However, we made two modifications to improve its performance. We developed three 3-D voxel models for the same image sequence. The first 3-D model is based on the basic algorithms in our approach, which are similar to those of Eisert's approach. The second and third 3-D models are both based on modified approaches (the modifications that result in the second 3-D model are a subset of the modifications that result in the third 3-D model.). Experimental results show that the performance of the two 3-D voxel models based on the modified approach (VM5b and VM5c) is significantly better than that of the 3-D voxel model based on the basic approach (VM3a), in terms of the size of the encoded 3-D data and the quality of reconstructed images by re-projecting the 3-D voxel model back to image planes for the same camera viewing positions of the original

images. Furthermore, the experimental results for 3-D model coding and residual coding show that the quality of the 3-D voxel model greatly impacts the coding efficiency of our multi-view image coding scheme as anticipated.

For the 3-D voxel model coding, we proposed two approaches: 3-D wavelet-based SPIHT coding scheme and H.264-based coding scheme. Both of the two approaches can exploit not only the 2-D correlations of image frames/slices (correlations along the x-dimension and the y-dimension), but also the correlations among neighboring image frames/slices (correlations along the z-dimension). Experimental results show that the data to represent the 3-D voxel model was significantly compressed using these two coding schemes. Furthermore, in general the H.264-based coding scheme outperforms the 3-D SPIHT coding scheme, according to the obtained rate-PSNR curves for the same 3-D voxel model. There is no doubt that many newly introduced or specified features of H.264 have helped improve the coding efficiency for our 3-D voxel model. To our knowledge, we were the first to apply the H.264 video coding standard to 3-D data compression. The experimental results we have obtained illustrate us the potential of the techniques involved in the H.264 standard beyond the range of video coding.

Another important aspect of the 3-D data coding is the characteristics of our 3-D voxel model—a vast majority of the voxels within the predefined volume are not on with the 3-D object surface and can be marked as "useless". Therefore, two problems emerge from this observation. First, we must find a way to identify the "useful" and "useless" voxels in encoding the 3-D voxel model. Second, how can we utilize this observation to

improve the coding efficiency? Our solution to the first problem is to label all the "useful" voxels and the label set is stored, compressed, and transmitted along with the 3-D voxel model as side information. With the label data, we can now assign the "useless" voxels the average color value of all the "useful" voxels to reduce the high-frequency energy for the improvement of the performance of the transformation used in the coding scheme. Therefore, the encoded 3-D data for the 3-D voxel model include both the encoded label data and the encoded pre-processed 3-D data.

The residual coding is an optional procedure in our multi-view coding system. However, the residual coding will be required for high-quality reconstruction of original images in many applications. We discussed in detail two aspects of the residual coding. The first aspect is that there are still some correlations among the residual images, since one voxel that contains incorrect color information will lead to correlative errors among all the considered images. Hence, we proposed to employ the 3-D SPIHT or H.264-based coding scheme to de-correlate the residual images. Experimental results show that the H.264-based residual coding outperforms the 3-D SPIHT residual coding. The second aspect regarding the residual coding is the residual regulation, by which the out-of-ranged residual data is mapped back to the valid range. We proposed two methods for the residual regulation, called residual splitting and residual rescaling. Experimental results show that the latter performs much better than the former.

In summary, the key difference of our proposed multi-view image coding scheme from existing multi-view image coding schemes is that instead of representing the 3-D

scene information of the images to be encoded by the mesh model as well as the texture data and encoding the mesh model as well as the texture data, we use a 3-D voxel model to represent the 3-D scene information of the considered images and then encode the 3-D voxel model for the purpose of storage and transmission. There are several advantages of the 3-D voxel model. First, the 3-D voxel model is much simpler than the mesh model in structure. Second, recovering the original images or generating synthetic images from the 3-D voxel model is straightforward by the re-projection of the 3-D model; meanwhile image reconstruction from the mesh model requires mapping the texture data to the mesh model. Third, since the 3-D voxel model is an extension from 2-D data to 3-D data, many existing techniques for the image/video coding can be applied for the coding of the 3-D voxel model. We have employed the H.264 coding standard (already applied for video coding) and the 3-D SPIHT coding scheme (already applied for video coding and volumetric data coding) in our experiments. Experimental results show the potential of our proposed multi-view image coding system. Furthermore, the coding efficiency of our scheme can be remarkably improved if the quality of the 3-D voxel model is improved, as shown in the experimental results in Section 5.3.4 and 5.4.3.

6 Conclusions

6.1 Summary

With the rapid development of modern computers and networks, a great deal of research has been focused on the transformation from 2-D visual applications to the 3-D visual world. Briefly speaking, there are two major areas of 3-D visual research. In the first 3-D visual research area, researches are focused on depicting the 3-D scenes using 2-D representations, in which multi-view image coding has become a key issue. Under the second area, researchers concentrate on reconstructing the 3-D scene from available multiple 2-D images.

These two research areas are closely related to each other, especially in the aspect of multi-view image coding. Recent studies have shown that in multi-view image coding, if the 3-D scene geometry information is given, the coding efficiency, decoding speed and rendering visual quality can be dramatically improved. Motivated by this exciting observation and related research problems in existing 3-D geometry based multi-view image coding schemes, this dissertation proposed and developed a multi-view coding system that operates directly in the 3-D space based on automatic 3-D scene reconstruction. The dissertation focused on key aspects of the automatic 3-D scene reconstruction and detailed coding scheme of the proposed multi-view image coding system.

This dissertation made two contributions in the field of automatic 3-D scene reconstruction. First, a new multistage self-calibration algorithm is proposed. Differing from previous approaches, where the optimization function is not explicit with respect to the camera intrinsic parameters, we derived a polynomial optimization function of the intrinsic parameters, which makes the optimization simple and insensitive to the initialization. Then based on a stability analysis of the intrinsic parameters, a multistage procedure to refine the self-calibration is proposed. We applied our method to both synthetic and real images. Second, we presented a new proof that there are only four possible solutions in recovering the camera relative motion from the essential matrix. The new proof concentrates on the geometry among the essential-matrix, the camera rotation, and the camera translation. In addition, we provided a generalized SVD-based proof for the four possible solutions in decomposing the essential matrix. We discussed the feasibility of SVDs for the essential matrix and provided the general form of the feasible SVDs. Then we verified that the multiple SVDs for the essential matrix lead to only four possible solutions to the camera relative motion.

We proposed a multi-view image coding system in 3-D space based on automatic 3-D scene reconstruction, which is an important kernel of this dissertation. Instead of coding the 3-D geometry information (usually a mesh model) and image (texture) date separately, as applied in many existing multi-view image coding schemes, we establish a unifying 3-D scene voxel model for all the available image views and then encode the 3-D scene voxel model for compression. Encoding of the image (texture) data

is no longer required in our proposed coding system. We studied in detail three crucial functional blocks in our proposed coding system: volumetric 3-D reconstruction, 3-D scene voxel model encoding and residual coding.

The volumetric 3-D reconstruction is crucial in our multi-view image coding system because the quality of the obtained 3-D voxel model greatly impacts the coding efficiency of our coding scheme. We made modifications to improve Eisert's approach for volumetric 3-D reconstruction. In terms of different versions of the volumetric 3-D reconstruction approach, we developed three 3-D voxel models. Experimental results show that the performance of the two 3-D voxel models based on modified approaches is significantly better than that for the 3-D voxel model based on the basic approach, in terms of the size of the encoded 3-D data, the quality of reconstructed images by re-projecting the 3-D voxel model back to image planes for the same camera viewing positions of the original images, and the coding efficiency of later 3-D voxel model coding and residual coding.

We proposed two approaches for the 3-D scene voxel model coding: 3-D wavelet-based SPIHT coding scheme and H.264-based coding scheme. Both of the two approaches can exploit not only the 2-D correlations of image frames/slices but also the correlations among neighboring image frames/slices. Experimental results show that the data size of the 3-D voxel model was greatly reduced using these two coding schemes. Besides, in general the H.264-based coding scheme performs better than the 3-D SPIHT coding scheme does, according to the obtained rate-distortion curves for the same 3-D

voxel model. This observation illustrates the potential of the techniques developed by the H.264 standard for coding applications that are beyond video coding. Another important aspect of the 3-D voxel model coding is the usage of label data that is necessary to distinguish the voxels that reside on the considered 3-D object surface (which we refer to as "useful" voxels) from other "useless" voxels. In detail, we label all the "useful" voxels and the label set is stored, compressed, and transmitted along with the encoded 3-D voxel model as side information. With the label data, we assign the "useless" voxels the average color value of all the "useful" voxels to reduce the high-frequency energy for the improvement of the performance of the transformation used in the coding scheme.

The residual coding is an optional procedure in our multi-view coding system but will be required for high-quality reconstruction of original images in many applications. We discussed in detail two aspects of the residual coding. First, since there are still some correlations among the residual images, we proposed to employ the 3-D SPIHT or H.264-based coding scheme to compress the residual images. Experimental results show that the H.264-based residual coding outperforms the 3-D SPIHT residual coding. The second aspect is the residual regulation, by which the out-of-ranged residual data is mapped back to the valid range. We proposed two methods for the residual regulation, called residual splitting and residual rescaling. Experimental results show that the residual rescaling performs significantly better than the residual splitting.

In summary, we proposed a multi-view image coding system based on a 3-D scene voxel model that represents the images to be encoded in 3-D space, and employed the

H.264 coding standard (already applied for video coding) and the 3-D SPIHT coding scheme (already applied for video coding and volumetric data coding) in the proposed 3-D multi-view image coding system. Compared with the multi-view coding schemes that is simply an extension of 2-D coding techniques to 3-D case and does not explore the 3-D description of the considered images, the proposed multi-view coding system provides a 3-D scene voxel model that represents the images to be encoded in 3-D space. This 3-D voxel model does not only improve the coding efficiency, but also provide the availability of synthetic image generation. The latter feature is important in many developing multimedia techniques, such as virtual reality, video conferencing system, and distance education. Compared with existing multi-view image coding schemes using 3-D scene geometry, the proposed multi-view coding system uses the 3-D voxel model to describe the 3-D scene information, instead of the mesh model as well as texture data. There are several advantages of using the 3-D voxel model. First, the 3-D voxel model is much simpler than the mesh model in structure. Second, reconstruction of the original images or generation of synthetic images from the 3-D voxel model is straightforward by the re-projection of the 3-D model using the desired camera viewing positions; meanwhile image reconstruction from the mesh model requires mapping the texture data to the mesh model. Third, since the 3-D voxel model is an extension from 2-D data to 3-D data, many existing techniques for the image/video coding can be applied for the coding of the 3-D voxel model. Recent examples of these coding techniques include the H.264 video coding standard and 3-D SPIHT coding scheme, both of which we employed

in the proposed coding system. Experimental results show the potential of our proposed multi-view image coding system in the aspect of coding efficiency and flexibility for various multimedia applications.

6.2 Discussion and Future Research

There are several considerations to improve the performance of the proposed multi-view image coding system in 3-D space. First, both theoretical analysis and experimental results show that the coding efficiency of our coding scheme can be remarkably improved if the quality of the 3-D voxel model is improved. One possible improvement in the volumetric 3-D reconstruction is to apply additional post-processing to further remove noisy voxels and refine the continuity of the considered 3-D object surface, e.g., median filtering that is widely used in image processing. Second, in the volumetric 3-D reconstruction, we assumed that the considered images are calibrated. However, in many applications, only uncalibrated images can be acquired for automatic 3-D reconstruction. Hence, extension from the calibrated images to uncalibrated images is expected in the step of volumetric 3-D reconstruction in the proposed multi-view coding system for the purpose of meeting the needs of various applications. Third, we have just directly employed the 3-D wavelet SPIHT coding scheme and the H.264-based coding standard to the 3-D voxel model coding and residual coding. It is expected that the coding efficiency can be improved if both of the coding schemes are modified/optimized to adapt the characteristics of the 3-D voxel model data and residual data. For instance,

the statistical characteristics of the residual images are closely related to the performance of the corresponding part of the 3-D voxel model. In detail, if from some camera viewing position the 3-D voxel model is of good performance and hence the reconstructed image is of good quality, the residual data will be of small variation. And vice versa. Therefore, a residual coding scheme that is adaptive to the performance of the different parts of the 3-D scene voxel model (and hence the statistical characteristics of the residual data) is expected to improve the coding efficiency.

Besides the proposed 3-D SPIHT and H.264 coding schemes, warping the 3-D scene model to a 2-D plane is another possible approach for 3-D voxel model coding. The idea of warping the voxel model that represents the 3-D object surface to a 2-D plane is similar to the texture-map based method that is described in Section 2.2.2. For instance, Gu et al. [77] proposed to re-mesh an arbitrary surface onto a completely regular structure called *geometry image*. It captures geometry as a simple 2-D array of quantized points. The obtained geometry image can then be encoded using traditional image compression techniques.

REFERENCES

[1] S. E. Chen, "Quicktime VR –An image-based approach to virtual environment navigation", *Proc. of ACM Conf. Computer Graphics 95*, pp. 29-38, 1995.

[2] M. Levoy and P. Hanrahan, "Light filed rendering", *Proc. of ACM Conf. Computer Graphics'96*, pp. 31-42, 1996.

[3] S. Gortler, et al., "The lumigraph", *Proc. of ACM Conf. Computer Graphics'96*, pp. 11-20, 1996.

[4] H.-Y. Shum and L.-W. He, "Rendering with concentric mosaics", *Proc. of ACM Conf. Computer Graphics'99*, pp. 299-306, 1999.

[5] G. Miller, S. Rubin and D. Ponceleon, "Lazy decomposition of surface light fields for precomputed global illumination", *Proc. of Eurographics Rendering Workshop'98*, pp. 281-292, 1998.

[6] Y. Wu, L. Luo, J. Li and Y.-Q. Zhang, "Rendering of 3D wavelet compressed concentric mosaic scenery with progressive inverse wavelet synthesis (PIWS)", *Proc. of SPIE Visual Communications and Image Processing'2000*, vol. 1, pp. 31-42, 2000.

[7] I. Peter and W. Strasser, "The wavelet stream: Interactive multi resolution light field rendering", *Proc. of 12th Eurographics Rendering Workshop*, pp. 262-173, 2001.

[8] C. Zhang and J. Li, "Compression and rendering of concentric mosaics with reference block code (RBC)", *Proc. of SPIE Visual Communications and Image Processing'2000*, vol. 1, pp. 43-54, 2000.

[9] X. Tong and R. M. Gray, "Interactive view synthesis from compressed light fields", *Proc. of IEEE International Conf. on Image Processing'2001*, pp. 85-88, 2001.

[10]J. Li, H. Y. Shum and Y.-Q. Zhang, "On the compression of image based rendering scene", *Proc. of IEEE International Conf. on Image Processing'2000*, pp. 21-24, 2000.

[11]S. Wong, L. Zaremba, D. Gooden and H. Huang, "Radiologic image compression-a review", *Proc. of IEEE*, vol. 83, pp. 194-219, 1995.

[12]J. Hu, Y. Wang and P. Cshill, "Multispectral code excited linear prediction coding and its application in magnetic resonance images", *IEEE Trans. on Image Processing*, vol. 6, pp. 1555-1566, 1997.

[13]J. Wang and K. Huang, "Medical image compression by using three-dimensional wavelet transform", IEEE *Trans. on Medical Imaging*, vol. 15, pp. 547-554, 1996.

[14]M. Pratt, C. Chu and S. Wong, "Volume compression of MRI data using zerotrees of wavelet coefficients", *Proc. of SPIE Wavelet Applications in Signal and Image Processing IV*, vol. 2825, pp. 752-763, 1996.

[15]J. Luo, X. Wang, C. W. Chen and K. Parker, "Volumetric medical image compression with three dimensional wavelet transform and octave zerotree coding", *Proc. of SPIE Visual Communications and Image Processing*, vol. 2727, pp. 579-590, 1996.

[16]Z. Xiong, X. Wu, S. Cheng and J. Hua, "Lossy-to lossless compression of medical volumetric data using three-dimensional integer wavelet transforms", *IEEE Trans. on Medical Imaging*, vol. 22, no. 3, pp. 459-470, 2003.

[17]R. Y. Tsai and T. S. Huang, "A versatile camera calibration technique for high-accuracy machine vision metrology using off-the-shelf TV cameras and lenses", *IEEE Journal of Robot and automation*, vol. 3, pp. 323-344, 1987.

[18]O. D. Faugeras, Q.-T. Luong and S. J. Maybank, "Camera self-calibration: theory and experiments", *Proc. 2nd European Conf. Computer Vision., Lecture Notes in Computer Science*, vol. 588, pp. 321-334, 1992.

[19]M. Pollefeys and L. Van Gool, "Stratified self-calibration with the modulus constraint", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 707-724, 1999.

[20]H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections", *Nature*, vol. 293, pp. 131-135, 1981.

[21]R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 13-26, 1984.

[22]J. Weng, N. Ahuja and T. S. Huang, "Motion and structure from point correspondences with error estimation: planar surfaces", *IEEE Trans. on Signal Processing*, vol. 39, no. 12, pp. 2691-2717, 1991.

[23]M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool, "Automated reconstruction of 3D scenes from sequences of images", *Isprs Journal of Photogrammetry and Remote Sensing*, vol. 55, no. 4, pp. 251-267, 2000.

[24]P. Eisert, E. Steinbach and B. Girod, "Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views", *IEEE Trans. on Circuits and Systems fro Video Technology*, vol. 10, no. 2, 2000.

[25]D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin and W. Stuetzle, "Surface light fields for 3D photography ", *Proc. of ACM Conf. Computer Graphics'00*, pp. 287-296, 2000.

[26]H. Schirmacher, W. Heidrich and H.-P. Seidel, "High-quality interactive lumigraph rendering through warping", *Proc. of Graphics Interface2000*, pp. 87-94, 2000.

[27]M. Magnor, P. Ramanathan and B. Girod, "Multi-view coding for image-based rendering using 3-D scene geometry", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1092-1106, 2003.

[28]D. Cohen-Or, Y. Mann and S. Fleishman, "Deep compression for streaming texture intensive animations", *Proc. of ACM Conf. Computer Graphics'99*, pp. 261-268, 1999.

[29]S. M. Seitz and C. M. Dyer, "View morphing", *Proc. ACM Conf. Computer Graphics'96*, pp. 21-30, 1996.

[30]D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video", *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp572-588, 1994.

[31]S.-J Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video", *IEEE Trans. on Image Processing*, vol. 8, no. 2, pp. 155-167, 1999.

[32]W. Li, "Overview of fine granularity scalability in MPEG-4 video standard", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301-317, 2001.

[33]M. V. der Schaar and H. Radha, "A hybrid temporal-SNR fine granularity scalability for internet video", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 318-331, 2001.

[34]P. Eisert, E. Steinbach and B. Girod, "Multi-hypothesis, volumetric reconstruction of 3-D objects from multiple calibrated views", *Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing'1999*, pp. 3509-3512, 1999.

[35]W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm", *Proc. of ACM Conf. Computer Graphics'87*, pp. 163-169. 1987.

[36]H. Hoppe, "Progressive meshes", *Proc. of ACM Conf. Computer Graphics'96*, pp. 99-108, 1996.

[37]R. Pajarola and J. Rossignac, "Compressed progressive meshes", *Georgia Inst. Technol.*, Atlanta, GA, Rep. GIT-GVU-99-05, 1999.

[38]M. Magnor and B. Girod, "Fully embedded coding of triangle meshes", *Proc. of Vision, Modeling and Visualization'1999*, pp. 253-259, 1999.

[39]Y. Gao and H. Radha, "A multistage camera self-calibration algorithm", *Proc. of IEEE Conf. on Acoustic, Speech and Signal Processing'2004*, vol. III, pp. 537-540, 2003.

[40]Y. Gao and H. Radha, "There are four solutions in recovering the camera relative motion from the essential matrix", Submitted to *IEEE Trans. on Image Processing.*

[41]L. G. Shapiro and G. C. Stockman, "Computer vision", Prentice-Hall Inc., 2001.

[42]G. Karlsson and M. Vetterli, "Three-dimensional subband coding of video", *Proc. of IEEE Conf. on Acoustics, Speech, and Signal Processing'1988*, pp. 1100-1103, 1988.

[43]G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression", *IEEE Signal Processing Magazine*, November 1998, pp. 74-90, 1998.

[44]Z. Zhang, R. Deriche and O. D. Faugeras, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", *Artificial Intelligence*, vol. 78, pp. 87-119, 1995.

[45]Q.-T Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms and stability analysis", *The International Journal of Computer Vision*, vol. 1, no. 17, pp. 43-76, 1996.

[46]R. I. Hartley, "In defense of the eight-point algorithm", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580-593, 1997.

[47]R. I. Hartley, "Estimation of relative camera positions for uncalibrated cameras", *Proc. Second European Conf. Computer Vision*, pp. 579-587, 1992.

[48]P. R. S. Mendonca and R. Cipolla, "A simple technique for self-calibration", Proc. of *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 112-116, 1999.

[49]G. Roth and A. Whitehead, "Some improvements on two autocalibration algorithms based on the fundamental matrix", *Proc. of International Conf. on Pattern Recognition*, vol. 2, pp. 312-315, 2002.

[50]M. Magnor and B. Girod, "Model-based coding of multi-viewpoint imagery", *Proc. of SPIE Visual Communication and Image Processing'2000*, vol. 1, pp. 14-22, 2000.

[51]A. Said and W. A. Pearlman, "A new, fast and efficient image codec based on set portioning in hierarchical trees", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 3, 1996.

[52]M. Magnor, A. Endmann and B. Girod, "Progressive compression and rendering of light fields", *Proc. of Vision, Modeling and Visualization'2000*, pp. 199-203, 2000.

[53]S. Chen and L. Williams, "View interpolation for image synthesis", *Proc. of ACM Conf. on Computer Graphics'1993*, pp. 279-288, 1993.

[54]A. R. Calderbank, I. Daubechies, W. Sweldens and B.-L Yeo, "Wavelet transforms that map integers to integers", *Applied and Computational Harmonic Analysis (ACHA)*, Vol. 5, No. 3, pp. 332-369, 1998.

[55]J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients", *IEEE Trans. on Signal Processing*, vol. 41, pp. 3445-3462, 1993.

[56]B.-J Kim, Z. Xiong and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1374-1387, 2000.

[57]J. Xu, Z. Xiong, S. Li and Y. Zhang, "Memory-constrained 3-D wavelet transform for video coding without boundary effects", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, pp. 812-818, 2002.

[58]T. Sikora, "Digital video coding standards and their role in video communications", *Signal Processing for Multimedia*, J. S. Byrnes (Ed.), pp. 225-252, IOS Press, 1999.

[59]C. E. Shannon, "A mathematical theory of communication", *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, July and October, 1948.

[60]N. Ahmed, T. Natrajan and K. R. Rao, "Discrete Cosine Transform", *IEEE Trans. on. Computers*, vol. C-23, no. 1, pp. 90-93, 1984.

[61]S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, 1989.

[62]CCITT, "Draft revision of recommendation H.261", *CCITT SG XV, Document 572*, March 1990.

[63]ISO/IEC, "Information technology—coding of moving pictures and associated audio for digital storage media at up to bout 1.5 Mbit/s-video", *ISO/IEC 1172-2*, Geneva, 1993.

[64]ITU-T and ISO/IEC, "Generic coding of moving pictures and associated audio information—Part 2: Video", *ITU-T Recommendation H.262 – ISO/IEC 13818-2*, November 1994.

[65]ITU-T, "Video coding for low bit rate communication", *ITU-T Recommendation H.263*, version 1, November 1995; version 2, January 1998; version 3, November 2000.

[66]ISO/IEC, "Coding of audio-visual objects—Part 2: Visual", *ISO/IEC 14496-2*, April 1999; Amendment 1, February 2000; Amendment 4, January 2001.

[67]Joint Video Team of ITU-T and ISO/IEC, "Draft ITU-T recommendation and final draft international standard of joint video specification", *ITU-T Recommendation H.264 - ISO/IEC 14496-10 AVC*, March 2003.

[68]T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, 2003.

[69]C. Zeller and O. D. Faugeras, "Camera self-calibration from video sequences: the Kruppa equations revisited", *Research Report 2793*, INRIA, Feb. 1996.

[70]T. S. Huang and O. D. Faugeras, "Some properties of the E matrix in two-view motion estimation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1310-1312, 1989.

[71]Z. Zhang, Q.-T. Luong and O. D. Faugeras, "Motion of an uncalibrated stereo rig: self-calibration and metric reconstruction", *Research Report 2079*, INRIA, June 1994.

[72]B. K. P. Horn, "Recovering baseline and orientation from essential matrix", *http://www.ai.mit.edu/people/bkph/papers/essential.pdf*, 1990.

[73]R. Hartley and A. Zisserman, "Multiple view geometry in computer vision", Cambridge University Press, pp. 238-241, 2001.

[74]W. Wang and H.-T Tsui, "An SVD decomposition of essential matrix with eight solutions for the relative positions of two perspective cameras", *Proc. of 2000 International Conf. on Pattern Recognition*, vol. 1, pp. 362-365, 2000.

[75]http://www.nt.e-technik.uni-erlangen.de/~eisert/reconst.html.

[76]http://www.cipr.rpi.edu/research/SPIHT/spiht3.html.

[77]X. Gu, S. J. Gortler and H. Hoppe, "Geometry images", *Proc. of ACM Conf. Computer Graphics'2002*, pp. 355-361, 2002.