View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Kentucky

University of Kentucky
UKnowledge

University of Kentucky Master's Theses

Graduate School

2009

3D RECONSTRUCTION USING MULTI-VIEW IMAGING SYSTEM

Conglin Huang
University of Kentucky, zhaoxp1346@yahoo.com

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation

Huang, Conglin, "3D RECONSTRUCTION USING MULTI-VIEW IMAGING SYSTEM" (2009). *University of Kentucky Master's Theses*. 600.
https://uknowledge.uky.edu/gradschool_theses/600

This Thesis is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Master's Theses by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

ABSTRACT OF THESIS

3D RECONSTRUCTION USING MULTI-VIEW IMAGING SYSTEM

This thesis presents a new system that reconstructs the 3D representation of dental casts. To maintain the integrity of the 3D representation, a standard model is built to cover the blind spots that the camera cannot reach. The standard model is obtained by scanning a real human mouth model with a laser scanner. Then the model is simplified by an algorithm which is based on iterative contraction of vertex pairs. The simplified standard model uses a local parametrization method to obtain the curvature information. The system uses a digital camera and a square tube mirror in front of the camera to capture multi-view images. The mirror is made of stainless steel in order to avoid double reflections. The reflected areas of the image are considered as images taken by the virtual cameras. Only one camera calibration is needed since the virtual cameras have the same intrinsic parameters as the real camera. Depth is computed by a simple and accurate geometry based method once the corresponding points are identified. Correspondences are selected using a feature point based stereo matching process, including fast normalized cross-correlation and simulated annealing.

KEYWORDS: Mesh simplification, Local parametrization, Surface curvature, Interpolation, Multi-view image reconstruction

Conglin Huang
05/2009

3D RECONSTRUCTION USING MULTI-VIEW IMAGING SYSTEM

By

Conglin Huang

Fuhua Cheng
Director of Thesis

Andrew M Klapper
Director of Graduate Studies

05/27/2009

RULES FOR THE USE OF THESIS

Unpublished theses submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments.

Extensive copying or publication of the thesis in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this thesis for use by its patrons is expected to secure the signature of each user.

Name	Date

THESIS

Conglin Huang

The Graduate School

University of Kentucky

2009

3D RECONSTRUCTION USING MULTI-VIEW IMAGING SYSTEM

THESIS

A thesis submitted in partial fulfillment of
the requirements for the degree of Master of
Science in the College of Engineering at the
University of Kentucky

By

Conglin Huang

Lexington, Kentucky

Director: Dr. Fuhua Cheng, Professor of Computer Science

Lexington, Kentucky

2009

Copyright© Conglin Huang 2009

ACKNOWLEDGEMENTS

The following thesis, while an individual work, benefited from the insights and direction of several people. I would like to thank all of the people who helped make this thesis possible.

First, I especially would like to thank my advisor and thesis director, Dr. Fuhua Cheng, for all of his enlightening guidance, inspiring encouragement, and tremendous support, and for providing me with the unique opportunity to work in the research area of graphics and computer vision. His knowledge of graphics and considerable experience often came to the rescue.

Next, I would like to thank my committee members, Dr. Yang and Zhang, for their acceptance of this task and for their helpful comments and suggestions. Each individual provided insights that guided and challenged my thinking, substantially improving the finished product.

In addition to the technical and instrumental assistance above, I received equally important assistance from my family and friends. Special thanks to my parents for instilling in me the desire to obtain the Master’s. Special thanks to my friends for their constant support.

TABLE OF CONTENTS

Acknowledgements iii

List of Figures vi

Chapter 1 Introduction 1

Chapter 2 Conceptual Development 5
  Background and Related Work 5
    Subdivision Surfaces 5
    Simplification 5
    Surface Interpolation of Irregular Meshes 6
    Multi-view Imaging 7
  Algorithms and Geometric Fundamentals 8
    The Loop Algorithm 8
    Parametrization of Loop Subdivision Surfaces 10

Chapter 3 Simplification & Interpolation 15
  Simplification 15
    Basic Idea 15
    Pair Selection 16
    Approximating Error with Quadrics 16
    Algorithm Summary 16
    Results 17
  Interpolation 17
    Basic Idea 18
    Local Interpolation 19
    Local Blending 21
    Curvature Computation 22

Chapter 4 3D Reconstruction Based on Multi-view Imaging System 25
  Early Trials 25
    Cylindrical Mirror 25
    Square Glass Mirror 27
  Imaging System 28
    Overview 28
    Geometric Depth Computation 30
  Reconstruction 33
    Camera Calibration 33
    Correspondence 37

Chapter 5 Implementation Results 47

Chapter 6 Concluding Remarks 52

References.. 53

Vita.. 55

LIST OF FIGURES

Figure 2.1 Cylindrical Mirror Multi-view System. 7
Figure 2.2 Mirrors of different shapes. 8
Figure 2.3 Subdivision Masks for triangle Splines. 8
Figure 2.4 Surface patch near an extraordinary vertex with its control vertices. 10
Figure 2.5 The mesh of Figure 2.5 after one Loop subdivision step. 12
Figure 2.6 Three regular meshes corresponding to the three shaded patches. 12
Figure 2.7 Partition of the unit square into an infinite family of tiles 13
Figure 3.1 Edge contraction. 15
Figure 3.2 Simplification Results. 17
Figure 3.3 Gaussian Curvature. 24
Figure 4.1 Cylindrical mirror. 25
Figure 4.2 Image taken using cylindrical mirror. 26
Figure 4.3 Square glass mirror. 27
Figure 4.4 Image taken using square glass mirror. 27
Figure 4.5 Double reflection. 28
Figure 4.6 Square Tube Mirror. 29
Figure 4.7 Overview of the multi-view imaging system. 29
Figure 4.8 Collinear relationship between a pixel and its corresponding points. 31
Figure 4.9 Geometric depth computation. 33
Figure 4.10 Image taken by the Square Tube Mirror. 42
Figure 4.11 Image area labeling. 42
Figure 4.12 Continuousness of the corresponding points on the same scanline. 45
Figure 5.1 Mirror calibration. 48
Figure 5.2 Reconstruction of a single, non-textured tooth. 49
Figure 5.3 Original object and feature point cloud (from different angles). 50
Figure 5.4 Original object and the disparity map. 51

Chapter 1 Introduction

The purpose of this thesis is to present a system of reconstructing the 3D image of dental casts. The system includes simplification and interpolation of the standard model and a single-camera based multi-view imaging reconstruction process.

Dental casts are widely used in dentistry to plan treatment for the patients. Three dimensional models provide a convenient and fast way to build prostheses. Furthermore, a 3D representation of dental casts is extremely useful for both doctors and patients. Some applications have already been used as a routine technique for dental examinations. Such as oral and maxillofacial radiology, including methods like computed tomography (CT), tomosynthesis [1] and tuned-aperture CT (TACT) [2]. Although the radiology techniques provide complete information of the jaw, the equipment is rather expensive. Moreover, the resolution is too low for 3D modeling.

Recently, many efforts have focused on computer-vision based diagnosis in dentistry. Goshtasby et al. [3] presented a system that uses a range scanner to reconstruct the cast. The scanner is based on white light. The system uses a string that moves over the cast so that the scanner can capture the shadow profiles to create the model. D. Laurendeau and D. Possart [4] designed a computer-vision based technique for the acquisition of jaw data from inexpensive dental wafers. That system was capable of obtaining imprints of human teeth. David Tasman and Allan G. Farman [5] presented a technique using a sequence of optical images. The approach uses a modified shape from shading (SFS) algorithm which uses camera calibration and perspective projection to compute the 3D points.

Our approach relies on the images that were captured by the camera, which means some part of the jaw or teeth will be hard to reach. To solve this problem, a standard model is needed. The model is obtained by scanning real teeth and gum plastic models. The laser scanner provides complex, highly detailed meshes in order to maintain a convincing level of realism. However, the full complexity of such models is not always necessary. So we applied a thinning process [6] to get simplified models.

The algorithm maintains surface error approximations using quadric matrices and iteratively contract vertex pairs to simplify models. The steps are as follows:

1. A fundamental error quadric K is computed for every point on the mesh which can be used to find the squared distance of any point in the space to the planes incident on the point. In other words, the cost of contracting a pair of vertices is computed as the sum of distances between the new point and the planes incident on the original vertices. The cost is represented as Q.

2. Vertex pairs are considered as edges in our implementation. Compute the optimal contraction target v for each pair. The point v minimizes the cost (Q) for both vertices of this pair. Place all the pairs in a heap keyed on cost with the minimum cost pair at the top.

3. Iteratively remove the pair of the least cost from the heap, contract this pair, and update the costs of all valid pairs involving the vertices in this pair.

In order to apply future process to the simplified model such as segmentation, reconstruction, and registration, curvature estimation is necessary. Curvature information can be easily obtained if the local parametrization is known. The method [7] used in our project is to interpolate the interested region of the given mesh with a C^2 -continuous, locally blended Loop subdivision surface [8] and use curvature of the locally blended Loop surface as an approximation to the curvature of the given mesh. The constructed surface faithfully reflects the shape of the interested region of the given mesh. Actually, since the locally blended Loop surface has continuous curvature everywhere, one can compute curvature for any point of the given mesh in any direction. The construction of the locally blended Loop surface requires two steps. First, a Loop subdivision surface that interpolate vertices of the interested region of the given mesh is constructed using a technique similar to the progressive interpolation technique for B-splines [9],[10],[11]. A blending technique similar to the one used for Catmull-Clark subdivision surfaces [12] is then applied to vicinities of the extra-ordinary points to ensure boundedness and continuity of curvature at the extraordinary points. Then each triangular face of the given mesh is parametrized using parametrization of the corresponding patch of the locally blended Loop surface.

Our imaging system consists of a digital camera and a square tube mirror in front of the camera. A bracket is built to support the camera and the tube mirror as well as the human jaw. The bracket has wheels so that each part of the system can be moved freely to adjust the distance between them. The mirror is made of stainless steel in order to avoid double reflections.

The imaging system for capturing 3D shapes is based on stereopsis. The classic computer stereo vision uses two cameras to take pictures of the same scene. They are separated by a distance - just like our eyes. A computer compares the images while shifting the two images together over top of each other to find the parts that match. The shifted amount is called the disparity. The disparity at which objects in the image best matches is used by the computer to calculate their distance. Although the system has only one camera, the reflected areas of the image are considered as the images taken by the virtual cameras. The square tube mirror creates four areas that are reflected only once, so the image can be considered as five views taken by five different cameras. The mirror not only gives us the multi-view images but also gives us advantageous properties that can be used in the depth computation. In most camera configurations, finding corresponding points requires a search in two dimensions. However, if the two cameras are aligned to have a common image plane, the search is simplified to one dimension - a line that is parallel to the line between the cameras. In our system, if the mirror plane and the optical axis of the camera are parallel, the virtual cameras and the real camera shares the same image plane. This will give us image rectification [33]. At this time, all corresponding points are at the same scan line. Using this property, a very simple but accurate geometry depth computation method (mentioned in Chapter 4) can be applied.

The virtual cameras have the same intrinsic parameters as the real camera so only one camera calibration is needed. Correspondences are selected by using a feature point based matching process. Building a 3D representation does not require the depth of all pixels, in fact, if the depths of all pixels in the image are computed, a large number of them will have incorrect results. Especially when dealing with objects with solid color and produces specular reflection like human teeth. So building the 3D model based on the feature points will give us better results and enough points for the model. Canny edge detection is used to identify the feature points. Fast normalized cross-correlation and simulated annealing are used for stereo matching of the feature points as well as the constraint conditions.

Chapter 2 Conceptual Development

Background and Related Work

Subdivision Surfaces

Subdivision surface is a method of representing a smooth surface via the specification of a coarser piecewise linear polygon mesh. The smooth surface can be obtained from the control mesh by recursively subdividing each polygonal face into smaller faces that better approximate the smooth surface. The ability to model arbitrary topology surfaces makes them more suitable than classical spline surfaces in some applications. There are subdivision methods using approximating schemes, such as the Catmull-Clark subdivision scheme [13] proposed in 1978, which is the generalization of bi-cubic spline surface. The Doo-Sabin subdivision method [14] is the generalization of quadratic spline surface. Later, the Loop subdivision scheme [8] was developed for triangular meshes which generalize the Box splines. There are also interpolating subdivision schemes that interpolate the given mesh, including the butterfly subdivision method [15] which was modified subsequently to generate smoother interpolation surfaces in [16]. An interpolating scheme for quadrilateral meshes was proposed in [17].

Simplification

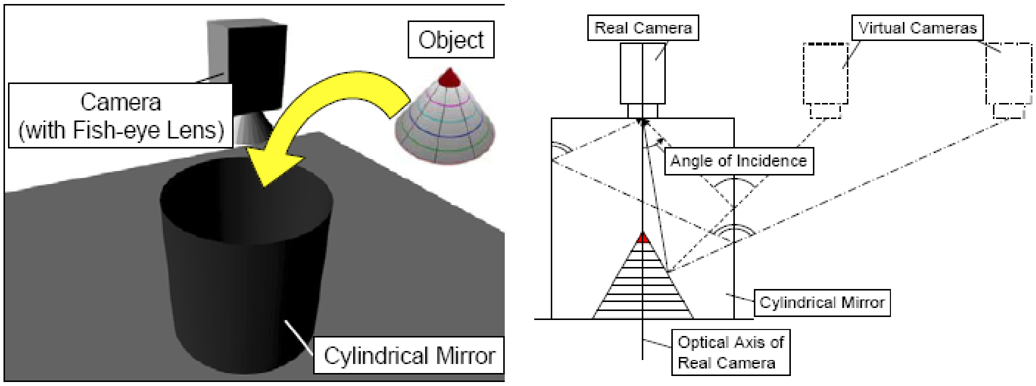
Surface mesh simplification is the process of reducing the number of faces used in the surface while keeping the overall shape, volume and boundaries preserved as much as possible. It is the opposite of subdivision. Over the past several years a large amount of works have been done on mesh simplification. The methods have various algorithms and can be classified into several different categories. Jarek Rossignac and P. Borrel [18] introduced a vertex clustering method which group vertices into clusters and determine a single representative vertex. Iteratively merge clusters into larger clusters until only a single cluster/vertex remains. The algorithm is fast and simple but open has bad output quality. William J. Schroeder [19] presented a decimation algorithm which iteratively removes vertices that pass a certain distance/angle criterion. Among all the simplification algorithm categories, edge collapse is considered as the most often used primitive operation. The vertices connected by an edge are collapsed into a single representative vertex. Michael Garland and Paul S. Heckbert [6] uses a remarkable error metric that can be used for efficient generation of multiple levels of detail for a given triangle mesh of arbitrary topology.

Surface Interpolation of Irregular Meshes

Interpolation is widely used in many computer graphics fields such as surface design and shape modeling. Plenty of publications have discussed the interpolation problem using various surface representations. Interpolation algorithms based on subdivision surfaces have also been developed as the appearance of recursive subdivision surfaces. There are methods which are required to solve a global system of linear equations, like [20, 21]. To avoid the computational cost of solving a large system of linear equations, in [22], an always-working method solved the problem by using a two-phase subdivision method. The method presented in [23] avoids solving a system of linear equations by using the concept of similarity. There is also an approach that [24] avoids solving a system of linear equations by using quasi-interpolation.

Multi-view Imaging

Normal computer stereo vision uses two cameras takes the same scene from different angles. Then the computer can calculate the distance of the object by comparing the two images and find the disparity of the object on the images. Multi-view imaging system uses only one camera but it is still able to calculate the object depth. In most cases the system uses specially designed mirrors to create virtual cameras. Using the scene that is captured by the real camera and the virtual cameras the computer can use the same scheme as the normal computer vision to calculate the object depth. Yuuki Uranishi introduced a multi-view imaging system [25] that uses a cylindrical mirror. The object is placed inside the cylindrical mirror, see Figure 2.1. A camera which uses a fish eye lens captures from the top of the mirror.



(a) 鳥瞰図 (b) 側面図

Figure 2.1 Cylindrical Mirror Multi-view System

Correspondence can be found on the radial lines of the captured image. Shree K. Nayar [26] presented a system that is similar. Instead of placing the object inside of the mirror, the object is placed outside on the other end of the mirror. The camera looks through a hollow cone that is mirrored on the inside. The mirror has different shapes, shown in Figure 2.2.

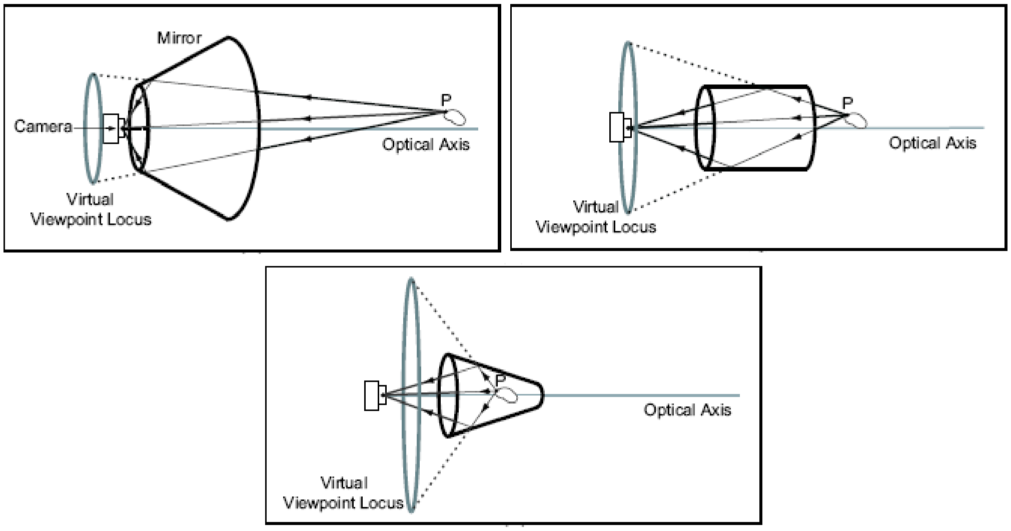


Figure 2.2 Mirrors of different shapes.

Algorithms and Geometric Fundamentals

The Loop Algorithm

This algorithm generalizes the subdivision of a regular triangular mesh [8]. A triangular mesh is a control point mesh whose faces are all triangles. Derivation of the generalized subdivision rules for Loop algorithm begins with an abstraction of the geometric properties of the subdivision masks. These masks are as follows (Figure 2.3):

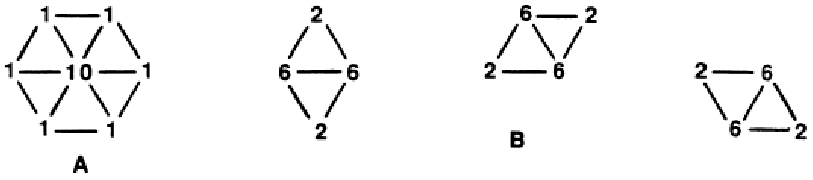


Figure 2.3 Subdivision Masks for triangle Splines.

Mask A generates new control points for each vertex, and masks of B generate new control points for each edge of the original regular triangular mesh.

The masks of B compute the new edge points as convex combinations of the vertices of the two triangles that share the edge. In an arbitrary triangular mesh, each edge will be shared by two triangles. Therefore, an obvious generalization is to leave this subdivision rule intact. Like the Catmull-Clark algorithm, generalization of a vertex point is more difficult.

To derive the new vertex point rule, consider mask A. The new vertex point $\bar{V}$, can be computed as a convex combination of the old vertex, and all old vertices that share an edge with it. Alternatively, this same point may be found indirectly as a convex combination of two points. These points are: V, the old vertex point, and Q, the average of the old points that share an edge with V. The new vertex point is computed as

$$\bar{V} = \frac{5}{8}V + \frac{3}{8}Q \tag{2.1}$$

This can be applied to an arbitrary triangular mesh.

In the special case of a regular triangulation, the algorithm is equivalent to binary subdivision of a surface. Three iterations of the algorithm based on the rules just described are shown in Figure 2.3. As subdivision proceeds, the triangular control point mesh becomes locally regular, except at a fixed number of extraordinary points. For this new algorithm, only extraordinary points correspond to vertices of the original mesh rather than its faces. The surface of this algorithm is locally C^2-continuous everywhere, except at the extraordinary points.

Parametrization of Loop Subdivision Surfaces

Jos Stam presented a parametrization method for Loop subdivision surfaces [27]. By using Loop subdivision rules any triangular mesh can be refined. In the limit of an infinite number of subdivisions, a smooth surface is obtained. The original Loop subdivision can only be parametrized by avoiding extraordinary vertices (whose valence N is not 6). The method provides a way to efficiently evaluate parametrizations near extraordinary points.

The initial mesh used in this method is assumed to be subdivided at least twice, in order to isolate the extraordinary vertices so that each face contains at most one extraordinary vertex. The situation around an extraordinary vertex of valence N is depicted in Fig. 2.5. The shaded triangle in this figure is defined by the $K = N + 6$ control vertices surrounding the patch. The extraordinary vertex corresponds to the parameter value u = 1. Since the valence of the extraordinary vertex in the middle of the figure is $N = 7$, there are $K = 13$ control vertices in this case.

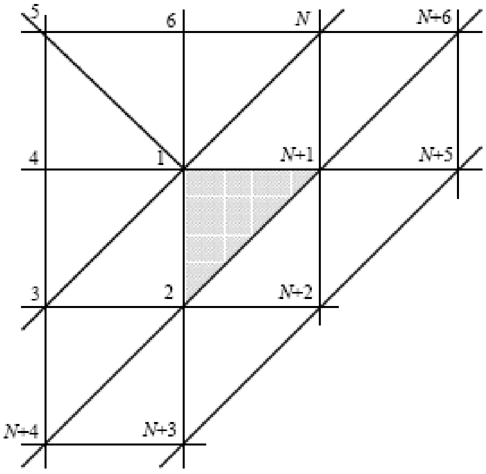


Figure 2.4 Surface patch near an extraordinary vertex with its control vertices.

The initial control points set is denoted by matrix $C_0^T = (c_{0,1}, \cdots, c_{0,K})$,where $K = N + 6$. After one subdivision, a new set of $M = K + 6$ control points (see Figure 2.6) will be generated. Subsets of these new vertices are the control vertices of three regular triangular patches. Therefore, three-quarters of the surface patch is parametrized. This new set of vertices is denoted by $C_1^T = (c_{1,1}, \cdots, c_{1,K})$ and $\overline{C}_1^T = (C_1^T, c_{1,K+1}, \cdots, c_{1,M})$. Then, the subdivision step is simplified to a multiplication process:

$$C_1 = AC_0, \tag{2.2}$$

where matrix A is a $K \times K$ *subdivision matrix*. It has the following block structure:

$$A = \begin{pmatrix} S & 0 \\ S_{11} & S_{12} \end{pmatrix}. \tag{2.3}$$

The additional points needed to evaluate the three triangular patches are defined using a bigger matrix $\overline{A}$ of size $M \times K$:

$$\overline{C}_1 = \overline{A} C_0, \tag{2.4}$$

where

$$\overline{A} = \begin{pmatrix} S & 0 \\ S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}. \tag{2.5}$$

The subdivision step of Equation 2.2 can be repeated to create an infinite sequence of control vertices: $C_n = AC_{n-1} = A^n C_0$ and $\overline{C}_n = \overline{A} C_{n-1} = \overline{A} A^{n-1} C_0$, $n \geq 1$.

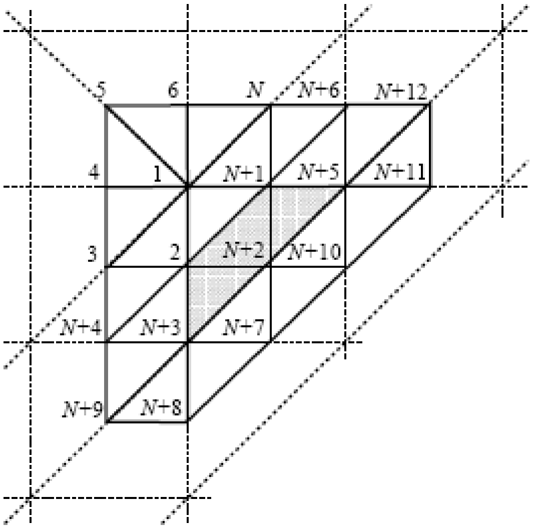


Figure 2.5 The mesh of Figure 2.5 after one Loop subdivision step.

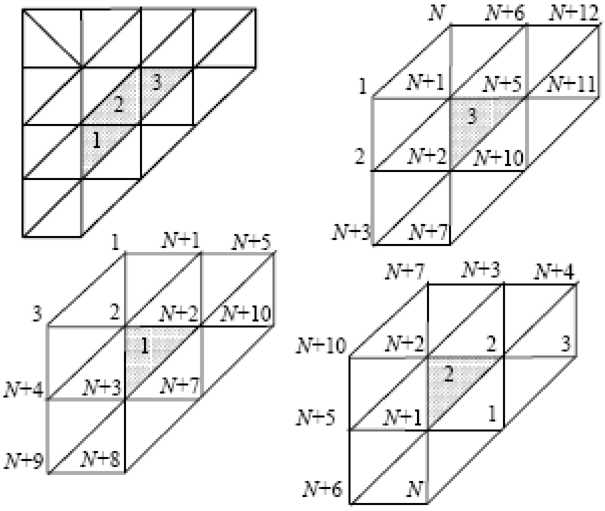


Figure 2.6 Three regular meshes corresponding to the three shaded patches.

As noted above, for each level $n \geq 1$, a subset of 12 vertices of $\overline{C}_n$ form the control vertices of a regular triangular patch. These control vertices can be defined by the following three 12×3 matrices: $B_{k,n} = P_k \overline{C}_n$, where P_k is a $12 \times M$ "picking" matrix and $k = 1,2,3$. Then, the surface patch corresponding to each matrix of control vertices is defined as:

$$s_{k,n}(v,w) = B_{k,n}^T b(v,w) = \overline{C}_n^T P_k^T b(v,w), \tag{2.6}$$

where $b(v,w)$ is the vector of basis functions. $\Omega = \{(v,w) \mid v \in [0,1], \quad w = \in [0,1-v]\}$ and $k = 1,2,3$.

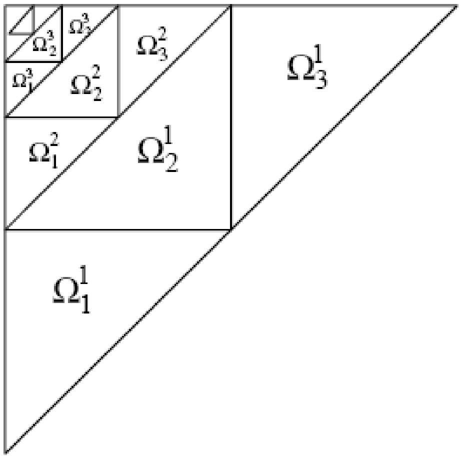


Figure 2.7 Partition of the unit square into an infinite family of tiles

As shown in Figure 2.8 we can partition the parameter domain into an infinite set of tiles $\{\Omega_k^n\}$, with $n \geq 1$ and $k = 1,2,3$. The surface patch is then defined by its restriction of each of these triangles:

$$s(v,w)\Big|_{\Omega_k^n} = s_{n,k}(t_{n,k}(v,w)) = C_0^T (P_k \overline{A} A^{n-1})^T b(t_{n,k}(v,w)), \tag{2.7}$$

where the transformation $t_{n,k}$ maps the tile Ω_k^n onto the unit square Ω:

$$\begin{aligned}
\mathbf{t}_{n,1}(v,w) &= (2^n v - 1, 2^n w), \\
\mathbf{t}_{n,2}(v,w) &= (1 - 2^n v, 1 - 2^n w) \quad \text{and} \\
\mathbf{t}_{n,3}(v,w) &= (2^n v, 2^n w - 1).
\end{aligned}$$

The eigenstructure of matrix A can be written as $A = V\Lambda V^{-1}$. Let $\hat{C}_0 = V^{-1}C_0$ be the projection of the initial control vertices onto the eigenspace of A and let $\Phi(v,w)$ be the K-dimensional vector of eigenbasis functions defined by:

$$\Phi(v,w)|_{\Omega_k^n} = \Lambda^{n-1}(P_k \overline{A} V)^T b(t_{n,k}(v,w)) \quad n \geq 1 \text{ and } k = 1,2,3.$$

So the basis function can be evaluated as:

$$\Phi(v/2, w/2) = \Lambda\Phi(v,w). \tag{2.8}$$

Now triangular surface patch can be written as:

$$s(v,w) = \hat{C}_0^T \Phi(v,w). \tag{2.9}$$

Chapter 3 Simplification & Interpolation

Simplification

Basic Idea

Since the scanning system cannot take images from every angle of the teeth of patients, building a standard model is necessary in order to maintain the integrity of the 3D representation. Every tooth as well as the upper and lower gum of the standard model is scanned separately by a laser scanner, which provides highly detailed results. Each tooth has about 10,000 vertices and about 20,000 faces. This large amount of data will make the future computation extremely slow and it would also be a waste of storage space. In order to reduce the complexity but also keep the accuracy of the model, a quadric error based simplification process is used.

The basic idea of the simplification algorithm [6] is based on the iterative contraction of vertex pairs. A pair contraction (Figure 3.1) moves the vertices v_1 and v_2 to the new position $\bar{\mathbf{v}}$, connects all their incident edges to v_1, and deletes the vertex v_2. Subsequently, any edges or faces which have become degenerate are removed.

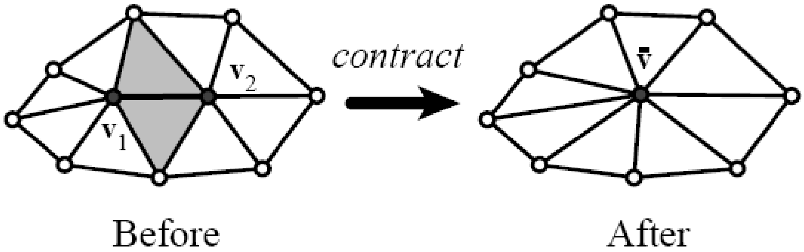


Figure 3.1 Edge contraction.

Pair Selection

We have chosen to select the set of valid pairs at initialization time, and to consider only these pairs during the course of the algorithm. In our case, all vertices that share one edge are considered as a pair.

Approximating Error with Quadrics

In order to select which pair to contract during the iteration, we need to define the cost of contract for each pair. The contract cost is defined as the error at each vertex. We associate a symmetric 4×4 matrix Q with each vertex, and we define the error at vertex $v = [v_x v_y v_z 1]^T$ to be the quadratic form $\Delta(v) = v^T Q v$. Note that $\Delta(v)$ is the error with respect to Q. For a given contraction $(v_1, v_2) \rightarrow \bar{v}$, we must derive a new matrix $\bar{Q}$ which approximates the error at $\bar{v}$. Here we chose to use the simple additive rule $\bar{Q} = Q_1 + Q_2$.

Algorithm Summary

The algorithm maintains surface error approximations using quadric matrices and iteratively contract vertex pairs to simplify models. The steps are as follows:

1. A fundamental error quadric K is computed for every point on the mesh which can be used to find the squared distance of any point in the space to the planes incident on the point. In other words, the cost of contracting a pair of vertices is computed as the sum of distances between the new point and the planes incident on the original vertices. The cost is represented as Q.

2. Vertex pairs are considered as edges in our implementation.

3. Compute the optimal contraction target v for each pair. v is the point that can minimize the cost (Q) for both vertices of this pair.

4. Place all the pairs in a heap keyed on cost with the minimum cost pair at the top.

5. Iteratively remove the pair of the least cost from the heap, contract this pair, and update the costs of all valid pairs involving the vertices in this pair.

Results

The algorithm provides efficiency as well as good results:

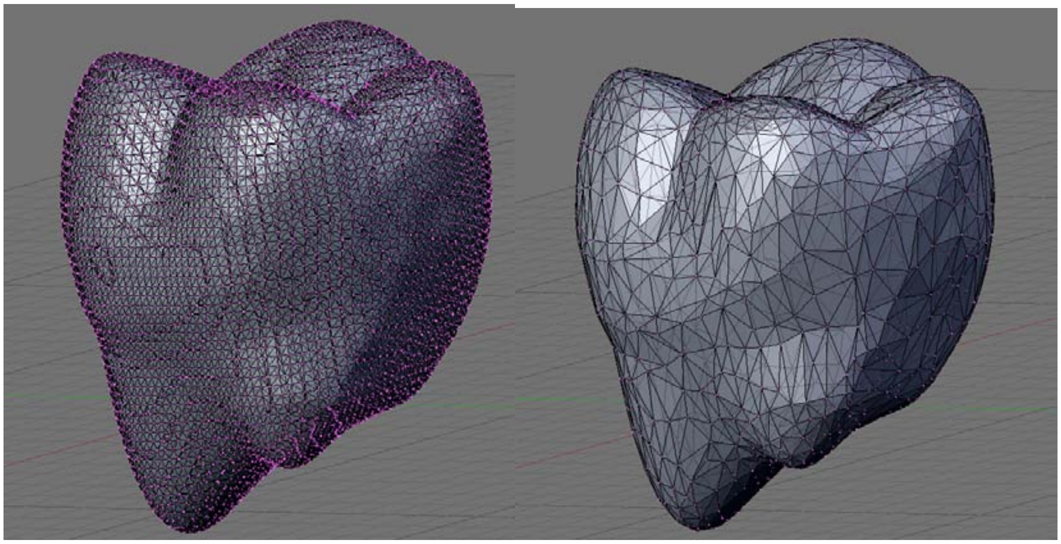


Figure 3.2 Simplification Results.

Interpolation

Interpolation can provide us the local parametrization of the teeth mesh, which can be used to get curvatures of the mesh surface. The interpolation technique guarantees that the resulting surface reflects the local shape of the mesh, including features such as edges and corners.

Basic Idea

First, we use a progressive interpolation technique [9],[10],[11] to find a Loop subdivision surface [8] that interpolates the interested region of the given triangular mesh. The idea is to view the given triangular mesh as the control mesh of a Loop subdivision surface, and iteratively upgrade locations of the vertices of this mesh until a new control mesh whose limit surface interpolates the given mesh is obtained. The limit of the iterative interpolation process has the form of a global method, but the control points of the limit surface can be computed using a local approach. Therefore, the interpolation technique enjoys the advantages of both a local method and a global method.

With the technique for evaluating Loop subdivision surfaces at arbitrary parameter values being available [27], we can find the parametrization of each triangular surface of the control mesh. The computations of the first and second fundamental forms are easy since the parametrization of the surfaces are known.

However, a standard Loop subdivision surface is only C1 -, not C2 -continuous at extraordinary points. To avoid infinite curvature at the extra-ordinary points, the surface is then modified by applying a local blending technique similar to the one used for Catmull-Clark subdivision surfaces [12] to vicinities of extra-ordinary points of the given mesh. The blending process is done by blending the Loop surface with a low degree polynomial defined locally over a characteristic map of the extra-ordinary point [28]. The modified Loop surface is C2 -continuous everywhere and, hence, has bounded and continuous curvature at the extra-ordinary points. This blending process does not change the value of the Loop surface at the extra-ordinary points. Therefore the modified Loop surface still interpolates all the vertices of the interested region of the given mesh.

Note that, with the parametrization known, we can compute not only the Gaussian and Mean curvatures at every point of the original mesh, but also curvatures in any direction at any point on the limit surface.

Local Interpolation

The process of constructing an interpolating Loop subdivision surface based on progressive interpolation technique is described below.

Given a 3D triangular mesh $M = M^0$. To interpolate the vertices of a region of M^0 with a Loop subdivision surface, one needs to find a control mesh $\overline{M}$ whose Loop surface passes through all the vertices of that region of M^0. Without loss of generality, we shall assume that region is M^0 itself. The job of finding the relationship between the vertices of $\overline{M}$ and the vertices of M^0 will be achieved through an iterative process, instead of a direct process.

First, we consider the Loop surface S^0 of M^0. For each vertex V^0 of M^0, we compute the distance between this vertex and its limit point V_∞^0 on S^0,

$$D^0 = V^0 - V_\infty^0 ,$$

and add this distance V^0 to get a new vertex called V^1 as follows:

$$V^1 = V^0 - D^0 .$$

The set of all the new vertices is called M^1. We then consider the Loop surface S^1 of M^1 and repeat the same process.

In general, if V^k is the new location of V^0 after k iterations of the above process and M^k is the set of all the new V^k 's, then we consider the Loop surface S^k of M^k . We first compute the distance between V^0 and the limit point V_∞^k of V^k on S^k

$$D^k = V^0 - V_\infty^k .$$

We then add this distance to V^k to get V^{k+1} as follows:

$$V^{k+1} = V^k + D^k .$$

The set of new vertices is called M^{k+1} .

This process generates a sequence of control meshes M^k and a sequence of corresponding Loop surfaces S^k . S^k converges to an interpolating surface of M^0 if the distance between S^k and M^0 converges to zero. Therefore the key task here is to prove that D^k converges to zero when k tends to infinity. This will be done in the Appendix.

Note that for each iteration in the above process, the main cost is the computation of the limit point V_∞^k of V^k on S^k. For a Loop surface, the limit point of a control vertex V with valence n can be calculated as follows:

$$V_\infty = \beta_n V + (1-\beta_n)Q ,$$

where

$$\beta_n = \frac{3}{11 - 8 \times (\frac{3}{8} + (\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n})^2)} ,$$

and

$$Q = \frac{1}{n}\sum_{i=1}^{n} Q_i \ .$$

Q_i are adjacent vertices of V. This computation involves nearby vertices only. Hence the progressive interpolation process is a local method and, consequently, can handle meshes of any size.

Another point that should be pointed out is, even though this is an iterative process, one does not have to repeat each step strictly. By finding out when the distance between M^0 and S^k would be smaller than the given tolerance, one can go directly from M^0 to M^k, skipping the testing steps in between.

Local Blending

With the construction of the Loop subdivision surface in the previous section, we can define a local (u, v) parameterization for each triangular face of the given mesh [27]. In case one of the vertices is an extra-ordinary point, then (0,0) is set at the extraordinary point. The limit surface $S(u,v)$ is C^2 -continuous for (u, v) different from (0,0) and C^1 -continuous at (0,0) . We assume each face has at most one extra-ordinary vertex here. If this is not the case, simply perform one subdivision on the constructed Loop surface first.

Then a low degree polynomial $P(u, v)$ is calculated to approximate the original Loop limit surface $S(u,v)$ near the extraordinary point. For simplicity, and also to avoid unexpected oscillations, quadratic polynomials are used here for $P(u, v)$. We calculate limit points on the subdivision surface after 3 levels of recursive subdivision. Knowing the values of the limit points and the (u, v) parameter values we can calculate the coefficients of $P(u, v)$ by a least-squares fit.

Using an approach similar to the one used for Catmull-Clark subdivision surfaces [12], the modified surface $\overline{S}$ is defined as:

$$S\ (u,v) = S(u,v)W(u,v) + P(u,v)(1 - W(u,v)),$$

where $0 < W(u, v) < 1$, is a C^2 continuous weight function that decays to zero at (0,0) , thereby canceling the irregularity of $S(u,v)$. Since the modified surface $\overline{S}$ coincides with P at the origin, $P(0,0) = S(0,0)$.

The following formula is chosen to define $W(u, v)$:

$$W(u,v) = \rho^2 (3\rho^2 - 8\rho + 6) \quad , \quad \rho = \sqrt{u^2 + v^2} / \lambda, \rho \leq 1 .$$

λ is the sub-dominant eigenvalue of the subdivision matrix related to an extraordinary vertex of valence n .

At the extraordinary point itself, the modified surface $\overline{S}$ coincides with P up to its second-order derivatives, due to the interpolation requirement $S(0,0) = P(0,0)$. Thus, for evaluating $\overline{S}$ and its derivatives at the extraordinary point, we only need to evaluate the polynomial P at (0,0) .

Curvature Computation

After applying local blending on each extraordinary point, we have the parametrization for each face that gives C^2 -continuity. Hence, we are able to compute the Gaussian and Mean curvature at all points on the original mesh:

The first fundamental form:

$$E = < X_u , X_u >,$$

$$F = < X_u , X_v >,$$

$$G = < X_v, X_v > .$$

The second fundamental form:

$$e = < N, X_{uu} >,$$

$$f = < N, X_{uv} >,$$

$$g = < N, X_{vv} > .$$

Normal of a vertex can be computed either using the standard approximation technique of averaging the normals of adjacent faces, or by acquiring the cross product of X_u and X_v at that vertex. It turns out that, as expected, getting the cross product of two directional derivatives based on the parametrization is much more precise.

Based on the equations of Weingarten, we obtain Gaussian curvature *K* and Mean curvature *H* as follows:

$$K = (eg - f^2)/(EG - F^2),$$

$$H = (eg - 2fF + gE)/(2(EG - F^2)).$$

The principle curvatures k_1 and k_2 are the roots of the quadratic equation:

$$k^2 - 2Hk + K = 0,$$

that is,

$$k = H \pm \sqrt{H^2 - K} .$$

Figure 3.3 shows the Gaussian curvature of a tooth:

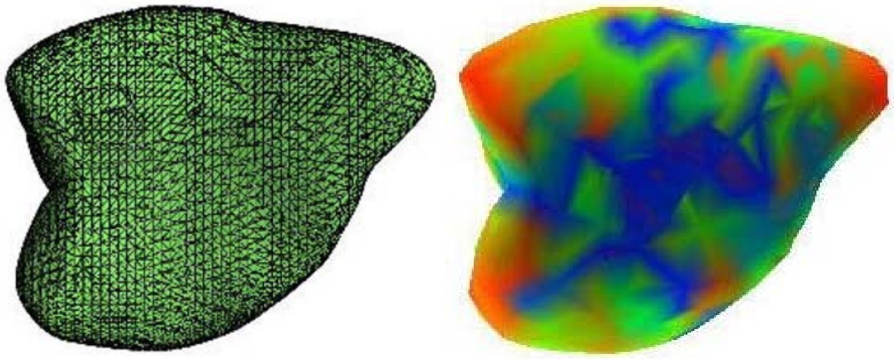


Figure 3.3 Gaussian Curvature.

Chapter 4 3D Reconstruction Based on Multi-view Imaging System

Early Trials

3D reconstruction has been widely used and has its applications in almost every industry. Many different methods have been developed and used in reconstruction of artifacts, buildings, industry measurement, human body, etc. All these technologies have their own advantages. Our case, which is building the 3D representation of human teeth, requires a safe, convenient and efficient way to reconstruct. Since image based 3D reconstruction satisfies these factors, plus the cameras nowadays can provide high resolution pictures even if they are small enough to be put into a human mouth, we choose computer vision method in our application.

Although cameras can be made fairly small, putting several cameras into a patient's mouth is not a convenience way. Therefore, to obtain multi-view image, we use mirrors to provide "virtual cameras" from the reflections.

Cylindrical Mirror

First we tried the cylindrical mirror in the previously mentioned Shree K. Nayar's work [26]. The mirror was made so both the rear and the near openings are the same size.

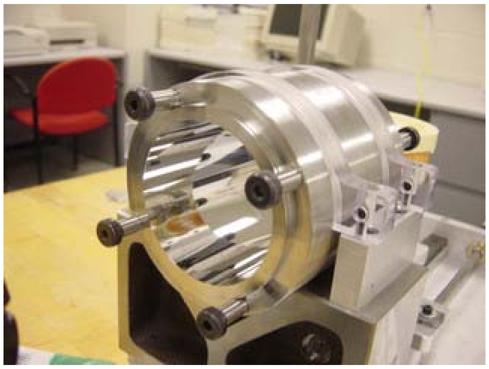


Figure 4.1 Cylindrical mirror.

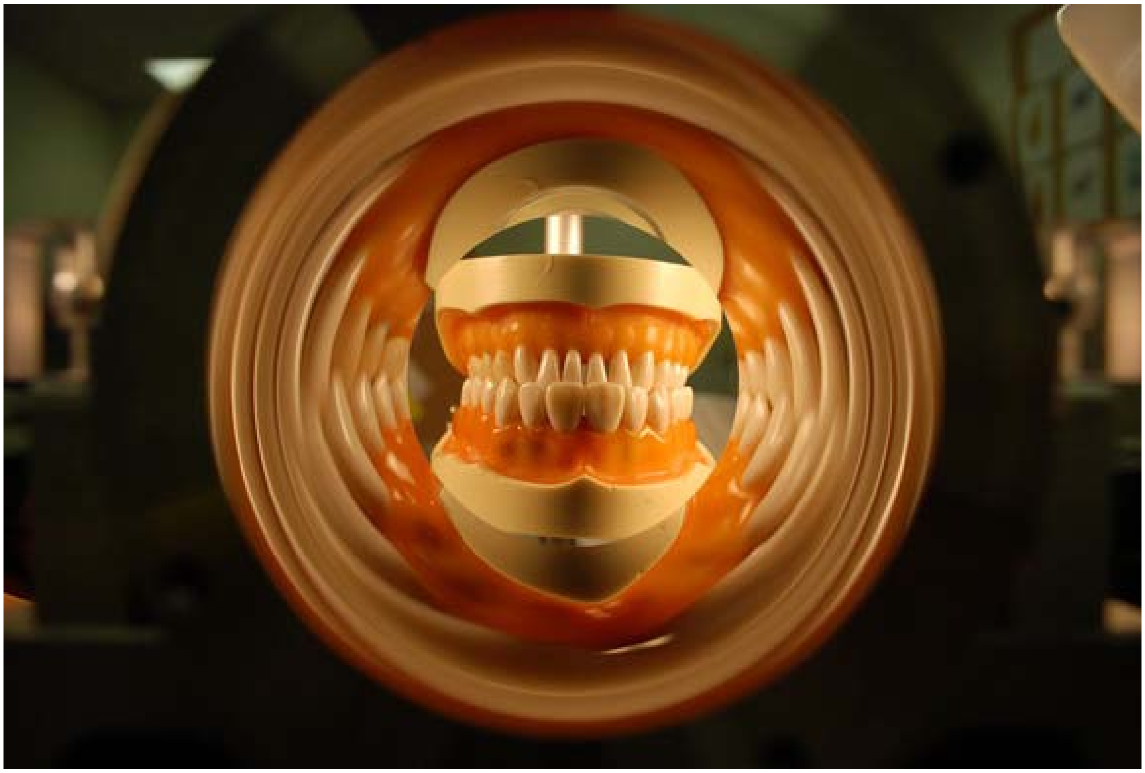


Figure 4.2 Image taken using cylindrical mirror.

Figure 4.1 is a picture of the cylindrical mirror and figure 4.2 shows the image that is taken using the mirror. The advantage of this mirror is that you can always find the corresponding points on the same diameter line of the image, for each radial slice of the captured image has its unique virtual cameras. This property requires that the optical axis must pass through the center of the mirror and be parallel to every mirror surface tangent plane. Although it sounds every easy to achieve, we found it very hard to calibrate. Furthermore, as we can see in Figure 4.2, the image is heavily blurred. This is because the distance between the object and the virtual camera are much longer than the distance between the object and the real camera, but the focusing distance of the real and virtual cameras are still the same. The blurring makes correspondence work very difficult. Due to the drawbacks of the cylindrical mirror, we decided to come up something new.

Square Glass Mirror

To find a design that is easy to calibrate and be able to provide us less blurring image is our goal. So we tried something shown in Figure 4.3: we taped four same flat, rectangular glass mirrors together to make them like a tube mirror. Figure 4.4 is the image taken using this mirror.

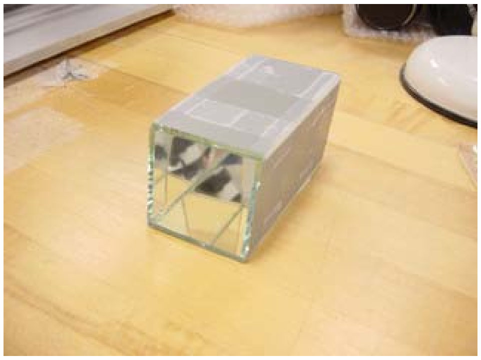


Figure 4.3 Square glass mirror.

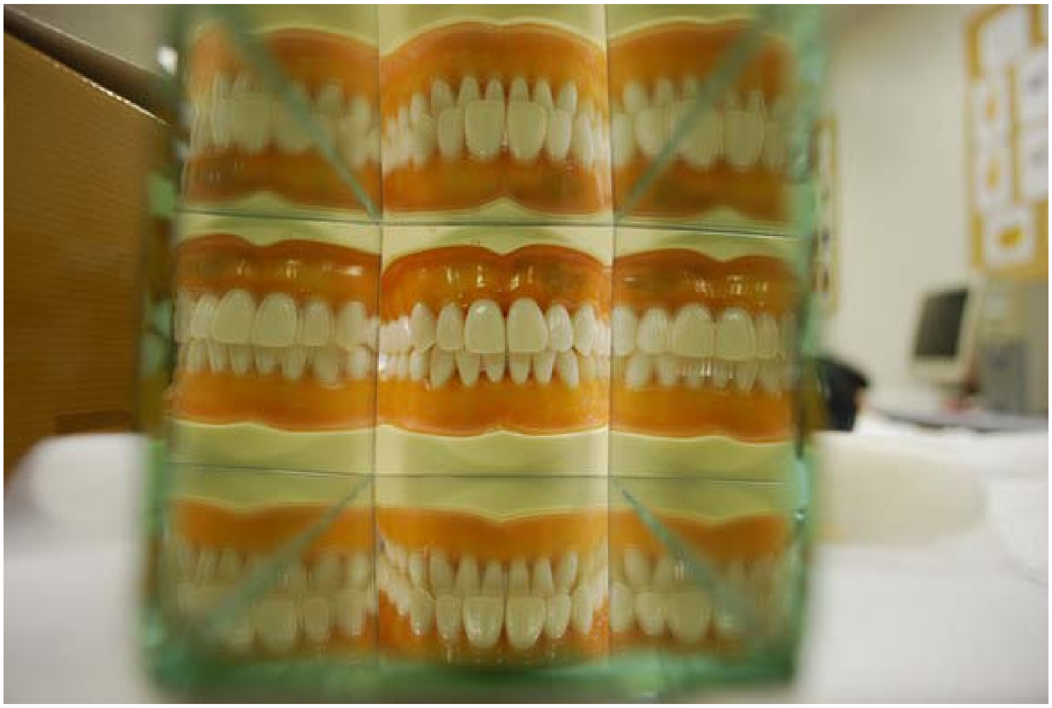


Figure 4.4 Image taken using square glass mirror.

This design gives us a much less complicated scheme because there are only four virtual cameras instead of many. Moreover, the reflected image looks much less blurred than the one reflected by cylindrical mirror. Still there is one drawback of this design. As you can see in Figure 4.4, the reflections have double images. This is because the surface of the mirror is glass. Shown in Figure 4.5, the glass surface will reflect part of the light, and refract the other light to the mirror surface. When the other part of the light being reflected back, they won't have the same trace as the part that is reflected by the glass surface.

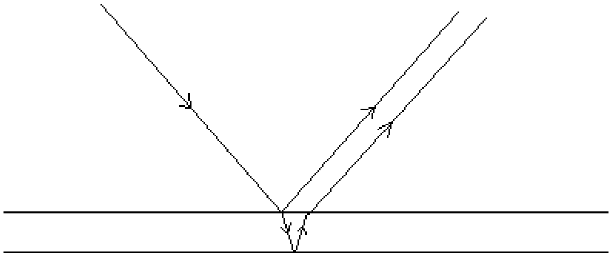


Figure 4.5 Double reflection.

Imaging System

Overview

To overcome the drawbacks of these early trials, we changed the material of the square mirror to stainless steel. Figure 4.6 shows the look of the square tube mirror and Figure 4.7 is an overview of the entire system. The camera, the mirror and the object are all on the same bracket with each of them has a standalone, movable support underneath. One can adjust the distances between them freely without calibrate the centers again.

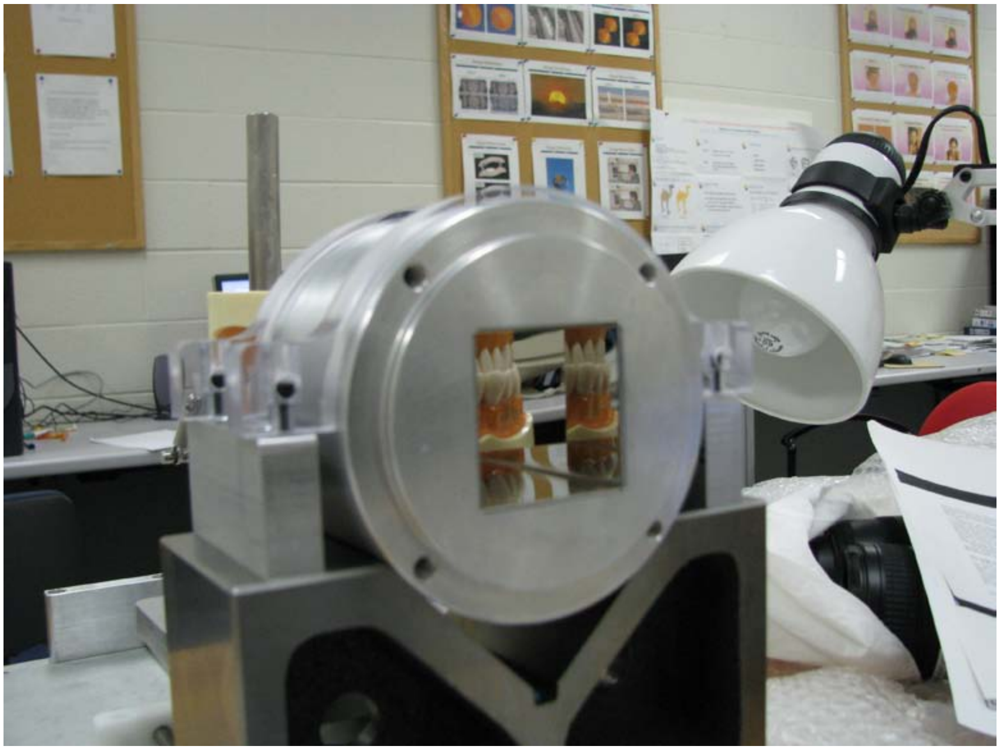


Figure 4.6 Square Tube Mirror.

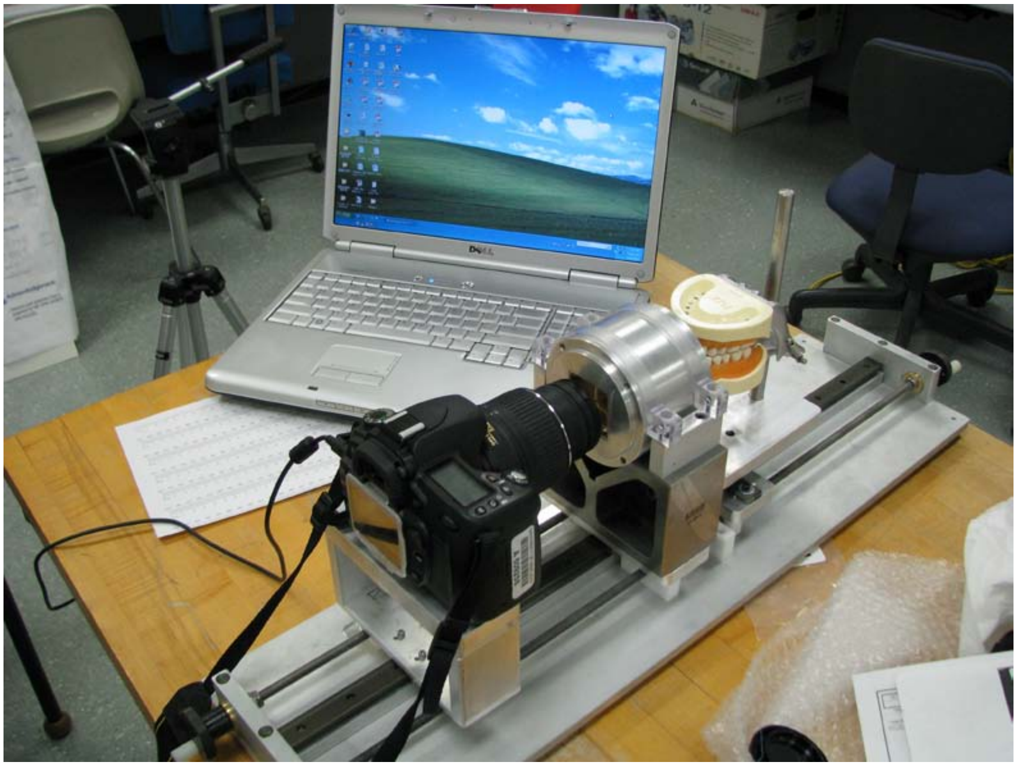


Figure 4.7 Overview of the multi-view imaging system.

Geometric Depth Computation

In our case, we can consider that the reflections are images taken by the virtual cameras. Since they are reflections, the geometry we have here posses some properties that we can take advantage of.

1. Since the virtual cameras can be considered as the reflections of the camera itself, the virtual cameras will have the same properties of the real camera. Once the position and the direction of the mirrors are known, we only need to do the calibration of the real camera to find the intrinsic parameters. Once the parameters of the real camera are known, the parameters of the virtual cameras are known.
2. Between two cameras there is a problem of finding a corresponding point viewed by one camera in the image of the other camera. In most camera configurations, the two cameras are aligned to have a common image plane, so that the corresponding points will be on the same scanline. This is called image rectification. In our multi-view imaging system, mirror reflection properties give the same advantage of image rectification without calibrating between the real and the virtual cameras. As shown in figure 4.8, the scene point and its corresponding points are always on the same scanline.

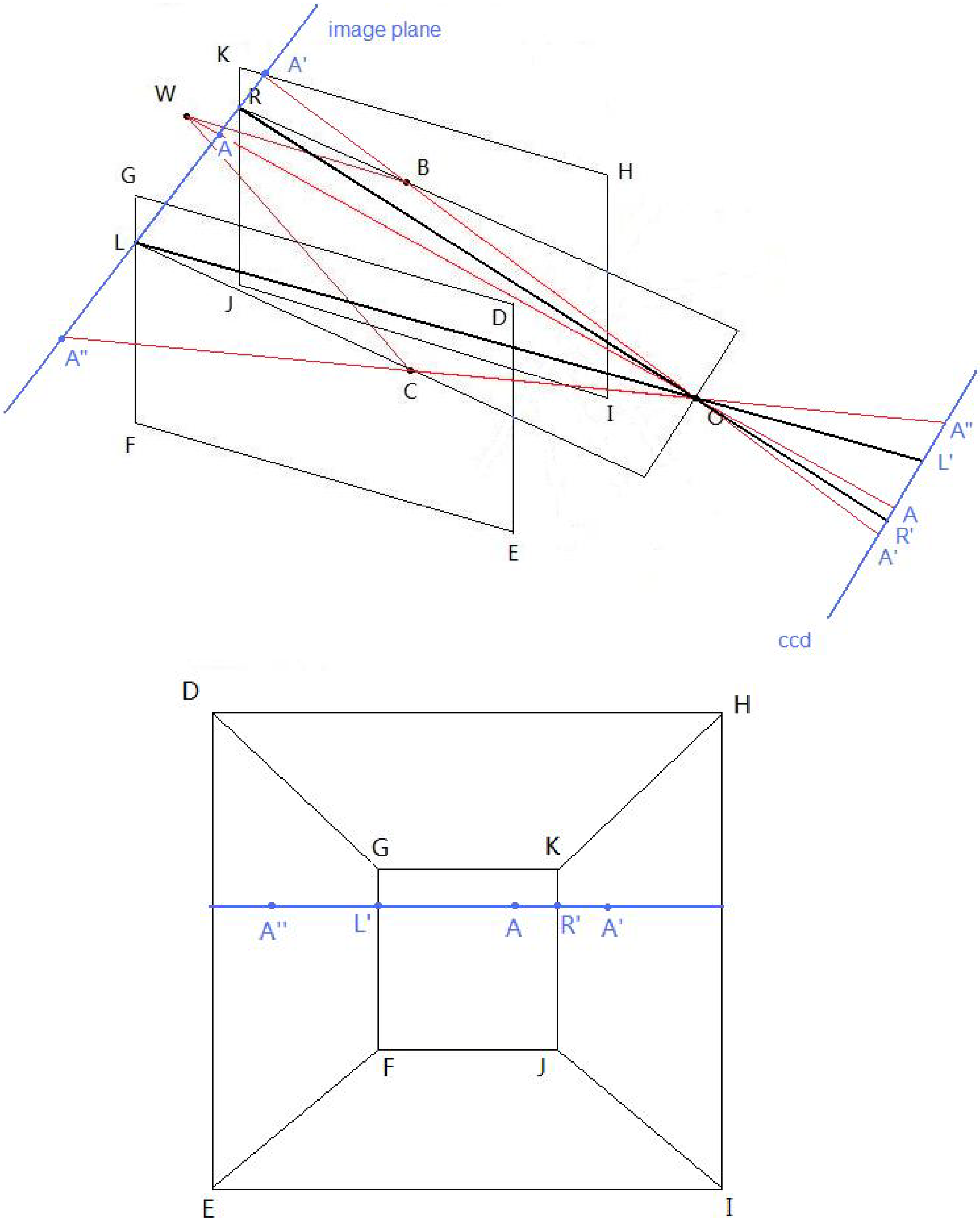


Figure 4.8 Collinear relationship between a pixel and its corresponding points.

With these special properties of our system, we found out a geometry based depth computing method. It is easier and faster than the classic stereo vision depth computing approach.

A cross section of the square tube mirror is shown in Figure 4.9. The optical axis lies in the center of the tube mirror. The angles between the optical axis and the mirrors are both g. O is the real camera center (center of projection), L and R are the virtual camera centers of left and right respectively. A is a point on the object. Assume that the length of AP is a’, and the length of AI is c’. The width of the mirror at the rear end (distance between the left and the right mirror) is w. Since OP is in the center, we have:

$$a'+c'=\frac{w}{2}+(d-h)\tan g\ . \tag{4.1}$$

B is the projection of A on the image plane, assume the length of BN is c. According to the properties of perspective projection, ΔOBN is similar to ΔOAI, we have:

$$\frac{c}{c'}=\frac{h}{d}\ . \tag{4.2}$$

Substitute (4.2) into (4.1): $a'+\frac{dc}{h}=\frac{w}{2}+(d-h)\tan g$. (4.3)

Since h can be obtained by an easy calibration process, c is known, if a’ can be computed, d is solved. Using triangle similarity, $\frac{a'}{b}=\frac{d-k}{k+x}$. (4.4)

Here, $b=\frac{w}{2}-h\tan g$. (4.5)

If we can get k and x, a’ is obtained.

Again, using triangle similarity, $\frac{b}{a}=\frac{k}{h-k}$, so $k=\frac{bh}{a+b}$. (4.6)

Looking closely at triangles XYL and XYO we can get: $x = 2b \sin g \cos g$. (4.7)

Substitute (4.4) (4.5) (4.6) (4.7) into (4.3) we get an equation with only d is unknown.

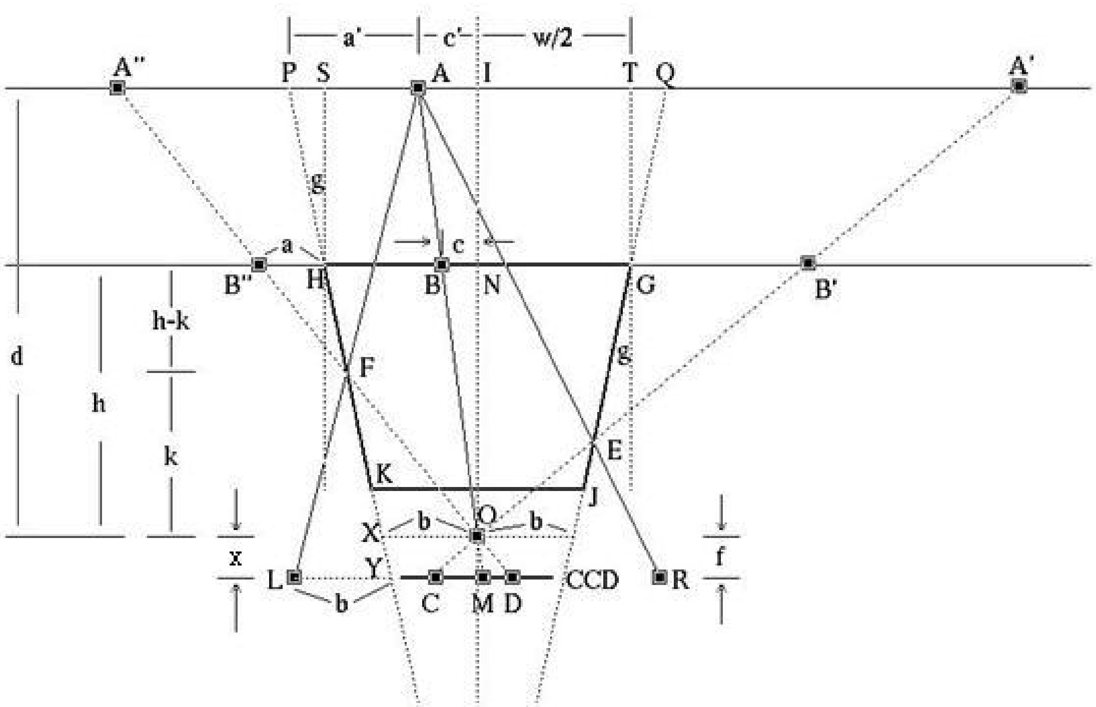


Figure 4.9 Geometric depth computation.

Reconstruction

Camera Calibration

Camera calibration is a necessary step in computer vision in order to obtain the parameters of the camera. In 1998, Zhengyou Zhang presented a technique for camera calibration [30]. It doesn’t require specialized knowledge of computer vision or 3D geometry. It provides good results and it is very easy to use. First we introduce the notation used in the calibration process.

1. Notation

A 2D and a 3D point are denoted by $m = [u, v]^T$ and $M = [x, y, z]^T$ respectively. $\tilde{m} = [u, v, 1]^T$ and $\tilde{M} = [X, Y, Z, 1]^T$ are the augmented vectors by adding 1 as the last element. A camera is considered as a pinhole, so the relationship between a 3D point *M* and its image projection *m* is given by

$$s\tilde{m} = A[R \quad t]\tilde{M} \,, \tag{4.8}$$

where *s* is an arbitrary scale factor. (R, t) is the rotation and translation which relates the world coordinate system to the camera coordinate system. It is called the extrinsic parameters. *A,* which is called the camera intrinsic matrix, is given by

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ & \beta & v_0 \\ & & 1 \end{bmatrix},$$

with (u_0, v_0) the coordinates of the principal point. The principle point is the intersection of the optical axis and the image plane. α and β are the scale factors in image *u* and *v* axes, which are related to the focal length f. γ the parameter describing the skewness of the two image axes.

2. Homography between the model plane and its image

In this calibration process, the camera needs to observe a planar pattern shown in a few different orientations. The plane which pattern lies on is called the model plane.

The model plane is assumed on *Z* = 0 of the world coordinate system. The i^{th} column of the rotation matrix *R* is denoted by r*i*. From (4.8), we have

$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=A\begin{bmatrix}r_1 & r_2 & r_3 & t\end{bmatrix}\begin{bmatrix}X\\Y\\0\\1\end{bmatrix}=A\begin{bmatrix}r_1 & r_2 & t\end{bmatrix}\begin{bmatrix}X\\Y\\1\end{bmatrix}.$$

For a point M on the model plane, we have $M=[X,Y]^T$ since Z is always equal to 0. In turn, $\tilde{M}=[X,Y,1]^T$. Therefore, a model point M and its image m are related by a homography H:

$$s\tilde{m}=H\tilde{M} \quad \text{with} \quad H=A[r_1 \quad r_2 \quad t]. \tag{4.9}$$

The 3×3 matrix H is defined up to a scale factor.

3. Constraints on the intrinsic parameters

A homography can be estimated by giving an image of the model plane. It is denoted by $H=[h_1 \quad h_2 \quad h_3]$ here. From (4.9), we have

$$[h_1 \quad h_2 \quad h_3]=\lambda A[r_1 \quad r_2 \quad t],$$

where λ is an arbitrary scalar. Using the knowledge that r_1 and r_2 are orthonormal, we have

$$h_1^T A^{-T} A^{-1} h_2 = 0, \tag{4.10}$$

$$h_1^T A^{-T} A^{-1} h_1 = h_1^T A^{-T} A^{-1} h_2. \tag{4.11}$$

These are the two basic constraints on the intrinsic parameters, given one homography. Because a homography has eight degrees of freedom and there are six extrinsic parameters (three for rotation and three for translation), we can only obtain two constraints on the intrinsic parameters.

4. Solving Camera Calibration

Here we ignore the lens distortion to make the computation simpler.

Let

$$B = A^{-T} A^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma (v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} & -\frac{\gamma (v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}. \tag{4.12}$$

Notice that B is symmetric, so here we define *b*, a 6D vector

$$b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T . \tag{4.13}$$

Denote the i^{th} column vector of H by $h_i = [h_{i1}, h_{i2}, h_{i3}]^T$. Then we have

$$h_i^T B h_j = v_{ij}^T b , \tag{4.14}$$

with

$$\begin{aligned} v_{ij} = [& h_{i1} h_{j1}, h_{i1} h_{j2} + h_{i2} h_{j1}, h_{i2} h_{j2}, \\ & h_{i3} h_{j1} + h_{i1} h_{j3}, h_{i3} h_{j2} + h_{i2} h_{j3}, h_{i3} h_{j3}]^T . \end{aligned}$$

Therefore, (3) and (4) can be rewritten as:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0. \tag{4.15}$$

If we have *n* images of the model plane, by stacking *n* such equations as (4.15) we have

$$Vb = 0, \tag{4.16}$$

where V is a $2n \times 6$ matrix. If $n \geq 3$ we will have in general a unique solution b defined up to a scale factor. In fact normally we take 7 - 15 pictures of the pattern and use around 10 images for calibration to obtain a more accurate result. The solution to (4.16) is the eigenvector of $V^T V$ associated with the smallest eigenvalue.

Once b is estimated, A can be computed.

Once A is computed, we can compute the extrinsic parameters for each image:

$$r_1 = \lambda A^{-1} h_1 ,$$

$$r_2 = \lambda A^{-1} h_2 ,$$

$$r_3 = r_1 \times r_2 ,$$

$$t = \lambda A^{-1} h_3 ,$$

here, $\lambda = \frac{1}{\left\| A^{-1} h_1 \right\|} = \frac{1}{\left\| A^{-1} h_2 \right\|}$.

Correspondence

As the camera parameters are known, the only challenge is to find the corresponding points between the images. Unfortunately, the reliable identification of corresponding points is a very difficult problem. Especially with objects that have solid color and specular reflection material, like human teeth. We use a feature point based approach to achieve better results.

1. Canny Edge Detection

The Canny edge detection [31] algorithm is known to many as the optimal edge detector. The purpose of the method is to detect edges with noise suppressed at the same time. The Canny Operator has the following goals:

(a) Good Detection: the ability to locate and mark all real edges.

(b) Good Localization: minimal distance between the detected edge and real edge.

(c) Clear Response: only one response per edge.

The approach is based strongly on convolution of the image function with Gaussian operators and their derivatives. This is a multi step procedure.

Consider the Gaussian function in two dimensions:

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad .$$

The directional derivative of *G(x,y)* along *n* is*:*

$$G_n(x,y) = \frac{\partial G(x,y)}{\partial n} = n \cdot \nabla G(x,y) \, ,$$

where $n = (\cos\theta, \sin\theta)^T$, $\nabla G(x,y) = (\frac{\partial G(x,y)}{\partial x}, \frac{\partial G(x,y)}{\partial y})^T$.

After the convolution filtering with *G(x,y), f(x,y)* has a directional derivative along *n*:

$$\frac{\partial [G(x,y) * f(x,y)]}{\partial n} = G_n * f(x,y) \, .$$

We change the direction of *n* in order to get the maximum of $G_n * f(x,y)$, which is the vertical gradient. From

$$\frac{\partial[G_n * f(x,y)]}{\partial\theta} = \frac{\partial[\cos\theta \dfrac{\partial G(x,y)}{\partial x} * f(x,y) + \sin\theta \dfrac{\partial G(x,y)}{\partial y} * f(x,y)]}{\partial\theta} = 0$$

we get

$$\tan\theta = \frac{(\partial G(x,y)/\partial y) * f(x,y)}{(\partial G(x,y)/\partial x) * f(x,y)},$$

$$\cos\theta = \frac{(\partial G(x,y)/\partial x) * f(x,y)}{|\nabla G(x,y) * f(x,y)|},$$

$$\sin\theta = \frac{(\partial G(x,y)/\partial y) * f(x,y)}{|\nabla G(x,y) * f(x,y)|}.$$

Therefore, the vertical gradient of $G(x,y) * f(x,y)$ is

$$n = \frac{\nabla G(x,y) * f(x,y)}{|\nabla G(x,y) * f(x,y)|} \quad .$$

Substitute *n* into the directional derivative equation:

$$|G_n * f(x,y)| = \left| \cos\theta \frac{\partial G(x,y)}{\partial x} * f(x,y) + \sin\theta \frac{\partial G(x,y)}{\partial y} * f(x,y) \right| |$$

$$= | \nabla G(x,y) * f(x,y).$$

Here we can see that the key of canny operator is to compute $\nabla G(x,y) * f(x,y)$. We separate the two dimensional convolution $\nabla G(x,y) * f(x,y)$ into two one dimension filter:

$$\frac{\partial G(x,y)}{\partial x} = kxe^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} = h_1(x)h_2(y),$$

$$\frac{\partial G(x,y)}{\partial y} = kye^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} = h_1(y)h_2(x),$$

where k is a constant,

$$h_1(\omega) = \sqrt{k}\omega e^{-\frac{\omega^2}{2\sigma^2}}, \ \ h_2(\omega) = \sqrt{k} e^{-\frac{\omega^2}{2\sigma^2}} .$$

Then we get the convolution of f(x,y) and these two templates separately:

$$E_x(m,n) = \frac{\partial G(m,n)}{\partial x} * f(m,n) = [h_1(m)h_2(n)] * f(m,n),$$

$$E_y(m,n) = \frac{\partial G(m,n)}{\partial y} * f(m,n) = [h_1(n)h_2(m)] * f(m,n).$$

The above-mentioned convolution can be acquired along the x axis and y axis respectively.

Let

$$A(m,n) = \sqrt{E_x^2(m,n) + E_y^2(m,n)}, \qquad \theta(m,n) = \tan^{-1}\frac{E_y(m,n)}{E_x(m,n)}.$$

Canny operator sets two thresholds to detect the edge points. The steps are as follows:

1. Compute the gradient mask and the direction of gradient for each point of the image.
2. Use non-maximum suppression to extract local extreme points.
3. Set two thresholds to extract the edge.

The edge points are considered as feature points. Because of the feature they have, it is much easier to find their corresponding points.

2. Constraint Conditions

Figure 4.10 shows an image taken by our system. In Figure 4.11 each area of the image is labeled. Area 0 is the real image; area 1, 2, ,3 , 4 are the images reflected once by the upper left, right, lower mirror respectively; area 5, 6, 7, 8 are the images reflected twice which we ignored since the geometry of these reflections are too complicated. Assume that on scanline L, we have two points A and B, and both have their corresponding points A1, B1 in area 3 and A2, B2 in area 2. A3 and A4 are the corresponding points of A in area 1 and 4. The corresponding points in our system should have the following constraints [32]:

(1). Ordering Constraint: For opaque surfaces the order of neighboring correspondences on the corresponding epipolar lines is always preserved. For example, if the index on the scanline l: $u_A < u_B$, then $u_{A1} > u_{B1}$, $u_{A2} > u_{B2}$. This is because the mirror reflection makes the image inversed.

(2). Disparity Limit: The search band is restricted along the epipolar line because the observed scene has only a limited depth range. For example, if we are looking for the corresponding point of A in area 2, we don’t need to search the entire scanline in area 2, we only need to search pixels in a certain threshold depending on the depth range.

(3). Variance limit: The difference of the depth computed by the corresponding point in each area should be less than a threshold. For example, A1, A2, A3, A4 each can be used as a corresponding point of A and compute a depth of A. We compute the variance of the four depth and it must be smaller than a threshold otherwise at least one of the depth is wrong.

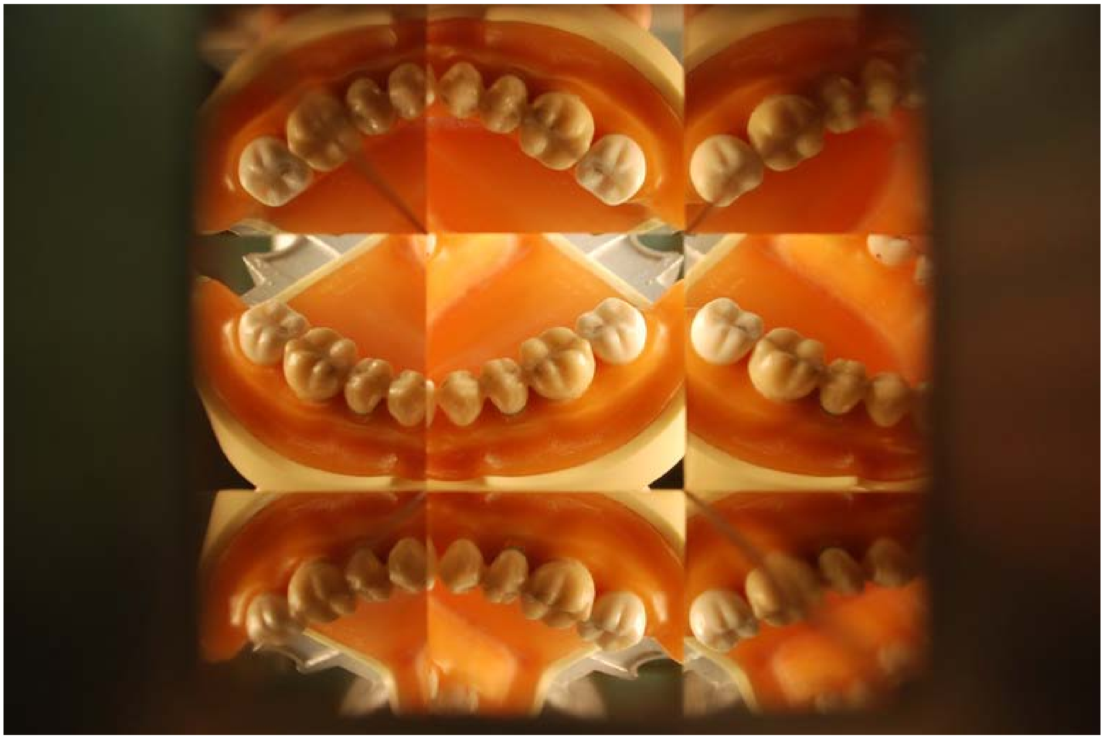


Figure 4.10 Image taken by the Square Tube Mirror.

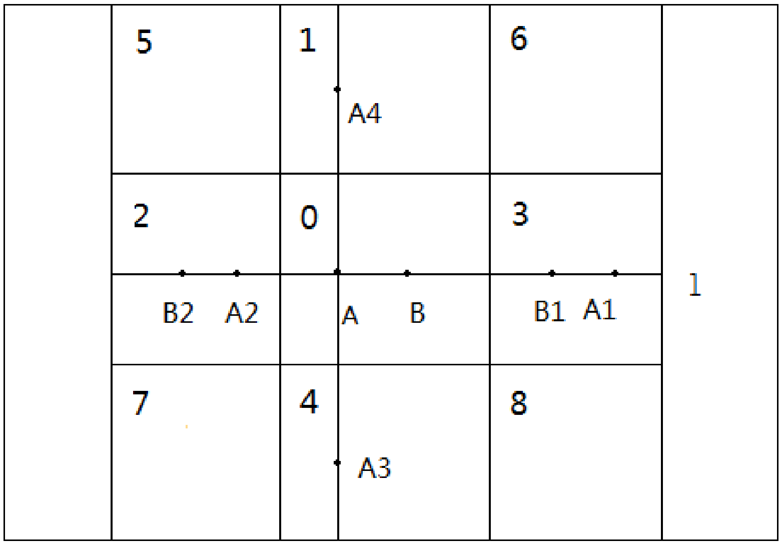


Figure 4.11 Image area labeling.

3. Stereo Matching

After the feature points are determined, the correspondence of these points can be found using stereo matching approaches. The constraints of the previous section can be used to reduce the computation complexity and refine the existing results. We use two intensity based approaches for stereo matching:

(1) Fast Normalized Cross-Correlation

Normalized cross-correlation [34] is an effective and simple method as a similarity measure. In our application, the mirror reflection factor gives intensity reduction to the reflected image, but normalized cross-correlation is invariant to linear brightness and contrast variations. This approach provides good matching results for our feature points.

The use of cross-correlation for template matching is motivated by squared Euclidean distance:

$$d_{f,t}^{2}(u,v)=\sum_{x,y}[f(x,y)-t(x-u,y-v)]^{2} ,$$

where *f* is the source image in the region and the sum is over *x,y* under the region of destination image *t* positioned at *u,v*.

Here we expand d^2:

$$d_{f,t}^{2}(u,v)=\sum_{x,y}[f^{2}(x,y)-2f(x,y)t(x-u,y-v)+t^{2}(x-u,y-v)],$$

where the term $\sum t^{2}(x-u,y-v)$ is constant. If the term $\sum f^{2}(x,y)$ is approximately constant then the remaining cross-correlation term

$$c(u,v)=\sum_{x,y} f(x,y)t(x-u,y-u) \qquad (4.17)$$

is a measure of the similarity between the source image and the destination image.

Although (4.17) is a good measure, there are several disadvantages to using it for matching:

1). If the image energy $\sum f^2(x, y)$ varies with position, matching using (4.17) can fail. For example, the correlation between the destination image and an exactly matching region in the source image may be less than the correlation between the destination image and a bright spot.

2). The range of $c(u, v)$ is dependent on the size of the region.

3). Equation (4.17) is not invariant to changes in image amplitude such as those caused by changing lighting conditions across the image sequence.

The correlation coefficient overcomes these difficulties by normalizing the image and feature vectors to unit length, yielding a cosine-like correlation coefficient:

$$\gamma(u,v) = \frac{\sum_{x,y}[f(x,y) - \bar{f}_{u,v}][t(x-u, y-v) - \bar{t}]}{\left\{\sum_{x,y}[f(x,y) - \bar{f}_{u,v}]^2 \sum_{x,y}[t(x-u, y-v) - \bar{t}]^2\right\}^{0.5}}, \tag{4.18}$$

where $\bar{t}$ is the mean of the destination image in the region and $\bar{f}_{u,v}$ is the mean of $f(x, y)$ in the region under the feature. (4.18) is what we referred as the normalized cross-correlation.

(2) Simulated Annealing

As mentioned in the previous section, the corresponding points in the source image and the destination image must lie on the same scanline. If we take a close look at the intensity profiles from the corresponding scanline of the image pair, the intensity profiles differ only by a horizontal shift and a local foreshortening [36].

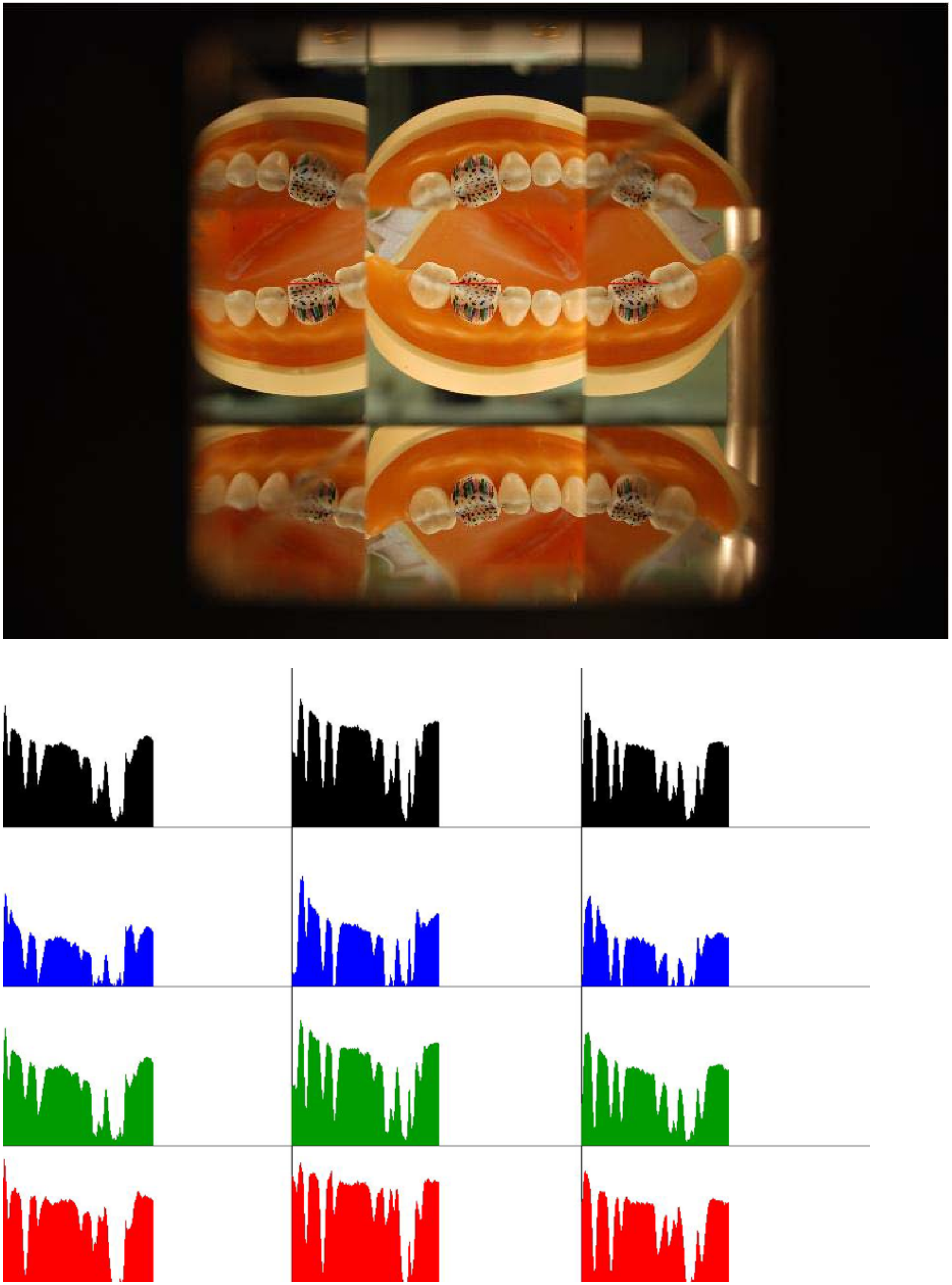


Figure 4.12 Continuousness of the corresponding points on the same scanline.

As shown in Figure 4.12, the similarity of the pair is continuous, which means an optimization process would be suitable. Actually, in 1987, Barnard [35] attempted matching the parallel stereo images using simulated annealing. He defined an energy function as:

$$E_{ij} = | I_L(i,j) - I_R(i, j + D(i,j)) | + \lambda | \nabla D(i,j) | ,$$

where $I_L(i,j)$ denotes the intensity value of the source image at *(i,j)*, and $I_R(i,k)$ denotes the intensity value of the destination image at the same row but at the k-th column; *D(i,j)* is the disparity value (or horizontal shift in this case) at the *ij*-position of the source image. So this is a constrained optimization problem in which the only constraint being used is a minimum change of disparity values *D(i,j)*.

Chapter 5 Implementation Results

The camera used for implementation is Nikon D80 with an AF-S DX NIKKOR 18-55mm f/3.5-5.6G VR lens. We chose the 18-55mm lens is because it has the shortest focusing distance (0.28m) among all non-wide-angle Nikon lenses. Shorter focusing distance means the object can be placed closer to the camera and also it makes the angle of view larger. The larger the angle of view is, the wider the reflection areas are. As shown in Figure 4.10, the lens provides reflected areas almost the same size as the directly captured area.

Before capturing the images of teeth, one must make sure the optical axis of the camera is parallel to the mirrors. After the calibration, the intrinsic parameters of the camera are known. If the extrinsic parameters of the real and virtual cameras can be computed, the orientation and position of the mirrors are known. As shown in Figure 5.1, we put a chess board in front of the mirror and took several pictures.

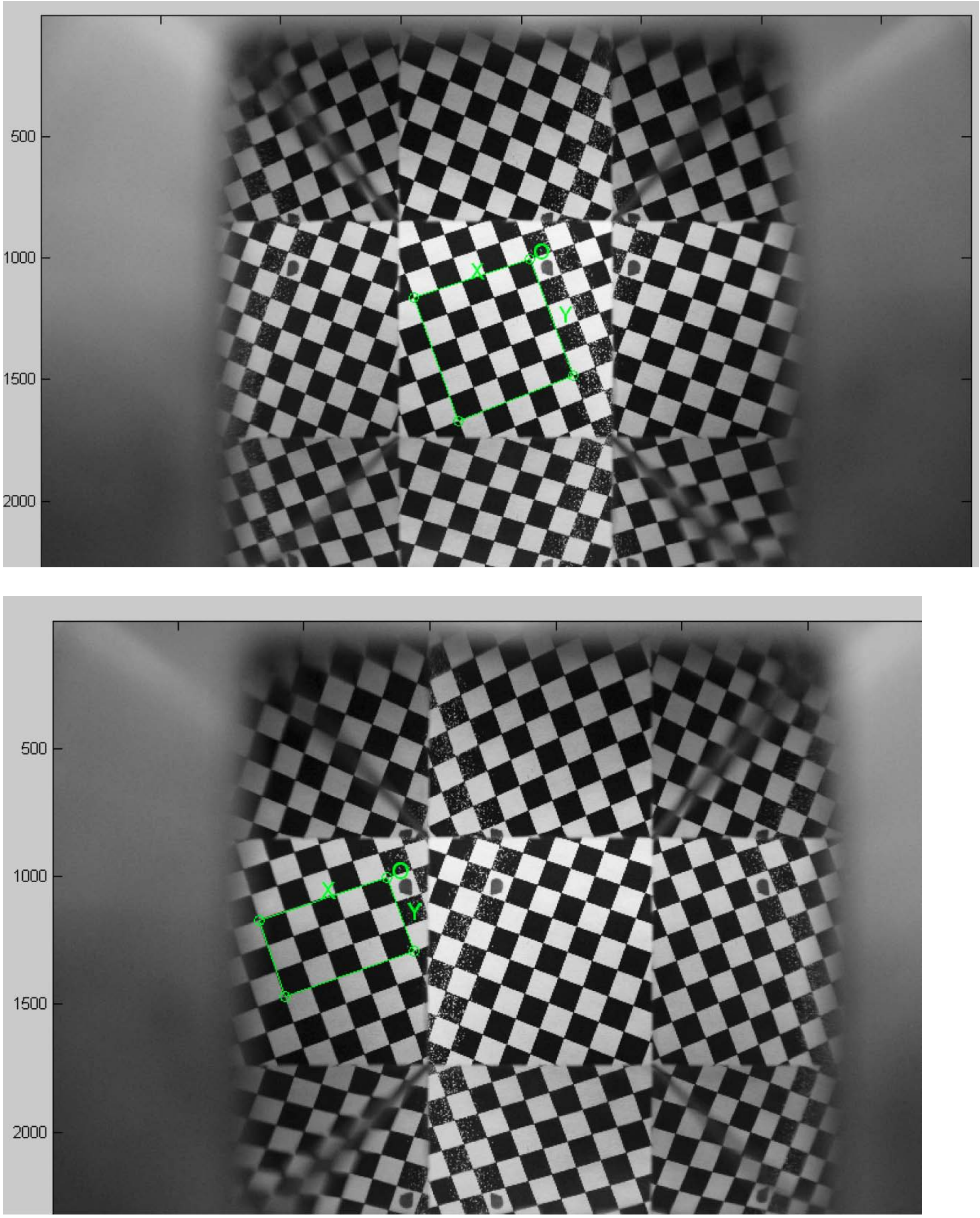
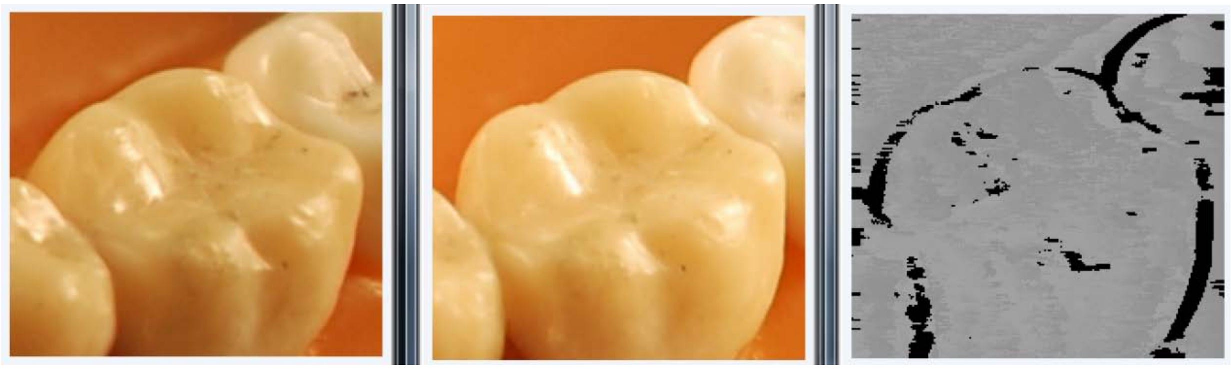


Figure 5.1 Mirror calibration.

From the region extraction, the extrinsic matrix of the real camera and virtual cameras are obtained. Notice that when doing the region extraction of the reflected areas, the image must be flipped because of the reverse of mirror.

At first we tried to reconstruct a single, non-textured tooth. As shown in Figure 5.1, because of the specular reflection and the solid colored surface the tooth had, the results are almost non-acceptable.



Reflected image Real image Depth map

Figure 5.2 Reconstruction of a single, non-textured tooth.

The challenge of reconstructing the human teeth is not only overcoming the surface they have, but also the small distance variation on the teeth compare to the focal length. This makes the results very vulnerable because an error of one pixel in the disparity will give the distance an error of 3mm. Even a large human tooth is only about 10-20mm long/wide. So adding more features on the surface is necessary. Figure 5.3 and 5.4 shows the results of a colored tooth.

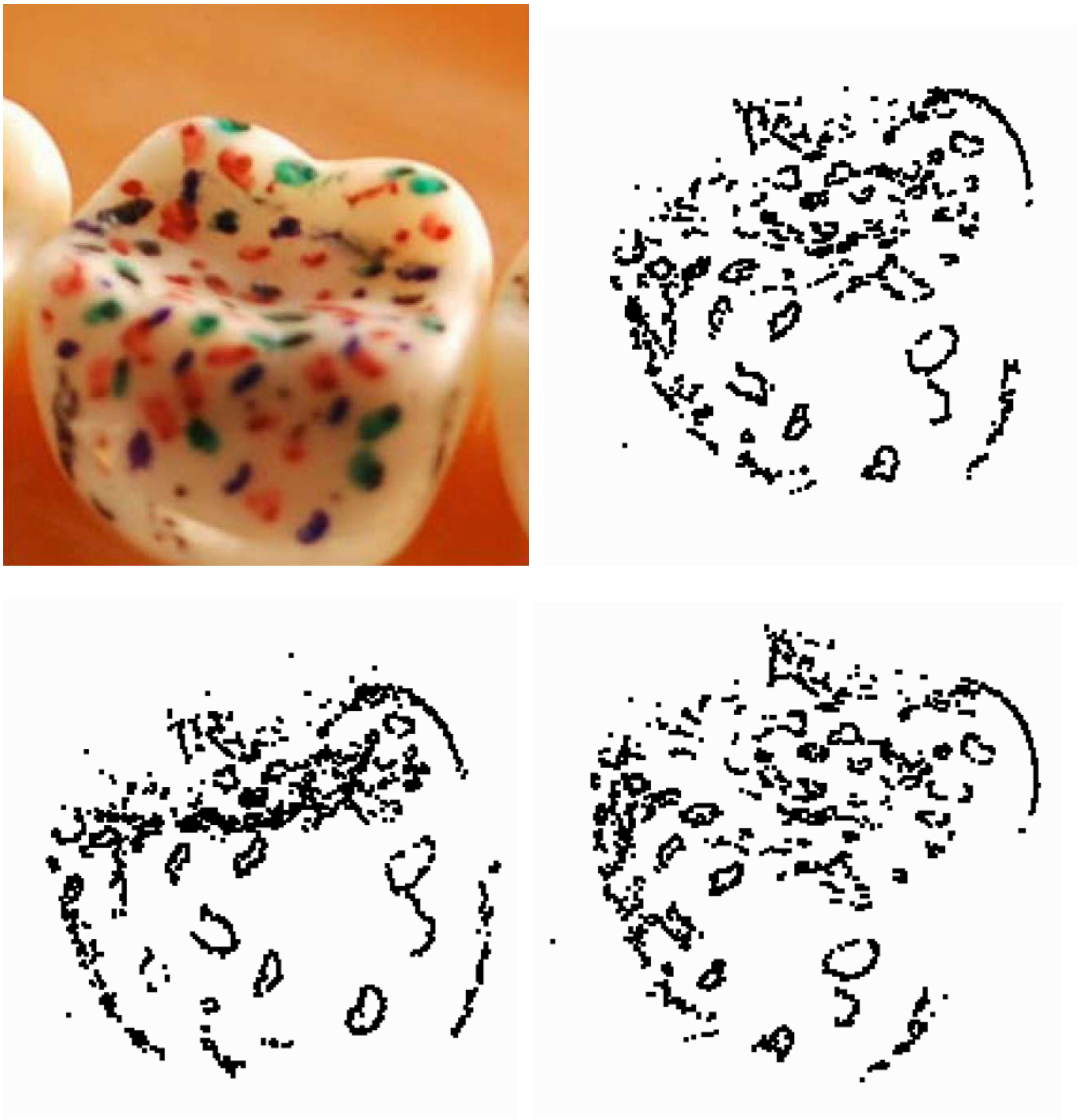


Figure 5.3 Original object and feature point cloud (from different angles).

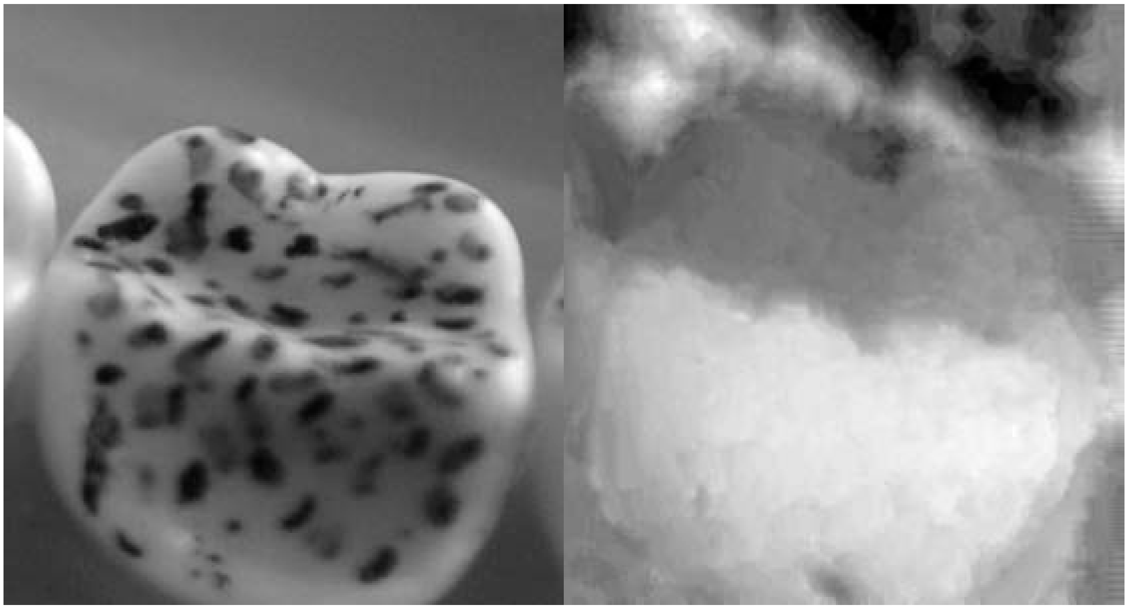


Figure 5.4 Original object and the disparity map.

Chapter 6 Concluding Remarks

A new system of reconstructing the 3D image of dental casts as well as the simplification and interpolation of the standard model are presented. The scanned standard model is simplified by an iterative vertex pair contraction process using quadric error metrics. Then in order to acquire the curvature of the teeth surface, an interpolation process is applied to obtain the local parametrization of each point on the surface. Once the local parametrization is obtained, the surface curvature of any direction is known. The multi-view imaging system uses a square tube mirror to capture multi-view images. Depth can be computed by finding the corresponding points in the real area and the reflected areas. Notice that it is very difficult to find the corresponding points for objects with solid color and specular reflection like human teeth. In this case, structured light can create variation for the intensity and gives many more feature points. Although good results are achieved so far, there is more work to be done. Features of each tooth must be found in order to differentiate them and identify which teeth are reconstructed during the process. Because the camera cannot capture all parts of the teeth, the known shape should be used to transform the standard model.

References

[1] P. F. van der Stelt and S. M. Dunn, “3D-imaging in dental radiography," in *Advances in Maxillofacial Imaging*, A. G. Farman, ed., pp. 367-372, Elsevier Science B. V., 1997.

[2] R. L. Webber, R. A. Horton, D. A. Tyndall, and J. B. Ludlow, “Tuned-aperture computed tomography (TACT). theory and application for three-dimensional; dento-alveolar imaging," *Dentomaxillofac. Radiol*. 26, pp. 51-62, 1997.

[3] A. A. Goshtasby, S. Nambala, W. G. deRijk, and S. D. Campbell, “A system for digital reconstruction of gypsum dental casts," *IEEE Transactions on Medical Imaging* 16, pp. 664-674, Oct 1997.

[4] D. Laurendeau and D. Possart, “A computer-vision technique for the acquisition and processing of 3D profiles of dental imprints: An application in orthodontics," *IEEE Transactions on Medical Imaging* 10, pp. 453-461, Sept 1991.

[5] Sameh M. Yamany, Aly A. Farag, David Tasman, Allan G. Farman: A 3D-reconstruction System for the Human Jaw Using a Sequence of Optical Images. *IEEE Trans. Med. Imaging* 19(5): 538-547 (2000).

[6] Michael Garland and Paul S. Heckbert. Surface Simplification using Quadric Error Metrics. In *SIGGRAPH '97 Conference Proceedings* (1997). pp. 209-216.

[7] Fuhua (Frank) Cheng, Fengtao Fan, Shuhua Lai, Conglin Huang, Jiaxi Wang, Jun-Hai Yong: Progressive Interpolation Using Loop Subdivision Surfaces. *GMP* 2008:526-533.

[8] Loop, C. T.: Smooth subdivision surface based on triangle, *Master's thesis, Department of Mathematics, University of Utah*, 1987.

[9] Qi, D.; Tian, Z.; Zhang, Y.; Zheng, J. B.: The method of numeric polish in curve fitting, *Acta Mathematica Sinica*, 18, 1975, 173-184 .

[10] de Boor, C.: How does Agee's method work? 1979 *Army Numerical Analysis and Computers Conference, ARO Report* 79-3, Army Research Office, 299-302.

[11] Lin, H.; Wang, G.; Dong, C.: Constructing Iterative Non-Uniform B-spline Curve and Surface to Fit Data Points, *SCIENCE IN CHINA*, Series F, 47, 2004, 315-331.

[12] Levin, A.: Modified subdivision surfaces with continuous curvature *SIGGRAPH, ACM Transactions on Graphics*, 2006, 1035-1040.

[13] Catmull E, Clark J, Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer-Aided Design* 1978, 10(6):350-355.

[14] Doo D, Sabin M, Behavior of recursive division surfaces near extraordinary points, *Computer-Aided Design* 1978, 10(6):356-360.

[15] Dyn N, Levin D, Gregory JA, A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control, *ACM Trans. Graphics* 1990, 9(2):160-169.

[16] Zorin D, Schroder P, Sweldens W, Interpolating Subdivision for Meshes with Arbitrary Topology, *Computer Graphics, Ann. Conf. Series*, 1996, 30:189-192.

[17] Kobbelt L, Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology, *Comput. Graph.* Forum 1996, 5(3):409-420.

[18] Jarek Rossignac and P. Borrel. Multi-resolution 3D Approximation for Rendering Complex Scenes. In *Geometric Modeling in Computer Graphics. Springer Verlag*, 1993. pp. 455-465.

[19] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of Triangle Meshes. In *SIGGRAPH ’92 Conference Proceedings* (1992). pp. 65-70.

[20] Halstead M, Kass M, DeRose T, Efficient, fair interpolation using Catmull-Clark surfaces, *Proc.SIGGRAPH 1993*, 47-61.

[21] Nasri AH, Surface interpolation on irregular networks with normal conditions, *Computer Aided Geometric Design* 1991, 8:89-96.

[22] Zheng J, Cai YY, Interpolation over arbitrary topology meshes using a two-phase subdivision scheme, *IEEE Trans. Visualization and Computer Graphics* 2006, 12(3):301-310.

[23] Lai S, Cheng F, Similarity based Interpolation using Catmull-Clark Subdivision Surfaces, *The Visual Computer* 2006, 22(9):865-873.

[24] Litke N, Levin A, Schröder P, Fitting Subdivision Surfaces, *Proc. Visualization* 2001, 319-324.

[25] Yuuki Uranishi, Mika Naganawa, Yoshihiro Yasumuro, Masataka Imura, Yoshitsugu Manabe, Kunihiro Chihara: Three-Dimensional Measurement System Using a Cylindrical Mirror. *SCIA* 2005: 399-408

[26] S. Kuthirummal and S. K. Nayar, "Multiview Radial Catadioptric Imaging for Scene Capture," *ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH*), Jul, 2006.

[27] Stam J, Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values, *Proc. SIGGRAPH* 1998.

[28] Reif, U.: A unified approach to subdivision algorithms near extraordinary points, *Computer Aided Geometric Design*, 12(2), 1995, 153-174.

[29] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330-1334, 2000.

[30] Canny, J., A Computational Approach To Edge Detection, I*EEE Trans. Pattern Analysis and Machine Intelligence*, 8:679-714, 1986.

[31] Marc Pollefeys, “Exploiting scene constraints”, Tutorial Notes

[32] Wikipedia, “Image rectification”, http://en.wikipedia.org/wiki/Image_rectification

[33] J. P. Lewis, “Fast Template Matching”, *Vision Interface*, p. 120-123, 1995.

[34] Barnard, S. T. (1987), Stereo Matching by Hierarchical, Microcanonical Annealing, *Int. Joint Conf. on Artificial Intelligence, Milan,* Italy, pp. 832-835.

[35] Bob Fisher, “Stereo matching”, *http://homepages.inf.ed.ac.uk/rbf*

Vita

- Date and place of birth: 10/15/1982, Chengdu, China
- Education: BS, Computer Science, Sichuan University
- Name: Conglin Huang