

**An Interactive 3D Geometric Model Acquisition and  
Registration System**

by

Yushuang Liu

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF  
**Master of Science**

in

THE FACULTY OF GRADUATE STUDIES  
(Department of Computer Science)

We accept this thesis as conforming  
~~to the required~~ standard

**The University of British Columbia**

May 2003

© Yushuang Liu, 2003

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of computer science

The University of British Columbia  
Vancouver, Canada

Date April 25<sup>th</sup>, 2003

# Abstract

The acquisition of geometric information from real world objects has now become a major way of modeling complex scenes and environments. Unfortunately, most optical methods for geometric model acquisition require the combination of partial information from different view points, in order to obtain a single, coherent model. This, in turn, requires the registration of partial models into a common coordinate frame, a process that is usually done offline. As a consequence, holes due to undersampling and missing information often cannot be detected until after the registration process.

This thesis tries to solve the 3D registration problem using graphics hardware to achieve an interactive speed, so that online next-best-view planning and hole detection during the scanning process can be conducted. The system implemented realizes effective model reconstruction and refinement by interactively using a stereo camera setup and a hardware accelerated depth-map based registration algorithm. The emphasis is put on the hardware accelerated 3D range registration algorithm which can be categorized as a variant of the well known Iterated Closest Point (ICP) algorithm. This work contributes to the field of 3D geometric model acquisition in that it proposes a fast method that leverages the function and structure of modern graphics hardwares. Technically, graphics hardware is used in two operations. The first is to compute a rigid transformation using modern graphics rendering pipeline. The second is to combine two rendered depth images to compute quickly the absolute difference of z-values of each projected points in the overlap region in the underlying two depth images, where the computation is used to obtain an error metric for a specific candidate transformation during numerical iteration. This error metric is based on depth images which are rendered by point-based rasterization.

This system currently performs roughly one registration per second, and is therefore fast enough for on-the-fly evaluation by the user. Given more time, the same method is also capable of producing full geometric models at higher quality.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Contents</b>	<b>iii</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Related Work . . . . .	4
1.2.1 3D Measurement Techniques . . . . .	4
1.2.2 Previous Registration Methods . . . . .	6
1.3 Contributions and System Overview . . . . .	8
1.4 Thesis Organization . . . . .	9
<b>2 Graphics Rendering Pipeline</b>	<b>10</b>
2.1 Rendering Pipeline Overview . . . . .	10
2.2 Coordinate Systems in 3D Rendering Pipeline . . . . .	12
2.2.1 Geometry Related Coordinate Systems. . . . .	13
2.2.2 Texture Coordinate System . . . . .	15

<b>3</b>	<b>Range Data Acquisition</b>	<b>17</b>
3.1	Digiclops Stereo Vision System . . . . .	17
3.1.1	Digiclops Stereo Processing . . . . .	18
3.2	Integrating Digiclops into the System . . . . .	20
3.3	Stereo Processing Environment . . . . .	23
3.3.1	Coordinate Systems Used in the Triclops Library . . . . .	23
3.3.2	Stereo Processing Parameters . . . . .	23
<b>4</b>	<b>Range-Based Registration</b>	<b>25</b>
4.1	Problem Specification and General Approaches . . . . .	26
4.2	Range-Based ICP Algorithm . . . . .	28
4.2.1	Existing ICP Variants . . . . .	29
4.2.2	Overview of Our Range Based ICP Algorithm . . . . .	35
4.2.3	Coordinate Systems Involved . . . . .	38
4.2.4	Parameter Space . . . . .	42
4.2.5	Surface Consistency Measure Function . . . . .	44
4.2.6	Optimization Algorithm . . . . .	46
<b>5</b>	<b>Hardware Acceleration</b>	<b>51</b>
5.1	Off Loading Geometric Computations onto GPU . . . . .	51
5.2	Accelerating Surface Consistency Evaluation Using NVIDIA Register Combiners Mechanism . . . . .	52
<b>6</b>	<b>Model Integration</b>	<b>56</b>
6.1	Merging Algorithm . . . . .	56
6.1.1	Voxel Grid Data Structure . . . . .	57
6.1.2	Voxel Model Integration . . . . .	58
6.2	Global Optimisation . . . . .	59

<b>7 Experimental Results</b>	<b>62</b>
7.1 Synthetic Range Image Test for Registration . . . . .	63
7.2 Real Model Acquisition Result . . . . .	64
<b>8 Conclusions and Future Work</b>	<b>68</b>
<b>Bibliography</b>	<b>70</b>

# List of Tables

3.1	Triclops stereo processing parameters, descriptions and values used in the range acquisition module. . . . .	24
-----	---	----

# List of Figures

2.1	Diagram of major stages in graphics rendering pipeline. . . . .	11
2.2	Coordinate systems in 3D rendering pipeline. . . . .	13
3.1	A picture of the Digiclops stereo vision system. . . . .	19
3.2	Image and world coordinate systems in the Triclops library. . . . .	24
4.1	Diagram of the control point matching strategy. . . . .	37
4.2	Three explicitly involved coordinate systems. . . . .	39
4.3	Possible outcomes for simplex deformation trial steps. . . . .	48
5.1	Diagram of texture compositing strategy with OpenGL NVIDIA register combiners extension. [Kil02] . . . . .	54
7.1	Experimental result of synthetic data registration. . . . .	63
7.2	A picture of the ceramics model unicorn. . . . .	65
7.3	Experimental results of the real data. . . . .	67



# Acknowledgements

I would like to thank my supervisor, Wolfgang Heidrich, for his support over the years, for his insightful ideas and the enlightening conversations we have had on the project, and for the opportunity he has offered me to work on this interesting project. Thanks also to my advisor, Dinesh Pai, for his valuable advice on my studies, which inspired me to enter the exciting area of computer graphics; thanks to Professor David Lowe for his kindly proofreading of this thesis; thanks to Don Murray for his great help in working with the device. Thanks also to David Pritchard, Daniel Archambault, Lisa Streit, Shuzhen wang, and Peng Zhao for their friendship and suggestions on my research; thanks to PointGrey Research Inc. and BC ASI for their financial support.

YUSHUANG LIU

*The University of British Columbia  
May 2003*

# Chapter 1

## Introduction

With the tremendous increase in PC power and the fast improvement of dedicated graphics hardware, the research scope of computer graphics has been greatly broadened over the past decade. Measurements of real world objects have been employed into each of the three traditional computer graphics research fields to produce more realistic effects, e.g. 3D scanning is used for modeling, motion capture is used for animation, and surface reflectance measurement is used for rendering specific materials. This thesis focuses on the topic of fast 3D geometric model acquisition from real world objects.

A practical approach for acquiring the 3D shape from a real object and thereby modeling the object is to obtain the surface information directly using a range finder. The surface information thus scanned is the so called range data which usually refers to range images or 3D point clouds. Range data is widely used to model real world objects and environments. However there exists a fundamental problem here, which is due to occlusions and/or limitation on field of views, one set of range data scanned from a single point of view contains only samples from part of the object surface. Therefore to acquire a complete model, multiple range data sets from good distributed point of views need to be scanned and then aligned into a common coordinate system. The process of aligning two or more three dimensional

data sets into one common coordinate system with rigid motions is a fundamental problem, namely 3D registration.

For range scanning tasks, although some researchers build their own range scanners for specific applications [RHHL02], there are several commercial range scanners available to choose which are normally easy to manipulate [Cyb]. In our project which aims at building a fast, near real-time model acquisition system by taking advantages of dedicated graphics hardware, we use a commercial stereo vision system, the Digiclops<sup>TM1</sup> [Res], to do the range scanning, and put more effort on accelerating 3D registration problem by using concurrent graphics hardware.

## 1.1 Motivation

3D modeling has applications not only in Computer Graphics but also in a wide variety of other fields:

- **Medical Visualization.** More and more visualization techniques have been used in medical diagnose and treatment, such as MRI, CT, Ultrasound. Often, different scans (either from different scanners or from different positions) have to be combined to form a 3D model so that physicians can see the internal organs on a computer screen.
- **Art Documentation and Study.** One good example is the Digital Michelangelo Project [LPC<sup>+</sup>00] at Stanford University, which involves scanning the sculptures and architectures of Michelangelo in ItalyPointGrey Research Inc. and BC ASI for their financial support; thanks to and postprocessing the data acquired there to produce electronic 3D geometric models. For those sculptures and architectures, nothing can be better documents than the 3D models obtained directly from the real objects. In addition, by producing a set of

---

<sup>1</sup>Digiclops<sup>TM</sup> is a trademark of Point Grey Research Inc.

3D computer models, these immortal masterpieces in art history can be made available to scholars worldwide.

- **Computer Aided Manufacturing.** One direct application of 3D model acquisition in this area is reverse engineering, which is the process of producing digital models of existing parts and then remanufacture them. This application problem arises from cases when the mechanical parts need to be remanufactured were hand-tuned to fit the existing machinery or the parts were designed by hand while the remanufacturing process needs a digital model.
- **Electronic Entertainment and Virtual Reality.** As video games are moving toward interactive 3D graphics, and film industry is relying on synthetic imagery more and more for producing special effects, the demands for digitized models are getting huge. At the same time, virtual reality as a way of simulationPointGrey Research Inc. and BC ASI for their financial support; thanks to real world environments and experiences is gaining more popularity. All of these application areas may require 3D geometric models digitized directly from real world objects or sculptures created by artists.
- **Electronic Commerce.** Online shopping has now been widely accepted and well recognized as a great trend in this century. More and more individuals as well as organizations are relying on it for shopping supplies and marketing products. If 3D models created from real products can be put on the web and made available to be explored interactively by customers, then the customers will definitely get a much better view of their interested product than only from 2D images.

Given so many applications of 3D modeling from real world objects in both industry and scientific fields, the current techniques are far from practical for most purposes. One of the difficulties concerns the scanning equipment which usually requires sophistications to use. Another comes from the registration process, since

most existing 3D free surface registration algorithms require intensive human interaction and they are usually very time consuming as well.

For most traditional 3D acquisition methods, the range data capturing and the model creation are separate stages of a pipelined process. Therefore, it is often very hard to make sure all relevant data has been captured in the first stage since the resulted model cannot be seen until after the registration process. Often users will find in the second stage that certain views have been missed, or the surface of the object is not covered in certain regions. In this case, users will have to go back to the first stage and capture the missing data. This may be difficult, for example, because the underlying object is no longer available, or because the whole setup needs to be recalibrated.

The goal of this particular project is to implement an interactive 3D geometric model acquisition pipeline which allows a user to monitor the geometric acquisition process and make data capturing decisions accordingly.

## **1.2 Related Work**

To acquire a 3D geometric model from real world object has been a long-standing research topic in the field of computational geometry and computer vision. Especially in recent years, with the great increment in PC power and dedicated graphics hardware broadening the application areas of real world 3D models, it is gaining more and more researchers' interests.

### **1.2.1 3D Measurement Techniques**

With the development of lasers, CCD's and high speed sampling and timing circuitry, methods of 3D shape measurement of real world objects have been evolving rapidly in recent years. Based on whether they interact with the object (direct contact with the object or project some kind of energy onto it) or not, these methods fall into two primary categories: passive and active methods.

Passive shape digitizing algorithms have been deeply explored for decades in the computer vision community. Several approaches, such as shape from shading and shape from silhouettes for single images, stereo algorithms for image pairs, and optical flow and factorization methods for video streams have been developed and applied to various applications. Although, general passive methods are computationally expensive, there are approaches within this scope suitable for real time use having been developed [FHM<sup>+</sup>83, Mat97, MBR<sup>+</sup>00]. This kind of approach has a limitation that it usually applies to scenes with considerate textures.

Various active 3D digitizing methods coming in a wide range of accuracies as well as costs have been developed. The literature provides an abundance of those. For a very nice taxonomy, refer to [RCM<sup>+</sup>01]. In short: active 3D digitizing systems are either contact sensor based or non-contact sensor based. Typical touch sensor based systems consist of a touch probe attached to a mechanical jointed arm or pointer. By touching the probe to the surface of the underlying object and measuring the joint angles and the lengths of the arms when the contact event is signaled, the position of the sensor head, which is also the position of the surface point, is recorded. Among various contact sensor systems, the Coordinate Measuring Machines (CMM's) are extremely precise and costly, which are currently the standard tool for making precision shape measurements in industrial manufacturing. A large majority of active digitizing systems are non-contact sensor based, which operate by projecting specific kind of energy waves onto the object and then measure the amount of transmitted or reflected energy or the time needed for the receiver to receive the energy bounced back from the interested object point. The energy waves can be both optical or non-optical including sonar, microwave radar (RAdio Detecting And Ranging), X-Ray, laser etc. For some variants of active non-contact range scanning systems, see the surveys on optical based methods [Bes88], methods based on time of flight [Inc], depth from defocus [SNN96], photometric stereo [HRG97], or projected-light triangulation [Cur97]. The real-time scanning-stripe system de-

signed by Gruss [GTK92] in 1992 falls in this scope, too. Comparing to passive methods, the active approaches require specific devices controlling and measuring the transmitted/reflected energy or the flying time of reflected energy and are thus more expensive. In addition they are normally faster, more robust and more accurate, and they do not require the existence of good scene textures to work.

### 1.2.2 Previous Registration Methods

Due to the limitations of the contact based methods (slow, expensive, clumsy, etc.), more and more 3D measurements are made by the non-contact methods which usually offer 3D range data as measurement result. For those model acquisition pipelines starting from range data, normally there are two major phases: first is the registration stage which is the process of bringing two data set into alignment, and then the registered views are integrated into a seamless 3D model.

Many different methods have been proposed for the registration of two 3D shapes. Some early methods rely on precisely calibrated mechanical equipment [SKSI90, SKSI91, VA86], such as a turntable and a robot arm to provide well controlled motion of the range finder or the subject. When using methods in this category, the underlying object and the sensor are moved relatively by a known amount in the data acquisition stage; and then in the merging stage, the view transformations provided by the apparatus are assumed to be accurate enough. Actually by using dedicated hardware and sophisticated calibration, this approach avoids the registration problem. Usually, it has a lower accuracy than the other approaches explained later, and the relative motions of object and sensor are restricted to calibrated motions.

Then in early 90's, a powerful 3D registration algorithm based only on geometry in the overlapping area, the Iterative Closest Point algorithms or the ICP algorithm in abbreviation, has been introduced by Chen and Medioni [CM91] and Besl and McKay [BM92]. Since then many invariants based on the basic idea of this algorithm have been proposed for many specific applications. The algorithm

starts from an initial guess of the rigid motion transformation between two range data sets, and then iteratively evaluates and improves the current transformation by searching for corresponding point pairs on the two data sets and optimizing a predefined similarity measure. ICP has now been a well established algorithm for aligning two or more 3D shapes, which is widely used for the registration of 3D range data sets obtained by 3D scanners. However most existing systems running ICP are slow and not easy to use, either because a lot of human interaction is needed for obtaining a rough alignment or because very little feedback is available in the data acquisition stage making good view planning almost impossible.

To improve the level of control and automation for 3D range registration process, extensive research work has been done. Maver and Bajcs [MB93], and Stamos and Allen [SA98] have proposed automated next-best-view systems to guide the user for good sensor poses. This kind of approach either involves intensive computational work or needs specific hardware to control the scanner positions. Another trend of automation registration process assumes that the relative location and orientation of the two scans to be aligned are completely unknown, and then by using surface features or exhaustive search, the optimized transformation can be found. These approaches either only deal with special objects or very time consuming and they are typically slow and not robust.

Recently, Rusinkiewicz and coworker enumerated and classified many invariants of the ICP algorithm, evaluated the effects of different approaches on the convergence speed [RL01]. Chapter 4 provides a summary of those ICP variants and the comparisons of their performances. Based on the survey, the authors proposed a real-time 3D model acquisition system which allows the user to see the continuously growing model on a computer screen at the same time when the hand held subject is moved and scanned [RHHL02]. The quick online feedback makes it possible for the user to oversee the scanning process, thus the next-best-view can always be decided actively and holes in the partial model can be filled interactively. Their system uses



a 60Hz. structured-light range finder and can align two range data sets in several tens of milliseconds, a whole model can be obtained within half an hour. Then, if necessary, a more accurate postprocessing can be conducted afterwards using the registration results from the real-time phase as initial guesses.

Besides the traditional ICP algorithm and its variants, there are other approaches to the range registration problem. Labsik et al proposed a new method for the registration of free-form surfaces which is not based on finding closest points [LRG00], instead their algorithm takes advantage of the modern graphics hardware and uses the depth buffer for determining the distance between the surfaces. Seen from a higher level, this method is still an ICP variant.

### 1.3 Contributions and System Overview

Our 3D model acquisition system aims at an interactive working mode, which although it might not be able to create high-resolution models as a variety of existing systems can do, it should be easy to manipulate for general users and fast enough for the user to get a preview of the ongoing partial 3D model. Therefore, view planning can be done while range data is scanned and thus data quality can be ensured. After the interactive range scanning and registration phases, if desired, a more precise but much slower post-process can be done to get a model with a higher precision.

Our 3D model acquisition system consists of three modules: the range scanning module, the 3D-3D range registration module and the model integration module. Since it is aimed at an interactive working mode, speed is the major concern for each of the three modules. For the range scanning, we choose a commercialized stereo vision system, the Digiclops camera, as the range finder, which works in real time. For the 3D-3D range registration process, we adopt a depth based ICP algorithm, which takes advantages of the modern graphics hardware for geometric computations and thus runs faster than common ICP algorithms. The fast converging downhill simplex method is used for the numerical optimization. Also

for this module, we proposed a hardware accelerated approach which combines two depth images into one to compute quickly the absolute difference of z-values of each projected point and thus greatly reduces the necessary amount of memory accesses for registration purpose. For the model integration module, a volumetric model is used, which is fast to add new data on and fast to render as well.

The major effort has been in the acceleration of the 3D-3D registration process using graphics hardware.

## 1.4 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 describes some highlighted points of the modern graphics rendering pipeline of which we take advantage for the implementation of our system. Chapter 3 presents the usage of the range scanner, the Digiclops camera in the model acquisition pipeline. Chapter 4 describes the design of the depth buffer image based 3D range registration algorithm. Some research results of the important invariants of the ICP algorithm have been summarized for a better understanding. Chapter 5 describes two approaches of using concurrent graphics hardware to accelerate the registration process, which are the major contributions of our work. Chapter 6 introduces the integration of the voxel 3D model and some possible ways of optimizing it offline. Chapter 7 presents the experimental results on both synthetic range data sets and real range images grabbed by the Digiclops camera. Chapter 8 makes a conclusion of this research work and suggests some ideas for further improvements of this system.

## Chapter 2

# Graphics Rendering Pipeline

Our model acquisition system is designed to take advantage of dedicated graphics hardware for 3D range data registration as well as 3D geometric model display tasks. In order to discuss the usages of graphics hardware, it is useful to have a closer look at the fundamental features of a graphics system on which we rely on for the implementations of our model acquisition algorithms. The application programming interface (API) we use to control the computer graphics tools is OpenGL (GL is the abbreviation for Graphics Library), but since most contemporary graphics hardware and APIs are variations of the traditional 3D rendering pipeline, we will keep the discussion independent of any specific programming interface. OpenGL is only chosen due to its open structure and the fact that it is well specified and documented.

### 2.1 Rendering Pipeline Overview

Because of the processing of information in 3D graphics rendering occurs in a sequential order, where results computed by the previous step are used as input for the current one, an abstract model named “rendering pipeline” has long been used to describe the sequence of operations applied to all original data sent through this processing “pipe”. The output of the pipeline is the final raster image drawn in the

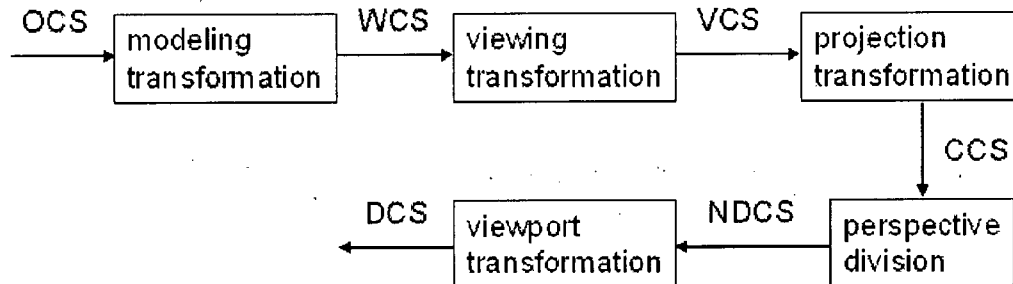


Figure 2.1: Diagram of major stages in graphics rendering pipeline. In the figure, OCS is the abbreviation for object coordinate system, WCS is for world coordinate system, VCS is for viewing coordinate system, CCS is for clipping coordinate system, NDCS is for normalized device coordinate system, and DCS is for device coordinate system which is also called window coordinate system.

frame buffer. Figure 2.1. depicts the major stages of the graphics rendering pipeline which applies to most contemporary graphics hardwares and APIs.

All geometric data sent directly for geometric operations are eventually described by vertices sitting in the object coordinate system which is a local coordinate system tied to the underlying object. If the input data are mathematically described by a few control points, such as smooth curves or surfaces, an evaluator stage will be responsible for approximating curves or surfaces with points before they are sent through the pipeline for geometric operations. Internally, a homogeneous coordinate system which is a 4-dimensional coordinate system denoted by  $[x, y, z, w]^T$ , is used

to represent vertices. All geometric operations deal only with homogeneous vertices and their associated data (e.g., normal vectors or colors). The major tasks of geometric operations are geometric transformations and their related calculations. As shown in the Figure 2.1, a series of operations are applied to bring vertices from their 3D object coordinate system to the pixel positions in the window coordinate system on the screen, which include:

- **Transformations.** Usually it includes modeling, viewing, and projection operations. When a scene object is sent through the pipeline, some combination of rotation, translation, scaling, reflecting, orthographic projection, and perspective projection transformations will be applied.
- **Clipping.** Objects or part of an object outside of the defined viewing frustum should not be seen by viewers so that they should be clipped or thrown away.
- **Viewport transformation.** This is the process responsible for setting up the correspondences between the transformed coordinates and the quantized screen pixel position in the window coordinate system.

In addition to those major operations which apply to all geometric primitives set through the pipeline, some optional steps may also get involved if enabled. Lighting calculations, texture mapping, and special plan clipping are some examples.

## 2.2 Coordinate Systems in 3D Rendering Pipeline

This work deals mostly with geometric primitives of vertices represented in and transformed among various coordinate systems in 3D and 2D spaces. Therefore it may help us get a clearer idea to look at more details of the coordinate systems involved in the rendering pipeline and the transformations that tie them together. Those coordinate systems would determine the various geometric representations of

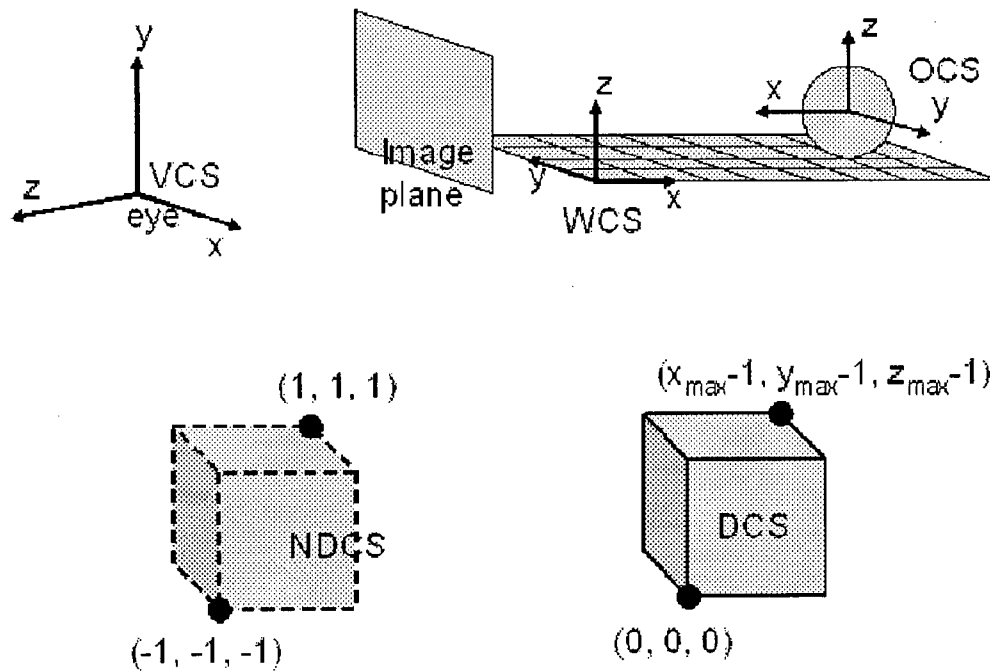


Figure 2.2: Coordinate systems in 3D rendering pipeline.

a vertex while it is transformed within the pipeline before being drawn as a pixel on the screen.

### 2.2.1 Geometry Related Coordinate Systems.

Figure 2.2 shows the five coordinate frames the geometric calculations use in the rendering pipeline in the order the operations occur in a drawing process. A series of mathematical transformations convert the 3D coordinates of geometric primitives in their individual object coordinate system (OCS), which is the input to the rendering

pipeline, to their corresponding pixel positions in the window coordinate system or the device coordinate system (DCS), which is the final output from the pipeline. Four intermediate coordinate systems including a world coordinate system (WCS), a viewing coordinate system (VCS), a clipping coordinate system (CCS), and a normalized device coordinate system (NDCS) sit inside the pipe appearing in the order as they are mentioned here. When a vertex goes through the rendering pipeline, its representation changes from one coordinate system to the next one until it is rendered into the frame buffer as a pixel.

Originally, all geometric primitives are represented in their individual object coordinate systems. For positioning and orientating all objects in a scene and keep tracking the moving objects, a world coordinate system is employed. The transformation closely related to this coordinate system is the modeling transformation which is responsible for setting scene objects properly for view. The viewing transformation brings all vertices from the world coordinate system to the eye coordinate system or the viewing coordinate system which always stitches to the eye point and oriented accordingly. There is a duality in the nature of viewing and modeling transformations, that is moving the eye in some way can be viewed as moving the object oppositely. Therefore both viewing transformation and modeling transformation modify the same modelview matrix. However, when rendering a scene, the viewing transformation must precede the modeling transformation in order to make sure that the incoming object's coordinates are properly transformed to the eye coordinate system.

The clipping coordinate system coming after the projection transformation is so named because the standard clippings happen right here when geometries are represented in this coordinate system. A viewing frustum is defined by the projection transformation applied to the scene, which action is analogous to choosing a lens for a camera when shooting a picture using a real camera. Intuitively, only those objects inside this frustum can be seen by the viewer and thus need to be drawn. The

standard clipping refers to the clipping of scene against the border of the viewing frustum. Usually, two types of projection transformations are provided to control how the viewing frustum is specified and how the objects are projected onto the computer screen. One is the perspective projection which mimics how things are perceived by human eyes or pinhole cameras. The most prominent property of perspective projection is the foreshortening effect which is the further an object is to the eye (or the pinhole of camera), the smaller it appears to be. To acquire photo realistic effects of 3D objects, it has to be perspective projection. The other type of projection is the orthographic projection which preserves objects' real size in 3D by mapping them directly onto the screen. By using orthographic projection, although the rendered objects may look distorted, they actually reflect the real or scaled measurements of objects in all 3 coordinates. Orthographic projection is therefore widely used in architectural and computer-aided design applications.

After the standard clipping, viewing volume is transformed to a unit cube through a perspective division step which is to divide the three coordinate values by the fourth coordinate,  $w$ . The unit cube is named normalized device coordinate system. Finally, the viewport transformation maps the scene onto the computer screen by transforming the normalized device coordinate system into the quantized window coordinates or the device coordinate system, which stage is often referred to as Rasterization. By manipulating the dimensions of the viewport, the final image can be enlarged, shrunk or stretched. The visible part of the scene are now rendered as a 2D image into the frame buffer, the quantized  $x$  and  $y$  coordinates of a point are the row and column indices to its corresponding pixel, and its  $z$  coordinate is saved in the depth buffer.

### **2.2.2 Texture Coordinate System**

Besides the six aforementioned coordinate systems, the texture coordinate system is another important coordinate frame used in computer graphics to indicate how



textures should be mapped to the geometry. Depending on the nature of textures, texture coordinate can be one, two, three or even four dimensional. With the fourth coordinate typically given the value 1, four dimensional texture coordinates is usually interpreted as homogeneous texture coordinates. When a texture mapped scene is drawn, the object coordinate of a vertex determine the position of its rendered image and the normalized texture coordinates of it are used to look up its associated texel in texture memories. There are two ways of assigning texture coordinates to vertices. One is to explicitly assign texture coordinates when the geometry is defined, the other is automatic texture generation which is to compute the texture coordinates automatically. This feature plays an important role in this work. If this automatic texture generation mechanism is used, one or more texture coordinate components can be computed as the distance between the vertex and an arbitrary pre-defined plane in object or viewing coordinate systems. Besides, a specific mode for generating texture coordinates in a spherical environment map is also available.

Texture coordinate calculations happen before the projection transformation through the manipulation of a  $4 \times 4$  homogeneous texture matrix, which makes texture coordinates perspectively correct and makes it possible to apply geometric transformations to textures as well. No matter what the dimensions of a real texture is, all texture coordinates are within the range of  $[0, 1]$ . Texture coordinates outside of this range can either be clamped to 0 or assigned the repeated value of its border.

## Chapter 3

# Range Data Acquisition

Our 3D model acquisition system begins with a range data scanning step. To make it possible for the system to work at interactive rates, choosing a proper range data acquisition device is the first crucial decision to make. A commercially available stereo vision imaging system, the Digiclops Stereo Vision System [Poi01a] provided by Point Grey Research Inc. [Res], which provides real-time 3D digital image capture is used for the range scanning purpose in our system.

### 3.1 Digiclops Stereo Vision System

As explained in Chapter 1, the literature provides an abundance of different methods for obtaining 3D geometry from real world objects. Based on whether the measurement is made by interacting with the object or not, these methods follow two directions: passive and active sensing. Passive methods are mainly explored in the computer vision community, which include shape-from-shading for single image, stereo triangulation for pairs of images, and optical flow and factorization methods for video streams, etc. The Digiclops Stereo Vision System is based on stereo triangulation technique and thus falls in the passive sensing category.

### 3.1.1 Digiclops Stereo Processing

As clarified above, the Digiclops camera is a commercially available stereo vision system which makes range measurements using stereo triangulation techniques. Having been explored by computer vision community for decades, the purpose of the stereo vision technique is to perform range measurements using images obtained from slightly offset cameras. There are three major steps in performing general stereo processing, which includes:

1. Establish correspondences between image features grabbed by different views of same scene points. The Digiclops accompanying software establishes correspondence between raw images using the Sum of Absolute Differences correlation method.
2. Calculate the relative displacements between those feature correspondences in image coordinate system.
3. Determine the 3D locations of the scene features relative to the local cameras coordinate system, using the knowledge of the camera geometry.

A picture of the Digiclops camera is shown in Figure 3.1. With a size of  $15.5cm \times 15.5cm \times 5.0cm$ , it can be easily fixed onto a tripod or held by hand. As shown in the picture, on hardware part, instead of making up of only two cameras, it is a trinocular stereo system consisting of an onboard three-camera module. The reason for using multiple cameras is to make matching between images more robust. The geometric setup of the Digiclops camera is like a letter *L* with the three cameras situated on the vertices of a right angle triangle. Imposed by the geometric features of the imaging system, the epipolar constraints applies to the stereo system, which transforms the two dimensional search for correspondences into a one dimensional one when establishing correspondences between pixels in different images. In addition, the *L* setup of cameras is to align the epipolar lines with the image rows and

columns, and to reduce the image deformations due to rectification. On software part, the Digiclops camera comes with an accompanying software system, the Digiclops and Triclops<sup>TM1</sup> libraries, which in combination with the three camera module, perform range measurements.

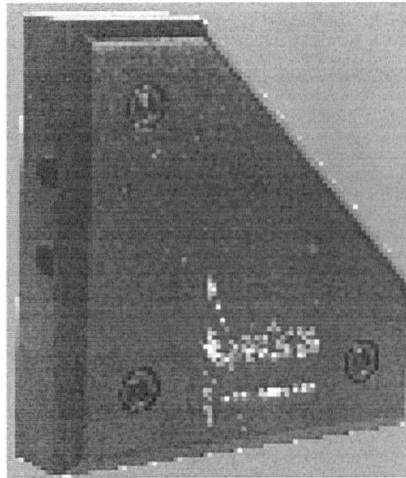


Figure 3.1: A picture of the Digiclops stereo vision system.

When making range measurements, the camera module obtains three images of the scene object from slightly different view points simultaneously, which are then digitized and stored in the memory of the computer. While the images appear quite similar, from the knowledge of stereo vision, there's a shift between image pixels corresponding to the same scene point in different images. The closer the object point is, the larger the shift of its corresponding images appear to be. When the software system processes raw data, correspondences are firstly established between features captured in two offset images. Based on those correspondences, the shifts

---

<sup>1</sup>Triclops<sup>TM</sup> is a trademark of the Point Grey Research Inc.

are then calculated for all established corresponding point pairs. Finally, by using the knowledge of the camera's geometry, it is possible to determine the distances between the scene points and a reference plane which is the plane the three camera determine. With the resulted depth/disparity image at hand, it is possible to calculate the 3D position of each point in the image relative to the origin of the local camera coordinate system.

As for the stereo processing strategy of the three raw images obtained simultaneously by Digiclops, the image from the camera located at the square corner of the right angle triangle is taken to be the reference one. This reference image is then correlated horizontally with the image on the left and vertically with the one on the top to get two depth/disparity maps, which are then merged together to produce one dense fused depth map. This approach of correlating both horizontally and vertically has the advantage of introducing redundancies which allow us to obtain more reliable results. For example, based on those redundancies, we can verify whether two matches are compatible or not by testing if the matched points in the left and top images lie on the same diagonal epipolar line. If this correlation is trustable by a predefined criteria then the two matches measure the same depth and we can be almost absolutely sure that they are correct because the searches for the two corresponding points are made on completely different areas horizontally and vertically. Otherwise, at least one match is false and we can optionally select the better one or ignore them both.

### **3.2 Integrating Digiclops into the System**

According to the classification of 3D range scanners explained in chapter 1, the Digiclops camera is a passive digitizer using stereo triangulation algorithms to make 3D range measurements, which on hardware part is a three-camera module. The design of using three onboard cameras dramatically reduces the difficulty involved in using general stereo vision systems, which as a result makes the Digiclops camera

very easy to manipulate for even general users who know nothing about stereo and 3D imaging. Basically, it functions simply as a hand-held camera and can be used to capture 3D real world objects by pointing the lenses to the objects and monitoring the results displayed on a computer screen.

There are several practical advantages for choosing the Digiclops camera [Poi01a, Poi01b] as the range scanner in our model acquisition system. Firstly, the Digiclops camera consists of three on board progressive scan CCDs that allow for full 3D ranging of moving objects without interlacing problems found in standard NTSC CCDs. Secondly, the system is calibrated to a high precision prior to shipping, both lens distortions and misalignments have already been compensated and known as system constants. In this way we can avoid having to perform extensive calibration algorithms by ourselves when conducting 3D measurements. Thirdly, the combined software API, the Digiclops and Triclops SDK's, provides real-time range images using stereo vision technology, which allows users to accurately measure the relative distance of every valid pixel in an image and generate dense depth maps fast and accurately. Finally, the device acquires color of the object at the same time as range information making the registration of textures and geometries done in one process, which is a task not many range scanners can tackle.

The Digiclops camera operates at a frame rate of about  $16Hz$  for a image resolution of  $640 \times 480$  from each camera, and its resolution can be as high as  $1024 \times 768$ . Instead of using the disparity maps output from the Digiclops directly, our registration algorithm starts from 3D point clouds. Therefore, in addition to the CPU time required to do other calculations besides the stereo processing, the frame rate for generating new point clouds is in the order of  $10Hz$  on a  $1.6GHz$  Pentium 4 machine. Although this rate cannot be considered true real-time yet, an interactive model acquisition can be achieved if the subject is moved slowly.

Ideally, we are expecting to realize a 3D video camera out of the Digiclops,

which can be used to capture real world 3D shapes in real-time while the Digiclops camera is held in the user's hand and moved around the underlying object slowly and statically. However, there are practical problems arising when the scanner is used in this way. First of all, its operating rate is not fast enough for real-time capturing so that the user has to be very careful when changing view points. Secondly, in order to make our range based registration algorithm work, two successive images should have good overlapping area. But usually the subject is about half a meter away from the camera, so when the user tries to move the Digiclops camera around the subject by hand, slight tilts of the equipment may result in very large displacements of the scene images, the subject may even get out of the field of view of the cameras on board. Finally, due to the limitation of stereo techniques, the Digiclops fails to produce correct 3D information for points along discontinuities in the scene. This makes the range registration using 3D measurement calculated directly from original data maybe erroneous. This problem mainly manifests itself along silhouette edges.

To get around the above mentioned problems, we have adopted an alternative acquisition strategy to acquire the range data from the underlying object, which includes two major approaches. One approach is a solid colored background has been employed so that a color segmentation algorithm can be used to get rid of the erroneous sample points along the silhouette together with all other background points; the other approach is to fix the Digiclops camera on top of a tripod and rotate the subject instead of the range finder when making range measurements. To make the movement of the subject easier, we actually put it around the center of a turntable covered by the black background, and rotate the table and thus the subject. In this way, once the position of the subject relative to Digiclops is fixed, it wouldn't be moved very far from there since the motion of the subject is basically rotate about the a line about to pass its center.

### 3.3 Stereo Processing Environment

Each group of the three raw images acquired by the Digiclops system has to be processed in order to get the desired 3D range data. The Triclops library accompanying the Digiclops camera is the software SDK responsible for the generation of depth maps from the raw data, which is the ultimate result of the Triclops API. Those depth maps can then be used to generate range information in other formats, e.g. the 3D point clouds in our application.

#### 3.3.1 Coordinate Systems Used in the Triclops Library

When talking about 3D information and operations, there is always a coordinate system involved. Only when relative to the coordinate system, the measurements and representations are meaningful. The Triclops library represents raw data in its image coordinate system, the origin of which is at the top left corner of the digitized image. This coordinate is only used by the library functions in our system and thus can be seen as opaque by the user. The depth maps output from the Triclops library are represented in its world coordinate system, which when seen at the level of the whole acquisition system is actually the local sensor coordinate system and/or the individual object coordinate system. The two coordinate systems used for Digiclops data processing by the Triclops library are shown in Figure 3.2 [Poi01b].

#### 3.3.2 Stereo Processing Parameters

The Triclops data processing block takes raw images obtained from the Digiclops camera module and produces depth images out of them. Further characteristics of depth maps may vary depending on a number of stereo processing parameter settings. The Triclops SDK allows specifying several of the characteristics stereo processing may use. The brief descriptions and the values of the parameters we use in our range finding module have been summarized in Table 3.1. The values of all parameters are chosen by experiments.



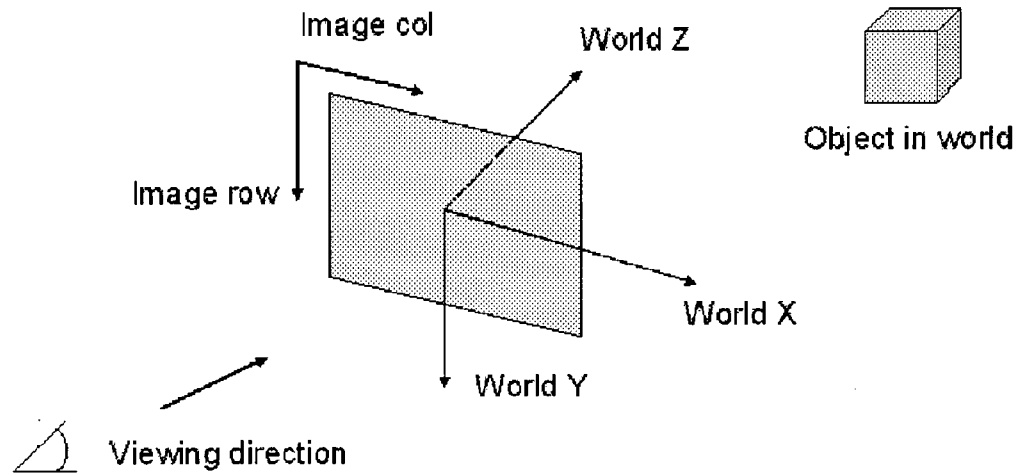


Figure 3.2: Image and world coordinate systems in the Triclops library.

Table 3.1: Triclops stereo processing parameters, descriptions and values used in the range acquisition module.

Stereo Parameters	Description of the Parameter	Value of the Parameter
Image size or resolution	A few supported raw image sizes. Larger ones consume more time; smaller ones contain less data.	640 × 480,
Disparity range	Pixel range of distance measurements done in images.	0 ~ 220 (close to the largest)
Mask size	Window size for stereo processing	13
Preprocessing	Does image unpacking, smoothing, rectification and edge detection.	edge detection mask size: 11
Validation	Supported methods for verifying measurement correctness	surfaceValidation: on textureValidation: off surfaceValidationSize:100 surfaceValidationDifference: 0.9 uniquenessValidation: off
Regions of Interest	Regions of image processing done.	The full image.
Subpixel Interpolation	Allowing establishing correspondences to subpixel accuracy, and thus offer more precise measurements.	on strictSubpixelValidation: off

## Chapter 4

# Range-Based Registration

Having obtained 3D data from the range scanner, in our case the Digiclops camera, the next step is to align the range data sets scanned from different point of views and thus sitting in different local sensor coordinate systems to a common global world coordinate system. This process is the so called 3D-3D registration. Researchers from the computer vision community have attempted to formalize the description of this fundamental problem: "Given 3D data in a sensor coordinate system, which describes a data shape that may correspond to a model shape, and given a model shape in a model coordinate system in a different geometric shape representation, estimate the optimal rotation and translation that aligns or registers the model shape and the data shape minimizing the distance between the shapes and thereby allowing determination of the equivalence of the shapes via a mean-square distance metric." [BM92] Usually, a range data registration strategy consists of two major steps: correspondence selection in which candidate correspondences between data sets are chosen, and rigid motion estimation in which the rigid motion optimizing a predetermined surface similarity measure function between the two range data sets are estimated. The application problem confronted in our project is the point-set matching problem without correspondences.

As mentioned in Chapter 1, there exists a well established algorithm, the

ICP algorithm, for registering 3D geometric models based on geometry only (these are some ICP variants using additional information besides geometry, e.g. color and intensity [JK97]). Although our algorithm tackles the registration problem from a different approach than the classical ICPs, seen from an abstract level of view, it is still an ICP variant. Therefore, taking a closer look at the common processing phases involved in basic ICP algorithm may lead to a well understanding of our approach.

## 4.1 Problem Specification and General Approaches

The most popular used strategy for extracting the surface structure of an real world object and thereby modeling the 3D object these days is to obtain multiple range images from various viewing positions and orientations, and then to merge them together to reconstruct the complete surface. Due to the involvement of multiple range views, corresponding transformations that register each of these range images sitting in their individual local sensor coordinate systems into the global coordinate system need to be determined so that all of them can be expressed in a unique modeling coordinate system.

More formally, the range registration problem can be stated as: given  $N$  range images of an object in a scene, each one being a piece of the 3D structure of the object seen from a particular viewpoint and thus sitting in its individual sensor coordinate frame, the registration process is to find  $N$  corresponding transformations  $T_1, T_2, \dots, T_N$ , that specify the positions and orientations of each local sensor frame relative to the unique global modeling frame which we call the reference coordinate system. Given that range image  $i$  ( $i = 1, \dots, N$ ) consists of a set of 3D points (or vertices of a mesh)  $S_i$  represented in the local sensor coordinate system  $i$ , the transformation  $T_i$  transforms  $S_i$  into the unique coordinate system,  $S'_i = T_i(S_i)$ , where  $S'_i$  is the representation of the point set in the global coordinate system. By transforming all point sets of the  $N$  range images, a new set of 3D points which is

the union of all transformed range data sets  $S'_i, \dots$ , mathematically

$$S = \bigcup_{i=1}^N T_i(S_i) = \bigcup_{i=1}^N S'_i \quad (4.1)$$

can be generated. This new set of points is the surface boundary 3D model of the object reconstructed from the multiple range views [BL93] of the subject.

The existing methods used for solving the 3D-3D registration problem fall into two main categories. One is to rely on precisely calibrated mechanical equipment to determine the transformation between different views. This approach actually avoids the problem of searching for the appropriate transformation by blindly accepting the registration transformation provided by the acquisition apparatus [SKSI90, SKSI91]. Precisely calibrated turntables and robot arms are the most frequently used equipments for registration purpose [SKSI90, SKSI91, VA86]. The advantage of this approach is that it circumvents the complex registration process. But usually, the accuracy of this kind of methods is much lower than the methods falling in the second category and the acquisition systems are expensive and not flexible. The second category of registration methods derives the transformation between range images from the information contained in the range images as well as some additional information provided by the range data acquisition system. In this kind of systems, the transformation parameters are gradually updated and refined until according to a predetermined criteria, the views are precisely registered. Normally, a feedback function measuring the quality of the current registration is needed. Also, in most cases, an estimation of the registration between the image to be registered and the reference one is part of the information available. Theoretically, if a set of point pairs in which two points in different images corresponding to the same surface point on the subject can be precisely matched, then the registration problem is transformed to solving a group of equations. But practically, the correspondence information is not usually available, which makes

finding the correspondence a sub-problem of the registration process. Two possible solutions exist. The first one pairs points by matching features from the two underlying views, in which both kinds of features that are either parts of the scene or synthetic features that are added on to the scene manually have been used by researchers [SM92, Chu96, ea95, JH97, FH86]. Using scene features for correspondence matching directly is highly feature dependent, normally it greatly limits the application area of the system since very few objects contain desirable features, while using added on features destroys the surface structure of the original scene. In addition, it spends a large percentage of computation time on extracting the invariant features [CHC99], which is another major drawback. The other possible approach to the point matching problem compares the differences in the structure of the surface across views, which is a process operated over the entire surface or over a set of chosen control points. The ICP method, the most popular registration method used nowadays, is within this category.

## 4.2 Range-Based ICP Algorithm

The ICP algorithm, originally introduced by Chen and Medioni [CM91] and then extended by Besl and McKay [BM92] in early 1990's, has now become the dominant method for registering partially overlapping 3D shapes based on geometry only (some approaches may use additional information other than geometry, such as color and intensity). Its main application is for aligning multiple range data outputs of 3D scanners obtained from different point of views and thus sitting in different local sensor coordinate systems into one common coordinate system.

ICP works with two data sets, usually two meshes (but not necessarily), one is the reference and the other is the data set to be registered to the coordinate system in which the reference range data set sits. Starting from an initial guess of their relative transformation, the algorithm interactively improves the intermediate registration results between the two underlying overlapping surfaces by searching for the unique

transformation that optimize a predefined surface consistency measure. Guided by the internal mechanism of the chosen numerical optimization algorithm, the process keeps operating until by a predetermined criteria a solution has been found or the maximum number of iterations has been reached (in this case, the algorithm fails). At each iteration, the quality of the current transformation is firstly evaluated by generating corresponding point pairs based upon the current relative position of the two surfaces and calculating the predefined surface consistency measure function based on the matched point pairs. Then the registration is gradually improved using the numerical strategy. The guideline for the optimization process is actually the numerical algorithm.

Four sub-problems need to be solved in an ICP process, which are:

- initializing the transformation parameters which is the starting point for the process to improve;
- generating corresponding point pairs based on which the current registration can be evaluated;
- coming up with a good surface consistency measure function which is used to rate the level to which the two pieces of surfaces could be aligned;
- finding a suitable numerical optimization strategy which is the guide for the whole process.

Although a crucial step, the initial guess generation problem is usually not addressed in the ICP algorithm itself. When applying the ICP algorithm, it is almost always assumed that a starting point is ready. All the other three sub-problems are normally included in the ICP algorithm.

#### **4.2.1 Existing ICP Variants**

Researchers have been working on different applications of the ICP algorithm and proposed many variants based on the basic strategy affecting all phases of the al-

gorithm, refer to papers [FH86, SM92, JH97, DWJ97, CHC98, CHC99, Pul99a]. Recently, Rusinkiewicz and his coworker have studied the existing variants thoroughly and done lots of comparisons among them [RL01]. They used a six-stage scheme to describe the basic algorithm and classify those ICP variants, which can be seen as the invariant for all the variants. The effects of different approaches adopted by the underlying variants in one or more stages on the speed performance and result accuracy have been explored. These comparison results can be valuable guidelines for building new ICP variants for new application requirements. The six-stage scheme and some of Rusinkiewicz's results are summarized below:

### **Control Points Selection**

Control points are the points sampled from one or both data sets which will then be matched to points in the other data set. Since it is those control points that will be used for the following registration stages, they need to be very representative for the whole range data set. Five existing selection strategies, including a new one they proposed in the same work, have been examined, which are:

- Besl's approach [BM92], which is to use all available points as the control points;
- Turk's uniform sub-sampling method [TL94];
- Masuda's random sampling approach [MSY96];
- Weik's approach [Wei97] which takes advantage of the associated intensity of the range data in addition to the geometry information, selecting points with high intensity gradient;
- author's new feature based sampling strategy [RL01] which is to choose control points based on their associated normals, the goal is to make the normals of the selected points span as large an angular space as possible.

Considering the sampling direction can be sampling from only one data set or from both [GRB94], there are actually two approaches for each of the above mentioned five. Author's comparisons show that the convergence performance of all those approaches for scenes with a good distribution of normals is similar. However, for some difficult scenes, it is sometimes the small features that actually determine the correct registration. To use feature based methods is usually the way to get the correct registration for those kinds of scenes. Author's normal space sampling method is the only one among the six that successfully aligned such a difficult test scene in their experiments. For the effects of sampling directions, they found that it depends on the adopted matching strategy as well as on the overlapping condition of the two data sets. If the matching strategy is a symmetric one and the two range images have good overlap, the sampling direction doesn't really matter, otherwise to sample from both data sets is a little bit superior in terms of both convergence speed and registration accuracy.

### **Point Matching**

The goal of this stage is for each control point to find the corresponding point in the other range data set so that both of the two points correspond to the same surface point. However there are far less information available for this pairing. Different matching strategies have been proposed. The strategies investigated by the authors fall into two major categories:

- Projection based algorithms, which include "normal shooting" [CM91], "reverse calibration" [Neu97, BL95] and project-search [Pul99b, Wei97, DWJ98, BS97]. The "normal shooting" approach is to project the control points in the direction of their normals and find the point at the intersection, while "reverse calibration" is to project the control points of the other data set from the sensor's position and find the intersection point. The project-search strategy is to project the control points to the other data set along the sensor's projec-



tion ray and then find the corresponding point according to various predefined metrics.

- Non-projection based algorithm, which refers to the matching scheme of finding the closed point in the other data set [BM92].

Similar to the control points selection approaches, if we restrict the pairing to only those pairs in which the two points are compatible to each other according to a specific metric [GRB94, Pul99a], then we can get another matching method from each of the above mentioned approaches.

As to time performance, the project-search method shows a significant advantage, while the closest-point algorithm is very robust although it might not be fast. If the pairing is not restricted to compatible pairs, all the algorithms can be accelerated by a  $k - d$  tree data structure and/or closest-point caching [Sim96].

### **Weighting and Rejecting Point Pairs**

After the control points are matched, different weights are assigned to all pairs in order to give pairs that are most likely to be correctly matched more importance and to reduce the influence of outliers. The step that immediately follows weighting, rejecting point pairs, is closely related to weighting, which may reject certain point pairs completely. The intention for the rejecting point pairs step is to get rid of the erroneous points which may be produced in the range scanning process or when matching the control points. Four weighting methods have been examined in Rusinkiewicz's work:

- Constant weighting, which is actually ignore this step and treat all point pairs acquired in the previous steps equally.
- Treating the point pairs with smaller point-to-point distances as more trustable ones, the larger the distance is, the smaller its assigned weight is [GRB94].

- Weighting according to compatibility of normals or colors [GRB94].
- Weighting based on some statistical analysis on the range scan process.

From his experiments, Rusinkiewicz concludes that normally the weighting strategy will not affect the convergence rate much, and that the effects will be highly data-dependent.

As to the step of rejecting outliers, actually, it can be seen as assigning a 0 weight to certain point pairs. Three kinds of rejection strategies have been explored in Rusinkiewicz work:

- Threshold the point-to-point distances. The threshold can be a distance value, a percentage out of all participating pairs [Pul99a] or some value generated from statistics [MSY96].
- Rejecting point pairs that are not consistent with neighboring pairs, usually the consistency here refers to point-to-point distance [DWJ98].
- Rejecting point pairs with one point on surface boundaries [TL94].

Comparisons show that outlier rejection strategies don't not really affect the convergence rate although it might affect the accuracy and stability of the registration algorithm. However, the strategy of excluding boundary point pairs is very useful for eliminating wrongly paired points when aligning partially overlapped range data sets.

### **Surface Consistency Measure and Optimization**

The surface consistency measure is a function based on current alignment of the two underlying data sets, the value of which indicates to what extent both of the two data sets can be representations of the same physical surface under the current registration. At each optimization iteration, this consistency measure is first

evaluated based on the matched point pairs and then improved. So that from the mathematical point of view, the goal of the registration process is actually to optimize the surface consistency measure function. The difference between its current value and the predetermined stop criteria is the source of the force that pushes the numerical optimization process to keep searching in the parameter space. The authors explored the effects of two different surface consistency measures (which is named error metric in their paper) on the performance of the registration process. The two surface consistency measures are:

- Sum of point-to-point distances over the entire overlapping area. The most intuitive approach uses the value of this sum only, while some other approaches may add on color [JK97] differences or differences based on information of other kinds. For this consistency measure, closed-form solution exists, several numerical methods can be used to solve it [ELF97].
- Sum of point-to-plane [CM91] distances, where the plane is the plane contains the corresponding point and perpendicular to the normal associated to this point. The optimization of this consistency measure is a non-linear problem, where there's no closed-form solution exists.

Experiments show that to adopt different numerical optimization methods does not change the accuracy and stability very much [ELF97]. So that the main concern of choosing a suitable numerical algorithm is the convergence rate. As to the several existing ways of formulating the searching scheme [Sim96, MSY96, BL93, BM92, CM91], since more complex ones tend to converge slowly, only two of them are explored by the authors, which include standard "select-match-optimize" scheme with [BM92] and without [CM91] extrapolation. Experiments also show that the point-to-plane measure is dramatically superior than the point-to-point measure. Although the extrapolation strategy during the optimization process increases stability and avoids problems caused by overshoot, it might consume more computing

time.

Based on their experiments, the authors concluded that for a fast ICP variant which might be suitable for real-time applications, it should use project-search strategy for point matching, a point-to-plane surface consistency measure and adopt the standard “search-match-optimize” ICP iteration [RL01]. They proposed and implemented a new fast ICP algorithm adopting the above mentioned strategies, which is very promising for real-time applications. For other stages where different approaches do not change the convergence rate much, their method adopted the simplest ones which include random control points selecting, constant weighting and a distance threshold based outlier rejecting step.

#### 4.2.2 Overview of Our Range Based ICP Algorithm

With the exception of the work done by Rusinkiewicz et al [RL01], the performance of the 3D-3D registration task has so far been too low for interactive applications. As mentioned in the previous subsection, Rusinkiewicz achieves the interactive rates by only using a random subset of points on the model for registration. For the same purpose, we go along a different route: taking advantages of the contemporary graphics hardware and rendering pipeline which are extremely efficient for geometric computations. Considering the fact that the registration problem involves mostly computations that the graphics hardware is optimized for, we adopt a depth based ICP method that by using the rendering pipeline offloads the geometric computations to the dedicated graphics board.

Our algorithm is generated from the intuition that if two pieces of a 3D shape match (by match, I mean represent the same surface in the same coordinate frame) , then their overlapping regions should match in all directions, specifically, their depth images match in the overlapping area. By looking at the two depth images, the expensive task of explicitly finding corresponding points in the two 3D data sets have been avoided. Instead, we only need to do the searching in one

direction which is along the depth. There are other researchers attempted solving the 3D registration problem from this direction [LRG00]. With these in mind, we render the two range data sets and use the rendered depth buffer images for determining a one dimensional distance based consistency measure function which will be optimized in the registration process. This algorithm makes use of graphics hardware for both tasks of transforming the geometry from one coordinate system to another, and for evaluating the surface consistency measure function using the rasterization part of the graphics hardware. We can categorize our method using Rusinkiewicz' framework for ICP algorithms [RL01]

- **Control Points Selection.** All points in the current range data set obtained from the range scan module are used as the control points for computing the surface consistency measure.
- **Point Matching.** The projection-based approach is adopted. Each time when we evaluate the current transformation, we render the data set of selected control points in the reference coordinate frame via the transformation under an orthographic projection, and then pair each visible control point with the point in the other range image along the same orthographic projection ray, as shown in Figure 4.1. By passing the underlying range data through the rendering pipeline, we utilize the fast modern graphics hardware to carry out the complex geometric calculations. The reason why we choose an orthographic projection over a perspective one is two fold: firstly, the perspective foreshortening effect is avoided in this way, so that the resolution of the rendered image is independent of the distance from the viewer; secondly, it simplifies the matching strategy. To pair up the points along the same orthographic projection ray, we can simply match the points in the two depth images with same row and column indices. While under a perspective projection, this simple pairing method does not have any meaningful physical interpolation.

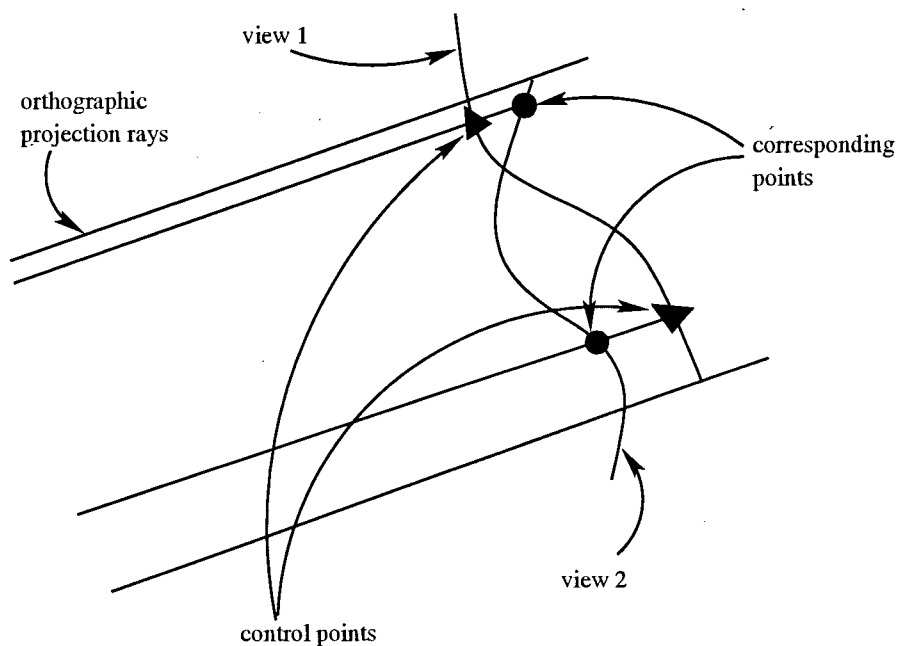


Figure 4.1: Diagram of the control point matching strategy, which is to match points in different rendered depth maps along same orthographic projection rays.

- **Point Pair Weighting.** We treat all generated point pairs equally by simply ignoring this step, all pairs are now assigned a weight of 1. Note that not all points from one range map have a corresponding point in the other, since the object may not cover the whole image and the stereo algorithm may not be able to recover the depth at every pixel, which means we would have to throw away the control points that can not be paired up by the point matching strategy.
- **Outliers Rejection.** In our model reconstruction system, no explicit rejection method has been used during the registration process. However, as mentioned in Chapter 3, a color segmentation method is used to get rid of

background points. This step, at the same time, eliminates most of the erroneous points along surface discontinuities employed by the limitation of stereo processing.

- **Surface Consistency Measure.** A sum of point-to-point distance based function is used as our consistency metric. Due to the fact that for a sum of point-to-point distances, the contribution of non-overlapping pixels to the consistency measure is equivalent to pixel pairs that have been perfectly aligned. It is not enough to consider the distances only, therefore the overlapping area has been employed to our surface consistency measure function as well. Details are covered in the “Surface Consistency Measure” subsection.
- **Optimization.** We use the standard “select-match-optimize” iteration scheme and a multidimensional numerical optimization method named “downhill simplex” [PTVF94] to find the local minimum around an initial starting point in the parameter space. Details of the “downhill simplex” algorithm will be described later on in subsection “Numerical Optimization”.

#### 4.2.3 Coordinate Systems Involved

The registration problem itself is originated from 3D representations of the same physical surface in different coordinate systems. So that to make our registration goal more clearly, we need to take a closer look at all the coordinate systems involved in our model acquisition system and find out how they are related to each other. There are all together three categories of coordinate systems explicitly involved in our model acquisition pipeline, which include a global world coordinate system, a series local sensor coordinate systems, and the window coordinate system of a computer screen. Figure 4.2 illustrates the coordinated systems mentioned here.

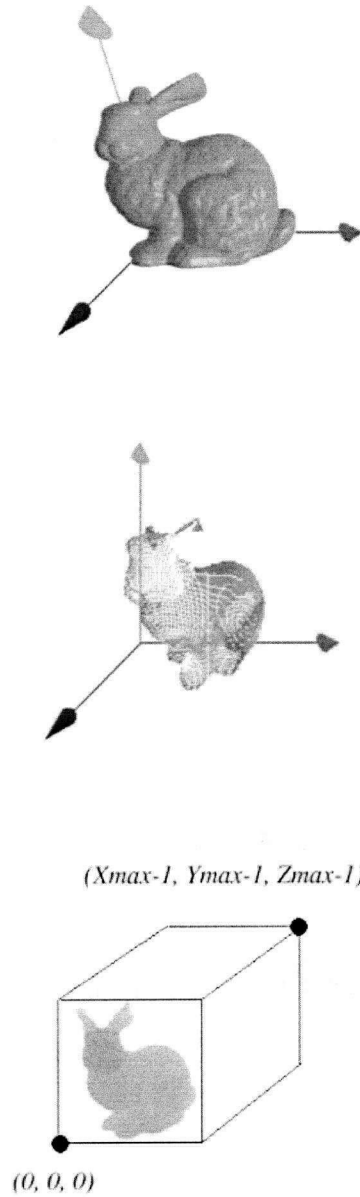


Figure 4.2: Three coordinate systems explicitly involved in the model acquisition pipeline. The top one is the unique global world coordinate system which is located at around the geometric center of the underlying subject, the middle one is one of the local sensor coordinate systems which are glued to each individual sensor position, while the bottom one is the window coordinate system which is a quantized coordinate system on a computer screen.



### **Local Sensor Coordinate Systems**

Those are the coordinate systems which the Triclops library functions use for the measurements of the underlying world object and the representations of the resulting 3D range data output. Each scanned 3D data set is represented in its individual local sensor coordinate system whose origin is always located at the pinhole position of the current reference camera. As explained Chapter ??, in our model acquisition system, in order to obtain enough range data for reconstructing a complete 3D model, instead of fixing the subject and moving the camera to different viewing positions, we actually fix the camera and moving the subject. This approach makes the data acquisition process much easier to manipulate, which is mathematically equivalent to the moving camera approach. Because of the duality, we can always think in the way that the camera moves around the fixed subject, therefore sensor coordinate systems keep changing during the range scanning process although the actual range finder is fixed. In our system, the local sensor coordinate systems are right-handed with a real dimensional measurement of the object in meters.

### **Global World Coordinate System**

Our destination, the 3D geometric model, lies in this space. It is fixed on the subject somewhere close to its geometry center. Without losing generality, we actually choose the world to be the translated local coordinate system of the first range image whose origin is located at about the center of the subject. As same as the local sensor system, it is right-handed and with a real measurement in meter. By using such a global coordinate frame, we can acquire a model of real size.

### **Window Coordinate System**

All the range data sets are rendered in this coordinate system for display on the screen and the surface consistency measure is also evaluated in this quantized coordinate system. The window coordinate system is actually an intermediate frame

employed by the rendering pipeline we use for registration. Once the registration process is finished, the range data will be transformed directly from its original local coordinate system to the global one and the window coordinate system will decrease. Unlike the above mentioned two, it is a left-handed system which is quantized into pixels in both  $x$  and  $y$  dimensions and scaled to the range of  $[0,1]$  in depth or the in dimension  $z$ .

Since that all the local sensor coordinate systems and the global one use the same measurement and that they all use the right-handed convention, there is only a rigid body transformation relating each of the them to one another. Specifically, one such transformation will bring one local coordinate system to the global coordinate system. Our registration process is to find these rigid body transformations for each of the local sensor coordinate frames so that all valid range data sets can be merged into the global coordinate system. Although there is a left-handed system, the window coordinate, involved in the registration process, it is actually transparent to the user, which is used by the OpenGL rendering pipeline and the evaluation of the surface consistency measure only. During the registration, both the reference data set and the current data set under registration are rendered into this window coordinate system using same rendering environment. So that all the geometric calculations can be carried out implicitly through rendering process and then the surface consistency measure function can be evaluated based on their rendering results in this quantized system too. Considering that they are both rendered using the same rendering parameters, if the comparison shows that the two rendered images align well in this frame, then it means the same rigid transformation aligns the local coordinate frame and the reference frame to the same accuracy as well.

#### 4.2.4 Parameter Space

Since this 3D acquisition pipeline starts from raw data obtained by cameras, are there any intrinsic camera parameters need to be found in the registration process? The answer is no. All parameters need to be recovered are the six parameters for rigid body transformation. Although the 3D scanner we use, the Digiclops camera, is actually a stereo vision system, which is made up of three CCD cameras, we don't need to consider any of the intrinsic camera parameters related to the vision system, such as the calibration factors caused by shape of the sensor or lens distortions. As already mentioned in Chapter 3, the system is precisely calibrated before shipping and all the intrinsic stereo parameters that matter are fixed and can be treated as system constants [Poi01a, Poi01b]. The Triclops SDK provides functions to transform points from the image coordinate system to object coordinate system which is the local sensor coordinate system mentioned in previous subsection. Therefore, the image coordinate system is actually invisible in the registration and integration processes and can thus be ignored. All parameters we need to find out in the registration process are the six rigid motion parameters describing rotations and translations of the sensor coordinate frame relative to the reference coordinate frame. So that our registration problem is actually a six dimensional optimization problem with each dimension in the parameter space representing one rigid motion parameter.

#### Representation of Rigid Motion Transformation

Euler angles are very popular used in defining 3D rotations. The basic idea of it is to factor a rotation matrix as a product of sub-rotations about the three major coordinate axes. The form of the factorization depends on the specified ordering of the three successive rotations, which in turn is determined by the needs of the application. The approach of factoring rotation as Euler angles makes it very easy to compose the rotation matrix using OpenGL API, which can be implemented by

basically issuing three standard OpenGL commands in sequence. There are various conventions for the sub-rotation ordering, such as “ $yzx$ ” convention and “ $zxy$ ” convention. We use the “ $xyz$ ” (sometimes named “pitch-roll-yaw”) convention, which is popular used in the computer vision community and aviation engineering. Under this convention, the rotation matrix is factorized as  $R = R_x(\theta_x)R_y(\theta_y)R_z(\theta_z)$ , where the subscripts  $x, y, z$  represent the axis around which the sub-rotations are conducted, and  $\theta_x, \theta_y, \theta_z$  are values of the sub-rotation angles around the corresponding axis.

As explained previously, our registration process starts from 3D point clouds, and the goal is to optimize the six degree rigid motion parameters between a local sensor coordinate system and the reference global coordinate system. Among those parameters, three are for translations along three coordinate axis and three are for the rotations about the three coordinate axis. The position of the range finder in the global world coordinate system which is abstracted as the origin of the local sensor coordinate system, is expressed by a translation vector,  $\vec{t} = (t_x, t_y, t_z)^T$ , while the relative orientation of it can be described by three angles  $R(\theta_x, \theta_y, \theta_z)$ , where the subscript  $x$  denotes the sub-rotation angle around the x-axis, similar meaning applies to the subscripts  $y$  and  $z$ . Therefore, a rigid motion of a 3D point  $\vec{p} = (p_x, p_y, p_z)^T$  relative to the reference global coordinate system can be given by a linear transformation  $T = R\vec{p} + \vec{t}$ . However, this representation has a disadvantage coming from its physical interpolation which is the expression of a rotation followed by a transformation. Note that the transformation is now conducted in the new, already rotated coordinate system. So that for an object point far away from the origin, small variations in one or more rotation angles may result in a large displacement of the object point, although the translation vector may be unchanged or very close to the previous one. This is a undesired feature in our work since the sampled surface points are almost always far from the origin of the global coordinate, which is somewhere close to the geometry center of the object. If the transformation

is instead described by a rotation  $R$  and a preceding translation  $\vec{t}$  (the rotation center is thus the translated origin), the effects of rotation and translation would be much better distinguished, and thus the above mentioned problem will be avoided. We have therefore decided to adopt this slightly different transformation scheme, which can be mathematically expressed as  $T = R(\vec{p} + \vec{t})$  [NK99, LHS01]. The six rigid motion parameters  $(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$  are those need to be recovered by the registration process. For each range image we acquire in its individual local sensor coordinate system, the six parameters have to be recovered in order for the data to be aligned to the model sitting in the global coordinate system. We offload these matrix transformations onto the graphics hardware using the OpenGL API for further calculations.

#### 4.2.5 Surface Consistency Measure Function

Throughout the range data registration process, it is necessary to compare two surface representations to evaluate to which degree they could refer to the same physical surface. A numerical surface consistency metric is employed for this purpose, which acts as the objective function to be optimized by the numerical optimization algorithm. Different local surface alignment metrics have been explored by researchers [Hub01]. In our approach an overlapping region based surface distance function is used.

For the registration process, we store the partial model that we have acquired so far in a voxel grid. From this partial model we can easily compute a dense range map by rendering a single OpenGL point for every occupied voxel. Similarly, every input range image can be converted into a range map seen from another view by simply re-projecting the individual points in the range data set.

As explained in the algorithm overview section, we match points along orthographic projection rays, where the pair wise distance is simply the distance between two corresponding points in  $z$  direction (in camera space). An intuitive alignment

metric is the sum of pair wise distances over the overlapping region in the two rendered range images, mathematically represented as:

$$consistency_1(T) = \sum_{\vec{p} \in S_o} d(\vec{p}, C(\vec{p}, T)) \quad (4.2)$$

where function  $d()$  is the 3D Euclidean distance between two points  $d(\vec{p}_1, \vec{p}_2) = \|\vec{p}_1 - \vec{p}_2\|$ ,  $T$  denotes the current transformation to be evaluated, and  $S_o$  (a function of the current transformation  $T$ ) denotes the overlapping area between the two depth images,  $C$  is the correspondence function that yields the point  $\vec{q}$  on the second surface, which is the projected corresponding point to point  $\vec{p}$  under the current transformation  $T$ .

Intuitively, the larger the accumulated distance value is, the larger the value of the surface consistency measure function will be. Therefore, the transformation yielding the best registration will be the one with the smallest consistency measure value. However, there's a problem using this consistency measure function. Usually, the larger the overlapping area is, the more corresponding point pairs will be taken into account, and thus the larger the consistency measure tends to be. While on the other hand, larger overlapping areas often indicate better alignment. By taking this into consideration, we normalize this consistency measure by the number of pixel pairs that have contributions to the accumulated distance to get an improved consistency metric.

$$consistency_2(T) = \frac{consistency_1(T)}{\|S_o(T)\|} \quad (4.3)$$

Where  $\|S_o(T)\|$  is the number of matched point pairs in the overlapping area. However, there is another problem arising from the second definition. When there is no corresponding point found for a control point, the point is simply ignored; while when the two paired points are perfectly aligned, the distance between them is 0. Thus the contributions of the two opposite cases to the consistency measure are identical! In order to encourage larger overlap regions and distinguish the above

mentioned two cases, we employ a threshold  $\epsilon$  into our consistency measure function:

$$\text{consistency}(T) = \begin{cases} C_1 & \text{if } \|S_o\| = 0; \\ \frac{C_2}{\|S_o(T)\|} & \text{if } 0 < \|S_o\| < \epsilon; \\ \frac{\sum_{\vec{p} \in S_o} d(\vec{p}, C(\vec{p}, T))}{\|S_o(T)\|} & \text{if } \|S_o\| \geq \epsilon. \end{cases} \quad (4.4)$$

where  $C_1$  and  $C_2$  are simply two predetermined constants. These two constants and the threshold have to be carefully chosen so that the condition

$$C_1 > \frac{C_2}{\|S_o(T)\|} > \frac{\sum_{\vec{p} \in S_o} d(\vec{p}, C(\vec{p}, T))}{\|S_o(T)\|} \quad (4.5)$$

is always true. Here, we choose  $C_2 = \|S_o(T)\|$ , and  $C_1 = 1.5$ , since the condition  $d(\vec{p}, C(\vec{p}, T)) < 1.0$  always holds for depth buffer images whose values have been normalized to  $[0, 1]$ .

#### 4.2.6 Optimization Algorithm

The discussions in subsection 4.2.4 have made it clear that the registration problem confronted in this project is a multidimensional optimization problem. Multidimensional here means the number of parameters we are optimizing is more than one. We choose the downhill simplex method [PTVF94] as the numerical optimization method for this task, which simply slides downhill in cost in a self-contained manner that almost does not put any limitations on the cost function. In our application, the surface consistency measure function acts as the cost in the downhill simplex algorithm. This algorithm requires no derivatives at all, only function evaluations are needed. Although in terms of the number of function evaluations it requires, downhill simplex method appears not very efficient, it is often the most time efficient method when the cost function is not very complex. Named “downhill simplex”, this

method has nothing to do with a simplex which is a geometric figure consisting of one more vertex than the number of dimensions of the space it spans. The simplex here refers to a figure in the parameter space and the naming simply uses the deformation of a simplex in the parameter space to figuratively describe the searching procedure. Generally only simplexes without degeneracy are considered valid. Each vertex of a simplex in an  $N$  dimensional parameter space can be represented by an  $N$ -dimensional vector which when viewed outside of the downhill simplex algorithm and within our 3D registration application is the representation of a six dimensional rigid motion transformation.

As same as most of multidimensional optimization algorithms, downhill simplex method requires an initial guess to get started. Then guided by its inner mechanism, the algorithm has the ability to go downhill until it converges at a local minimum. The starting point is actually  $N + 1$  points spanning a simplex in the  $N$  dimensional parameter space. The overall force pushing the simplex to a local minimum is the goal to replace the vertex with the highest cost with a vertex with a lower cost. Working iteratively, each iteration starts by a sorting procedure, which is to sort the vertices according to their cost. There are four types of deformation actions that a simplex can take in each iteration: reflection, extraction, contraction and multidimensional contraction. Before all actions except for the multidimensional contraction can really happen, a trial deformation of the simplex is conducted and the cost of the potential vertex has to be tested to see if the trial deformation is feasible. Only the deformations that get rid of the highest point can be really made out. The possible simplex deformation actions in a two dimensional space are depicted in Figure 4.3 and summarized below:

- A reflection is to try the point that is reflective of the highest vertex against its opposite face (not necessarily two dimensional). Reflection reserves the inner volume of the simplex and thus maintain the degeneracy of the simplex.
- An expansion happens when the previous action reaches a lower cost point.



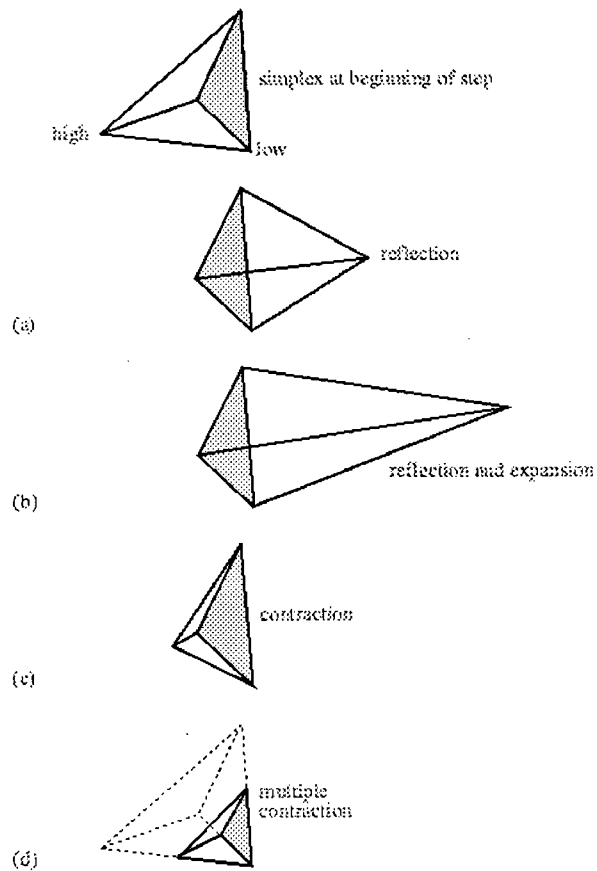


Figure 4.3: Possible outcomes for simplex deformation trial steps: a) A reflection. b) An expansion following the reflection. c) A one dimensional contraction. d) A multidimensional contraction [PTVF94].

It is the trial deformation of expanding the simplex further along the same direction as the previous successful reflection or expansion, which makes the searching space expanding.

- A contraction trial happens when the expansion can not be applied. The simplex contracts along the reversed direction of its preceding successful reflection, expansion or contraction.
- A multidimensional contraction applies if none of the above trials has found a

feasible point. This is the action that the simplex contracts along all directions towards the lowest vertex, which may make the process converge too fast to a local minimum. Lensch et al [LHS00] proposed a random displacement of the highest cost point within the simplex using the simulated annealing method to replace the multidimensional contraction deformation. Although this approach avoids the fast local convergence problem in most cases, it is too slow to be suitable for our interactive modeling project. Here we keep the multidimensional contraction step, as our starting points are close to the real optimization (refer to Chapter 3), the local convergence problem does not seem to affect the registration result much.

To start the downhill simplex method,  $N + 1$  vertices are needed to initialize a simplex. In our approach, one of them is the initial guess of the rigid motion, and the other  $N$  points can be carefully deduced from the single initial point. We simply double one of the  $N$  original parameters in turn while keeping others unchanged, which can be mathematically represented as  $P_i = P_0 + P_0^i$ , where  $P_0$  is the initial guess of the rigid motion and  $P_0^i$  is its  $i^{th}$  component,  $P_i$  is the  $i^{th}$  vertex of the starting simplex.

To terminate the process, however, we have to take all the dimensions and thus all the simplex vertices into consideration. Press et al's termination criteria is used here, which is to end the iteration when the cost of all simplex vertices are so similar that further iterations won't improve it much or when the number of deformation trials has reached a predefined maximum number. The downhill method is then restarted immediately using this result as the initial guess until the results of two successive downhill simplex optimization processes do not differ much. The pseudo code of this algorithm is listed below:

```

Initialize;
While not terminate
{  sort all vertices according to their costs,
   do a reflection trial step,
   if (the trial cost < the lowest cost)
       do an expansion trial step,
   else if (the trial cost > the second highest cost)
       do a one diemnsional contraction trial step,
       if (the trial cost > the highest cost)
           do a multidimensional contraction.
}

```

The functionality of each deformation trial step is two-fold: one is to return the cost value of the potential point; the other is to replace the highest point with the trial point if the trial cost is lower than the highest cost, or to keep the previous shape and cost otherwise.

For our interactive model acquisition application, the subject is moved slowly and in a consistent manner before a fixed range finder, so that we can use the transformation parameters obtained from the previous frame as the starting point and usually this approach leads to an acceptable solution according to the predetermined criteria. If the motion of the subject is too fast or not in a consistent manner and the two successively obtained range images do not have good overlap or the rigid transformations for two successive range data sets differ too much, the registration process may fail, and the algorithm has to be reset to a new starting situation. (for example by roughly aligning the partial model by hand to the next image).

## Chapter 5

# Hardware Acceleration

The current generation of graphics hardware is quite flexible in its programmability, and its use is no longer limited to the traditional rendering and animation purposes [Kil02]. Our work of using graphics hardware for registration purpose is inspired by the fact that the graphics hardware is specifically designed to out-strip the performance of conventional CPU hardware for the class of computations that graphics hardware is dedicated to. Our model acquisition system accelerates the registration process by taking advantages of the contemporary graphics hardware functionality from two separate aspects.

### 5.1 Off Loading Geometric Computations onto GPU

The contemporary graphics hardware today is comparable to CPUs in design and transistor complexity, so that they are often called GPUs. They are positioned to improve the performance of graphics computations which are highly parallel processes with very pipelineable algorithms. Consider that the registration process involves mainly geometric computations and that graphics hardware is designed to accelerate exactly those kind of computations, we can thus bring graphics hardware that traditionally was specialized for conventional 3D rendering and animation operations to bear on our registration task for accelerating the geometric computations.

We therefore off load the geometric primitives, here are the 3D point clouds output from the range acquisition module, onto the graphics board. Then, by rendering the 3D points into proper coordinate frames, the mathematical computations that carry out the 3D transformations are thus done through the rendering pipeline by specifically optimized hardware. The two range data sets used for the evaluation of the surface consistency measure function, which are two depth images, are generated as the rendering results.

This approach, on one hand saves CPU time for other tasks by off loading all geometric computations to the dedicated graphics board, on the other hand paralyzes the complex geometric computations which is the reason for developing graphics hardware by carrying out the math calculations implicitly through the graphics rendering pipeline.

## 5.2 Accelerating Surface Consistency Evaluation Using NVIDIA Register Combiners Mechanism

The most time consuming process for our 3D range registration algorithm is the multidimensional numerical optimization which includes several tens of function evaluations of the alignment metric. For each such evaluation, we need to use not only the depth image rendered from the current point set but also the depth image rendered from the reference geometry, which means we need to access two pieces of memory for the most demanding operation. Note that we already use graphics hardware to accelerate the registration process by generating the two range images in the first place. However, we can go further. If the two depth images can be somehow combined into one, then accessing one piece of memory would be enough to evaluate the alignment metric, that would help speed up the registration process. However, the traditional OpenGL environment for combining pixel primary colour with several texture colours are lacking.

This can be achieved with the help of a specific piece of graphics hardware, the NVIDIA register combiners [NVI02] or a similar programmable hardware, which provides a flexible way of combining several parameters to arrive at a final pixel colour. Supporting at least two versatile general combiner stages and a special final combiner stage which is meant to perform a colour sum and fog computation, the register combiners sit in the OpenGL rendering pipeline as a rasterization processing stage operating in parallel to the traditional OpenGL texture environment. The Register Combiners mechanism allows a developer to specify a number of inputs, perform addition, multiplication or dot products, and produce an output, which may in turn be used as the input to a subsequent combiner. Inputs can be any of the texture colours, vertex colors, or even constants. When the register combiners are used for rasterization purpose, the traditional texture environment will be ignored and its state will not be changed by the register combiner stage. Because the developer has control over the inputs to its register set and operations performed on them, the register combiner mechanism can be used to help implement some functions in hardware for very sophisticated shading techniques. Some tasks requiring multi-pass rendering with the traditional rasterization functionality can be implemented in fewer passes by using the register combiners. This is where our idea of combining the two depth images into one with rendering pass comes from. A diagram depicting how a complex texture is composited by this specific piece of hardware using OpenGL extension is shown in Figure 5.1.

Our accelerating strategy starts by encoding the depth information into a rendered color image by applying a one dimensional linear texture whose coordinate has been automatically generated as the distance to a predefined  $xy$  plane. And then map the texture coordinates to  $z$  direction of the object during the rendering process. After that, the absolute distances of the depth values between corresponding points of the reference depth map and the current one are calculated implicitly by using the register combiners texture compositing mechanism. Two general register combiner

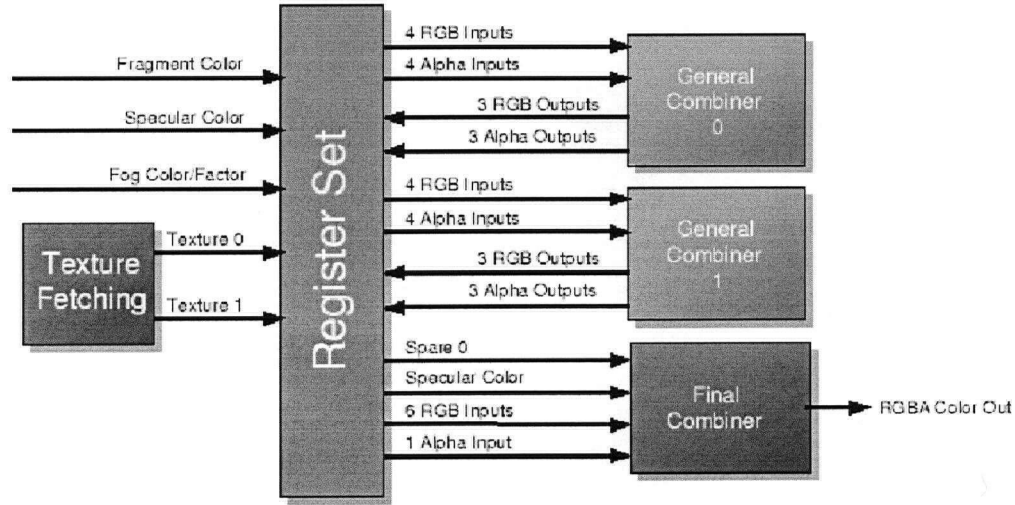


Figure 5.1: Diagram of texture compositing strategy with OpenGL NVIDIA register combiners extension. [Kil02]

stages and one global constant color are used in our implementation.

The mathematical expression of the distance function (see subsection Surface Consistency Measure) is the absolute difference between the two depth values stored in the same pixel position of two different depth maps. To calculate the absolute difference, we take advantage of the “mux” operation provided by the register combiners, for which the resulted pixel value is determined by a special alpha channel value:

$$\text{mux}(AB, CD) := (\text{Spare0}[\alpha] \geq 0.5) ? AB : CD \quad (5.1)$$

Where  $AB, CD$  represent the two texture operands, and  $\text{Spare0}$  is one of the registers in the register combiners' register set. In general combiner stage 1, an intermediate image is obtained by programming the combiners carefully. The pixel values

in the intermediate image are generated as:

$$(tex1 + (1 - tex2))/2, \quad (5.2)$$

which will then act as the special alpha values mentioned above. Since we have

$$0 < tex1, tex2 < 1 \quad (5.3)$$

the resulted pixel values fall into the same range as well. Then in general combiner stage 2, the intermediate image is mapped to

$$\begin{aligned} A &= 2 * (tex1 + (1 - tex2))/2 - 1 \\ C &= -2 * (tex1 + (1 - tex2))/2 + 1 \end{aligned} \quad (5.4)$$

and both  $B$  and  $D$  are assigned constant 1. By subscribing  $A, B, C, D$  into equation 5.1, the pixel wised values of the absolute distances between  $tex1$  and  $tex2$ ,  $|tex1 - tex2|$ , can be generated.

Finally, after the mux operation is applied, and the pixel wised absolute differences are generated in the rendering process, the surface consistency measure function can be evaluated by reading back this rendered image and summing up the per-pixel differences over the entire overlapping area.



## Chapter 6

# Model Integration

In the previous chapters, we have shown how the information for building a 3D geometric model could be obtained by the range scanning and the registration processes in our model acquisition system. This chapter addresses the problem of how this information can be properly merged together to build realism representation of the object. Since our system is aiming at an interactive working mode, realistic is not the only concern. To give the user a quick enough feedback about how the model looks like currently so that the user can verify the model and detect holes during the acquisition process is a must. Also, the model acquired in the interactive phase of the system is trading off accuracy for speed, therefore the uneven distribution of the registration inaccuracy may result a significantly different model although the individual range data set is accurately registered. So if desired, the same registration algorithm can later be used offline as a post-process method or some other global optimization strategies can be applied to produce a model with a higher accuracy.

### 6.1 Merging Algorithm

Traditionally, computer graphics uses triangles as rendering primitives, which approach has large costs on setup and rasterization for scenes consist of large amounts of geometry. In order to decrease such costs and improve the efficiency for surface

rendering and display, the approach of point rendering which uses points as display primitives for continuous surfaces has been proposed [IW85]. Given the goal of a live feedback of the model being scanned and reconstructed, some way must be found to display the constantly accumulated partial model at an acceptable frame rate. Modeling algorithms commonly used these days for display, simplification, and progressive transmission of meshes are too slow to be suitable for real-time use. For an in-depth survey of the existing merging algorithms, refer to paper [Cur97]. Usually, those approaches fall into two major categories: merge by operating only on point clouds [HDD<sup>+</sup>92, NAK98] or merge using the connectivity information as well [TL94, BL96].

We therefore adopt a 3D grid data structure and a point rendering approach which draws raw range sample points directly without reconstructing the polygonal surface mesh. This method does not generate the surface model with the highest quality that is possible these days, however it is faster and feasible to display the constantly increasing model on current hardware when the scanning process is on going, so that the user can see which areas of the model have been scanned and where there are holes need to be filled. By using this approach, we sacrifice the quality of the 3D model under reconstruction in order to get a reasonably fast merging and display rate for a live feedback. Some more accurate merging algorithms may be applied offline after enough range data have been obtained and roughly aligned to get a surface reconstruction with a higher quality.

### 6.1.1 Voxel Grid Data Structure

As stated previously, our real-time scanning system adopts a 3D grid model and a point rendering method for interactive model integration and display. An octree hierarchical data structure is used to achieve the compression of regions of empty space that is not occupied by any surface point. The whole octree represents the model we have so far acquired interactively. In each occupied octree cell, the 3D

location of the surface point in the form of a vector of 3 components, the RGB colour data of the point, and an associated rough normal vector are stored. For the grid resolution, theoretically, a cell size on the order of the spacing of the range data output from the range scanner is optimal. Smaller grid cells use more memory, slow down the rendering frame rate and be more sensitive to noise, yet it won't result in a better quality limited by the resolution of the input range data from the range finding process. Considering the purpose for using point rendering, which is to trade off accuracy for an interactive frame rate, we adopt this optimal cell size or a cell size a little bit larger than this one.

### 6.1.2 Voxel Model Integration

Our voxel model merging method discards all the connectivity information and performs surface merging based on the point sample themselves only. Two major tasks are tackled in this process, one is to quantize the points to be merged to a 3D regular voxel grid and the other is to merge the quantized points into the grid stored in the octree. Since surface data is accumulating rapidly, in order to maintain an acceptable rendering rate, the number of primitives that must be rendered has to be restricted. Here we simply combine all points in one grid to one point and discard all redundant, overlapping 3D data, which idea is similar to Rossignac and Borrel's mesh simplification method. As to color, in the range acquisition stage, we don't control lighting conditions at all, so that colors obtained for the same surface point may vary a little bit from view to view. The color information stored in the acquired model is only roughly true. Therefore, we can randomly choose the color of one point falling into an octree cell as the color of this cell or use a mean color in some sense. For the use of the associated rough normal vector, as specified in Chapter [?], our registration process uses the depth image rendered from the current partial model as the reference frame, but since the model is not closed yet and also there might be some holes on the surface, part of the depth image may not be rendered from

the points on the front surface which is desired. Instead, it maybe the rendered image of points on the back surface seem through the existing holes or from the side that has not been closed yet. This will lead to significant registration errors for the reference frame may contain wrong geometric information. In order to get around this problem, we apply a back surface culling algorithm when rendering the reference from the partial model. This approach makes sure that only those points on the front surface will get rendered into the reference frame. A rough normal associated with each surface point is maintained at each occupied voxel mainly for applying this algorithm. We dynamically update this rough normal by the averaged viewing direction from which this cell has been seen so far. As surface samples coming from multiple range images scanned from different point of views may map to the same voxel of the model, the rough normal associated with each voxel changes as the point data accumulating into the octree storing the model. Although this rough normal is sometimes far from the real one, it works well enough for our back face culling algorithm.

## 6.2 Global Optimisation

Usually, for multiple range view registration systems a gap may appear between the first and the last views due to the error accumulation between successive range images. This is caused by the local nature of both the registration process and the geometric information available in each individual view. In our interactive registration system, we are able to trade off accuracy for speed. So a slower offline post-process using the same strategies may be applied to reconstruct a final 3D model of a higher quality.

Registration algorithms can be classified into two categories, pairwise registration and global registration. Pairwise registration algorithms register two data sets at a time while global registration algorithms perform registration process on multiple data sets simultaneously. Our depth based range registration algorithm is

a pairwise registration technique, but its usage in this system is not limited to pairwise registration only. As mentioned previously, we use the rendered depth image of the current partial model instead of an individual range image as the reference, which actually consists of multiple range data sets. In this way we are actually registering one range data set to the combination of all data sets aligned to the model previously simultaneously. But still an uneven distribution of the registration error may result, some global optimisation strategy is needed to decrease the accumulated error due to the local nature of the registration process.

In our model acquisition pipeline, after the interactive registration pass, we have already gained more preliminary information for the under constructed 3D model, which include

- the user is confident that enough range data sets have been acquired by the range finding module from the interactive working mode (i.e. no holes left unfilled, and the model can be closed);
- 3D registration results obtained by the interactive process are available as very good initial guesses for the numerical optimization algorithm running in the post-processing module;
- a preliminary voxel model stored in an octree is available.

Based on the available information, the improvements of the 3D model can be approached from several different approaches: one is to use a better rigid motion prediction obtained from the interactive registration process as the starting points of the numerical optimization, which not only decreases the number of iterations required for registering one image speeding up the numerical process, but also improves the registration accuracy; the second approach is instead of registering the current range image in sequence, we can do a global optimization step by taking the

full model, deleting all point contributed from one view, and then re-registering that view into the model. By applying this global registration strategy, the individual range data set is sort of registered to all the other data sets simultaneously. As views are randomly picked and treated like this, we can gradually improve the registration of all views. Finally, we can use a voting scheme to determine outliers in the geometric model (e.g. erroneous data provided to us by the stereo algorithm in the range finding module). We do this by assuming that only those voxels containing surface points contributed from a certain minimum number of different views (2-4 usually) yield good results for us.

Given all the preliminary information, the post-process improving the model from the above mentioned three approaches can be run completely automatically.

## Chapter 7

# Experimental Results

Before describing the test results, some practical issues concerning the implementation are discussed.

**Range Data Acquisition:** we use the Triclops library which is the software development kit shipped with the Digiclops. The output from this process is the corresponding 3D point cloud ready for registration using rendering pipeline.

**Test system:** our model acquisition system is implemented on a system running Redhat Linux 7.3, with Intel Pentium 41.6GHz processor and a GeForce 3 GPU. The first part is the scanning process including the generation of 3D point cloud, which operates at 3-4Hz, and the later one for registration and integration operates at about 1Hz on average. This could be accelerated further by using newer hardware or a dual-processor machine (since the stereo matching and the registration part are independent of each other).

**Model Integration:** for a quick rendering, we use a voxel model stored in an octree model for the interactive mode. After a range image has been registered to the global coordinates system its points are merged into this octree.

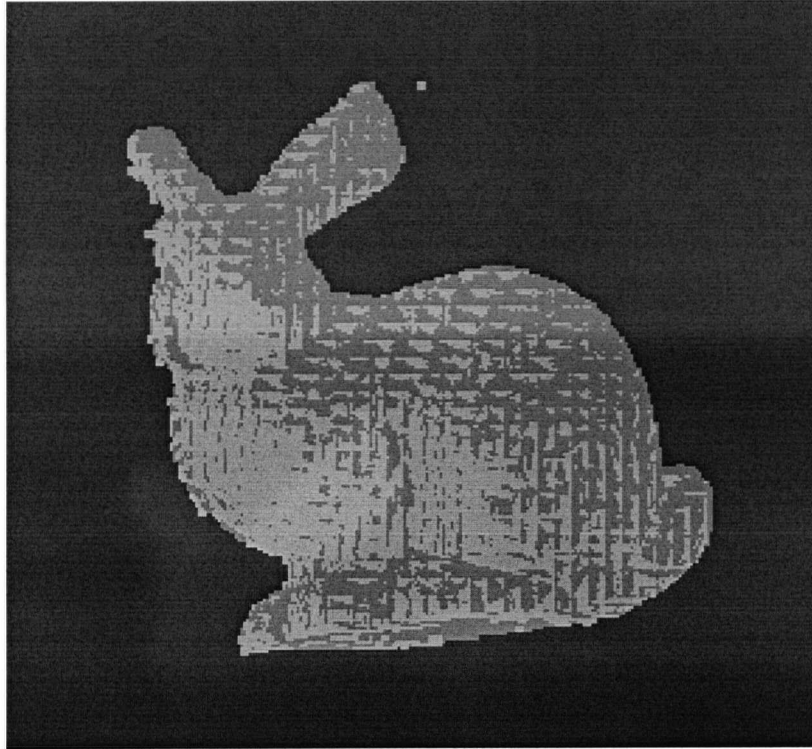


Figure 7.1: Experiment result of the synthetic data registration. A picture of surface points from two different views registered to the global coordinate system.

## 7.1 Synthetic Range Image Test for Registration

We tested our range based 3D-3D registration algorithm on a pair of synthetic range images created from a polygonal geometric model orthogonally mapped on the  $x-y$  plane. The second range image is generated from the same model transformed by a rotation vector of  $(-5.0, 15.0, 5.0)$  (rotation parameters are in degrees) followed by a translation of 5 percent of its bounding box size along all three major axes. Both original range data sets consist about 17,000 points. The merged partial model is shown in Figure 7.1. Here, we use different colors to distinguish between



pixels coming from different scans. The z-fighting of the two scans indicates a good alignment. For the computation of per-pixel difference, both implementations using hardware acceleration and calculating in software directly have been tested and compared. To demonstrate the algorithm's ability to converge, we adopted various starting guesses which are all not very close to the true value. The standard deviation for all the six rigid motion parameters have been calculated, among which the standard deviations for rotations are less than 0.5 degree and for translations are less than 2 percent of the bounding box size of the subject. Considering the window size used for calculations, the results are satisfying.

From these experiments, we can make the following observations: (1) The algorithm is capable of aligning range images accurately. (2) The fully hardware accelerated implementation has a lower accuracy relative to the implementation that computes the per-pixel difference in software. This is because the GPU we use has only eight bits for each color channel-the most recent generation of graphics hardware should not have these problems. (3) The fully hardware accelerated implementation is faster although there is not too much difference. As mentioned above, the starting position for this test is not close to the true value, the registration would be faster for better initial guesses.

## 7.2 Real Model Acquisition Result

We applied the model acquisition system to a textured ceramics model (Figure 7.2). During the acquisition process, the subject is moved slowly relative to the Digiclops, the newly scanned data set is registered to the partial model, and the updated model is rendered to the screen at the same time. For better control without occluding the object, we actually put the subject onto a plate and rotate the plate instead of the subject itself or the camera. After it is rotated for a whole circle, all the side views have been captured, the subject is then rotated about the axis that is roughly normal to the previous axis to get the data at the bottom of the model which can

not be acquired by the circular scanning. By watching the immediately updated model, the user can find holes where no data has been acquired, and do more scans of that area to fill the holes. The movements have to be slow enough that successive scans overlap each other in a not too small area so that the registration result from previous views can be used as the starting point for the registration of the current view.



Figure 7.2: A picture of the ceramics model unicorn.

Figure 7.3 shows several results of the 3D model acquired from this subject. The top image shows the result for a voxel grid of resolution  $100 \times 100 \times 100$ , while the bottom left image has been generated with a resolution of  $200 \times 200 \times 200$ . The bottom right one includes the global optimization step. As can be seen, the stereo algorithm combined with our registration method works quite well. There are, however, problems with very sharp features, such as the horn of the unicorn. Here the stereo data is very sparse, so that it is hard to capture the horn completely. In addition, the sparseness of the data also causes problems with the alignment.

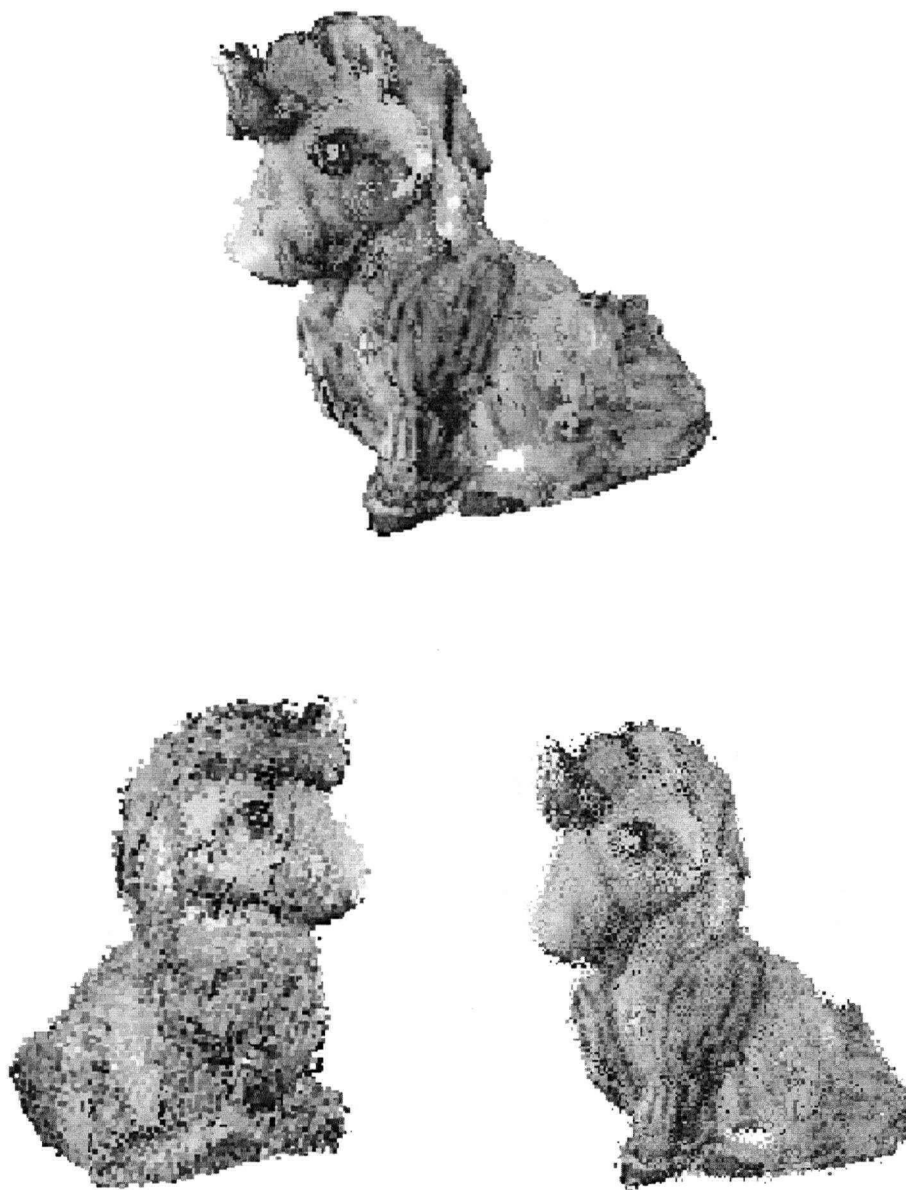


Figure 7.3: Results of the real world object model acquisition. Bottom two: a low (left) and a high (right) resolution results without global optimization. Top one: high resolution model with global optimization.

## Chapter 8

# Conclusions and Future Work

In summary, we have presented a 3D geometric model acquisition pipeline which consists of three modules: a range scanning module, a 3D-3D registration module and a model integration module. The system is aimed at an interactive working mode, so that efficiency is the major concern for each of the three modules. For the range scanning, we choose the commercialized stereo system, the Digiclops, as the range finder, which works in real-time. For the 3D-3D registration process, we adopt a depth based ICP algorithm, which takes advantages of the modern graphics hardware for geometric computations and thus runs faster than common ICP variants. The fast converging downhill simplex method is used for numerical optimization. An important contribution of this system is a hardware accelerated approach which combines two images into one and thus greatly reduces the necessary amount of memory access for the demanding computation task of surface consistency evaluation. For the model integration module, we use a volumetric model which is fast to add new data on and fast to render as well. The major effort, however, has been in the acceleration of the 3D-3D registration process using graphics hardware.

The results that we obtained using our system are quite encouraging. in the future, we would like to further improve the quality of the final model, for example by performing a better color filtering to reduce shading irregularations. We also hope

to make use of a multi-processor machine to improve the performance to about 5 frames per second. At that point, the initial guess for the new positions will be more accurate (due to faster update rates), so that the increase in performance will also manifest itself in an increased precision.

# Bibliography

- [Bes88] P. Besl. Active optical range imaging sensors. *Machine Vision and Applications*, 1, 1988.
- [BL93] G. Blais and M. D. Levine. Registering multiview range data to create 3d computer objects. Technical Report TR-CIM-93-16, Center for Intelligent Machines, McGill University, Montréal, Québec, Canada, November 1993.
- [BL95] G. Blais and M. Levine. Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8), 1995.
- [BL96] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH*, 1996.
- [BM92] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992.
- [BS97] R. Benjema and F. Schmitt. Fast global registration of 3d sampled surfaces using a multi-z-buffer technique. In *Proceedings of 3-D Digital Imaging and Modeling*, 1997.
- [CHC98] C. Chen, Y. Hung, and J. Cheng. A fast automatic method for registration of partially-overlapping range images. In *Proceedings of 3-D Digital Imaging and Modeling*, 1998.
- [CHC99] C. Chen, Y. Hung, and J. Chen. Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1229–1234, 1999.
- [Chu96] C.S. Chua. 3d free-form surface registration and object recognition. *International Journal of Computer Vision*, 17:77–99, 1996.

- [CM91] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings of IEEE Conference on Robotics and Automation*, 1991.
- [Cur97] Brian Curless. *New Methods for Surface Reconstruction from Range Images*. PhD thesis, Stanford University, 1997.
- [Cyb] Cyberware. <http://www.cyberware.com>.
- [DWJ97] C. Dorai, J. Weng, and A. Jain. Optimal registration of object views using range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10), 1997.
- [DWJ98] C. Dorai, J. Weng, and A. Jain. Registration and integration of multiple object views for 3d model construction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 1998.
- [ea95] K. Higuchi et al. Building 3-d models from unregistered range images. *Graphics Models and Image Processing*, 57(4):315–333, 1995.
- [ELF97] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: A comparison of four major algorithms. *Machine Vision and Application*, 9(5/6), 1997.
- [FH86] O. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *International Journal of Robotic Research*, 5(3), 1986.
- [FHM<sup>+</sup>83] O. Faugeras, B. Hotz, H. Mathieu, Z. Zhang T. Viéville, P. Fua, E. Théron, L. Moll, G. Berry, P. Vuillemin, and C. Proy. Real time correlation-based stereo: Algorithm, implementations and applications. Technical Report RR-2013, INRIA, 1983.
- [GRB94] G. Godin, M. Rioux, and R. Baribeau. Three-dimensional registration using range and intensity information. In *Proceedings of SPIE: Videometric III*, volume 2350, 1994.
- [GTK92] A. Gruss, S. Tada, and T. Kanade. A vlsi smart sensor for fast range imaging. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 1992.
- [HDD<sup>+</sup>92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of SIGGRAPH*, 1992.



- [HRG97] G. Taubin H. Rushmeier and A. Guézic. Applying shape from lighting variation to bump map capture. In *Proceedings of Eurographics Rendering Workshop*, 1997.
- [Hub01] D. F. Huber. Automatic 3d modeling using range images obtained from unknown viewpoints. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, pages 153–160, May 2001.
- [Inc] 3DV Systems Inc. <http://www.3dvsystems.com/>.
- [JH97] A. Johnson and M. Hebert. Surface registration by matching oriented points. In *Proceedings of 3-D Digital Imaging and Modeling*, 1997.
- [JK97] A. Johnson and S. Kang. Registration and integration of textured 3-d data. In *Proceedings of 3-D Digital Imaging and Modeling*, 1997.
- [Kil02] M. J. Kilgard. Graphics hardware functionality for geometric computations with opengl. *Circa*, 99, 2002.
- [LHS00] H. Lensch, W. Heidrich, and H.-P. Seidel. Automated texture registration and stitching for real world models. In *Proceedings of Pacific Graphics '00*, October 2000.
- [LHS01] H. Lensch, W. Heidrich, and H.-P. Seidel. Silhouette-based algorithm for texture registration and stitching. *Graphical Models*, 63(4):245–262, July 2001.
- [LPC<sup>+</sup>00] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D Scanning of Large Statues. In *Proceedings of Siggraph*, pages 131–144, July 2000.
- [LRG00] U. Labsik, S. Roman, and G. Greiner. Depth based registration of free-form surfaces. In *Proceedings of the 2000 Conference on Vision and Modeling and Visualization*, pages 121–128, November 2000.
- [IW85] M. levoy and T. Whitted. The use of points as display primitives. Technical Report TR 85-022, 1985.
- [Mat97] Y. Matsumoto. A portable three-dimensional digitizer. In *Proceedings of 3-D Digital Imaging and Modeling*, 1997.

- [MB93] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine intelligence*, 15(5), 1993.
- [MBR<sup>+</sup>00] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-based visul hulls. In *Proceedings of SIGGRAPH*, 2000.
- [MSY96] T. Masuda, K. Sakaue, and N. Yokoya. Registration and integration of multiple range images for 3-d model construction. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996.
- [NAK98] M. Bern N. Amenta and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *Proceedings of SIGGRAPH*, 1998.
- [Neu97] P. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Proceedings of International Conference on Shape Modeling and Applications*, March 1997.
- [NK99] P. J. Neugebauer and K. Klein. Texturing 3d models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum*, 18(3):245–256, September 1999.
- [NVI02] NVIDIA Corporation. *NVIDIA OpenGL Extension Specifications*, 2002. Available from <http://www.nvidia.com>.
- [Poi01a] PointGrey Research. *DIGICLOPS Installation Guide and Camera Control API Command Reference*, 2001.
- [Poi01b] PointGrey Research. *TRICLOPS Stereo Vision Software Development Kit (SDK)*, 2001.
- [PTVF94] W. Press, S. Teukolsky, W. Vetterling, and N. Flannery. *Numerical recipes in C : the art of scientific computing*. Cambridge Univ. Press, 2nd ed. edition, 1994.
- [Pul99a] K. Pulli. Multiview registration for large data sets. In *Proceedings of 3-D Digital Imaging and Modeling*, 1999.
- [Pul99b] K. Pulli. *Surface Reconstruction and Display from Range and Color Data*. PhD thesis, University of Washington, 1999.

- [RCM<sup>+</sup>01] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno. A low cost 3d scanner based on structured light. In *Proceedings of Eurographics*, September 2001.
- [Res] Point Grey Research. <http://www.ptgrey.com/>.
- [RHHL02] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *Proceedings of Siggraph*, pages 438–446, July 2002.
- [RL01] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings of Third International Conference on 3D Digital Imaging and Modeling*, 2001.
- [SA98] I. Stamos and P. Allen. Interactive sensor planning. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998.
- [Sim96] D. Simon. *Fast and Accurate Shape-Based Registration*. PhD thesis, Carnegie Mellon university, 1996.
- [SKSI90] Y. Sakaguchi, H. Kato, K. Sato, and S. Inokuchi. Generatio of 3-d models based on image fusion of range data, November 1990.
- [SKSI91] Y. Sakaguchi, H. Kato, K. Sato, and S. Inokuchi. Acquisition of entire surface data based on fusion of range data. *IEEE Transactions*, 74(10):3417–3422, October 1991.
- [SM92] F. Stein and G. Medioni. Structural indexing: Efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.
- [SNN96] M. Watanabe S. Nayar and M. Noguchi. Real-time focus range sensors. *IEEE Transactions on Machine Pattern Analysis and Machine Intelligence*, 18(12), 1996.
- [TL94] G. Turk and M. LevoyY. Zipped polygon meshes from range images. In *Proceedings of SIGGRAPH*, 1994.
- [VA86] B. C. Vemuri and J. K. Aggarwal. 3-d model construction from multiple views using range and intensity data. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 435–437, 1986.

- [Wei97] S. Weik. Registration of 3-d partial surface models using luminance and depth information. In *Proceedings of 3-D Digital Imaging and Modeling*, 1997.