3D Object Modelling via Registration of Stereo Range Data

Kenneth Haugaard Møller

Kongens Lyngby 2006
Master Thesis IMM-Thesis-2006-08

Abstract

Stereo vision has several advantages over other 3D imaging methods, but still it is mainly active solutions that are established on the market of commercial 3D modelling equipment. However, papers have recently been published presenting real time stereo matching on the GPU. So with the increasing demand for cheap 3D scanners and the advances of computer power along with new possibilities of efficient image processing on graphics hardware, the time has come to explore the full potential of stereo vision.

Assuming real time stereo range data is available, this thesis is a feasibility study of whether real time stereo can produce good enough results for creating an online preview of the scanning process. Being able to see the incrementally building model as it is scanned acts as a sort of online view planning and has huge advantages in flexibility and the amount of time used when doing 3D scanning.

To test this, a system for acquisition of real time stereo data has been built. The implemented stereo matching algorithm is based on summation of different support region levels to ensure robustness and maintain the distinct features of the objects topology. Assembling the model is done via registration of the range data using the Iterative Closest Point algorithm, and finally a simple and fast way of merging the aligned data, suitable for incremental integration of a real time system, is presented.

Problems, limitations and advantages of such a system are discussed along with proposals and needs in order to obtain a fully operational 3D modelling system. Finally, the system is tested on a variety of objects differing in shape and texture, and the good results are presented.

Keywords: 3D modelling, 3D scanning, stereo matching, stereo correspondence, range data registration, Iterative Closest Point, real time preview.

Resumé

Stereo vision har mange fordele frem for andre 3D scanningsmetoder, men alligevel er det hovedsagligt de aktive løsninger der har etableret sig på det kommercielle marked for 3D scanningsudstyr. Imidlertid er der for nyligt blevet udgivet artikler der foreslår at lave realtids stereo sammenligning på GPUen. Så med den stigende efterspørgsel på billige 3D scannere og computerkraftens fremskridt, samt de nye muligheder for at lave effektiv billedbehandling på grafikkortet, er tiden inde til udforske stereo visions fulde potentiale.

Forudsat at realtids stereo information er tilgængelig, udgør dette afgangsprojekt en forundersøgelse omkring hvorvidt realtids stereo kan producere tilstrækkelig gode resultater til at skabe et online preview af scanningsprocessen. Hvis det er muligt at se den gradvis opbyggende model efterhånden som den bliver scannet, kan det fungere som en slags live planlægning af scanningen, og det har store fordele vedrørende fleksibilitet og den tid en 3D scanning tager.

Et realtids system til optagelse a stereo data er blevet opbygget, for at teste ovenstående. Den implementerede stereo algoritme er baseret på at summere forskellige vinduesstørrelser for at sikre robusthed og samtidig bevare de enkelte detaljer på objektets overflade. Den komplette 3D model fås via registrering af stereo dataene ved hjælp af ICP-algoritmen, og til slut præsenteres en simpel og hurtig metode til at sammensmelte det registrerede data, som er egnet til gradvis integration af modellen i et realtidssystem.

Problemer, begrænsninger og fordele ved et sådan system bliver diskuteret sammen med forslag og nødvendigheder i forbindelse med at bygge et fuldstændig færdigt 3D scanningssystem.
Til slut testes systemet på en række objekter med varierende form og tekstur, og de pæne resultater præsenteres.

Nøgleord: 3D modellering, 3D scanning, stereo sammenligning, registrering af 3D data, Iterative Closest Point, realtids preview.

Preface

This thesis has been prepared at the Image Analysis Section of the Department for Informatics and Mathematical Modelling, IMM, at the Technical University of Denmark, DTU, as a partial fulfilment of the requirements for acquiring the degree Master of Science in Engineering, M.Sc.Eng.

The extent of the project is equivalent to 45 ECTS point and ran over one year ending in February 2006. Additionally in this period, two courses were followed, corresponding to a sum of 10 ECTS points, and one month was spend on vacation followed by a period of one and a half month of illness.

It is assumed that the reader understands the fundamentals of image processing and has some knowledge of computer vision.

Kenneth Haugaard Møller, February 2006

[kenneth.h.moeller@gmail.com]

Acknowledgements

Many people have contributed directly to my work in this thesis in the form of discussing ideas, providing data or proof reading, but certainly also indirectly in the form of mental support. So my acknowledgements go to the following people:

First and foremost, I thank my supervisors Jens Michael Carstensen and Henrik Aanæs for contributing with ideas and moral support throughout this thesis. It has been exciting to dig deeper in the world of stereo vision and 3D modelling, which I find to be very interesting areas of image analysis.

Keld Dueholm, for providing thoughts, input and the big overview regarding stereo camera calibration.

Rune Fisker and the helpful production department at 3Shape, for providing high accuracy "ground truth" models for evaluation.

My good friend Søren Bo Hansen, for providing help with illustrations. Hopefully we can spend a little more time together in the future to come.

My buddy Kenn Tornslev, for 2½ good years on DTU, and for contributing with ideas, discussion, proof reading and constructive criticism of the work done in this thesis.

My dear family, for love and support in numerous ways. I hope to see you more often in the near future where I am not as busy as I have been lately.

Last but not least my beloved Birgitte, for educating coaching talks on motivation and discipline along with all the love and support you give me, and for proof reading the "strange" words of my thesis. I know I have neglected you very much lately, but I will make up for the lost time.

Contents

Chapter 1

Introduction to 3D Imaging Systems

In this chapter, the world and technology of 3D imaging systems is reviewed and questions like "where, by who and why is it used?", are answered.

1.1 Computer Vision

The last two-three decades of accelerating development of computers, together with the dramatic improvements in cost and performance of cameras, has spawned a new and very attractive research area called computer vision.
Computer vision covers the technology of equipping a computer with a sensing device, mostly optical, and thereby making it able to "see" the environment, extract information, interpret it and in some cases make a robot react on it.
As demands for automating trivial human working processes are ever increasing, several different business areas has seen the great potential of computer vision, and therefore puts a lot of research into this field.

1.2 The Extra Dimension

Within computer vision a separate branch has evolved into an enormous research area of it self, namely three dimensional (3D) imaging, which constitutes the technology of extracting the 3D information of a scene.
The motivation for doing research in this area is the ability to do 3D models of an object or scene, but also through 3D models to do easier segmentation of a scene.

1.2.1 2D Computer Vision Systems

Traditional and older computer vision systems are based only on the two dimensional information in an image. Having only access to this collection of pixel intensities, tasks such as tracking, measuring, recognition and classification of objects can be very difficult for computers. Common for all these tasks are that they all start out by separating objects through image segmentation. To do this successively the computer is dependent on either well defined object borders or advanced object models. In uncontrollable environments these segmentation methods can fail due to lighting, shadows or simply different colour.

Figure 1.1: A scene with background (house and lawn) and two objects (woman and child) (From [41]).

As an example, looking at Figure 1.1, the human mind has no problem segmenting the image into two persons (objects) and a background. With the brain full of prior knowledge, we can easily classify the persons as a child and a woman, and come with rough estimates of their height.
For a computer, this task, which takes the human brain a split second, would cause difficulties already in the initial segmentation process. Notice for example

the pants or left arm of the woman, which has fairly the same pixel intensity as the neighbouring background.
Even if segmentation is successful, measuring how tall the persons are can cheat the computer as they, because of the perspective image capturing process, appear to have the same height.

1.2.2 3D Imaging Systems

With 3D imaging systems, which in addition to the pixel intensity information includes depth perception of the scene, the computer is given the ability to see in three dimensions and thereby interpret the world in a more advanced way than by a mere two dimensional image.
The extra information of depth, from Figure 1.1, would look like in Figure 1.2.

Figure 1.2: The depth map of the scene from Figure 1.1 (brighter is closer and darker is further away) (From [41]).

With the 3D geometric interpretation of the scene the "objects" stand out from the background, and the task of doing segmentation and measuring the persons (if camera calibration is known), is suddenly much easier.

When doing segmentation of a scene, which is a common operation, the 3D information can provide important information, compared to a traditional 2D image where texture can cause severe problems for a segmentation algorithm, as the segmentation with respect to 3D space is totally decoupled from texture-based segmentation. Of course the texture of the scene can also cause problems for the 3D reconstruction, but that is a different problem.

1.3 Applications of 3D Imaging

3D imaging is mainly used for 3D modelling of a scene or an object. Through 3D modelling, follows a wide range of purposes, such as navigation, quality control, digital archiving and so on.
The areas, in which these application purposes are used, are very wide spread. They span from military to medicine and from archaeology to the entertainment industry.
A few examples of 3D imaging are included in this chapter to show the diversity of possible application fields.

Cultural Heritage Saving

In the academic year of 1998-99 a team, mainly from Stanford University, spent their time in Italy, scanning works of the famous Michelangelo [10],[20]. This process of digitizing cultural artefacts ensures that the maintenance of invaluable objects is kept true to the original and the possibility of reconstruction, if necessary. Also, it enables scientists, archaeologists or just commonly interested persons to inspect the work in more detail than usually possible and for an infinite period of time.

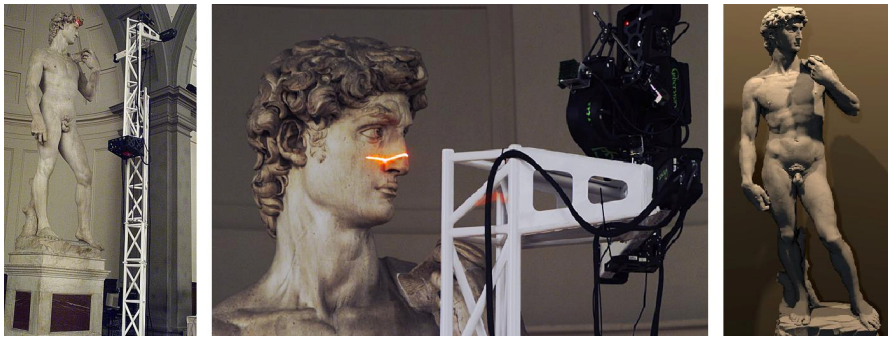

Figure 1.3: The five meter tall statue of David and the scanning setup (Left). Scanning of the face (Middle). The final CG-rendered model of David (Right) (all pictures from [10]).

Space Exploration

Already with the Mars Pathfinder expedition in 1997, NASA brought the technology of computer vision into the world of space exploration. The rover "Sojourner" was equipped with a 3D imaging system to navigate the vehicle safely through the treacherous terrain of Mars' surface.

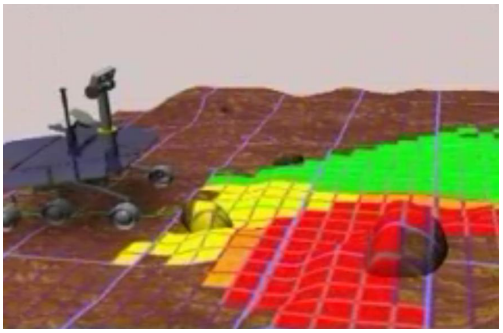

Figure 1.4: After obtaining a 3D impression of the terrain in front, the rover decides which path to follow as a function of their complexity and safety level (from [25]).

From a 3D interpretation, the surroundings are segmented into different zones categorized by their safety based on the number and size of obstacles, steepness of the terrain, etc... From this information the vehicle chooses the safest possible route to navigate through the terrain.
The rovers named "Spirit" and "Opportunity", in the more recent missions to Mars, are also equipped with a 3D imaging device for navigation.

Entertainment

In the movie industry, conceptual art design is often done in clay or other materials that are easy to work with. The models of creatures, artefacts or entire scenes are then scanned into a computer, animated, CG-rendered and composited into pictures or movie sequences.

Figure 1.5: In the Lord of the Rings, several creatures were scanned from real models and then animated in a computer (all pictures from [11]).

Also in archaeology sites or with bigger fossils, 3D measurements by scanning are often done before proceeding work, to easier recreate the different stages if necessary.
The car industry knows that the technology has great capabilities in obstacle detection and warning systems, as well as range estimation for easier parking, and such systems are slowly beginning to show up on the market.
For the military, there are huge advantages in using the technology for navigating autonomous vehicles on ground, in water or airborne reconnaissance missions.
In the biometrics world 3D imaging is used for 3D face scanning and recognition, and also in the surveillance industry it is used for security purposes.

1.4 Active vs. Passive Range Perception Methods

All 3D imaging methods have their pros and cons in certain application areas; they all need special constraints and have different limitations. Therefore it is hard to categorize the methods and say which is better.
One category that is very clear though, is that of which it is an active or passive method.

The strict definition is that if a system emits power in any frequency range in order to require the range, it is an active system, whereas the system is passive if it only needs to observe the scene, to acquire the range.

Active methods are the most developed and widely used.
For many years sonar and radar have existed as robust methods for determining range using respectively sonic and electromagnetic waves. They are however, not very accurate methods and mostly used for long range purposes.
More recent methods include Lidar which estimates range, velocity and position through analysis of the reflections of pulsed laser light.
Two more traditional methods are laser sheets and structured light. Using laser sheets is the most common. By sweeping a laser line over a scene, images are captured for specific angles, and the scene can be triangulated. Structured light is the method of projecting coded images onto a scene in order to acquire an entire scene triangulation through a single image acquisition process.
Common to the methods is that they require expensive equipment and, in the visual range cases, suffer from the active illumination, making them disadvantageous in uncontrolled or crowded environments. Also, methods of infrared structured light, for use in populated areas, have been presented.

Passive methods include structure from motion, stereo vision or multi-view vision, which respectively use one, two or more cameras to perceive range. Common to these techniques is that they require simpler and less expensive equipment. And in addition, off course, they are passive, meaning that they can be used freely in urban environments without bothering anybody.
There are however, considerable disadvantages of the passive methods. They require more photometric assumptions than active solutions and have high computational costs.

Naturally, from obvious reasons, passive methods are preferred if possible.

1.5 Commercial Systems

Most commercially available 3D imaging systems come from the active branch, as a result of the amount of research put into this specific area.

3Shape [1] and Cyberware [9] provides full automatic stand alone 3D scanners for the dental and hearing aid industry. The scanners are laser based with a rotating scaffold and can scan objects with maximum dimensions of 50*50*50 mm.

Konica Minolta produces a series of scanners suited for objects of sizes from 0.1-2 meters in diameter and a working range of 0.5-5 meters. The scanner produces a 2.5D range map from a laser sweep, so for object modelling or scene compositing the scanner or object has to be moved and the scans stitched together.

Polhemus [11] has a product called “FastSCAN” that comes in two models. They are handheld laser sheet scanners with a FASTTRAK system to determine position and orientation of the hand-scanner enabling complete 3D modelling. Its mechanically fixed cameras only tolerate a fixed working distance interval.

In passive products, TYZX [41] has developed the “Deep Sea V2”, which is a stereo head providing real time range data, with their specialized stereo processing unit.

Pointgrey [27] has several passive solutions. The “Bumblebee” and “Digiclops” are two camera stereo systems, while “Triclops” is a three camera multiview system. They provide streamed range data in different resolution and frame rate.

Common for all the commercial systems, is that they suffer from one of two drawbacks. Half the products only constitute a depth perception device, and

therefore can't really be used without additional software to align scans for modelling purposes, that is. The other half which are complete systems, for 3D modelling, are very expensive and suffers from inflexibility.

Clearly there is a need for a cheap and flexible 3D modelling system, which the average computer user or amateur sculptor can afford.

1.6 Research in Real-Time 3D Imaging Systems

Common for a lot of the commercial systems is that they require some sort of view planning to assemble complete 3D scans. The range data from different views is then coarsely registered by interactive point picking, prior to automatic precision alignment.
The alternative to view planning is a rotating scaffold which, from the mechanical calibration, provides the coarse alignment, making the method automatic. A rotation of an object, though, doesn't necessarily give enough information to constitute a complete scan.
Either way, if parts of the object are missing after the scanning process, it is first noticed at the final rendering. At this point, it is necessary to go back in the scanning process to acquire the missing views. This can be very problematic, time consuming and for some purposes not even possible, thus calling for more flexibility in the scanning process.

A way to handle this is to make the 3D acquisition real-time, and provide feedback of how much of the object that has been scanned. This, off course, sets demands concerning the 3D imaging process and high data acquisition rates, but the feedback only have to be a coarse preview of the model that is being scanned.
The real-time preview can then be evaluated online to see if parts of the object have been missed, in order for the operator to cover them.
After the scanning process the operator will be certain that the entire object have been scanned and an offline model can be rendered in high quality based on the collected data.

Lately an increasing amount of research has been put into this field of online 3D model acquisition with real-time preview.
Rusinkiewicz et al. [29] proposed a real-time system consisting of a pc-projector and a camera. Dense range acquisition was acquired by projecting time coded patterns which, assuming slow movement of the object was decoded over time and triangulated. Through fast registration and volume integration a real-time preview was provided.

Jaeggli et al. [15] proposed a somewhat similar system, with an incrementally built model preview, but with a different and adaptive scheme for the projected patterns, which is supposed to allow fast movement of the scanned object.

Unfortunately, these projects suffer in the way that they have an active depth perception device.

1.7 The Future of 3D Imaging

Looking at the commercial systems available it is clear that there is a need for cheaper and more flexible scanners that can be used for multiple purposes and object sizes. Also, to make the technology of 3D imaging available to the common man, as 2D scanning is today, several features have to be considered in the new generation of 3D imaging devices.

Low Cost

The key requirement to all applications is that it has to be cheap. This means no specialized equipment or parts with heavy power consumption. Also cheaply upgraded or replaced.

Passive Range Acquisition

Preferably the range acquisition method is non intrusive so it can be used for multiple purposes in various environments. If high precision requirements are present, though, a laser solution might be necessary.

Real-Time Preview

To avoid the slow process of view planning and reacquiring of missing parts, a real-time solution, with an incremental preview of the object, is definitely preferable.

Flexibility

To allow usability in outdoor as well as indoor environments, as well as objects of arbitrary size, a flexible solution is wanted and not some stationary custom designed built system for fixed object dimensions. Hand scanners would be advantageous as they could be moved around big objects and cover tight spaces that traditional devices wouldn't cover.

User Friendliness

To allow for example amateur sculptors, and not only engineers, access to the technology the systems have to be easy to setup, calibrate, upgrade and use.

Chapter 2

Motivation and Objectives

During the last five years, graphic cards have evolved tremendously and tasks like heavy image processing can now be implemented in hardware freeing up CPU-power for other purposes.
Yang et al. [42] have taken advantage of these advances in graphic card technology, and implemented a stereo vision algorithm in graphics hardware for real-time tele-conferring purposes.

This chapter is an outline of why these recent advances have lead to the work of this thesis and what the objectives of the project have been.

2.1 3D Modelling using Stereo Vision

The by far most often used method for 3D imaging is some sort of active lighting technique, which is expensive, inflexible and impractical in a lot of ways.
Clearly, there is a need for cheaper and more flexible 3D scanners and with the increasing power of graphic cards and their possibility of solving stereo real-time, why not try and exploit the advantages of the passive stereo method in object modeling.

With the rapidly falling prices and increasing qualities of commercial cameras, stereo vision would be the obvious choice to fill this space of need for cheapness.

Previously, the stereo method was considered too computational heavy, and therefore only suited for real-time purposes in specialized and expensive hardware. But as computers become ever more powerful and graphic cards have matured to be able to handle heavy image processing tasks, the time has come to really explore the potentials of stereo vision.

Stereo vision has several advantages over existing 3D imaging systems and meets all the demands of the future generation of 3D scanners.

- Since stereo is a passive method, it has huge advantages in the flexibility of working areas, and can also easily be configured to various working distances and sizes of objects.

- A neat thing about stereo is that it calculates range based on the original texture content of the scene. This means that the texture is given "for free", and doesn't have to be captured separately (out of sync with range acquisition) as in e.g. structured light systems.

- A stereo system has no mechanical parts, which is a huge advantage, in terms of construction, durability and cost.

- Stereo systems are relatively easy to calibrate and upgrading is straight forward. Also the stereo algorithm is easily transferred to another system of cameras e.g. infrared.

- With the advances in graphic cards and computer power it is possible to do real-time stereo and incremental model preview

- Last but not least, Moore's Law works in favour of the stereo technology, as its two biggest hurdles in the competition with active solutions, namely precision and computation tasks, are rapidly getting smaller with the continuing increases in camera resolution and processing power.

Theoretically, as stereo can be implemented real-time in graphics hardware, a system producing dense real-time 2.5D range data, could be built. As the CPU is free for other purposes, it could run a parallel real-time registration algorithm aligning the consecutive range maps into a common 3D coordinate system, and merge the data into a coarse preview of the so far built model. Since the system would be running real-time huge amounts of redundant data would be collected

allowing better estimation of outliers and regions of uncertainty, along with the online preview assisting the scanning process.
All data would be stored, and after all angles of the object have been covered, an offline high quality model would be rendered from all the collected data using global optimization models.

As far as it is known to the author, no research or tests have been done earlier, with a scanner design of this type. The individual technologies to do it are developed, but they have not been combined in such a way before.

2.2 Project Description

This thesis is about building a 3D modelling system of this type, but as a complete and fully operational system is considered to be outside the scope of what is possible with the time available, the project has to be limited in some sense.
The project is a feasibility study of combining the existing technologies, with the overall objective of trying to answer the following questions:

- Can the stereo method provide range data of sufficient quality, to be used for 3D modelling?

- Would it be possible to produce a real-time preview of the model with sufficient quality to act as an online view-planning tool, during the scan process?

To answer these questions, the project needs to be split into smaller pieces, resulting in the following partial goals:

- The construction of a stereo acquisition setup.
- Develop a stereo algorithm, suitable for implementation in graphics hardware.
- Develop a registration algorithm for aligning the range data.
- Develop a method for incrementally updating and visualization of the 3D model.
- A discussion of the future aspects of making the system fully operational and real-time.

2.3 Thesis Overview

Given the outline of the objectives, it was natural to divide the thesis into five parts:

Experimental Setup and Calibration, where the built system is presented. The calibration of the cameras is described along with a discussion of the variables to be considered, in order to capture high-quality stereo data.

Depth Perception via Stereo Vision, is the section concerning the basic theory and issues to consider in stereo vision, along with details of the implemented algorithm.

Registration via ICP, concerns the technique of aligning the stereo range data into a common 3D model suitable for real-time preview. The implemented algorithm is presented along with visualization possibilities.

Experimental Results, evaluates the system. The complete 3D modelling system is tested on different objects.

Discussion, propositions and needs for future work, and a conclusion is given in this last part.

2.4 Project Overview

To ease the comprehension of what this thesis is about, a schematic version of the intended system is shown in Figure 2.1.

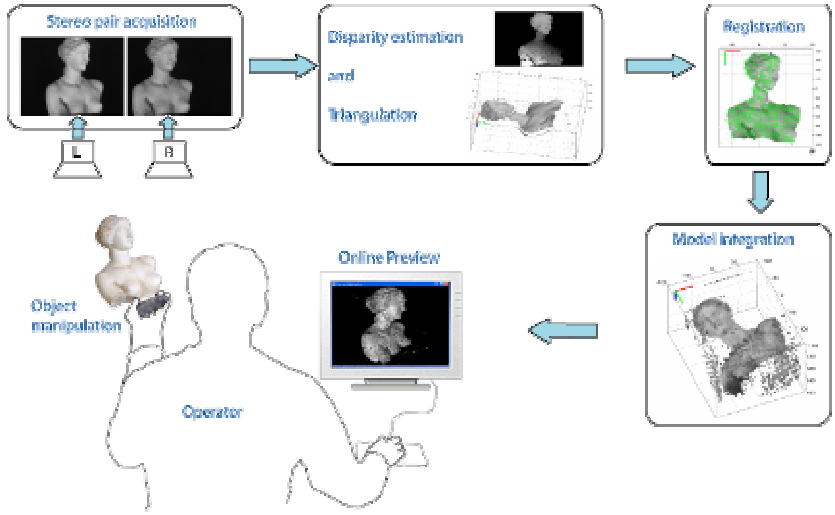

Figure 2.1: Diagram showing the different modules of the system constituting the feedback loop of online model previewing.

The system operator rotates and translates the object under the two cameras, with one hand, while the image pair sequence is captured.
With the eventual implementation of a real-time system, a preview of the incrementally built 3D model can be viewed online on the monitor and oriented with the mouse to evaluate what parts of the object are missing.
Constituting a full feedback loop, view planning is done online and the scanning process is easily completed.

2.5 Terminology

The words, range map, range image, depth map, depth image, height map are used interchangeably dependent on the specific application, but all refer to an explicit function (2.5D), which denotes a uniform sampled x-y grid with individual function values f(x,y).

Part I

Experimental Setup and Calibration

Chapter 3

System Design

This chapter presents the system, as it has been built, with a description of the individual parts and a discussion of important parameters. The system, which is seen in Figure 3.1, makes it possible to capture live stereo image sequences and thereby test the stereo- and ICP algorithm.

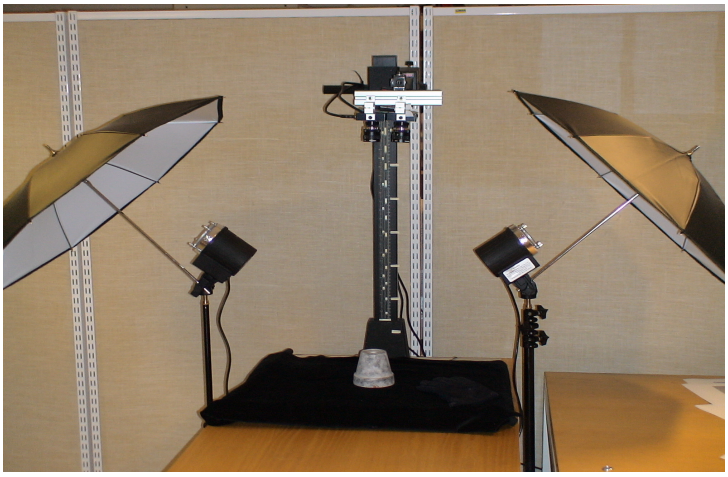

Figure 3.1: The 3D modelling setup.

3.1 Cameras

The system consists of two digital IEEE-1394 DragonFly™ cameras from Point Grey Research [27], connected to the pc through an IEEE-1394 PCI-card. As the data flow from the cameras are 100% digital, there is no resampling of the image data, which would introduce further noise into the images.
The CCD, of the DragonFly™ model, is 8 bit gray scale, has a resolution of 1024x768 pixels, and can stream this quality to the pc without compression at a frame rate of 15 fps. For further information, consult the technical reference at the Point Grey website [27].

The cameras are positioned relatively close to each other, approx. 70 mm apart, in a fronto parallel alignment. This is to prevent perspective distortion having too much influence in the correspondence search and obtain a big effective viewing frustum.
As the working distance, for object modelling, in this setup is set to be around 0.25-0.5 m for objects of 50-200 mm in diameter, this gives a B/D relationship of 7/25 - 7/50 ~ 1/3 – 1/7.

Currently the cameras are equipped with a pair of C-mount lenses from PENTAX. The lenses have a focal length *f* of 8.5 mm.

3.2 Field of View and Frustrum Resolution

These lenses combined with the 1/3 inch CCD's give the following field of view properties:

Field of view properties	
Diagonal	~50°
Horizontal	~31°

The CCD's unit pixel size is 4.65*4.65 μm. From the cameras relative position and camera constants, the lateral and range resolution of the system is calculated and shown in Figure 3.2.

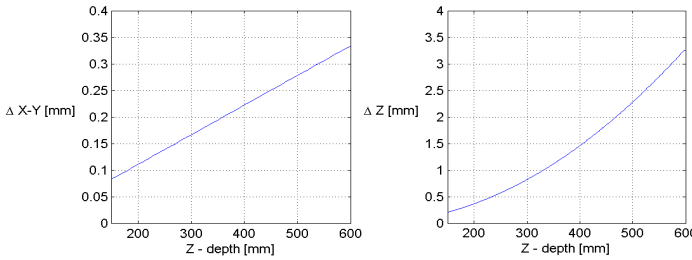

Figure 3.2: The lateral resolution (left) and the range resolution (right), both as a function of depth.

It is seen that the lateral resolution increases linearly, whereas the increase in depth resolution is quadratic as a function of the working distance. It, is also noticeable that the lateral resolution is a magnitude of a factor 10 bigger than the depth resolution.

3.3 Lighting Conditions and Camera Settings

As the stereo vision concept is based on the textural content of the scene, lighting conditions play an important role in achieving high quality 3D reconstruction.
First of all, the stereo matching is easier solved if corresponding points also have similar intensities. Second of all, if the registration module uses texture, the intensities of temporal correspondences also must be similar in the consecutive images.
These two constraints call for sufficient illumination in order to capture the true intensity and not a shaded version of the texture. As seen in Figure 3.1 of the setup, this is achieved by two bright lamps with umbrellas causing diffusion.

It is assumed that all scanned objects have surfaces with more or less uniform reflectance properties. Thus, plenty of diffuse illumination will result in the object appearing in its true colours when captured from different angles.

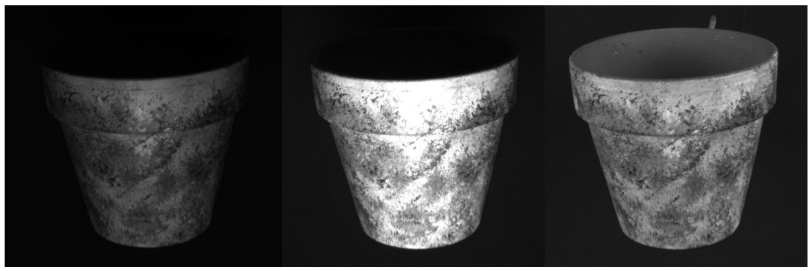

Figure 3.3: Three different lighting conditions of a pot: Shaded (left), saturated (middle) and reasonably lit (right).

In Figure 3.3 three examples of the captured object texture is seen. If only a single illumination source is present, the surface is shaded and the texture appears to be darker on the sides of the pot (left). Using a more powerful light source doesn't solve the problem, but only causes saturation and more unbalanced texture intensities (middle). Using several diffuse illumination sources gives a true textured image of the pot (right).

Also it is assumed that the surfaces of the objects don't have specular reflection properties.

Figure 3.4: A specular surfaced object, captured under direct (left) and diffuse (right) illumination.

Ideally, diffuse lighting conditions should handle specular surfaces in a reasonable way. But as seen in Figure 3.4 (right) the three diffuse light sources still pose a problem as they are "visible" in the captured image, showing themselves as three vertical bright lines on the round edge of the object. Compared to an image captured with a direct light source (right), the diffuse result is rather good though.

As lighting conditions may be different for each capturing session, settings in the camera can be adjusted to cope with these variations. The settings that can be varied include: Integration time (shutter time), gain, bias, and exposure. All settings are equal for both cameras and can be varied in the image acquisition software. Automatic adjustment has not been used.

The typical lighting conditions in laboratories are neon tubes, so the shutter time must be a multiple of 10 ms., otherwise pixel intensities would vary temporally in the captured images as a consequence of not integrating over a full period of the fluctuations.
As default, the integration time is set to the minimum of 10 ms., which from experiments was found to be small enough to avoid any significant blur when an object is moved during live capture, and long enough to visually give a reasonable signal to noise ratio.

Even though the cameras have equal settings, this doesn't mean that the amount of light captured is equal. This is commented further in the single camera calibration chapter.

The strong diffuse spots (2*300W) are very dominating light sources, making the illumination conditions constant. Thus, shadows from human activity or different external illumination as a function of the time of day become insignificant.

3.4 Image Acquisition Software

For acquisition of stereo image pairs a capture program has been developed from the PGR SDK-library in C++ using OpenGL for visualization. The program controls initialization and synchronization of the cameras and off course also the image pair capturing.

The program delivers live images from the cameras together with a zoomed version defined by the user.
All camera settings can be adjusted freely when running live except from the integration time that can only be adjusted in intervals of 10 ms. due to the eventual presence of neon tubes.
When it comes to the actual image capturing, the software has several functionalities:

1. Single image state. Pressing a key captures an image pair which is saved in bmp-format.

2. Aperture/focus calibration state. Image pairs are captured automatically at 2 fps. and stored in bmp-format (overwritten) in a specific calibration folder. This state was made for external calibration purposes.

3. Live streaming state. Images are captured at 7.5 fps and written to a "raw"-file. When live capturing is stopped, the "raw"-files are disassembled into the individual images in bmp-format.

All images from the cameras have time stamps, so all pairs are checked if they were synchronously captured. If not, they are discarded.
The images also have a sequence number, so it is possible to notice if an image was skipped due to buffer overrun.

Before capturing images for object modelling the camera settings should be adjusted to fit the object and the lighting conditions, so the full dynamic range of the cameras are used and the CCD doesn't saturate.

3.5 Uniform Coloured Background

As the purpose of the project is object modelling, it is reasonable to assume that the object colour is known. Therefore, images are captured against a uniformly coloured background to easily segment out the object through a simple thresholding of the images.
Currently, as mostly bright toned objects are scanned, a black velvet cloth is used. To avoid problems with the operators hand (a non-rigid object) holding the artefact of interest, this of course also must be covered with a glove in the same colour as the background. A pair of tongs or a wrench of some sort can also be used.

3.6 Summary and Discussion

The designed stereo acquisition setup has been presented and the environmental variables such as lighting and background were discussed. The image capturing software has also been outlined along with the variable camera settings.

The B/D relationship of 1/3 – 1/7 of the cameras is rather ill-conditioned. This means that the two projections of a world point intersect at a small angle, and

therefore the depth estimation is coarse and has a big uncertainty attached to it, with this setup.
Choosing wider angled lenses would give a better B/D relationship, on the cost of spatial resolution at a given distance. But looking at the resolutions as they are for this setup, both laterally and in the depth direction, this relationship is a little oblique in favour of the lateral resolution, so wider angled lenses properly would strengthen the relationship and level out the resolution difference in the system. A software solution has been chosen, in stead, to strengthen the depth resolution. This will be discussed in the stereo implementation chapter.
A solution of converging cameras could also have been chosen, but due to the perspective distortion issues, geometrical practicalities and for the sake of simplicity, the parallel axes solution was preferred.

The challenge of creating diffuse illumination could maybe be solved in a better way with for example a light tent.

Concerning both the performance of the stereo algorithm but also for reasons of segmenting out the object from the background, colour cameras, would give a major advantage and much more information. Any background colour could be used as long as it is not in the object, resembling the blue screen technique known from special effect movies and the weather on TV.

Chapter 4

Single Camera Calibration

This chapter describes how the individual cameras are calibrated, what is achieved by the calibration and why it is important.

4.1 The External Parameters

To begin with the external parameters such as aperture stop, focus and coarse directional alignment are handled.

4.1.1 Aperture Stop

As the algorithm depends highly on texture, preserving the high frequency content of the images is necessary. Therefore, to avoid too much blurring, an as high aperture stop k as possible is desired, meaning an as small diameter of the aperture as possible. On the other hand, letting less light come through to the CCD, will decrease the Signal to Noise ratio, which is not wanted.

Based on a given working distance, 0.5 m., and the formula of the *circle of confusion*, the amount of blurring can be seen as a function of the aperture stop and object distance, see Figure 4.1.

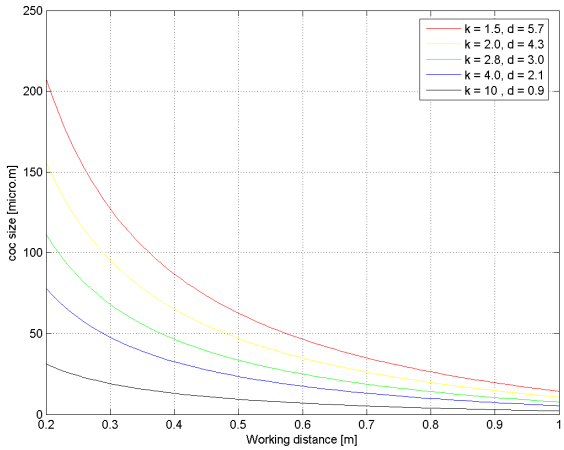

Figure 4.1: Blurring of a point on the CCD as a function of working distance and the aperture stop.

To help setting the aperture stop, a calibration board was produced and a matlab program written. The image acquisition software is set to run in aperture/focus calibration state with the calibration board placed in the typical working distance from the cameras and within both fields of view. The matlab program loads the captured images continuously and after the user have marked two regions and profiles the plot of Figure 4.2 is shown and updated continuously.

From the plot it is possible to check the dynamic range of the cameras via the histograms and evaluate if the high frequency content is blurred in the image profiles.

With the programs running the gain of the cameras are set to the minimum possible of approximately 2dB. Then the aperture stop of the left camera is adjusted to the maximum value where the different intensities of the histogram still are distributed nicely over the entire dynamic range, see Figure 4.2 (middle).
An aperture stop of approx. 9-10 was found to be suitable.
Checking with the graph of circle of confusion, it is seen that in a working distance of 0.3-0.5 m., the blurring ranges from approximately 10-20 μm. roughly corresponding to 2-4 pixels on the CCD, which has a unit pixel size is 4.65 * 4.65 μm.

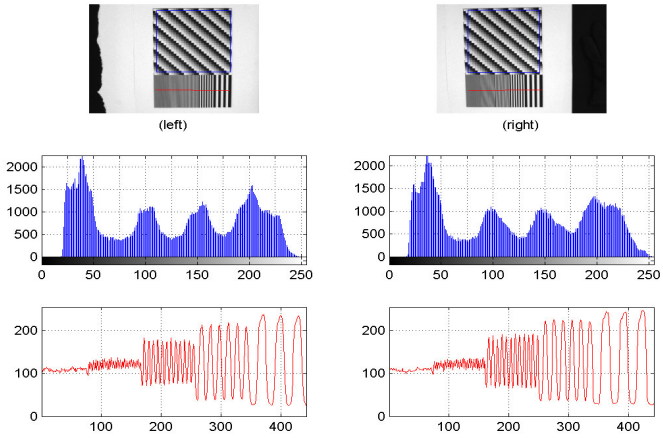

Figure 4.2: The matlab program to assist setting the aperture. Two images of the calibration board (top). Histograms of the intensity in the blue square (middle). Profile view of the red lines (bottom).

After adjustment of the left camera, the aperture stop of the right camera is adjusted so the intensity histogram approximately matches the left camera.

4.1.2 Focus

After having adjusted the aperture stop, and with the matlab calibration program still running, the camera constant c can be adjusted by focusing the lenses.
The profile plots of the calibration board from Figure 4.2 (bottom) can be used to find the setting giving the sharpest edges of the low frequencies and just higher peaks of higher frequencies. Also the zoom function of the image acquisition software can be used to judge the focus by the sharpness of the edges of either the calibration board or the checker board.

4.1.3 Directional Alignment

After the cameras have been focused, they have to be coarsely aligned.
The cameras or lenses are not perfectly built, so their optical axes are not perfectly perpendicular with the print boards or their mechanical mounts, which when building the setup is placed in parallel position. Therefore a rough coarse alignment satisfying the epipolar constraints by eye sight is done to get a more effec-

tive image overlap in the resulting stereo images. This is done with the image acquisition software.

4.2 The Internal Parameters

Following the external parameters, the internal can be calibrated, one camera at a time.

Jean-Yves Bouguet, from California Institute of Technology, has developed the "Camera Calibration Toolbox for Matlab" available at [5]. The toolbox can handle both calibration of single cameras and stereo setups, and is roughly based on papers from Zhang [45], Heikkilä [12] and Tsai [39].

The toolbox assumes the pinhole camera model, with the common parameters of camera constant, principal point, skew and lens distortion, and uses a planar checker board with constant sized squares, for determining the camera parameters.
The calibration procedure is outlined coarsely, while a more detailed description is available on the homepage [5].

To retrieve the internal camera parameters, images are taken of the checkerboard in different poses so as much of the viewing frustum is covered as possible. As the individual calibrations are preparations for the stereo calibration, the checker board poses must be captured by both cameras, synchronously, with the entire checker board in the viewing field of both cameras. The images, captured by the left camera, are seen in Figure 4.3.

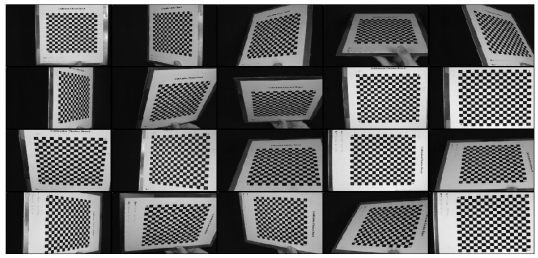

Figure 4.3: The 20 checker board calibration images from the left camera.

For each image, the four corners are annotated in the same order each time and by interpolation the rest of the checker corners are automatically predicted, see Figure 4.4.

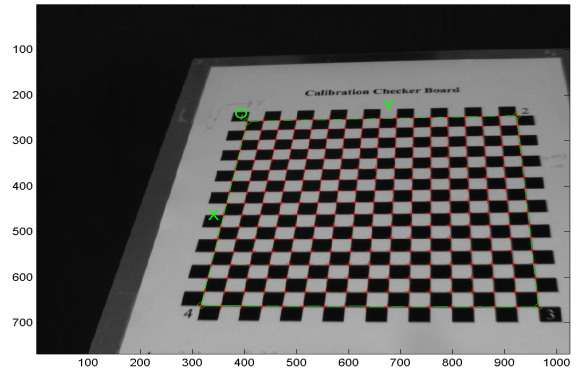

Figure 4.4: Annotated corners (green circles) and the predicted checker corners (red crosses).

In each image, an automatic corner optimization is done with the predicted positions as start guesses. The resulting positions are seen in Figure 4.5.

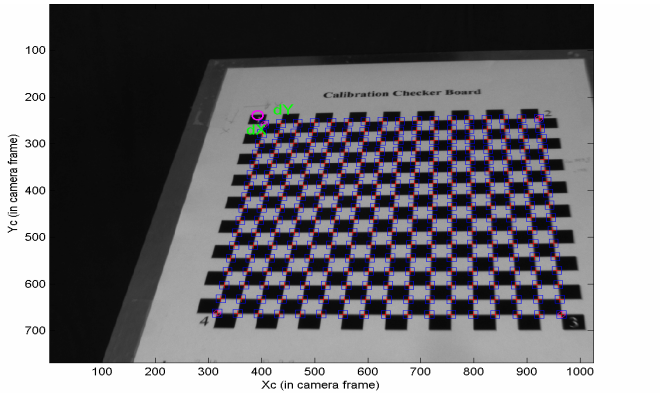

Figure 4.5: The optimized positions of the checker board corners (blue squares).

As all images have been annotated, the camera now possesses projections of thousands of points in the viewing frustum on which the calibration is based.

From an initial guess of the checker board poses and camera model parameters, a gradient descent-based optimization of the residual back projection errors is initialized which results in the following calibration data of the left camera:

```
Calibration results after optimization (with uncertainties):
Focal Length:      fc = [ 1850.66833   1851.71448 ] ± [ 2.22771   2.25858 ]
Principal point:   cc = [ 512.71470   292.53701 ] ± [ 3.96466   3.95634 ]
Skew:           alpha_c = [ 0.0 ] ± [ 0.0 ] => angle of pixel axes = 90.0 ± 0.0
degrees
Distortion:        kc = [ -0.24051   0.27434   0.00068   0.00152  0.00000 ]
                      ± [ 0.00944   0.08902   0.00044   0.00047  0.00000 ]
Pixel error:      err = [ 0.29114   0.29898 ]
```

The camera constant $c=f_c$ (called Focal Length in this case) is given in pixels and with two numbers as the pixels of the CCD have slightly different dimensions in x and y-direction. The uncertainty of roughly 2 pixels is insignificant as it is only a pro mille and means a microscopic scaling of the measured data.
As far as the principal point concern it is approximately 100 pixels off the image centre in vertical direction. The uncertainty of a little below 4 pixels is a little more than what to expect from a system like this.
There is no skew in the CCD, which also wasn't expected.
Looking at the lens distortion parameters k_c, the two first and the fifth coefficients are the 2nd, 4th and 6th order radial components, while the third and the fourth coefficients are tangential components. The radial component of the distortion model is visualized in Figure 4.6.

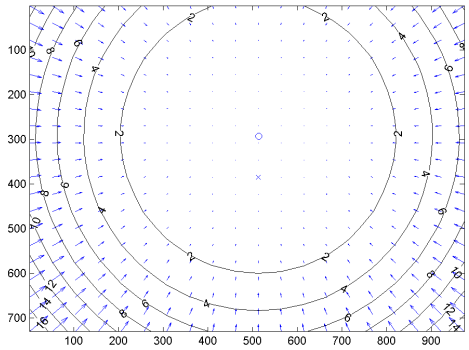

Figure 4.6: The radial component of the distortion model.

It is clear that the center of the distortion doesn't match the image center, as indicated by the principal point. In addition, it is seen that pixels are increasingly misplaced away from the center, up to 14-16 pixels in the extreme corners.

The tangential component of the distortion model is depicted in Figure 4.7.

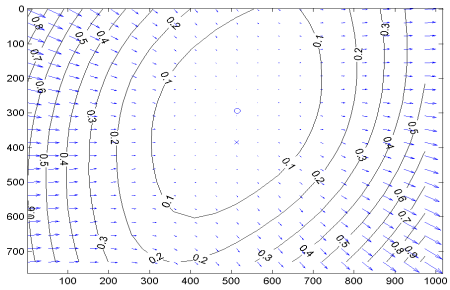

Figure 4.7: Tangential component of the distortion model.

In the tangential component, the displacement is not very bad. At all points it is under a tenth of the radial component, so it is not weird that the complete distortion model in Figure 4.8 looks mostly like the radial component.

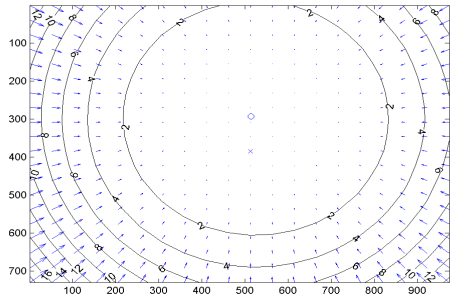

Figure 4.8: Complete distortion model.

Finally, the pixel error of the calibration is a measure of how well the calibration is. From earlier experiences and comparisons from papers a value of 0.29 is evaluated to be quite good as the calibration is part of a stereo calibration and therefore have not utilized the full viewing field.

As the calibration is finished the parameters are stored and the process is repeated with the images of the right camera.

4.3 Lens Distortion Compensation

To use the images for measuring depth with the stereo vision method they need to be corrected from the lens distortion.

This is done via a spatial output-to-input transformation of the image pixels based on the complete distortion model.

As the distortion can be up to several pixels it is very important to correct, otherwise the disparity estimation would be very erroneous resulting in bad 3D reconstruction.

4.4 Summary and Discussion

The method of calibrating each camera and thereby retrieving the internal parameters has been described. Also the cameras variables of aperture stop, focus and directional alignment have been presented and discussed.

To get a good model for the lens distortion, it is preferable that the checker board covers the entire viewing frustum of the camera. This is not possible though, with the current procedure, as the board also has to be visible in the other image, because of the later stereo calibration. This properly degrades the calibration compared to a normal single camera calibration.
Perhaps this process could be improved by first doing a full view calibration of each camera to determine the internal parameters, and then do an entirely different calibration of the stereo setup.

Chapter 5

Stereo Calibration

The Camera Calibration Toolbox for Matlab also has stereo calibration features. These are described briefly in this chapter along with the concept of epipolar geometry and image rectification, which are important topics of stereo vision.

5.1 Calibrating a Stereo System

From the toolbox, the stereo calibration module is initialized and the individual parameters of the two cameras are loaded (the previously calculated). Based on the calculated poses of the checker board for each camera, their relative position in space is plotted in Figure 5.1.

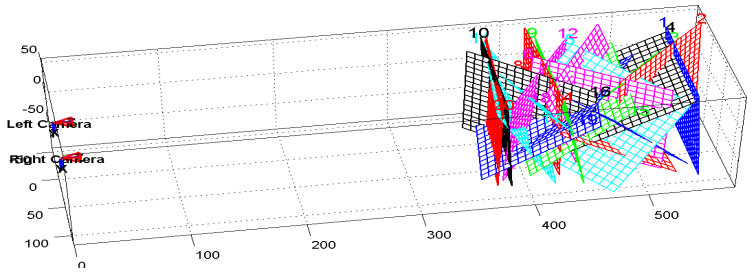

Figure 5.1: The stereo setup and the checker board poses.

The relative orientation of the stereo pair is given as:

```
Extrinsic parameters (position of right camera wrt left camera):
Rotation vector:             om = [ 0.02961   0.01467  -0.00624 ]
Translation vector:           T = [ -61.51409   -1.71693  0.84990 ]
```

With the projections of the checker board in both cameras, a combined global optimization is performed of both the relative orientation of the cameras, and the respective internal parameters. The results are as follows:

```
Intrinsic parameters of left camera:
Focal Length:       fc_left = [ 1848.06417   1847.27467 ] ± [ 1.72568   1.73341 ]
Principal point:    cc_left = [ 518.72535   296.06465 ] ± [ 4.11912   3.99732 ]
Skew:           alpha_c_left = [ 0.0 ] ± [ 0.0 ] => angle of pixel axes = 90.0±0.0
degrees
Distortion:         kc_left = [ -0.23883   0.29429   0.00019   0.00126  0.00000 ]
                            ± [ 0.01008   0.09978   0.00042   0.00044  0.00000 ]

Intrinsic parameters of right camera:
Focal Length:      fc_right = [ 1849.14560   1847.09691 ] ± [ 1.72646   1.73059 ]
Principal point:   cc_right = [ 518.34145   355.70242 ] ± [ 4.18237   3.74346 ]
Skew:          alpha_c_right = [ 0.0 ] ± [ 0.0 ] => angle of pixel axes = 90.0±0.0
degrees
Distortion:        kc_right = [ -0.20147   -0.01924   -0.00052   0.00100  0.00000 ]
                            ± [ 0.01211    0.12364    0.00036   0.00056  0.00000 ]

Extrinsic parameters (position of right camera wrt left camera):
Rotation vector:     om = [   0.0284   0.0157  -0.0061 ] ± [ 0.0028 0.0030 0.0001 ]
Translation vector:   T = [ -61.1623  -1.6435   1.1750 ] ± [ 0.0511 0.0445 0.3946 ]
```

For the left camera, all the parameters have changed a little. It is noticeable though, that the uncertainty of the camera constant has decreased, while it has increased for the principal point. For the right camera, compared with the initial parameter values, these trends are the same.

5.2 Epipolar Geometry

The main reason for going trough all these calibration stages is to estimate the epipolar geometry in the setup. The usefulness comes through the geometry of the setup, where a point in the left image x_l has its correspondence x_r positioned on a line in the other image. It is said that the corresponding point satisfies the epipolar constraint.

$$x_l^T \cdot F \cdot x_r = 0 \qquad (5.1)$$

where F is the fundamental matrix of the system, which can be derived directly from the relative orientation of the cameras.

The epipolar geometry is very useful when it is desired to find the correspondences in stereo images, as the task is reduced to a one dimensional search, as opposed to consist of the entire two dimensional image.
Correspondence search along these lines is very complex though, but can be efficiently simplified if the epipolar lines are altered to coincide with the horizontal scan-lines of the images. Thereby the correspondence search can be done for each disparity hypothesis, simply by shifting the entire target image before comparing with the reference image.

5.3 Image Rectification

Rectification is a warping of image pairs, making the epipolar lines coincide with the scan-lines in the images, and is based on the knowledge of the fundamental matrix.
Stereo pairs are rectified to make the correspondence search one dimensional along the horizontal scan-lines and thereby more efficient.
In a real-time system this rectification is then done on each incoming stereo pair. The warping function is only calculated once though, and then just applied to each image pair.

Figure 5.2 shows a non-rectified image pair with the same two scan lines marked with red in both images.

Figure 5.2: Non-rectified image pair with two pairs of corresponding scan-lines.

It is seen, that in the upper left section in "Blach's Turisttrafik", the second scan line does not go through the same points of the logo. Also, above the bus, the horizontal background lines are rotated differently with respect to the first scan line.
The effect is not as obvious as it could be, but this is due to the coarse alignment of the cameras, described in the calibration chapter. Without this initial viewing field alignment, the images can be severely rotated or translated relatively to each other, even though the camera casings where positioned in perfect parallel alignment.

Together with epipolar geometry, the rectification also adjusts the images for lens distortion, both radial and tangential, and is achieved by applying an affine transformation to the images.

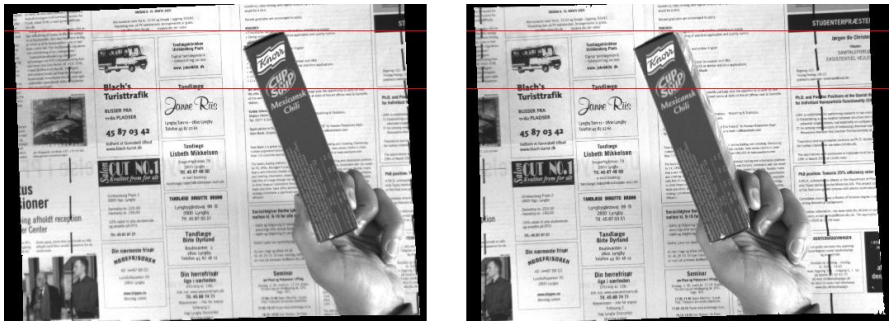

Figure 5.3: Rectified image pair with two pairs of corresponding scan-lines.

The result of the rectification is two images in which corresponding scan-lines pass the exact same points, if no occlusions are present, that is.
As the images are warped into alignment, some extra areas (not from the scene) are added, which shows as triangular regions at the image borders. As the background used for object modelling is black the areas are also set to black.

5.4 Summary

The results from the stereo camera calibration have been presented, along with the concept of epipolar geometry and its advantages, which is the motivation for doing the stereo calibration.

Part II

Depth Perception via Stereo Vision

Chapter 6

Introduction to Stereo Vision

Stereo vision has always fascinated and been interesting due to its passivity in obtaining depth information and its striking similarity to human vision.
During the last decade lots of research has evolved it into a well understood and robust computer vision method, which definitely has advantages over active methods in several areas. To understand the basic principles of stereo vision, a short review is given of how the human eyesight works.

6.1 The Human Visual System

Humans, as a lot of other animals (predators), have their eyes placed in the front of the head, which is called binocular vision. This makes the two views, obtained from the eyes, overlap and creates what is called a binocular field, which enables us to estimate range. This stands in opposition to the group of animals which typically have their eyes place on each side of the head, creating a 360 degrees field of view with very small binocular field (overlap), and therefore don't have the ability of perceiving range in the same way.

As our eyes are placed approximately 5 cm. apart, each eye captures its own view of the world. When the views are projected on to the back of the retina, the actual three dimensional information of the world is "lost" and reduced to parallax shifts in a couple of two dimensional images. The closer an object is positioned to the person the bigger is the parallax of this object in the two different views.

The two slightly different images of the world are transmitted to the brain and processed in the primary visual cortex. Here the parallax shifts in the images combined with prior knowledge of the three dimensional world are united into the three dimensional view we see with both our eyes open.

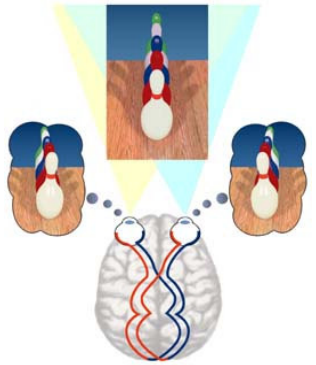

Figure 6.1: The human visual system combines two slightly different images into a 3D perception of the scene (From [26]).

An easy way of demonstrating the parallax shift is by holding a finger in front of your face, and looking at it with alternately the left and right eye closed. When placing the finger close to the nose the shift from left to right view is very large, but moving the finger as far away as possible, the shift becomes smaller and smaller.

Without binocular vision, i.e. using only one eye, we can still determine depth to some extend. This is because the brain has a lot of prior knowledge of our world (it has been trained for years), and can make use of perspective, shading, motion, size and so on to determine approximate depth or range to objects.
Still we need the binocular vision when catching or reaching for something, driving, pouring water, threading a needle, etc. Try for example to hold a pencil in each hand (or use the index fingers), stretch the arms and with one eye open make the pointed ends of the pencils meet. This is hard, but with two eyes it is no problem.

The process used by the human visual system to achieve this stereoscopic fusion, however, is not well understood. So to gain insight about the operation of the human visual system and to be able to produce autonomous systems that are able to passively perceive depth, a lot of research has been put into this field of computer vision.

6.2 A Mathematical Model of Binocular Vision

Simulating binocular vision on computers is generally referred to as stereo vision, and as the human visual system is a very complex system (and far from fully understood) several simplifications have to be made.

First of all, the cameras used have a lower resolution than the eyes. This is both due to the actual resolution of the cameras available today, but also to limit the computational tasks involved in stereo vision.
Next the cameras are usually placed in a fixed position relative to each other. This ensures that the two cameras can be inter-calibrated, which results in a huge reduction in the computational tasks (for details see the stereo calibration chapter).

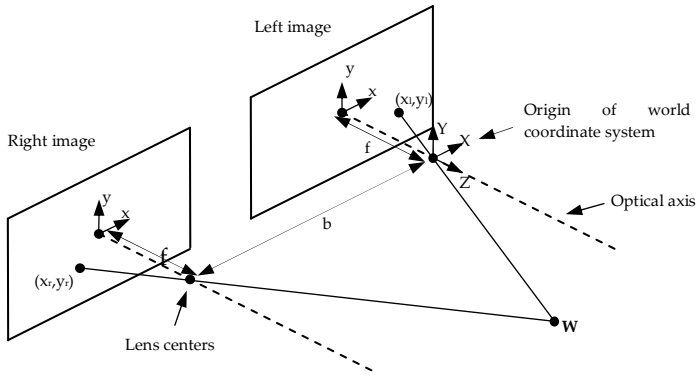

Figure 6.2: The geometric relationship in fronto parallel stereo vision.

To simplify the geometric understanding, the cameras are typically placed in a fronto parallel alignment, see Figure 6.2. This means that the image sensors lie in a common plane (their optical axis are parallel) with a distance b between the focal points (baseline) and the reference coordinate systems of the two image planes are rotationally aligned.

In the left and right image we have the projections (x_l,y_l) and (x_r,y_r), respectively, of the same physical world point W:

$$W = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (6.1)$$

Considering the geometry, it is possible to calculate the position of *W* with respect to the world coordinate system with reference in left camera's focal point:

$$W = \frac{b}{d} \begin{bmatrix} x_l \\ y_l \\ f \end{bmatrix} \quad (6.2)$$

where *f* denotes the camera constant of the cameras, *b* the baseline, and $d = x_l - x_r$ is the parallax shift, in the x-direction, of the point W. The derivations of (6.2) can be found in appendix A.
It is seen, that besides the constants *f* and *b*, the only thing needed to calculate the three dimensional position of a projected point (x_l, y_l) in the left image, is the point itself and its parallax shift in the right image, namely *d*.
This *d* is referred to as the disparity of a point, and the according values of *d* to an entire image of a stereo pair, is called disparity image or disparity map.

So the depth and position of a world point can be solved, but to do this it is required to know the two corresponding projection positions of *W*. Solving this is called the stereo correspondence problem, stereo matching or simply the correspondence problem.

6.3 The Correspondence Problem

As the two views of a scene can be different in several ways, solving the correspondence problem can be very difficult. In addition, it is exhaustive to search through all the possible disparities for each image element, making the problem computational heavy. This is also the reason why stereo has not been more widely used in commercial applications.
None the less, the correspondence problem can be solved in various ways as described in the next chapter.

Chapter 7

General Stereo Considerations

In this chapter the complications in solving the correspondence problem, assumptions and limitations are discussed. Also the general theory for area-based solutions is described and some related work is reviewed.

7.1 Problems in Stereo Correspondence

Several problems occur when solving the stereo correspondence problem, as the appearance of corresponding points will differ in the two images. These problems can come from either the nature of the scene or object, the lighting conditions, imaging process or other factors.

7.1.1 Lack of Texture

As stereo vision is a 100% passive method of 3D imaging, the presence of textural features are essential in order to estimate the horizontal parallaxes and thereby calculate distance. If no texture is present in an area, the stereo matching will most likely fail and result in erroneous areas in the disparity map.

7.1.2 Repetitive Texture

When a texture is repeated, like in a grass field or a brick wall, several matches can be found to match the reference. This ambiguity often leads to wrong correspondences.

7.1.3 Occlusions

In real world scenes the presence of several objects or just one with a complex shape, will lead to occlusions when viewed from certain angles. These occlusions cause depth discontinuities and since one camera can see something the other one can't, the matching is almost condemned to fail.
Also, depth discontinuities often act as a kind of "powerful" texture, as high frequency textural content is created along the discontinuity edge. Dependent on the textural content, the area around such an edge often contain erroneous matches.

7.1.4 Perspective Distortion

As a function of the displacement b of the cameras, the projection of the scene will differ in the two images, causing features in the scene to appear slightly different. As b increases, this phenomenon, called perspective distortion, gets worse and the matching of features more difficult to solve.

7.1.5 Photometric Variation

In the image formation process, a pixels intensity (or colour), coming from a projected world point, is influenced by several things. The material of the object surface can have non-uniform reflectance properties, making the pixel intensity of the projected world point depend highly on the angle at which it is gazed with respect to the light sources. As all points useable in stereo vision is gazed at from two different angles, this can cause problems as the same point projected in the two images might not have the same intensity.

7.1.6 Lighting Conditions

To acquire good signal to noise ratios in the images, sufficient and lots of illumination is a necessity. However, spotlights can cause highlights and saturation of the images dependent on the surface reflectance properties, which is not wanted.

7.2 Assumptions, Constraints and Limitations

It is impossible with the given time limits to create a system that can handle every imaginable object or scene. Also, the choice of using stereo vision introduces some problems as described in the previous chapter.
As a consequence the project has to be limited in the sense of what objects it can handle and under which conditions. To stay within these limitations, assumptions have to be made and constraints need to be set.

First of all, as the system is based on stereo vision as the range perception device, objects with textured surfaces are assumed. The texture has to be relatively structured and not to finely grained. Also, the texture patterns cannot be to repetitive as it can result in ambiguous stereo matches.

Assumptions regarding the topology of objects also has to be considered, as occlusions and depth discontinuities are subject for erroneous stereo matches. As the stereo algorithm needs a support region to be robust, the algorithm itself introduces a smoothness constraint of the object surface. The order of the smoothness assumed is dependent on how big a support region is used.
So objects with smooth surfaces and preferably no discontinuities are assumed, as the stereo algorithm will introduce some blurring in the reconstructed surface.

To cope with problems of photometric variation issues, it is assumed that only objects with diffuse (preferably lambertian) surfaces are used with no specular material properties.

In addition, as the project is a stationary setup, assumptions concerning working distance, object movement and lightning conditions can also be made.

Last but not least, to limit the computational task and to simplify the complexity of the disparity search patterns, the stereo images are assumed to satisfy the epipolar constraints as described in the stereo calibration chapter.

7.3 Considerations Toward Registration

The purpose of the stereo matching is to produce 2.5D surfaces which through registration will constitute the 3D modelling. Therefore the surfaces need to have the sufficient properties needed for such an alignment. Even though registration

is later in the modelling pipeline, it has to be considered when choosing a solution for the correspondence problem.

To get high quality registration, smooth surfaces are needed which calls for dense disparity maps. Small errors in the disparity map (which are unavoidable) are acceptable as it is the idea that the same area of an object will be covered multiple times, thereby creating lots of redundant data available for error reduction. Of course, if the errors become too large they ruin the registration, which is unacceptable.
Single outliers are not paid special attention to, as these are easy to cope with in the registration.
Along with the smoothness constraint of the surfaces to be registered, the surfaces still need to have some distinct features in order for the ICP algorithm to perform well. So this introduces constraints on the topology of objects that can be successfully modelled, but also calls for a stereo reconstruction that does not blur the surfaces in a way, so the distinct features disappear.

In the registration part of the thesis, the above is explained in more detail.

7.4 Area- vs. Feature-based Methods

Method for solving the correspondence problem can roughly be divided in two categories, namely area- and feature-based methods.
Area-based methods are low-level as the matching is done with primitives like pixel intensities. If texture is present and the surfaces vary smoothly they create dense disparity maps.
The feature-based work on more abstract features, as corners edges etc… They can provide more precise disparities, but because of the sparse and usually irregular distribution of features in an image, some extra interpolation step is needed if dense disparity maps are required. Also the feature detection in the two images increases the computational costs significantly.

In the context of this project where dense disparity maps is a criterion, only the area-based methods were considered.

7.5 The Disparity Space Image

When solving the stereo matching problem, one of the stereo images has to be chosen as reference and the corresponding disparity image d will coincide with

the reference image I_{ref}. Then, ideally, the corresponding target image I_{tar} is given by

$$I_{tar}(x, y) = I_{ref}(x + s \cdot d(x, y), y) \quad (7.1)$$

Depending on whether the left or right image is reference, s is a sign factor of ±1, so the disparity values always are positive.
The goal is then to find the values of d, so equation (7.1) holds.
Solving the stereo correspondence problem in an area-based fashion is a process of three tasks.
In the following chapters it is assumed that the input images, I_L and I_R, are rectified (they uphold the epipolar constraints) and the parallax shift is along the x-axis in the images.

7.5.1 Matching Cost Computation

To do the actual stereo matching, that is comparing the pixels of the two images, calls for some kind of matching cost $C_d(x,y)$. For a given d the pixels in the reference image is compared to the corresponding pixels in the target image with respect to d, by some matching cost measure.
Several such measures have been proposed, the most normal being cross correlation (CC), squared differences (SD) and absolute differences (AD).
The two latter are dissimilarity measures as they operate with differences and a small difference denotes small dissimilarity. CC is a similarity measure as high correlation factor means high similarity.

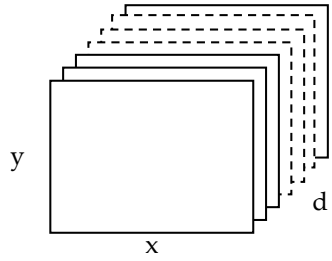

Figure 7.1: The disparity space image (DSI) consists of stacked matching cost images.

So for each *d*, a cost image is calculated and stacking these images produces what is referred to as the disparity space image *DSI(x,y,d)*, see Figure 7.1.

7.5.2 Cost Aggregation

In the stereo case, where only two images are available, the single pixel matching cost is not sufficient in order to do the final disparity computation. To avoid ambiguity and ensure robustness to noise and texture variation it is necessary to use a larger support region. In multi view approaches, like Yang et al. [44] using four cameras, the aggregation process can be discarded.
The aggregation of a fixed size support region is achieved by convoluting the DSI-volume with a box-filter, *w*:

$$DSI_{aggregated}(x, y, d) = DSI(x, y, d) \otimes w(x, y, d) \qquad (7.2)$$

Usually, the aggregation is done two dimensionally in the individual cost images but can also be done three dimensionally in the DSI-volume.

7.5.3 Disparity Computation

The objective of the preceding two tasks is to find an appropriate disparity value to each pixel in the reference image, in other words find the surface in the DSI-volume, which denotes the minimum cost, see Figure 7.2.

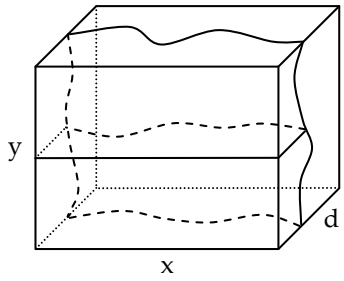

Figure 7.2: The minimum cost surface in the DSI-volume.

The simplest way to do this is by searching the DSI-volume in the d-direction and for each pixel assign the value of *d* that has the minimum cost, (7.3). This strategy is called “winner-takes-all”.

$$d(x, y) = \underset{d \in [d_{\min}, d_{\max}]}{\arg\min} \{DSI(x, y, d)\} \quad (7.3)$$

The "winner-take-all"-method is local in the way that each assignment of d to a pixel is independent of the surrounding assignments. Several other methods have been proposed, which uses global optimization of the minimum cost surface. Such methods can for example rely on either smoothness constraints or prior knowledge of the scene, but are as a consequence more computational intensive.

Additionally some refinement of the disparity map can be done, e.g. sub-pixel accuracy, filtering or outlier removal.

7.6 Related Work

A vast number of papers have been published with propositions on how to solve the stereo correspondence problem.

In [17], Kanade and Okutomi presented a method using an adaptive window size. To handle the effects of perspective distortion, their algorithm selects an appropriate window size based on variation statistics from the intensity images and the disparity map. Through an iterative process the disparity estimate for each pixel is updated, by choosing size and shape of the individual support regions until convergence.
Another proposition on how to handle perspective distortion comes from Kim and Park [18]. They use adaptive window warping, in a hierarchical matching process combined with probability theory, to compensate for the perspective distortion in certain regions.

In [33], Sun used fast cross correlation and rectangular sub-regioning to effectively calculate the matching costs. Furthermore, global optimization was used to extract the minimum cost surface from the DSI-volume.

Combining both area- and feature-based techniques Cochran and Medioni [2] suggested exploiting the advantages of both methods. They used edges to accurately locate depth discontinuities and remove the blur in these regions of the dense cross correlation-created disparity map.

In [37], Takita et al. used image pyramid-based Phase-Only Correlation techniques. The use of POC resulted in robustness against intensity variation and high sub-pixel accuracy estimation in a narrow baseline setup.

Also research has been done in hierarchical methods of image pyramid based coarse to fine searching techniques.

A full review of existing methods would be exhaustive and the reader is referred to Scharstein and Szeliski's taxonomy of area-based method [36].

The methods mentioned in this chapter are all clever and have their specific advantages. However, they are too advanced and computational complex to appeal to real-time applications.

7.7 Recent Research in Real-Time Stereo Vision

Before, and as late as five years ago, stereo was not considered suitable for real-time applications as a consequence of the heavy computational tasks involved in solving the correspondence problem.
However, recently Yang et al. have published a series of papers [42],[43] and [44] proposing to use the graphics hardware in computers to do the stereo matching, thereby freeing the CPU to do other tasks. They did an OpenGL implementation and the results are reasonable real-time frame rates.

Chapter 8

The Implemented Stereo Algorithm

In this chapter the implemented stereo algorithm is described and implementation issues along with considerations are reviewed and discussed.

The implementation is done in Matlab 7.01 with single computational heavy functions implemented as mex-files in Visual C++.

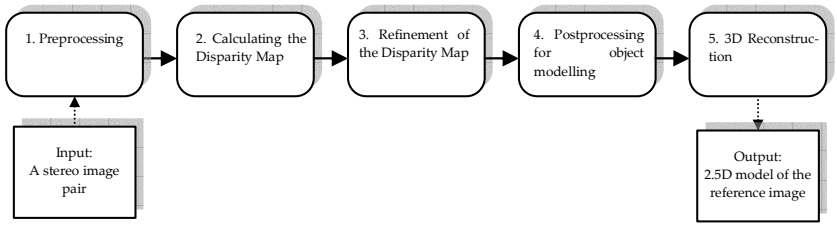

Figure 8.1: Block diagram of the implemented stereo algorithm.

The outline of the algorithm consists of five steps, as seen in Figure 8.1.

As the implementation is an offline version of an intended real-time system, this was considered when choosing the right type of algorithm. Based on the works of Yang et al. [42],[43],[44] and the taxonomy of Szeliski and Scharstein [36], an algorithm suitable for eventual real-time implementation was chosen. The algorithm is a traditional area-based method producing dense depth maps and, naturally inspired by Yang's graphics hardware method but with several modifications, which also should be possible to implement in graphics hardware.

8.1 Preprocessing

As the disparity estimation stage of the stereo algorithm assumes images satisfying the epipolar constraint to coincide with the scan-lines, the images have to be processed.
Based on the calibration data, this is done via a spatial transformation function provided with the Camera Calibration Toolbox for Matlab. It rectifies the image pairs and at the same time corrects for lens distortion.

8.2 Calculating the Disparity Map

To assist the different steps of calculating the disparity map, the stereo pair "Venus" has been chosen from Middlebury College's Stereo Vision Research Page [24]. The homepage has several stereo and multi-view images provided with ground truth ([32]) disparity maps, and is widely used in stereo vision research.

Figure 8.2: The Venus stereo pair (left and middle) and the sub-pixel accuracy ground truth disparity map of the left image.

The disparities in this image pair range from 20 (white), which is close to the camera and 2 (black), which is far from the camera.

8.2.1 Matching Cost

As matching cost for the stereo correspondence, the absolute difference measure AD was chosen. When the left image is reference (*s=1 in* (7.1)), the matching cost for a given *d* is formulated as

$$AD_d(x, y) = \left| I_L(x, y) - I_R(x + d, y) \right| \qquad (8.1)$$

and gives a measure of how well the pixels in the two images match when the right image is shifted *d* pixels relative to the left. As mentioned, AD is a dissimilarity measure, so low value means low cost, thus high similarity between the pixels.

The absolute difference can be implemented on graphics hardware by drawing the two images as textures in the frame buffer and then have a fragment shader computing the absolute difference of each pixel. The absolute difference image is transferred to a texture, and the frame buffer is ready to compute the matching cost for a new value of *d*.

Experiments have also been done with the SD measure, but the results didn't seem to be significantly different.

8.2.2 Aggregation

Cost aggregation is achieved by averaging the local values of a given support region two-dimensionally in the matching cost image. In the case of AD this is called mean-of-absolute-differences MAD and is defined as

$$MAD_d(x, y) = \frac{1}{r \cdot r} \sum_{m=-r/2}^{r/2} \sum_{n=-r/2}^{r/2} AD_d(x + m, y + n) \qquad (8.2)$$

where *r* denotes the size of the window that constitutes the support region.

The reason for choosing this type of aggregation is, again, that it can be efficiently implemented in graphics hardware. The GPU easily produces down sampled versions of textures (mipmapping) to avoid aliasing problems when rendered from afar. Applying the following filter, recursively, gives the different mipmap-levels (support region-levels) J^i of a texture (matching cost image) J^0:

$$J_{u,v}^{mml+1} = \frac{1}{4}\sum_{q=2v}^{2v+1}\sum_{p=2u}^{2u+1} J_{p,q}^{mml} \qquad (8.3)$$

where *(u,v)* and *(p,q)* are pixel coordinates and *mml* is the mipmap-level.
As a consequence, the GPU only generates textures with down sampling factors of 2^{mml}, giving support regions of 1-2-4-8-16-32... pixels.
Because of the way the GPU handles the aggregation stage, the method is often referred to as the mipmap-level method, and terms like mipmap-level 0-5 denotes the support regions of 1-32 pixels.

The problem is usually to choose the size of this support region, or the right mipmap-level. It must be large enough to include enough texture variation to avoid an ambiguous matching and be robust, but also small enough to avoid the effects of projective distortion and maintain the finer details of the surfaces.

In Figure 8.3 the calculated disparity maps (with the left image as reference image) for the individual mipmap-levels 0-5 are shown.

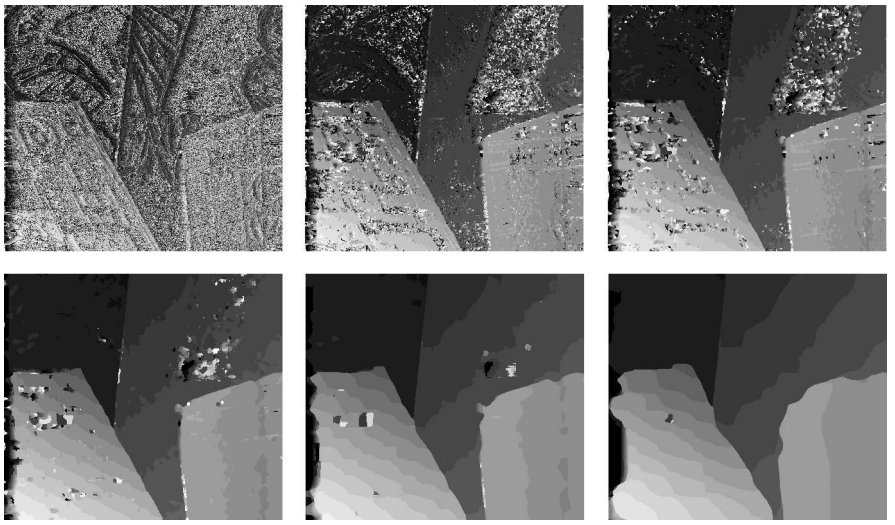

Figure 8.3: Disparity maps of the left Venus image for the individual mipmap-levels 0 (upper left) through 5 (lower right).

From Figure 8.3 it is obvious that the mipmap-levels give very different results. Level 0 is practically useless as it appear only to be noise. At the higher levels, the result resembles the ground truth with well defined edges, but is still very noisy. At the levels 4 and 5, the disparity map is fairly nice and smooth with little noise, but details as e.g. the edges are blurred heavily.

Instead of using a fixed size support region, a combination of the levels is desired, as both the precision and detail of the smaller support regions and the noiseless robustness, in less textured areas, from the larger support regions is wanted.

To achieve this, the levels are be combined by summation. This can be done by producing a DSI-volume for each wanted aggregation level and then collapse the volumes into one by averaging the corresponding cells in the volumes. By doing this both the robustness of the larger support regions can be achieved, while still preserving the distinct details of the surface, as these are needed for the ICP to work and of course to get a meaningful rendering of the model.

In Figure 8.4, the disparity maps of the combined DSI-volumes are shown in a cumulative sense. That is, the first disparity image is MML 0. The second is the average of the DSI-volumes of MML 0 and 1. The third the average of MML 0,1 and 2 and so on, up to the average of the DSI-volumes of MML 0, 1, 2, 3, 4 and 5.

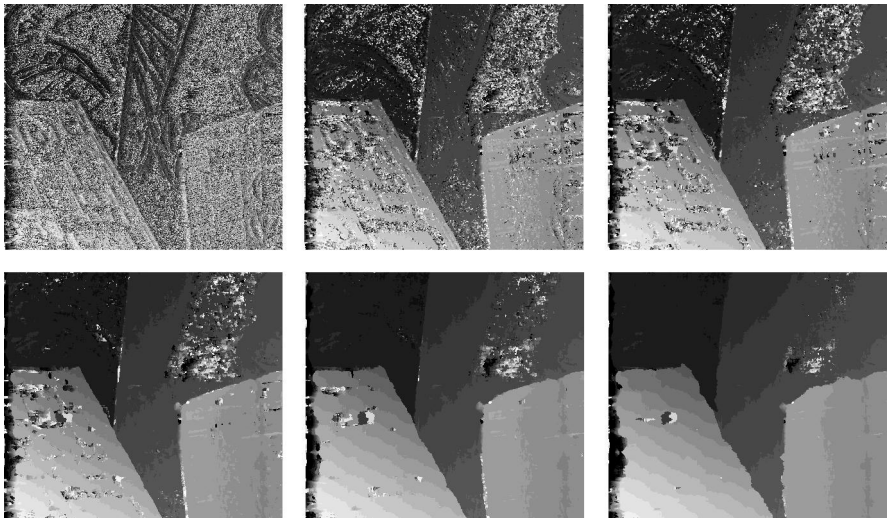

Figure 8.4: The disparity maps of the cumulative DSI-volumes from MML 0 (upper left) through 5 (lower right).

It is seen that as more mipmap-level DSI-volumes are averaged, the resulting disparity map looks more like the ground truth. The disparity map of all six levels is a fairly good estimation of the ground truth. It still has some erroneous regions, but the edges are sharper than when using a single mipmap-level.

Averaging the DSI-volumes of different mipmap-levels can be seen as doing aggregation with a large symmetric filter, where, as a consequence of the summation, pixels are weighted differently, see Figure 8.5.

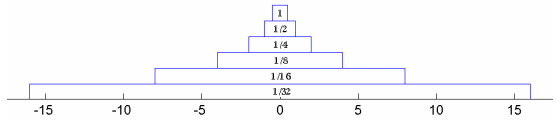

Figure 8.5: A schematic 2D illustration of the filter, with weight factors for each pixel in the region.

Summing up the weights for each pixel in the support region, results in the spatial filter profile of Figure 8.6.

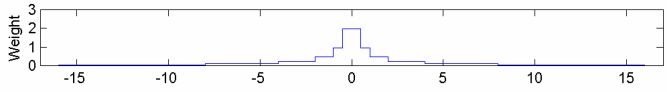

Figure 8.6: Filter profile of the summed mipmap-levels.

Pixels closer to the centre receive more weighting, and pixels away from the centre less weighting. This ensures the robust and high-frequency preserving characteristics of the method.
A support region structure of this type can be approximated in matlab, by summing six gauss kernels with individual sigmas, corresponding to the size of each support region, giving the mml-filter seen in Figure 8.7.

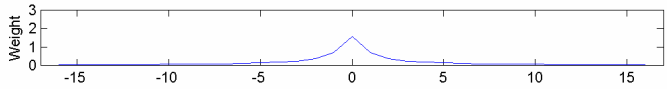

Figure 8.7: The matlab approximation of the mml-filter (mipmap-levels 0-5).

This filter was used in the matlab stereo implementation, as it is easy to generate, for a random selection of mipmap-levels and by having just one filter containing all six support regions, reduces the filtering tasks of each cost image significantly. Furthermore, the filter is symmetric and thereby separable which reduces the computational tasks in the convolution of a cost image.

Using the mml-filter (0-5) to do aggregation gives the resulting disparity map shown in Figure 8.8.

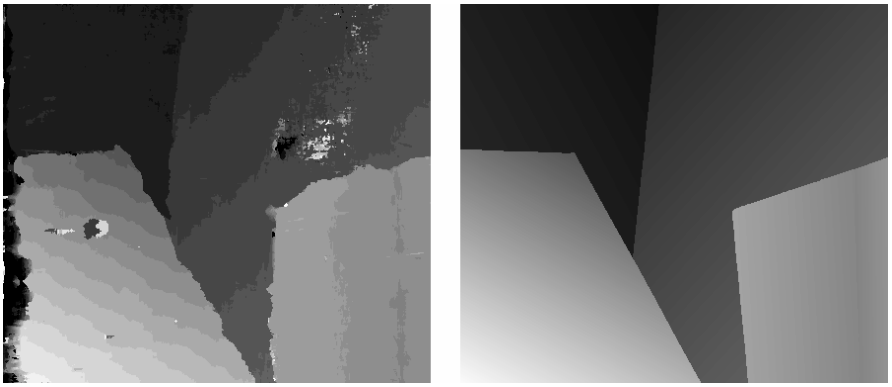

Figure 8.8: Final disparity map using the mml-filter with all 6 levels (left). The ground truth disparity map (right).

Overall, the result of using the mml-filter is a good approximation of the ground truth. The structures are preserved and edges are somewhat straight. However, there is some regions, which are very affected with outliers.

The smaller mipmap-levels of 0 and 1 try to capture high frequency detail. From the constraints of the algorithm, it is assumed that the topology of the objects contains a relative amount of smoothness. By exploiting this, the smaller mip-map-levels can be discarded in the aggregation, resulting in less weighting given to the centre pixels and its nearest neighbours, thus introducing some smoothness in the resulting disparity maps, as seen in Figure 8.9.

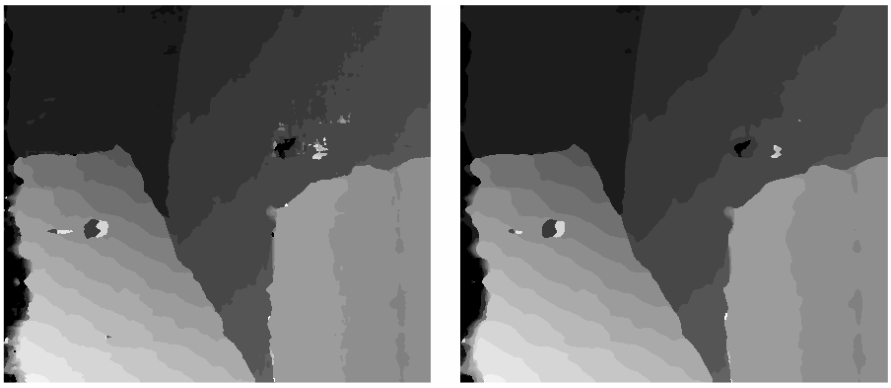

Figure 8.9: Disparity maps using the mml 1-5 (left) and mml 2-5 (right).

By discarding either 1 or 2 of the smaller levels a smoother and less noisy disparity map is obtained. It is bets seen in the upper right quadrant of the disparity maps, where the noise is significantly reduced compared to Figure 8.8. This gives better conditions for the ICP algorithm to perform successfully.

8.2.3 Disparity Computation

From the combined mipmap-level DSI-volume the final disparity is estimated with the "Winner takes all" method, as described in the previous chapter.

8.3 Refinement of the Disparity Map

To clean up the disparity map some additional features have been implemented in the stereo algorithm.

8.3.1 Cross Checking

Cross checking is an efficient way of removing possibly erroneous disparity assignments in discontinuity regions with occlusions, repetitive or non-textured areas, where the disparity estimation can be attached with high uncertainty.

The method relies on the fact that the two cameras see different versions of the scene, so what is seen by the left camera is necessarily not in the right cameras field of view. Thus the disparity maps of each image (as reference) are slightly different; however, the features that can be seen by both cameras must be consis-

tent. So the reliability of disparities can be validated through a two view consistency constraint.

With the DSI-volume of the left camera already calculated, the corresponding DSI-volume of the right camera is easily obtained in the following way: In a copy of the DSI-volume, each matching cost image is shifted its corresponding value of d (in pixels) to the left.

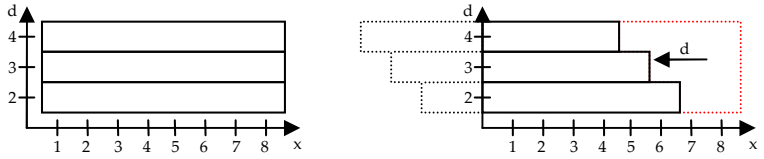

Figure 8.10: A two dimensional sketch of a DSI-volume (in the x-d plane) with the left image as reference (left), and the corresponding DSI-volume of the right image as reference (right).

The principle is shown in Figure 8.10 for a couple of one dimensional images of eight pixels each. The DSI-volume with the left image as reference (Figure 8.10 left) has a depth of 3 as d_{min}=2 and d_{max}=4. When the AD cost images are shifted respectively 2, 3 and 4 pixels an empty space (red stipples) remains in the DSI-volume with right image as reference. As the matching cost used is AD, the empty space is just filled with high numbers.
The right disparity map can now be calculated from the right DSI-volume.

Having calculated both disparity maps, they can be cross checked for consistency. From each pixel in the left image, the corresponding position in the right image is found based on $d_L(x,y)$. From these positions in the right image the mapping is reversed this time based on $d_R(x,y)$. The end positions, in the left image are subtracted from the respective initial positions, resulting in a perturbation measure δ.

$$\delta = \left| d_L(x,y) - d_R(x + d_L(x,y)) \right| \qquad (8.4)$$

Based on the resulting perturbations a pixel can be discarded if the displacement error is larger than some threshold. This is an effective tool for removing outliers in the disparity map, because if a pixel has a high perturbation value, it can be assumed that either one or both estimates (of left and right DSI-volume) are in-

correct. As the pixel therefore is attached with a high uncertainty it can be discarded. The large amount of redundant data, from capturing real-time, allows this careless deletion of erroneous points as they most likely will be scanned again from a different angle.

8.3.2 Sub-Pixel Accuracy

For some applications pixel level disparity correspondence may be sufficient. In case of the designed system it is not, though.
The system has a B/D relationship of approximately 1/3 – 1/7 in a working distance of 0.25-0.5 m from the camera head, which means that the depth resolution, in this region, is 6-10 times coarser than the corresponding lateral resolution. This means that calculating pixel level disparity maps, results in very staircase like 2.5D triangulations, which eventually will cause difficulties in the ICP algorithm, as the reconstructed surface isn't smooth.
The solution is to improve the resolution in the depth direction, by doing sub-pixel estimation of the disparity map. There are several methods for achieving this sub pixel accuracy.

A simple way of handling this would be to low-pass filter the disparity map and thereby introduce a smooth surface structure. This method, though, would remove the detail of high frequency content and is therefore inappropriate.
The brute force method, of intensity interpolation, consists of up-sampling the image data and then perform correlation on the new images of higher resolution. This method though, is not very convenient for our purposes, as it heavily increases the computational tasks of matching depending on how big accuracy is wanted.
Phase Correlation techniques can also be applied as Takati et al. have done it in [37]. Disparity is determined from the phase of the fourier spectrums of the aggregation windows. This method is computational complex, and also not well suited for the current problem. For more in-depth information of these two methods, see [38].

A more suitable solution is matching cost interpolation. The assumption of the method is that the captured scene is a continuous signal which, through the imaging process, has been sampled. As a consequence thereof the DSI-volume is also a sampled version of the continuous disparity space image. This leads to the idea of doing low order interpolation in the DSI-volume around the initially found, pixel level based, disparities and thereby reconstructing the continuous disparity estimates in this region.

Off course the theory only holds if the actual frequency content in the continuous scene does not exceed what the interpolating function is able to estimate. In other words, the captured images have to be sampled at a high enough frequency (the Nyquist theorem).
Szeliski and Scharstein has done some work [34][35], where the frequency characteristics of the continuous DSI-volume is examined, with help from the point spread function of the imaging process and Fourier transform. They can then determine the order of interpolation needed to get the correct precision in terms of sub-pixel accuracy.

The assumptions of this method comply with the system constraint of objects having smooth surfaces.
In the implementation, a 2nd order interpolation is believed to be sufficient. So in the DSI-volume, for each pixel, a quadratic fit is made from the matching costs on either side of the initially found disparity, P_a and P_b, and the matching cost of the disparity itself Pm, see figure Figure 8.11. But if necessary, a higher order function could be fitted to more neighbouring points.

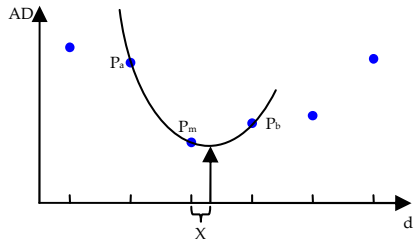

Figure 8.11: For each pixel in the DSI-volume, a 2nd order polynomial is fitted around the matching cost values (blue) of the disparity with minimum cost.

The minimum of the parabola can be found, analytically, with the following expression:

$$X = \frac{P_a - P_b}{2 \cdot (2P_m - P_b - P_a)} \qquad (8.5)$$

where the result X is the correction of the initially found disparity, in terms of disparity units (pixels). For further details see Anandan [2].

Looking at the Venus data the effect is obvious. With only pixel based disparity estimation the oblique surfaces in the image suffer from a "staircase" effect shown in Figure 8.12.

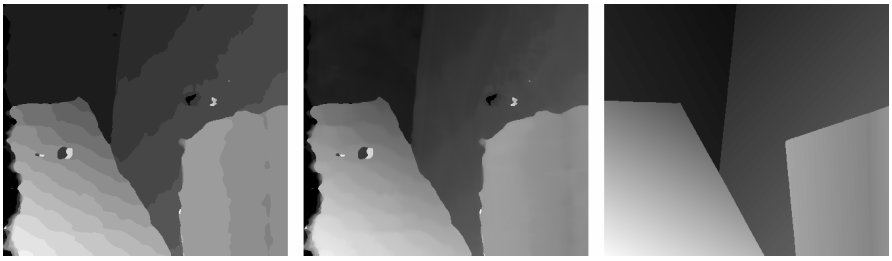

Figure 8.12: Three disparity maps of the Venus data. Pixel level based (left), sub-pixel accuracy (middle) and the true disparity map (right).

However, the sub-pixel estimate posses' smoother surfaces and resembles the ground truth data much better.

The sub-pixel disparity estimation is done in both DSI-volumes before cross checking.

8.4 Postprocessing for Object Modelling

In this project, stereo is used for 3D modelling, and since the object doesn't fill the entire field of view this can be exploited, as prior knowledge of the object is available on behalf of the uniformly coloured background.
Also, certain information has to be extracted from the disparity map, before it is triangulated, in order to prepare for the registration.

8.4.1 Dealing with Known Discontinuities

As discussed earlier, discontinuities can act as a "strong" texture and draw erroneous disparities to it. The effect is especially noticeable when doing object modelling against a uniformly coloured background.

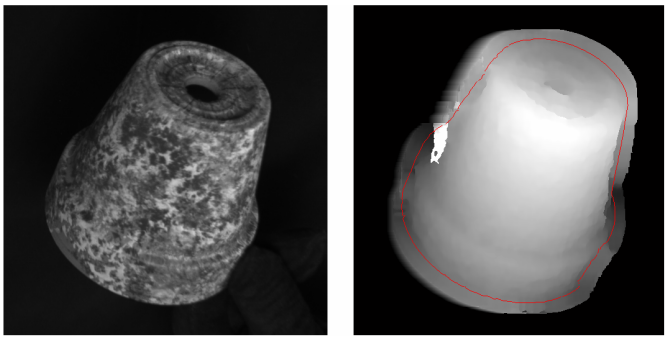

Figure 8.13: A textured pot (left) and the calculated disparity map (right) with the border of the object marked with red.

The strong textural discontinuity of the border has a strong and unwanted effect in the stereo matching as it dominates in these regions, causing false disparity estimations both inside and outside the object, see Figure 8.13 (right).
Assuming that the background colour doesn't appear in the object, this prior knowledge can be exploited through segmentation, and disparities outside the object border can be discarded, as they are expected to be false, see Figure 8.14.

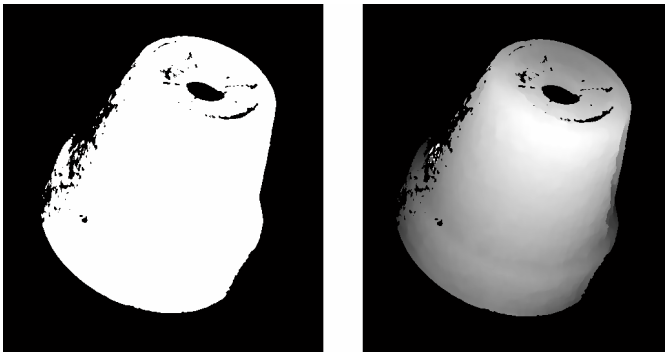

Figure 8.14: The binary mask of the original image (left) and the resulting disparity map when applying the mask (right).

The resulting disparity map looks much better, even though pixels have also been discarded for example on the bottom of the pot, which is unwanted but a result of the method. The trick is of course to only segment out the background, and this is difficult in many cases when objects possess various colours and intensities (well textured).

Having colour cameras would make this tool more efficient, as the background colour could be more distinct (like blue screen techniques)

Examining the disparity maps of Figure 8.13 and Figure 8.14 in detail, it is obvious that pixels in the inside border region, also are influenced by the "strong" textural discontinuity and thereby are unreliable. The effect is most visible near the right border of the pot, and as a consequence all these pixels also have to be discarded.
Due to the camera geometry, the parallax shift is in the horizontal direction, and therefore the inside region problem is only relevant for vertical border regions.

The inside region, is estimated based on the background/object segmentation (Figure 8.14 (left), by convoluting the binary image with two opposite edge-filters, see Figure 8.15,

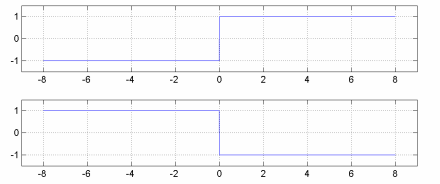

Figure 8.15: The filters applied in order to find the respectively left (top) and right (bottom) object border of the binary mask.

The filters highlight the vertical edges of each side of the object. As shown in Figure 8.16.

Figure 8.16: The vertical edges of the binary object/background mask.

Stretching the edges in the parallax direction, gives a rough estimate of the region affected by the texture discontinuity, see Figure 8.17

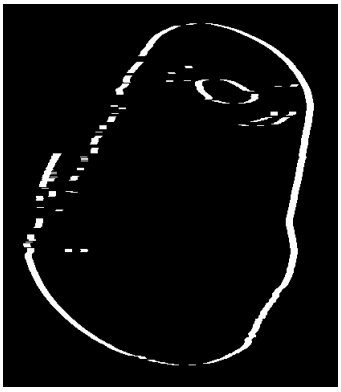

Figure 8.17: The vertical borders are stretched in the parallax direction and combined to a single binary edge mask.

The amount of stretching is determined on behalf of the mipmap-levels used. From experiments it was found that stretching the edges half the size of the biggest support region used (radius of the aggregation filter), gave good results.
The combined edge mask is inverted and multiplied with the disparity map.

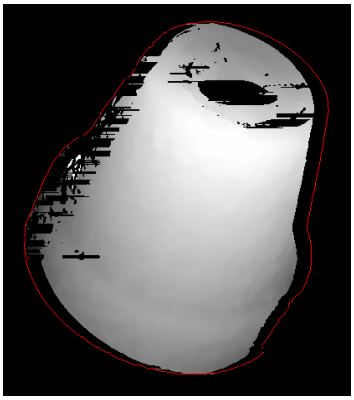

Figure 8.18: The resulting disparity map, where regions inside the object border (red) has been reduced.

In Figure 8.18 the effect is seen. The disparity map of the object has been eroded in the parallax direction, to remove the unreliable pixels.

8.4.2 Estimating the Object Border

The ICP algorithm needs knowledge about which points are on the border of the object. This information is easier to find before the disparity map is triangulated, as the shape then still is represented in 2D.

To determine these border pixels, a binary image of the disparity map (Figure 8.18) is eroded with a circular binary filter of eight pixels in diameter.

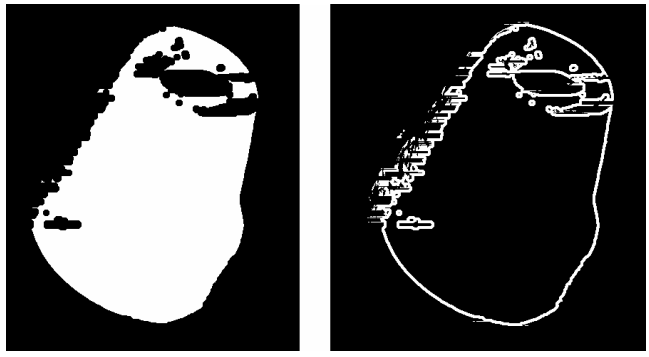

Figure 8.19: The eroded binary disparity map (left), and the disparity map edge (right).

In Figure 8.19 (left) the eroded disparity map is seen. This mask is subtracted from the binary disparity map resulting in the edge map of Figure 8.19 (right).
Each pixel is then marked as being a border point of the object or not.

8.5 3D Reconstruction

Given the camera parameters from the calibration and equation (6.2) the final disparity map is triangulated into a range map of 3D world coordinates (having exact size) with reference in the focal point of the left camera.
The result is a point set (point cloud) as depicted in Figure 8.20.

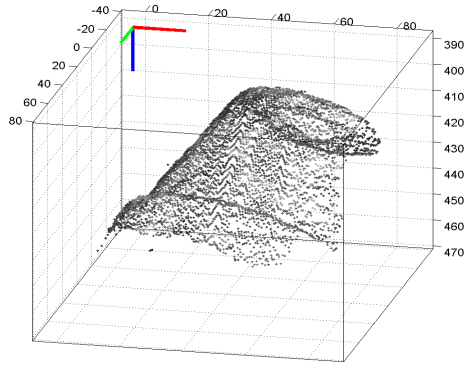

Figure 8.20: The 3D point set cloud (decimated) resulting from triangulating the disparity map of the "pot".

Through matlab, an interpolated grid can be applied to the point set.

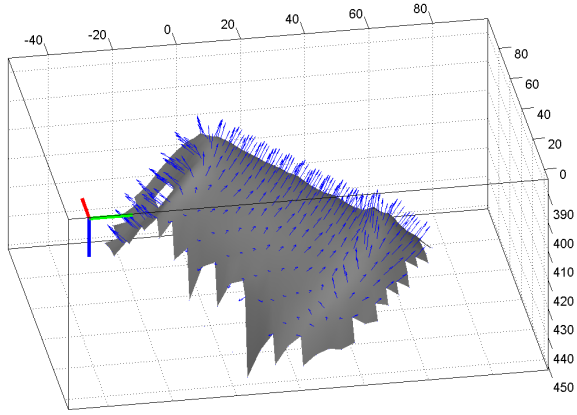

Figure 8.21: An explicit surface is drawn through the point set, which estimates the normal in each point (the surface is decimated to reduce the number of normals in the visualization).

The interpolation function automatically calculates the normal of each point which is useful in both the following registration, but also for shaded visualization, see Figure 8.21.

In Figure 8.22, the result of stereo matching, refinement and triangulation is seen. A 2.5D textured model (with surface normals, not shown) of the object captured in the stereo image pair.

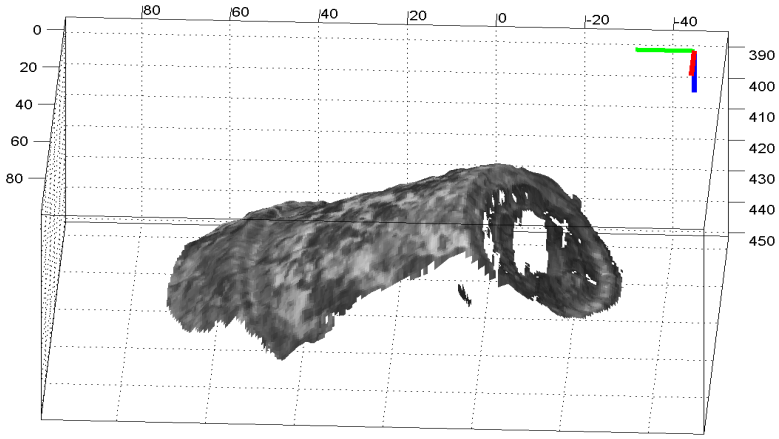

Figure 8.22: The textured 2.5D model of the stereo data.

8.6 Summary and Discussion

A stereo algorithm which produces dense disparity maps, has been developed and implemented in matlab. The algorithm is typically referred to as the multiple mipmap-level method, as it combines different aggregation levels by summation, thereby achieving the robustness of large support regions and finer detail of small support regions.
The calculated disparity maps are refined through sub-pixel accuracy in order to increase the depth resolution, and cross-checking to eliminate erroneous disparity assignments.
Finally the border of the object is estimated in the disparity map before it is triangulated into a 2.5D reconstruction and normals are determined.

The method is suitable for real-time implementation in commodity graphics hardware.
The sub pixel accuracy refinement of the disparities results in a higher resolution in the range direction of the system. As recalled, from the calibration chapter, the

depth resolution was in the area of 2 mm. in a working distance of 0.5 m. By estimating the sub pixel estimates and setting the perturbation constraint of cross-checking to ½ pixel the depth resolution can be reduced to 1 mm. This is the usual setting that has been used and in the experimental chapter, the depth accuracy will be tested to see if it really is in the order of 1 mm.
Lower perturbation has also been tried but e.g. ¼ pixels results in too sparse disparity maps as the noise level of the system is reached.

A dilemma of the stereo algorithm is whether it should emphasize on producing dense surfaces or remove possibly erroneous disparity assignments to get a more correct but sparse surface.
Erroneous pixels can cause difficulties for the ICP-algorithm. This is also the case for sparse disparity maps, so it is a balance that has to be kept.
As too much error removal in the stereo matching also removes true pixels, inevitably, it is better to let the *max_dist* in ICP along with other constraints deal with correspondences that are not so likely, and can cause problems in the fine alignment stage.

Part III

Registration via ICP

Chapter 9

Introduction to Registration

To produce a full 3D model of the 2.5D shapes coming from the stereo module, the individual point sets have to be properly aligned with respect to each other.

This chapter describes the theory behind automatic registration using the Iterative Closest Point (ICP) algorithm and the problems that occur in shape alignment are discussed along with the assumptions and constraints needed to handle these problems.

9.1 Registration

Registration covers the topic of aligning images or 3D shapes so their overlapping regions match.

The problem is relatively simple if corresponding points (control points) in the two objects are known. In that case, an optimization method (e.g. Marquardt), or even better, a closed form solution can be applied to find the transformation parameters T and R that minimize some error metric (e.g. the 2-norm) of the control points:

$$\arg\min_{T,R}\left\{\sum_{i=1}^{N}\left\|x_i-(R\cdot p_i+T)\right\|^2\right\} \quad (9.1)$$

where x_i and p_i are control points in respectively, the reference shape X, and the target shape P (new shape to be registered) and N is the number of control points. Since interactive point clicking is not an option in real-time scanning, the control points are not known prior to the registration. Thus a different approach must be taken.

9.2 The Iterative Closest Point Algorithm

A registration method has been proposed by both Besl & McKay [4] and Chen & Medioni [6], which has become a dominant method for aligning three dimensional shapes based only on the geometry, namely the Iterative Closest Point algorithm.

ICP is capable of handling triangle meshes, parametric surfaces, implicit surfaces etc., and also in this case point sets, which makes it a very flexible tool for shape alignment.
The method is an iterative process, which aligns the shapes in small steps until convergence, and normally consists of six looped stages, as seen in Figure 9.1.

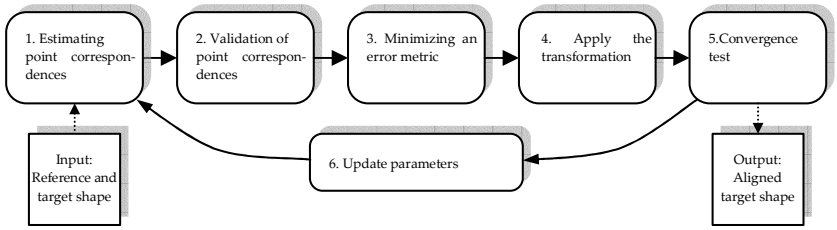

Figure 9.1: Block diagram of the ICP algorithm.

1. Estimating point correspondences

Between the two shapes, acceptable corresponding control points have to be established from the target shape to the reference shape.

2. Validation of point correspondences

The estimated control points are tested for satisfying certain constraints, and rejected if not.

3. Minimizing an error metric

Based on the estimated control points, an R-T transformation is calculated which minimizes some error metric.

4. Apply the transformation.

The calculated transformation is applied to the target shape.

5. Convergence test

The result of the iteration is evaluated. Based on the convergence state, the iterations can either be stopped or continued by going back to step 1 via step 6.

6. Updating parameters

Constraint parameters are updated.

For details concerning variations of the different stages, see Rusinkiewicz and Levoy's "Efficient Variants of the ICP Algorithm" [30].

9.3 Problems, Constraints and Assumptions

For the ICP-algorithm to work properly, some issues have to be considered.

9.3.1 Overlapping Regions

First of all, to make the automatic registration possible, it is a necessity that the two shapes share an overlap. That is, they must both include the same partial region of the scanned object. The bigger overlap, the better, and this is the reason why the system has to be real-time; so data is acquired with a high frame rate resulting in maximum overlap because of the minimal movement of the object between consecutive image pairs.

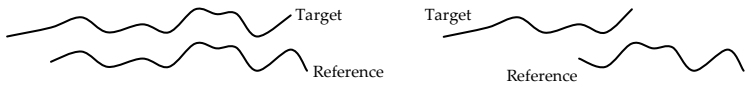

Figure 9.2: Two shapes with respectively big (left) and small (right) overlap.

In Figure 9.2, the shapes having big overlap (left) have good conditions for successful alignment, whereas the small overlap (right) will result in a less precise alignment and possibly erroneous.

Regarding the overlap an issue has to be considered when estimating control points. If selected points in the target set are outside the overlap region, they are forced to choose a wrong point in the reference set, see Figure 9.3.

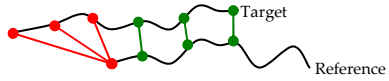

Figure 9.3: The non-overlap region can result in erroneous control points (red).

This situation (50% overlap) will result in a wrong estimation of the transformation, because of the large number of incorrect control points. The problem can be solved by introducing a constraint that tests if a control point pair includes the border of the reference shape which it isn't allowed.

9.3.2 Object Topology

The object topology strongly coheres with the choice of starting guess of the target sets position. A reasonable starting guess is necessary for each target set in order to find the global minimum, and not get stuck in local minima. How good the starting guess has to be is difficult to say and is totally dependent on the surface topology. If the surface topology is very jacked or noisy the starting position has to be very close to the true position in order to converge, whereas a smoother surface with bigger more distinct features can handle a coarser starting guess.

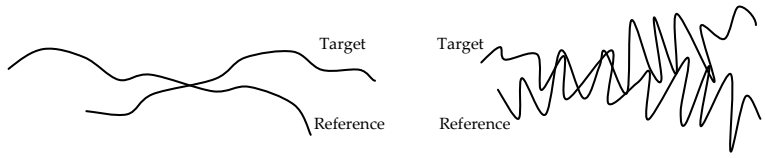

Figure 9.4: Two examples of shapes with respectively smooth (left) and jacked (right) surface.

In the example of Figure 9.4, finding "correct" point correspondences are more probable on smooth shapes, as opposed to either noisy or high frequency shapes, where it undoubtedly will result in wrong correspondences and thereby a fatal alignment.
Again the use of real-time data acquisition is assumed to assure good enough starting guesses for objects with relatively smooth surface topology.

A necessity for the ICP-algorithm to converge, when only applying geometry data, is that the surfaces need to have distinct features, unlike for example coplanar surfaces, cylinders or spheres, see Figure 9.5.

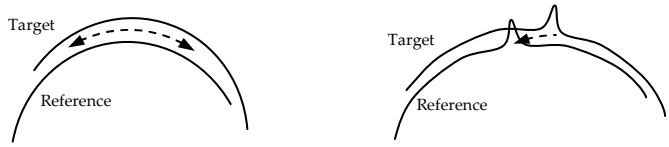

Figure 9.5: Two examples of overlapping regions. (left) Circular, with no features. (right) Circular, with a small distinct feature.

The circular shapes will never reach the true alignment, as the target will keep sliding back and forth on the reference surface. Minor features on the surface, can save the problem, though, as seen in Figure 9.5 (right).

To handle geometrically featureless surfaces, the surface feature of point colour (intensity) can be incorporated in the registration algorithm. This is an obvious possibility as textured surfaces already are a strong assumption of the system.

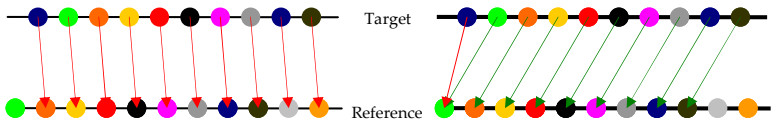

Figure 9.6: Point correspondence in traditional geometric ICP (left) and combined geometric and colour ICP (right).

As seen in Figure 9.6, colour ICP handles the point correspondence better in geometrically featureless areas, as colour dissimilarity is punished and a pixel further away is chosen as the correspondence.

9.3.3 Handling Outliers

To prevent outliers having disastrous influence in the error measure minimization, when estimating the optimal transformation, special care have to taken when selecting point correspondences. Therefore certain constraints can be set, which the correspondences have to obey.

For example as shown in Figure 9.7, where an outlier in the target set (red line) has been selected as a control point.

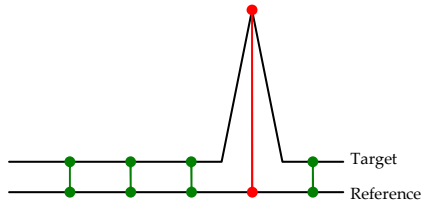

Figure 9.7: Influence of outliers in the target set.

When minimizing the residuals the outlier will undoubtedly cause big problems, because, as the correspondence distance is large compared to the other control points, it will be dominating in the least squares solution.
A way to deal with this type of problems is to introduce a constraint of maximum distance on the control point correspondences.

Another example could be outliers in the reference set, which would be attractive to a lot of control points in the target set, see Figure 9.8 (left).

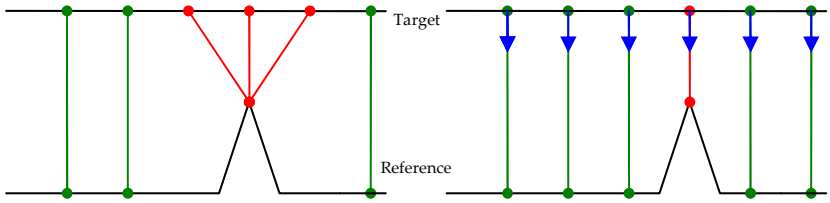

Figure 9.8: Control point correspondence in the basic manner (left) and in a projection based manner (right).

This situation can be handled by choosing control point correspondences with respect to the normal (blue arrows) of the shape. In other words, projecting the target control points onto the reference shape in the normal direction, and then choose the nearest point of the intersection, see Figure 9.8 (right).

To assure good convergence conditions, the shapes need to be well defined and smooth. In the case of point sets, this means that they need to be dense and not too affected with noise. The demands are met by the stereo algorithm, which produces dense disparity maps with emphasis on smoothness and error removal.

At last, it should be mentioned, though, that no matter how well the control points are estimated, the algorithm will only converge towards the optimum alignment, and not the 100% correct one. This is a consequence of the overlapping regions not being identical, but sampled versions of the 3D object. Therefore the alignment can only be optimized in some sense, giving a possibility of error accumulation during the registration, which results in drifting of the aligned shapes.

9.4 Related Work

ICP is a widely used method for various applications.

Lu et al. [21],[22] used ICP for aligning 2.5D face scans into a complete 3D face model.
Martins et al. [23] integrated texture information in ICP registration for object modelling.

Johnson and Kang [16] used ICP with texture information for registration of partial scenes into complete 3D virtual environments.

Chapter 10

The Implemented Registration Algorithm

In this chapter, the details of the implemented registration algorithm are described, in particular, the modifications of the ICP-algorithm, which was needed for it to perform well for the purpose of object modelling. Also, possible extensions and the limitations and problems of the algorithm are discussed.

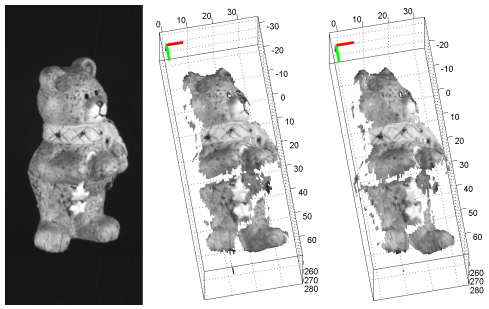

Figure 10.1: The left image of a pair from the "bear" sequence (left) and its corresponding (middle) and consecutive range map (right).

To visually assist the flow through the different stages of the algorithm, consecutive range maps from the "bear" sequence was chosen.

10.1 Object vs. Camera Coordinate System

The object coordinate system has been chosen to coincide with the camera coordinate system. Thereby, the orientation of the object, captured in the first image pair, will determine the objects orientation in the object coordinate system as the first stereo measurement data is kept as it is, and the following 2.5D measurements are appended to this reference.

10.1.1 Global Starting Guess

During the registration process, the incoming target surfaces are always aligned to the previously registered surface. As the object is rotated, the incoming target surfaces still lie in the camera coordinate system and therefore, after a few registered surfaces, are not found in the neighbourhood of the last surface registered.

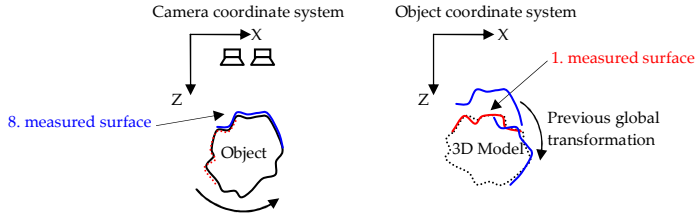

Figure 10.2: The object being rotated in the camera coordinate system (left) and the 3D model being build in the Object coordinate system (right).

To cope with this, the estimated global transformation, of the previous target surface ($R_{global,prev}$ and $T_{global,prev}$), must be applied to each incoming target surface, acting as a coarse (global) starting guess, see Figure 10.2.
The alternative is applying a transformation to the entire model, for each incoming target surface, resulting in a dynamically changing model coordinate system. This is disadvantageous though, as it takes more computations to apply the transformation to the entire model than to one individual point set. Also, in terms of an eventual real-time implementation, the real-time preview is easier to operate if the model coordinate system is stationary and do not change with each alignment of a new target surface.

The global starting guess of the target surface is therefore just the position and orientation of the previous surface. If data is acquired at a high enough rate, as

assumed, the movement of the object is therefore minimal and successful alignment should be possible, if no peculiarities of the object surface is present, such as repetitive geometry or no features at all.

10.1.2 Local Starting Guess

The previous global transformations are a necessity for each incoming target shape, but in addition a second starting guess can be added to the incoming target shape (which has been transformed with the previous global transformation). The previous local transformation ($R_{local,prev}$ and $T_{local,prev}$), can also be applied to the target shape, based on the assumption that the object is manipulated in a steady and continuous motion so the relative transformation of the next target point set is approximately close to the previous.

Additionally the target shape is translated so its centroid matches the centroid of the reference shape.

Figure 10.3: The three stages of estimating the coarse alignment of the target shape (coloured) w.r.t. the reference shape (textured): The previous global transformation (red). The previous local transformation (green). Translation to match the centroids (blue).

Obviously the first starting guess, the previous global transformation, is the by far most important, but the previous local transformation and translation of the target shape also helps in attaining the optimum starting point for the iterative fine alignment, see Figure 10.3.

10.2 Finding Point Correspondences

The difficulty of automatic registration is to find the appropriate point correspondences, which for every iteration ensure convergence towards the optimum alignment. The point correspondences are also called control point sets, and are denoted

$$C_h(p_h, x_h) \quad h = 1 \ldots H, \tag{10.1}$$

where C_h is a point pair consisting of a point p_h from the target set P and where x_h the corresponding point in the reference set X (h is not the index number in the respective sets).

$$p_h \in P \quad P = \{p_i\}, \qquad i = 1 \ldots I \tag{10.2}$$

$$x_h \in X \quad X = \{x_j\}, \qquad j = 1 \ldots J \tag{10.3}$$

The points p_h are randomly distributed in the target point set, and for each point p_h in the control set, the closest point in the reference point set is found as the point, which minimizes some error metric. In this implementation, the 2-norm has been chosen, which for geometrical vectors denote the Euclidean distance.

$$x_h = \min_{x_j \in X} \left| p_h - x_j \right|_2 \tag{10.4}$$

A control point set of the most likely correspondences between the target and reference point set is now established.

10.2.1 Colour ICP

To cope with the problems of featureless shapes and get better performance in the fine alignment process, the basic control point search has been modified a little.

Because the stereo acquisition process also provides texture information of the range data, the intensity information of each point can be used to increase robustness when the geometric properties alone have problems.

The point intensities of the target and reference point sets, are respectively denoted $I(p_i)$ and $I(x_j)$.
To exploit this, the correspondence estimate of the target control points is expanded to be the point x_j, which minimizes the error metric below.

$$C_h(p_h, x_h) = \min_{x_j \in X} \left\{ \left| p_h - x_j \right|_2 + \lambda \left| I(p_h) - I(x_j) \right| \right\} \qquad (10.5)$$

This means, that for a correspondence to be established, the intensities of the two points also need to have similar intensity to some extent depending on the weighting parameter lambda.
As lambda is small, the Euclidean distance to points in the reference point set is all dominating, but as lambda increases the absolute texture difference gets more influence in the selecting the closest control point.

10.2.2 Control Point Validation

In order to avoid wrong correspondences and the harmful influence of eventual outliers, some constraints are made to validate the control points. If the constraints are not satisfied the control points are discarded.

To test if the selected target control points are outliers, the Euclidean distance between the control pair pairs must satisfy the following equation:

$$\left| C_h \right|_2 = \left| p_h - x_h \right|_2 \le d_{\max} \qquad (10.6)$$

where d_{max} is the maximum Euclidean distance allowed between control point pairs. The value of d_{max} is updated after each iteration and is dependent on how close the two point sets are positioned. To begin with, though, d_{max} is a very high value to allow all control point matches.

Another constraint, which has to be satisfied for each control point pair, is based on the assumption of using the overlapping areas for the registration. The control points selected, have to lie inside the overlapping region to make sense, otherwise they will only contribute negatively when computing the transformation to minimize the error metric of the control points.
A way to control this is by checking each corresponding reference control point if it lies on the boundary of the reference point set. From the stereo module, all

boundary points have been marked for this purpose. If this is the case, the chosen target point most likely is not part of the overlapping region.

Because of the movement of the object, the target point set includes new regions which are not present in the reference point set. Often for target points in this new region, the closest reference control point lie on the boundary, and thereby it can be checked if selected target control points lie in this region.

10.3 Estimating the Optimum Transformation

Given the estimated control point pairs p_h and x_h, a transformation have to be estimated which minimizes the distance between the point pairs, in the sense of some error metric. As the shapes are of absolute size no scaling is required in the optimization, only translation and rotation as in (9.1).
Popular closed form solutions to this problem have been proposed by Horn et al. which used ortho-normal matrices [14], and Horn using unit quarternions [13].

The implemented method for determining the absolute orientation is based on the work of Arun et al. [3], and the method uses singular value decomposition to find the least-squares fit of the two control point sets.

The control point sets are related by:

$$x_h = R \cdot p_h + T + n_i \qquad (10.7)$$

where n_i is a noise-vector.
Estimating R and T, is done by minimizing the residual error Σ^2 in the least-squares sense:

$$\Sigma^2 = \sum_{h=1}^{H} \left\| x_h - (R \cdot p_h + T) \right\|^2 \qquad (10.8)$$

The centroids of the two sets are calculated as

$$p_c = \frac{1}{H} \sum_{h=1}^{H} p_h \qquad (10.9)$$

$$x_c = \frac{1}{H}\sum_{h=1}^{H} x_h \qquad (10.10)$$

The centroids are subtracted from the respective control points:

$$p_h^n = p_h - p_c \qquad (10.11)$$

$$x_h^n = x_h - x_c \qquad (10.12)$$

The translation can now be decoupled from equation (10.8), resulting in the following residual to minimize in order to estimate *R*.

$$\Sigma^2 = \sum_{h=1}^{H} \left\| x_h^n - R \cdot p_h^n \right\|^2 \qquad (10.13)$$

This equation can be minimized using SVD in the following way:

$$M = \sum_{h=1}^{H} p_h^n x_h^{nT} \qquad (10.14)$$

Using SVD on *M* gives:

$$M = U\Lambda V^T \qquad (10.15)$$

And R is calculated as:

$$R = VU^T \qquad (10.16)$$

Having found R, T is finally calculated by:

$$T = x_c - Rp_c \qquad (10.17)$$

The calculated R and T are applied to the target point set (Figure 10.4 (bottom)), and after each applied transformation, independent closest point correspondences are found, to check if the average distance of the two point sets actually was reduced. If not, the transformation is reversed and a new iteration of finding control point pairs and calculating the least squared fit is initiated.

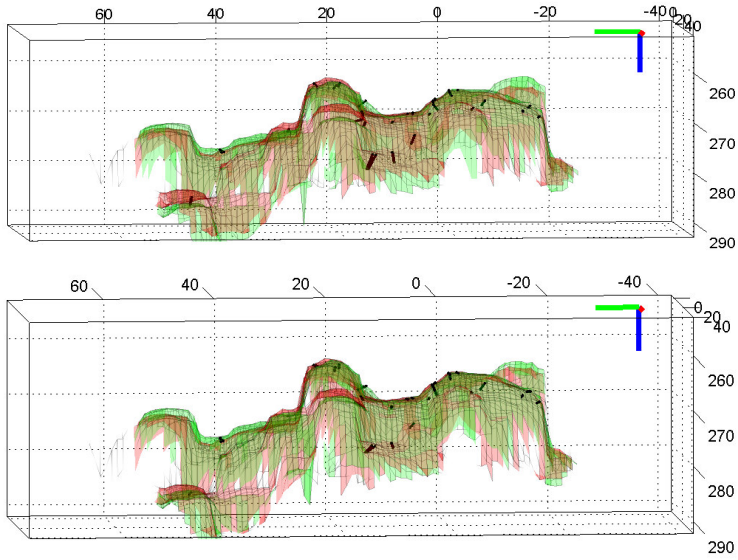

Figure 10.4: The two shapes with estimated control points (top), and the shapes after the transformation which minimizes the control point distances (bottom).

Figure 10.4 shows the first iteration of aligning the point sets from Figure 10.1. The control point distances (black) is clearly shorter after the transformation and that the alignment is closer can be seen both near the "foot" in the lower left part of the plots and near the "paws" and the "ear" in the upper right region.

10.4 Stopping Criteria

The procedure of iterative alignment is repeated until one or more stopping criteria are satisfied making the algorithm stop and start over with a new target point set.

As mentioned, after each applied transformation (R and T) the mean absolute difference ($MAD_{after\ RT}$) between the two point sets is estimated. The MAD-measure is based on randomly selected points in the target set and their geometrically closest correspondences in the reference set.
If the distance is below the expected value of the system, $dist_{expected}$, the iterations are stopped.
Additionally, if the reduction of the $MAD_{after\ RT}$ for three iterations in a row has been below a certain threshold, $dist_{reduction}$, the alignment is believed to have converged to a stationary point and the iterations are stopped.
As a safety stop, there is a limit to how many iterations the algorithm can carry out, I_{max}. This is to ensure that the algorithm doesn't enter an endless loop if the two previous criteria, for some reason, are never satisfied.

10.4.1 Estimating the New Starting Guesses

When the iterations are stopped, the resulting control point set is used to calculate the total transformation of the target point set, from its initial position in the camera coordinate system to the position of the iterated alignment. The calculated R_{global} and T_{global} denotes the global transformation of the target point set, and is stored to use it as a starting guess for the next target point set.

The transformation of the control point set between the reference point set and the aligned target point set is also calculated and denoted the local transformation (R_{local} and T_{local}) of the target point set. This is to be used as an additional starting guess of the next target point set.

10.5 Parameter Updating

After each iteration, a few parameters are updated to ensure good conditions for the convergence of the alignment.

Naturally, d_{max} is updated as a consequence of the point sets moving into closer alignment.

$$d_{\max,new} := d_{update\ factor} \cdot MAD_{after\ RT} \quad (10.18)$$

The value assigned to d_{max} is based on the $MAD_{after\ RT}$−measure, and from experiments it was found that a factor of 2 gave good results. The decreasing manner of d_{max} ensures that outliers and points outside the overlap region are not selected as control points.

In a similar manner, lambda is updated to give the texture more influence as the algorithm iterates.

$$\lambda_{new} := \lambda_{update\ factor} \cdot \lambda_{prev} \quad (10.19)$$

The value of lambda is low in the beginning, in order to let the geometry dominate in the initial aligning phase. But gradually the texture is given more control in the control point matching process, as lambda is increased. There is a maximum limit, λ_{max} , to prevent the texture from getting full control of the control point matching, which would corrupt the alignment process.

The global and local transformations are also updated to use for starting guesses for the next target point set.

$$R_{global,prev} := R_{global} \quad (10.20)$$

$$T_{global,prev} := T_{global} \quad (10.21)$$

$$R_{local,prev} := R_{local} \quad (10.22)$$

$$T_{local,prev} := T_{local} \quad (10.23)$$

10.6 Convergence

The iterations of the two point sets of the bear are seen in Figure 10.5. The plot shows the control point distances before (red) and after (blue) each applied transformation, the texture distance (green) of the control points and the lambda parameter (magenta).

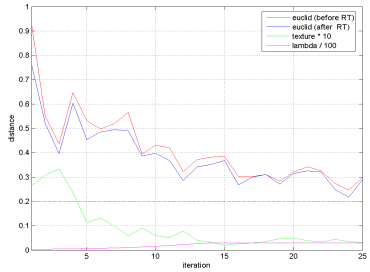

Figure 10.5: The control point distances during the iterations and the lambda parameter.

In the first iterations the control point distances decrease fast, but then the distance reduction fades out as the fine alignment stage begins. Even though the curve is jacked, due to the random nature of selecting the points, the distance clearly converges to a minimum. As lambda is increased the texture distance is also reduced, but then flattens out as lambda becomes constant. The iterations can of course be continued, but the graphs flatten out at the current levels.

Inspecting the independently calculated distance (with more points) of the two point sets after each transformation, is more interesting and seen in Figure 10.6.

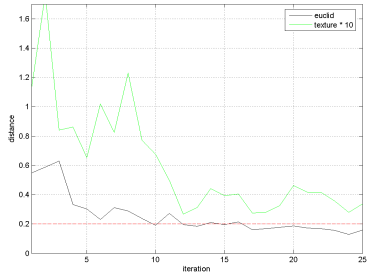

Figure 10.6: The independently calculated distance between the point sets after each transformation (black) and the corresponding texture distance (green).

The "true" distance of the point sets decrease in a more smooth fashion than the control points, and it is noticeable that it drops lower than the control point distance. This is because the independent distance is a clean Euclidean distance between the point sets and not influenced by texture in the closest point search, which also explains the somewhat larger texture differences in the closest point correspondences.
The plot indicates that for each iteration the alignment estimate gets better and the iterative alignment ends with a distance of the sets just below 0.20 mm.
In general, the presented system usually obtains a distance of 0.15-0.30 mm., when aligning consecutive point sets of smooth surfaces with characteristic features.

The iteration sequence can be seen more graphically in Figure 10.7, where the two sets are plotted in their relative position after every second transformation.

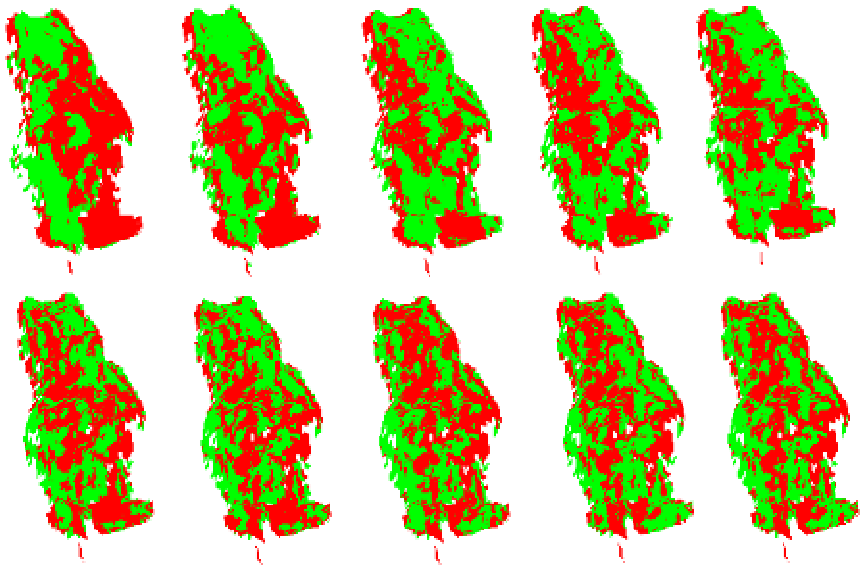

Figure 10.7: The relative position of the reference set (red) and target (green), for every second iteration from 0 (starting point – upper left) through 18 (final alignment – lower right).

In the beginning, big uniform coloured patches are present, indicating that the point sets are not well aligned. But as the iterations count, the coloured areas become more spotted indicating fine alignment of the overlapping areas. The

small coloured spots is a result of two points sets being in fine "perfect" alignment where the noise of each set wobbles on top of each other.

In Figure 10.8 the final alignment is seen, where the reference point set is textured and the target is coloured green.

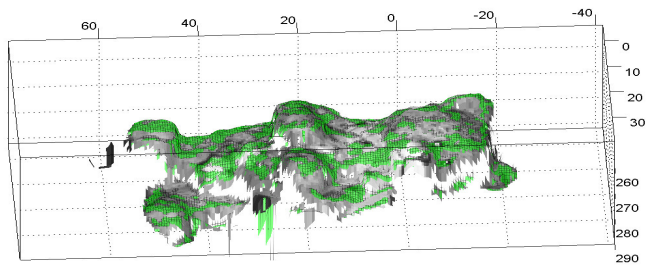

Figure 10.8: The final alignment position of the reference (textured) and the target (green).

10.7 Summary and Discussion

The outline of the implemented registration algorithm has been presented. The algorithm uses the Iterative Closest Point algorithm for aligning the point sets coming from the stereo module. The ICP algorithm relies on finding "true" control points in the two sets, minimizing an error measure of the control points, applying the found transformation and then start over in an iterative fashion.
The algorithm is based on constraints in the correspondence search and incorporates texture for handling "shapeless" objects.

The algorithm performs fine, but the convergence is rather slow, as many iterations are needed. Usually around 20 iterations are needed to reach the generally obtainable alignment distance of 0.15-0.30 mm. between two consecutive point sets.
One way to improve the performance would be to do a sort of weighting of the control points when doing the TR optimization. The weighting should be based on the certainty of the individual points, somehow calculated in the stereo matching process, and could maybe contribute with better convergence as unreliable points would have less influence.
Also, some clever way of controlling lambda would be preferable, and using scale spaced textures could be a possibility of additionally increasing the performance.

Typically a constant number of 20-30 points were used for the RT-estimation, which was found to give good results. Some adaptive scheme of using less points in the coarser alignment phases, and then increase the number as the surfaces gets closer to each other could be beneficial.

The possible build-up error from aligning only after the previous target point set is also an area that could be improved. Perhaps by using several of the last point sets or maybe even the entire model.

No real optimization of the closest point search has been done, so the efficiency could be boosted dramatically by using a Kd-tree for the closest point search and is essential in a real time implementation.

Chapter 11

Model Integration and Visualization

In order to do online preview of the rapidly growing amount of data in a real-time scanning process, there is a need to integrate this data in a fast and reasonable way, which can produce an incrementally building preview of the model. Since the CPU is busy doing registration, there is no processor time for advanced triangulation and mesh merging of the registered point sets.

This chapter describes how the registered point sets are integrated into a common 3D model and how the preview model can be visualized in an efficient way.

11.1 The Frequency Volume

The integration of the point set data has been done using a voxel grid. Each aligned data set is uniformly sampled to fit into the voxel grid coinciding with the object coordinate system. The value of each voxel is incremented for every point quantized into it. In this way, a frequency volume is obtained, which is very suitable for incremental updating.

Voxel grids of similar size are created for the normal and intensity data; the normal volume, of course having three values (x,y,z) for every voxel. As the fre-

quency of each voxel-cell is known, the normal and intensity of each voxel can be calculated as a running average.

The voxel grid size is usually set to 0.25 mm. in order to match the systems lateral sampling size in the average working distance. A grid size of 0.50 mm. can also be used if a not as heavy model is desired (only 1/8 the number of points), as it was experienced that often this was sufficient for visual purposes.

11.2 Splatting

To visualize the voxel grid model, a method referred to as splatting has been used. The method relies on the voxel grid being sufficiently dense and having a small grid size. Each occupied voxel is then rendered as a single circle or square. If the size of the individual rendered points are set right and the normals of each voxel is also applied the illusion of a smooth surface is created, see Figure 11.1.

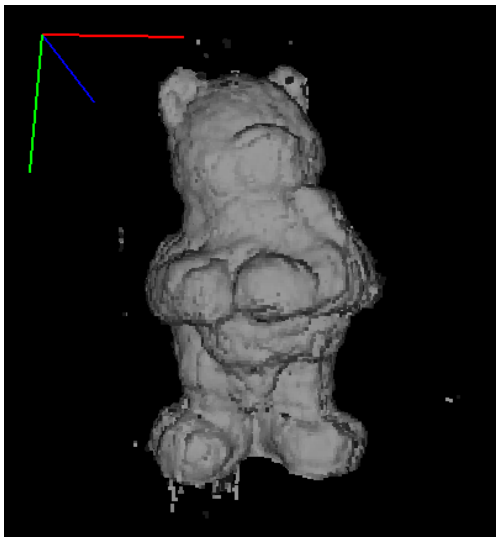

Figure 11.1: The bear model rendered with the splatting method.

The rendered model of the bear looks good and in this case the method of splatting is definitely suitable for an online preview of the incrementally building model.

Taking a closer look at the model, the structure of the voxel grid and individual point renderings can be seen, see Figure 11.2.

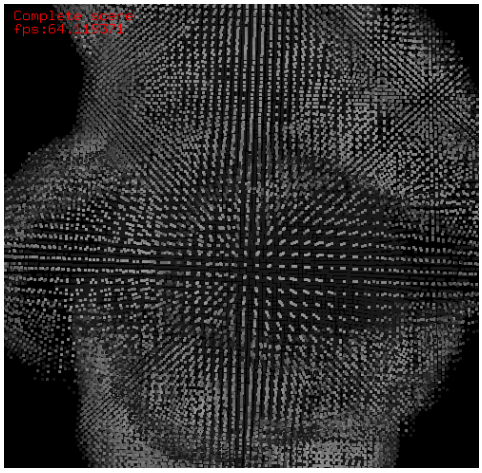

Figure 11.2: A zoom version of the bear model reveals the individual voxel renderings.

The uniform voxel grid structure can clearly be seen in the image above. So dependent on the view or zoomed distance of the model, a suitable size of the point renderings must be made. Normally the point size is set to the grid size of the voxel grid or a little higher assuring a small overlap and no holes in the rendered model.

For details regarding the method of splatting the reader is referred to the paper of Rusinkiewicz and Levoy [31].

11.3 OpenGL Visualization Software

In the model visualization software, developed in OpenGL, different features have been incorporated both to get a good rendering of the model but also to help evaluating and inspecting the errors of the models.
First of all it can show the model rendered with shading and texture. In addition there is the option of viewing the individual point sets. This is helpful when inspecting how successful the registration of the individual point sets have been and evaluating if there is drifting errors in the model.

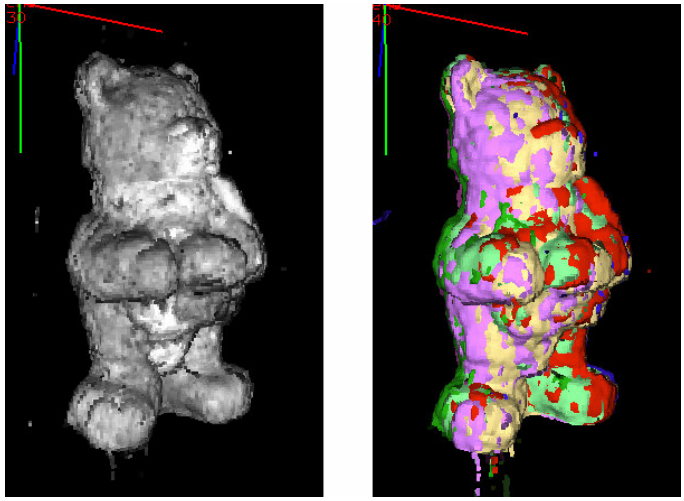

Figure 11.3: The textured and shaded model of the bear (left). The individual point sets of the model (right).

With texture added to the bear model the resemblance of the real model is even better, see Figure 11.3 (left). Despite the noise, the rendering looks like the real physical object.
In Figure 11.3 (right), the individual point sets are rendered in different colours, so they are easier separated visually. This is a strong tool for evaluating the registration results.

Part IV

Experimental Results

Chapter 12

3D Object Modelling

In this chapter the complete modelling system is evaluated. Different object are reconstructed in 3D and both the final voxel grid model but also the alignment of the point sets is examined.

12.1 The Bear

In the ICP implementation chapter the bear was used to demonstrate the algorithm. In this chapter the assembled 3D model is inspected in detail. The bear sequence consists of 35 images pairs constituting a full rotation of the object.

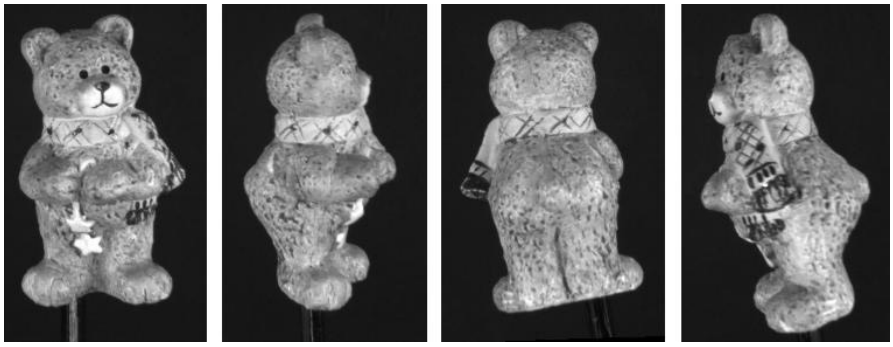

Figure 12.1: Four images from the bear sequence

To compare the real object with the 3D reconstruction, the voxel grid model is viewed from the same four angles as in Figure 12.1.

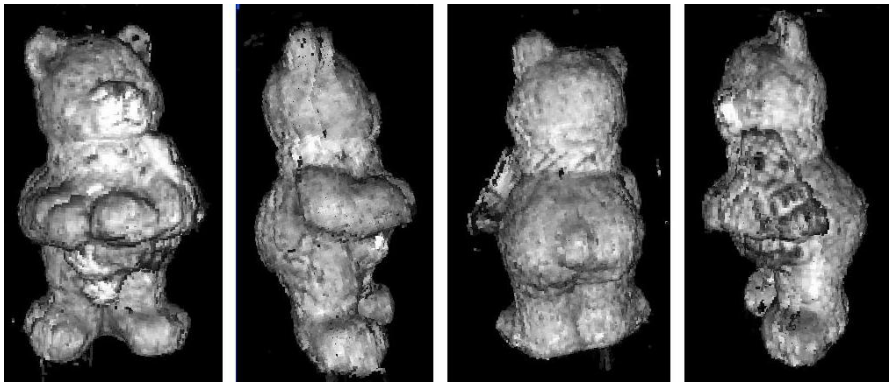

Figure 12.2: The reconstructed model of the bear in different poses.

The reconstruction looks very convincing when compared to the original images, even though some textural detail has been lost. The fine model coheres with a small registration error for all point sets of approximately 0.15-0.20 and as seen in Figure 12.3 the different point sets are positioned with nice wobbling overlaps.

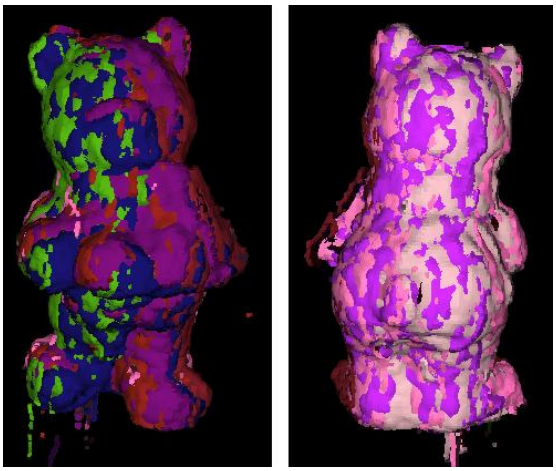

Figure 12.3: The individual point sets of the model.

However, examining the head closely in Figure 12.2 (image two from left), it can be seen that the bear has two right ears. This is due to the drift error which clearly can be seen if only the first and last point set of the sequence is plotted, as in Figure 12.4.

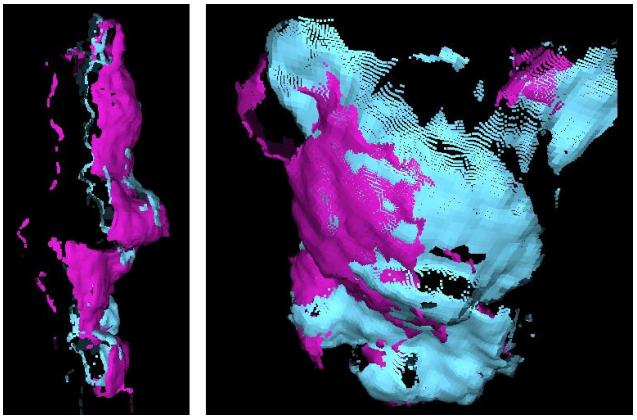

Figure 12.4: The first and last point set of the model, in profile (left) and close-up of the head and ears (right).

The two point sets are not aligned very well, resulting in some features of the bear appearing two times. The ear parts of the point sets don't coincide at all, due to the drift error.

12.2 A Small White Statue

The white statue is an object similar to gypsum or stone artefacts found in museums, and because of the very texture lacking surface, not much was expected from it. The captured image sequence is a full rotation of 44 image pairs.

Figure 12.5: The small white statue.

The reconstruction, though, actually proved to be quite good considering the lack of texture. Figure 12.6 shows four different poses of the 3D model.

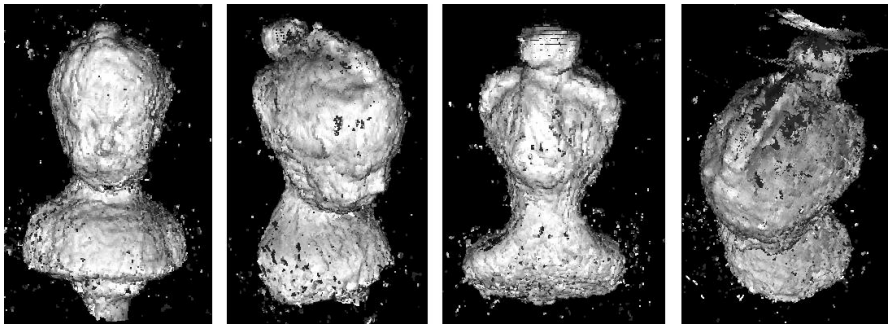

Figure 12.6: Four different poses of the reconstructed statue.

The voxel grid is really noisy in this case. It cheats a little though, as the ratio of outliers to model voxels actually is really small, the outliers just dominate when rendered directly.
Imagining a real time system the right image of Figure 12.6 would give the system operator a good view of the missing regions to scan on top of the statue.

The registration error for this model was also quite low, 0.20-0.30 mm, and this is also the impression you get when examining the point set alignments of Figure 12.7.

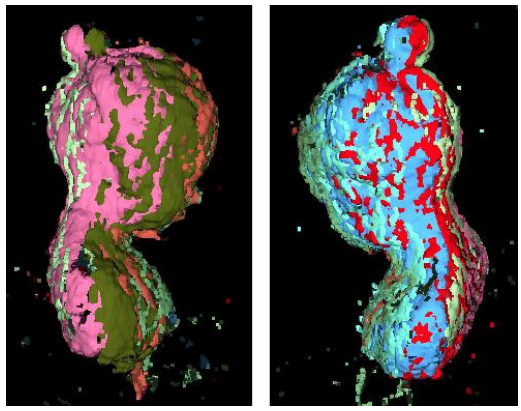

Figure 12.7: The individual point sets from two different views.

At first sight, the model doesn't seem to suffer from drifting, but examining the first and last point set, it is clear that the registration algorithm again has introduced a small error.

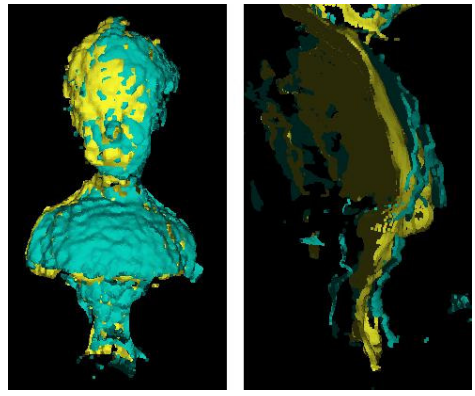

Figure 12.8: The first and last point set, front view (left) and profile of the chest region (right).

The registration looks fine in the head region, but looking at the chest area profile the misalignment from first to last point set is revealed, Figure 12.8.

12.3 Custom Made Object of Styrene Plastic

3Shape scanners [1] provide models with a point density of approximately 0.15 mm. and the points measured with an accuracy of 20 µm. Therefore it was found reasonable to use such a model as "ground truth" and compare the stereo reconstructed model with it. In order to fit into a high precision dental laser scanner from 3Shape, an object was custom made from styrene plastic and textured with paint. Below, two images from the sequence are shown.

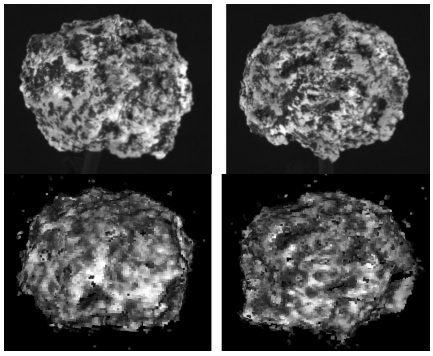

Figure 12.9: Two images of the image pair sequence (top) and the reconstructed model viewed from the same angles (bottom).

In Figure 12.9 (bottom), the reconstructed model is viewed from the same angles. As no real structures are present on the object, resemblance is hard to see, but looking closely will reveal that the model matches the images.

Looking at the individual point sets, see Figure 12.10, the alignment looks successful as indicated from the 0.5 mm. registration error of each point set.

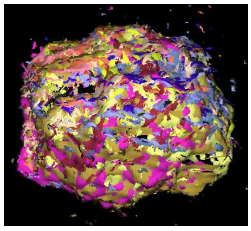

Figure 12.10: The individual point sets.

The comparison with the 3Shape model was done in the following way: The first point set of the reconstructed model is ICP-aligned to the 3Shape model, see Figure 12.11. Afterwards all other point sets are placed in their respective relative position to the first point set. For each point set, the distance to the 3Shape model is then calculated as the average Euclidean distance for a given number (1000) of points to the closest point in the 3Shape model.

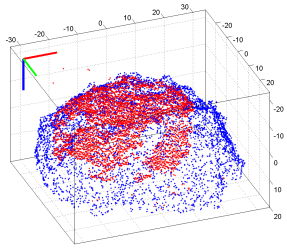

Figure 12.11: The decimated 3Shape model (blue) and the decimated first point set (red).

The results are seen in Figure 12.12.

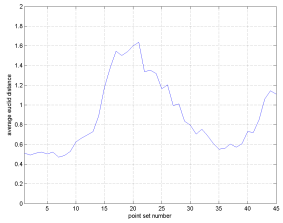

Figure 12.12: The average Euclidean distance from each point set in the model to the 3Shape model.

The average error of the first few point sets is approximately 0.5 mm. This is better than the expected 1mm. (perturbation: $\delta<½$), but as the point set number increases, so does the error distance. It is curious that it increases and decreases again, and the reason was found as it became obvious that the reconstructed model was scaled a little compared to the ground truth. It shows as the error is at minimum again at point set number 35, which is where the object has done a full rotation and is back where it lies close to the 3Shape model. This of course indicates the registration isn't that bad, despite the scaling. The reason for the scaling is maybe due to calibration issues.

12.4 A Round Pot

This sequence is of a textured pot which was rotated a little slower for the texture ICP to work. It contains 60 images constituting a full rotation.

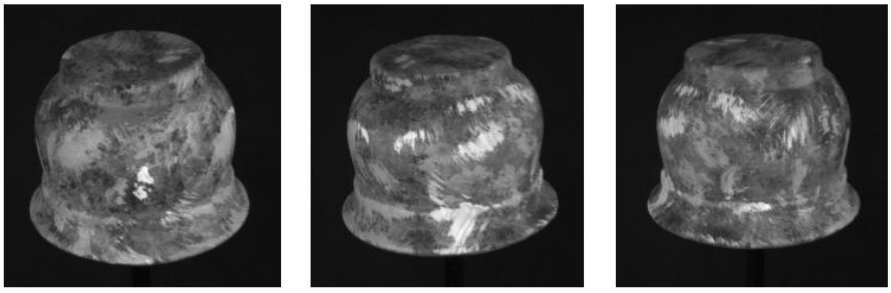

Figure 12.13: Three images from the pot sequence.

To begin with, the reconstructed model, with lambda set to 0 in the registration, is shown in Figure 12.14

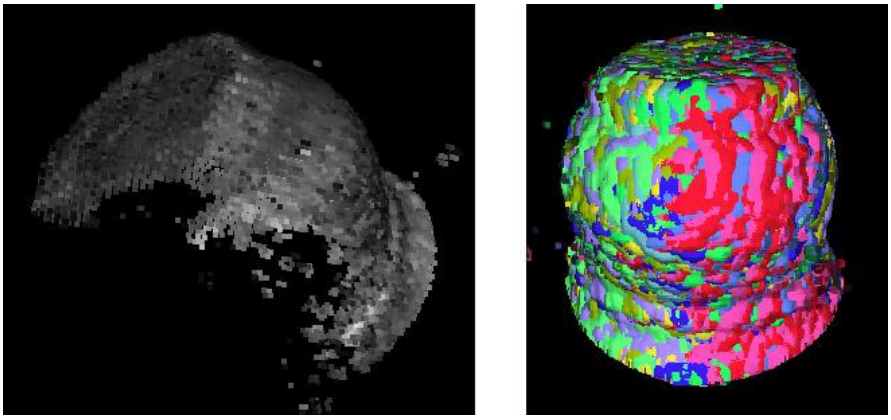

Figure 12.14: The reconstructed pot (left) and the individual point sets (right).

As the object is rotationally symmetric, the registration fails and the reconstructed model only consists of collapsed point sets. This example shows why the texture information is important, when there is a lack of geometric features in the object topology.

The results with lambda in its normal setting is shown below.

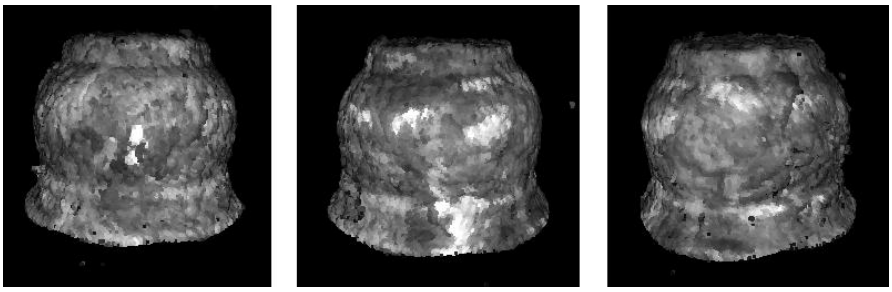

Figure 12.15: Three views of the reconstructed model, corresponding to the views in Figure 12.13.

The results using texture in the registration looks much better.

The pot could just fit into the 3Shape scanner, so a ground truth model was also provided. The procedure is the same as with the styrene plastic object and the point set distances are shown in Figure 12.16.

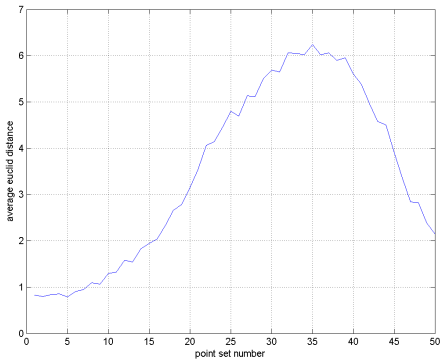

Figure 12.16 The average Euclidean distance from each point set in the model to the 3Shape model.

The tendency is the same, even though the measured distance of the ICP-aligned first point set is just below 1 mm.
The point set of the maximum distance is plotted in Figure 12.17.

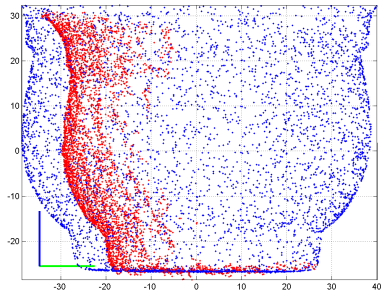

Figure 12.17: The 3Shape ground truth model (blue) and the point set of image 35 (red).

It is seen that the point set of image 35 lies inside the ground truth model as a result of the scaling of the reconstructed model.

12.5 Summary

The combined 3D modelling system has been tested on a variety of objects. The reconstructions of the tested models are of reasonable quality and can definitely be recognized even though some of them are surrounded by a lot of noise.

Some problems with a scaling of the model occurred though, but maybe this can be corrected by performing a more advanced calibration of the cameras as discussed in the calibration chapters.

Considering the results, stereo vision definitely has possibilities as a serious 3D modelling tool and from the rendered models it has also been proven that a simple integration of the captured range data, very well can act as an online preview of the scanned model.

Chapter 13

3D Face Modelling

To test the system for eventual use in the biometric world of face recognition, 3D face modelling was tried out.

Figure 13.1: The first, middle and last image of the face sequence.

A sequence of 12 images was captured of the authors head doing a 90 degrees pan. The neck ears and shoulders were removed manually from the images, to avoid problems in the registration as a consequence of a not 100% rigid object.

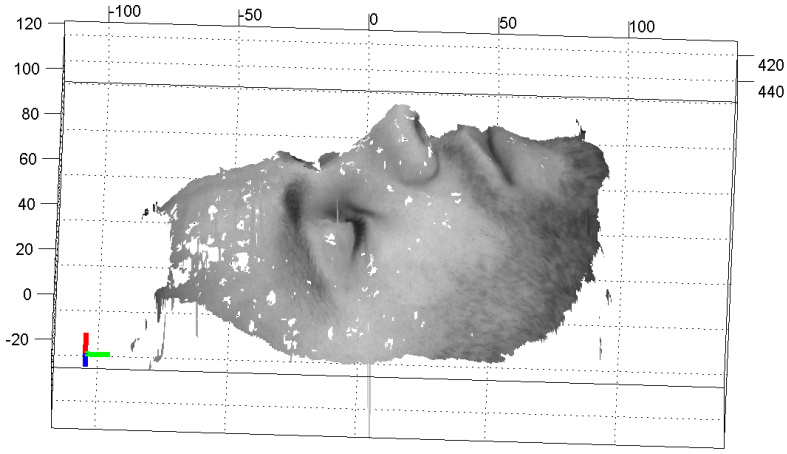

Figure 13.2: The Range data of one of the image pairs.

As seen in Figure 13.2, the 3D reconstruction was very successful even though it wasn't believed that the human skin contained sufficient texture.

The alignment of the consecutive face scans also went well, with a registration error of 0.5 mm between consecutive point sets.

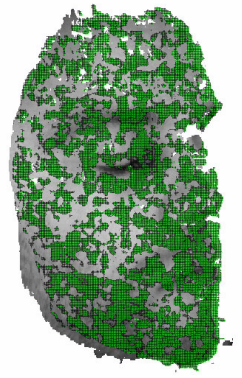

Figure 13.3: The registration result of two consecutive point sets.

The complete face model of the integrated face scans is seen in Figure 13.4.

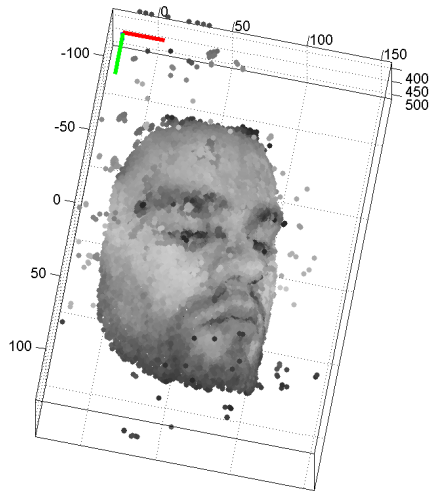

Figure 13.4: The raw voxel grid volume having integrated all 12 point sets.

The voxel grid is of course a little noisy, as nothing has been done to remove the outliers. Still, the integrated model is of good quality and strongly resembles the real face of the author, as seen in the different poses of the 3D face model in Figure 13.5.

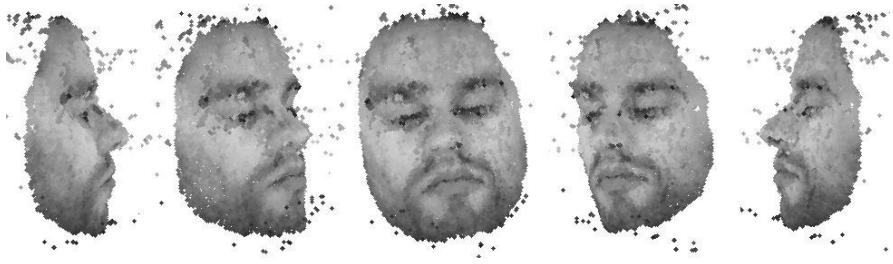

Figure 13.5: The integrated 3D face model viewed from five different angles.

The achieved results are really good, and it is believed that the quality is sufficient for 3D face recognition algorithms.

Part V

Discussion

Chapter 14

Future Work

This chapter discuss the areas where the algorithm needs improvement and the challenges of building a fully operational system for 3D modelling.

14.1 Algorithm Improvements

To provide a better model preview, either the individual range maps or the integrated model needs some noise elimination. A way to eliminate some outliers could be to exploit the frequency volume structure of the voxel grid, and for example only render the voxels which have been visited a certain number of times. This of course requires that the point sets are sampled into the exact same voxels and thereby sets hard demands for the registration algorithm, alternatively the grid size could be increased so it is certain that the point sets fall into the same voxels, but then the resolution of the model preview is sacrificed.

A way to improve the registration algorithm would be to introduce a weighted error minimization when estimating the optimum transformation of control points. The weighting should be based on the reliability of the points, which could be measured in the stereo matching process, either in connection with cross checking or simply based on the cost measure of the pixel.

Also in connection with the registration a kd-tree is necessary to speed up the closest point search.

If the algorithms were extended to handle colour images, it is also believed that the performance could be increased significantly. The range perception module and also the registration stage would draw big benefits from this extra information and presumably give better results. But of course it also triples the computational tasks of the stereo matching.

14.2 Real-Time Implementation

The next step of the project is a full implementation in C++ for the system to run real-time. The stereo matching can be effectively implemented in OpenGL on the graphics card, while the CPU can handle disparity map refinement, triangulation and registration.
With a running system the incrementally building real-time preview of the scanned object could be tested thoroughly.

14.3 High Quality Offline Rendering

The system only produces a coarse preview model, so to achieve a high quality model some heavy offline processing has to be done. The voxel grid itself is usually discarded but the transformations of each set of range data is kept to act as starting guesses of a global optimization. As a consequence of the online preview during the scan, the user is certain that all necessary data has been acquired and with the initial starting guesses, the offline rendering can run automatically. Several methods are commonly used.
Pulli [28] proposes a global multiview registration of all range data. As control points between the point sets are known (perhaps minor adjustments are needed) a global optimization of the set positions can be done instead of the alignment to the previously set which causes the drifting error.
When the point sets have reached a global equilibrium they can be delaunay triangulated and merged, as for example Turk and Levoy proposes [40]. The result is a common triangle mesh constituting the final model.
Another method is proposed by Curless and Levoy [8], who do volumetric integration of the range data and then estimates the final model by extracting an iso surface from the volume grid.

In an eventual real-time system it would also be beneficial to make use of all the redundant data collected during the scanning process. Surely it is an advantage

that the same region has been scanned several times, so this can be used to reduce the noise in the model.

As stereo range data always have some geometric distortion due to texture characteristics, the range map could be refined in an iterative deformation process. By shifting pixel-depths in a random fashion, based on the back projection error, a more correct range map could be obtained. This should of course be done prior to the global multi-view registration.

Chapter 15

Discussion

15.1 Summary of Main Contributions

The main objectives of this thesis were to find answers to the following questions:

- Can the stereo method provide range data of sufficient quality, to be used for 3D modelling?

- Would it be possible to produce a real-time preview of the model with sufficient quality to act as an online view-planning tool, during the scan process?

To answer these questions, a system for offline simulation has been built.

The system consists of a stationary stereo setup with two relatively high resolution cameras. Image pair acquisition software has been developed for real time image sequence capturing.
A stereo matching method for calculating the 3D data of the object has been implemented. The implementation is widely based on existing area-based methods of combining different support region levels, in order to combine robustness and precision.

For aligning the 2.5D range data into a common coordinate system, the ICP-algorithm has been used. The implemented registration algorithm achieves precise alignment of the data by, among other factors, using the texture of the objects and excluding possible outliers from the optimum transformation estimation.
Finally the aligned range data is integrated in a voxel grid suitable for incremental updating of new data.
The visualization is done as point splatting of every occupied voxel in the volume.

From experiments with the system, several kinds of objects were reconstructed in a successful manner. Even though a little noisy, the visual resemblance of the true objects are striking, and one have to remember that the models are meant as initial guesses to a high quality offline rendering. So, if models of the achieved quality can be obtained from a simple stereo algorithm, local registration and no significant noise reduction, then:

Yes, stereo can provide range data of sufficient quality to be used for 3D modelling.

And, since all algorithms have been implemented to suit real time purposes, then:

Yes, the quality of a 3D modelling system based on stereo vision, can provide a real time preview of the scanned model, in order to act as an online view-planning tool.

Having concluded on the main objectives, a 3D modelling system of this type has not been presented before, as far as it is known to the author. So it would be interesting to continue with the propositions discussed in the chapter of future work, in order to see exactly how much can be achieved with stereo vision as a 3D modelling tool.

15.2 Conclusion

In the future there will be a high demand for cheap and flexible 3D scanners. The next generation of scanners must be real time and provide an online preview on the scanned model in order for the operator to determine what regions are missing.
The stereo vision method meets the requirement of cheapness and with recent advances in image processing on graphics hardware, stereo matching can also be done real time now.

Through experiments, this thesis has proved that stereo vision to a great extent is suitable for 3D modelling, and that it also can meet the demand of a relatively high quality real time preview, when performing the scanning process.

With an achieved measuring accuracy in the order of 1 mm. (working distance=0.5 m.) assuming a well textured object, the system would definitely be suitable for 3D modelling of e.g. the initial clay models of CG rendered movie characters. The method would also present great advantages in for example creating virtual environments of urban scenarios or in the biometric field of 3D face recognition.

As the proposed 3D modelling method is based on stereo vision, eventually it will be a cheap way for the amateur sculptor to acquire a high quality 3D scanner, as it only requires e.g. two web cameras. The software could then be purchased on the internet and calibration done with a home printed checker-board. Of course it would take a lot of further development, but the possibility is there.

It is the conclusion of this thesis that stereo vision definitely has a big role to play in the future generation of low cost 3D scanners. The technologies to build a fully functional stereo vision-based 3D modelling system are there, they only have to be merged into a complete system.

Appendix A

Derivation of Stereo Triangulation Formulas

Given a fronto parallel stereo setup, as seen in Figure 16.1 (in 2D), we wish to calculate the three dimensional coordinates of *W* with respect to the world coordinate system (which has reference in the left lens centre). The left and right image has a local world coordinate system in their respective lens centers, which is indicated by the indices *l* and *r*, when referring to local world coordinates.

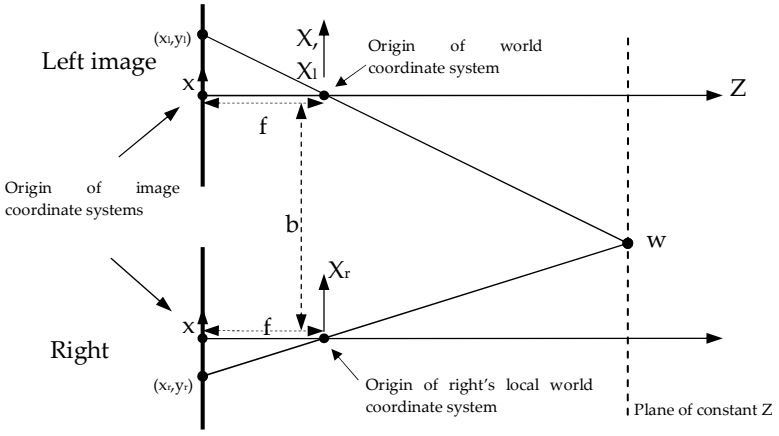

Figure 16.1: A 2D view of the geometry in fronto parallel stereo vision.

The calculations are based on w's projection in the left and right image, w_l and w_r, respectively.

$$w_l = \begin{bmatrix} x_l \\ y_l \end{bmatrix} \quad (16.1)$$

$$w_r = \begin{bmatrix} x_r \\ y_r \end{bmatrix} \quad (16.2)$$

$$W = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (16.3)$$

The relationships along the optical axes are as follows:

$$\frac{Z_l}{X_l} = \frac{-f}{x_l} \quad (16.4)$$

$$\frac{Z_r}{X_r} = \frac{-f}{x_r} \quad (16.5)$$

The point relationships hold:

$$X_l = X_r - b \quad (16.6)$$

and

$$Z_l = Z_r = Z \quad (16.7)$$

Substituting (16.4) and (16.5) in (16.6), Z can be isolated by using (16.7):

$$\begin{aligned} Z_l \frac{x_l}{f} &= Z_r \frac{x_r}{f} + b \\ Z \frac{x_l - x_r}{f} &= b \\ Z &= \frac{b \cdot f}{x_l - x_r} \end{aligned} \qquad (16.8)$$

(16.8) can now be substituted in (16.4) and *X* can be isolated:

$$\begin{aligned} X = X_l &= \frac{x_l}{f}\left(\frac{b \cdot f}{x_l - x_r}\right) \\ X &= \left(\frac{b \cdot x_l}{x_l - x_r}\right) \end{aligned} \qquad (16.9)$$

and by axis symmetry it holds that:

$$\begin{aligned} Y = Y_l &= \frac{y_l}{f}\left(\frac{b \cdot f}{x_l - x_r}\right) \\ Y &= \left(\frac{b \cdot y_l}{x_l - x_r}\right) \end{aligned} \qquad (16.10)$$

Setting $d=(x_l-x_r)$ we have:

$$W = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{b \cdot x_l}{d} \\ \frac{b \cdot y_l}{d} \\ \frac{b \cdot f}{d} \end{bmatrix} = \frac{b}{d} \begin{bmatrix} x_l \\ y_l \\ f \end{bmatrix} \qquad (16.11)$$

Bibliography

[1] 3Shape, 3D Scanners. http://www.3shape.com/fla_index.aspx

[2] P. Anandan. A Computational Framework and an Algorithm for the Measurement of Visual Motion. *International Journal of Computer Vision*, pages 283-310, vol. 2, no. 3, 1989.

[3] K. S. Arun, T. S. Huang, S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 5, pp. 698-700, Sep. 1987.

[4] P. J. Besl, N. D. McKay. A method for registration of 3D Shapes. *IEEE Pattern Analysis and Machine Intelligence*, 14 (2), pages 239-256, 1992.

[5] Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/

[6] Y. Chen, G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10 (3), pages 145-155, 1992.

[7] S. D. Cochran, G. Medioni. 3-D. Surface Description from Binocular Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 981-994, vol. 14, no. 10, October 1992.

[8] B. Curless, M. Levoy. A Volumetric Method for Building Complex Models from Range Images. *Proc. ACM SIGGRAPH '96*. 1996.

[9] Cyberware, Ear Impression 3D Scanner, Model 7G. http://www.cyberware.com/products/m7gInfo.html

[10] The Digital Michelangelo Project: 3D Scanning of Large Statues. http://graphics.stanford.edu/papers/dmich-sig00/

[11] FastSCAN Digital Scanners. http://www.polhemus.com/fastscan.htm

[12] J. Heikkilä, O. Silvén. 'A four-step camera calibration procedure with implicit image correction. *Proc. CVPR '97*, pages 1106-1112, IEEE, 1997.

[13] B. K. P. Horn. Closed-form solution of absolute orientation using unit quarternions. Journal of the Optical Society of America, vol. 4, pages 629-642, April 1987.

[14] B. K. P. Horn, H. M. Hilden, S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. Journal of the Optical Society of America, vol. 5, pages 1127-1135, July 1988.

[15] T. Jaeggli, T. P. Koninckx, L. V. Gool. Online 3D Acquisition and Model Integration.

[16] A. E. Johnson, S. B. Kang. Registration and integration of textured 3D data. *Image and Vision Computing*. Vol. 17. pages 135-147. 1999.

[17] T. Kanade, M. Okutomi. A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 920-932, vol. 16, no. 9, September 1994.

[18] J. Kim, J. Park. New Stereo Matching and 3D View Generation Algorithms using Arial Stereo Images. *Proc. 12th Int. Conf. on Geoinformatics*, pages 663-669, June 2004.

[19] Konica Minolta Scanners. http://www.konicaminolta-3d.com

[20] M. Levoy, et aller. "The Digital Michelangelo Project: 3D Scanning of Large Statues", Proc. SIGGRAPH '94, ACM, 2000, pp. 131-144.

[21] X. Lu, D. Colbry, A. K. Jain. Matching 2.5D Scans for Face Recognition. *Proceedings of ICBA*, pages 30-36, Hong Kong, China, July 2004.

[22] X. Lu, A. K. Jain. Integrating Range and Texture Information for 3D Face Recognition. *Proc. IEEE of WACV*, Breckenridge, Colorado, 2005.

[23] F. C. M. Martins, H. Shiojiri, J. M. F. Moura. 3D-3D registration of free formed objects using shape and texture. *SPIE Symposium on Electronic Imaging, Science and Technology*. San Jose, California, February 1997.

[24] Middlebury College's Stereo Vision Research Page. http://cat.middlebury.edu/stereo/data.html

[25] NASA, Jet Propulsion Laboratory, California Institute of Technology. Mars Exploration Rover Mission. http://marsrovers.nasa.gov/gallery/video/animation.html

[26] Optometrists Network, 3D Vision. http://www.vision3d.com/stereo.html

[27] Point Grey Research. http://www.ptgrey.com

[28] K. Pulli. Multiview Registration for Large Data Sets. *Proc. 3DIM*, 1999.

[29] S. Rusinkiewicz, O. Hall-Holt, M. Levoy. Real-Time 3D Model Acquisition.

[30] S. Rusinkiewicz, M. Levoy. Efficient Variants of the ICP Algorithm, *Proc. 3DIM*, 2001.

[31] S. Rusinkiewicz, M. Levoy. QSplat: A Multiresolution Point Rendering System for Large Meshes. SIGGRAPH 2000.

[32] D. Scharstein, R. Szeliski. High-accuracy stereo depth maps using structured light. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003),* volume 1, pages 195-202, Madison, WI, June 2003.

[33] C. Sun. Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques. *International Journal of Computer Vision*, pages 99-117, vol. 47, May 2002.

[34] R. Szeliski, D. Scharstein. Sampling the Disparity Space Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. pages 419-425, vol. 25, no. 3, March 2004.

[35] R. Szeliski, D. Scharstein. Symmetric Sub-Pixel Stereo Matching. *Proceedings of the 7th European Conference on Computer Vision-Part II*, pages 525-540, May 2002.

[36] R. Szeliski, D. Scharstein. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, pages 7-42, vol. 47 (1/2/3), 2002.

[37] K. Takita, M. A. Muquit, T. Aoki, T. Higuchi. A Sub-Pixel Correspondence Search Technique for Computer Vision Applications. *IEICE Trans. Fundamentals,* vol. E87-A, no. 8, August 2004.

[38] Q. Tian, M. N. Huhns. Algorithms for Subpixel Registration. *Computer Vision, Graphics and Image Processing*, pages 220-233, vol. 35, 1986.

[39] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation RA-3(4)*, pages 323-344, 1987.

[40] G. Turk, M. Levoy. Zippered Polygon Meshes from Range Images. *Proc. ACM SIGGRAPH '94*. 1994.

[41] TYZX – systems that see. http://www.tyzx.com

[42] R. Yang and M. Pollefeys. Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 211–218, 2003.

[43] R. Yang, M. Pollefeys, and S. Li. Improved Real-Time Stereo on Commodity Graphics Hardware. In *IEEE Workshop on Real-time 3D Sensors and Their Use (in conjunction with CVPR'04)*, 2004.

[44] R. Yang, G. Welch, and G. Bisop. Real-Time Consensus-Based Scene Reconstruction Using Commodity Graphics Hardware. *Proceedings of Graphics 2002*, pages 225-234, Beijing, China, October 2002.

[45] Z. Zhang. Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. *International Conference on Computer Vision (ICCV'99)*, pages 666-673, September, 1999.