Real-Time Registration of Video with Ultrasound using Stereo Disparity

Submitted in partial fulfillment of the requirements for

the degree of

Master of Science (MS)

in

Biomedical Engineering

Jihang Wang

B.S., Biomedical Engineering, Northeastern University (China)

Carnegie Mellon University
Pittsburgh, PA

May, 2012

Abstract

Enabling an ultrasound machine to scan the patient with an additional, new view of the external anatomy's 3D color surface combined with the traditional interior ultrasound image has the potential to revolutionize many areas such as plastic surgery and eye surgery that may benefit from visualization of ultrasound registered with the patient's surface. Critical to the implementation of this kind of visualization technique, in general, are the capabilities to accurately acquire the 3D point clouds of the exterior surface along with high resolution ultrasound data and then to register them properly into one coordinate system. This report proposes an image fusion device called Probe-Sight to realize the above functionality by mounting two video cameras on an ultrasound probe in a lightweight aluminum frame.

The motivation of the Probe-Sight device is discussed first, including its initial application of giving plastic surgeons better understanding of nerve regeneration after surgery. Some background knowledge for augmented reality and an early version of Probe-Sight are discussed next in the literature review. The report then demonstrates how Probe-Sight's stereo vision algorithms work by introducing the foundation of stereopsis including stereo camera calibration, epipolar geometry and stereo rectification. Different algorithms for matching the stereo points are then discussed, followed by our implementation using GPU based methods from the Open-source Computer Vision library (OpenCV). In addition, our image rendering framework is introduced, which registers both 3D exterior information and interior ultrasound data together in real time.

We report on the successful operation of our device, demonstrating a 3D rendering of an ultrasound phantom's surface with the ultrasound data superimposed at its correct relative location. Eventually, automated analysis of these registered data sets may permit the scanner and its associated computational apparatus to interpret the ultrasound data within its anatomical context, much as the human operator does today.

Acknowledgements

My life over the last two years of graduate school has been fantastic. I have many people to thank for this, and I am afraid I can only list a few here.

First I must thank my advisor, Dr. George Stetten, for the two years of mentorship. His visionary advice and constant encouragement meant a lot to me. He not only taught me lot of knowledge that I can't learn from the text book but also helped me to build the scientific research concept. He also helped me to write my first scientific paper and gave me lots of suggestions and revisions.

I want to also thank Dr. John Galeotti, who is a Senior Research Scientist in Robotic Institute. He was directly responsible for my project and gave me a lot of help in solving both software and hardware issues. I also learned useful knowledge about image processing and computer vision in his course *Methods in Medical Image Analysis*, which brought me many benefits in my later research.

During my times in graduate school, I had the opportunity work with and learn from many great people. I would especially like to thank Damion Shelton, Mel Siegel, Samantha Horvath, Ada Zhang, Vikas Shivaprabhu and Randy Lee, my collaborators and friends in the lab.

I am grateful to many of my friends in Pittsburgh. There are too many to name, but I would particularly thank Yuanfeng Jiao, who is another M.S student in BME department in CMU. I shared two years of companionship and laughter with him. We talked about many things in scientific research and advanced technology in our fields. We also helped each other to solve different kinds of problems in both research and life.

I would like to express gratitude to my parents, my grandpa, and my sister for many years of unconditional love and support. Though we are ten thousand miles away from each other, they were always with me when I needed them. Finally, I would like to thank Yue Zhong, my girlfriend. Thank you for your love, trust, and understanding. Thank you for giving me with the encouragement to pursue my dreams. I hope to stay with you soon in the near future.

Contents

List of Figures

Abbreviations

AR	**A**ugmented **R**eality
BM	**B**lock **M**atching
CSBP	**C**onstant **S**pace **B**elief **P**ropagation
FLTK	**F**ast**, L**ight **T**ool**k**it
GC	**G**raph **C**ut
GPGPU	**G**eneral **P**urpose **GPU**
GPUs	**G**raphics **P**rocessing **U**nit**s**
GUI	**G**raphical **U**ser **I**nterface
GVD	**G**raft **V**ascular **D**isease
NCC	**N**ormalized **C**ross-**C**orrelation
OpenCV	**Open** Source **C**omputer **V**ision Library
OpenGL	**Open G**raphics **L**ibrary
SAD	**S**um of **A**bsolute **D**ifferences
SGBM	**S**emi **G**lo**B**al **M**atching
SSD	**S**um of **S**quared **D**ifferences
VR	**V**irtual **R**eality

Symbols

$k1, k2, k3$	Lowercase $k1, k2, k3$ denotes the different orders of radius distortion coefficients
$p1, p2$	Lowercase $p1, p2$ denotes the different orders of tangential distortion
$\mathrm{R_r}, \mathrm{R_l}$	Uppercase R_r or R_l denotes the rotation matrix for either the right or left camera which transfers the coordinate system from 3D world coordinates to 3D camera coordinates
$\mathrm{R_{nr}}, \mathrm{R_{nl}}$	Uppercase $\mathrm{R_{nr}}, \mathrm{R_{nl}}$ denotes the rectified rotation matrix of each camera
$\mathrm{T_r}, \mathrm{T_l}$	Uppercase $\mathrm{T_r}$ or $\mathrm{T_l}$ denotes the translation matrix which transfers the coordinate system from 3D world coordinates to 3D camera coordinates
A_r, A_l	Uppercase A_r, A_l denotes the internal matrix of each camera
A_{nr}, A_{nl}	Uppercase A_{nr}, A_{nl} denotes the rectified internal matrix of each camera
P_r, P_l	Uppercase P_r, P_l denotes the perspective projection matrix for each camera
P_{nl}, P_{rl}	Uppercase P_{nl}, P_{rl} denotes the rectified the perspective projection matrix for each camera
O_r, O_l	Uppercase O_r, O_l denotes the center of each camera
P, p	Uppercase P denotes the point on world (object) coordinate system while Lowercase p denotes the point on camera coordinate system
e_r, e_l	Lowercase e_r, e_l denotes epipolar line for each of the images
R, T	Uppercase R, T denotes the rotation and translation matrix between two cameras
E, F	Uppercase E, F denotes the essential and fundamental matrix between two cameras
$R_\mathrm{r}, R_\mathrm{l}$	Uppercase $R_\mathrm{r}, R_\mathrm{l}$ denotes the rectified matrix for each camera
r_r, r_l	Lowercase r_r, r_l denotes the rectified coplanar matrix for each camera
R_{rect}	Uppercase R_{rect} denotes the rectified coplanar matrix between two cameras
$\mathrm{x}_r, \mathrm{x}_l$	Lowercase $\mathrm{x}_r, \mathrm{x}_l$ denotes the horizontal positions of the points in the right or left image

Chapter 1

1. Introduction

1.1 Motivation

Diagnostic sonography (ultrasonography) is an ultrasound-based diagnostic imaging technique used for visualizing subcutaneous body structures including tendons, muscles, joints, vessels and internal organs for possible pathology or lesions. In the history of ultrasound, many innovations have been developed since ultrasound was established as a medical image device in the 1960s, through the innovation of mechanical scanning. The first real-time ultrasound was developed in the late 1970s by placing three transducers on a rotating wheel, which allowed the system to keep track of where the transducer was pointing within the image plane. This early real-time B-mode led to a more rapid adoption of ultrasound as an established medical imaging technique. In the 1980s, flow analysis tools came to maturity through color flow imaging and quantitative Doppler modes. In 1990s, people made significant improvements in image quality with the introduction of real time compound techniques and harmonic imaging. Ultrasound has many advantages including its non-ionizing properties, low cost, and most importantly real time capabilities. Ultrasound has significantly impacted clinical segments within radiology, obstetrics, vascular or cardiology and created new markets of emergency medicine and intervention [I. Minin et al., 2010].

A serious limitation to ultrasound, in general, remains its local 2D nature. After the scan is done, the exact anatomical location of the scan at each moment in time is no longer available. External tracking of the probe has been used to record location. Figure 1.1 shows a 3D ultrasound system that uses an optical position sensor and a calibrated camera to track the probe's position.

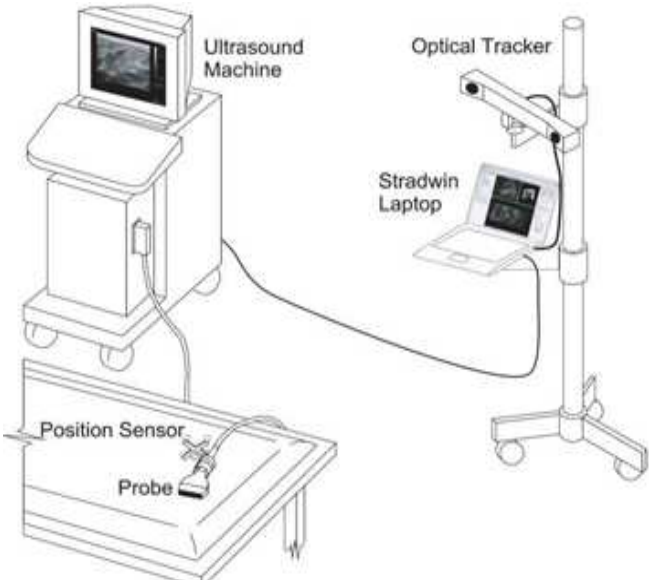


Figure 1.1: A 3D ultrasound system using an optical position sensor. (Three spheres attached to the probe are tracked by calibrated cameras. The laptop records the 2D ultrasound images together with the position and orientation of the probe at the point which each image was acquired)

(Image from R. Prager's paper *Three-dimensional ultrasound imaging*, 2010.)

However, this system has the major drawback of requiring the clinician to maintain an uninterrupted line of sight between the cameras and the tracked objects attached to the probe, which is so difficult to maintain [R. Prager et al., 2010]. 3D ultrasound does exist, but suffers either from low resolution for real time 3D scanners or slow scan rates for higher resolution 3D systems. Allowing an ultrasound probe to see both the exterior and interior of the patient may yield new applications in many medical fields. Camera-based computer vision is a fascinating topic nowadays, but despite decades of research and despite all the improvements made in the industry, computer vision has not yet been used much

in the medical field. Accordingly, our lab has developed a device called Probe-Sight, which enhances a 2D, high-resolution ultrasound probe with video cameras and computer vision. ProbeSight is capable of automatically determining the 3D position in anatomical coordinates of each 2D slice it acquires, without requiring a continually unobstructed line of sight to an external tracker. ProbeSight's real-time stereo vision algorithm and real-time rendering technique allow for interactive use.

1.2 Initial Application to Upper Extremity Transplantation

Based on a 2007 report from the National Limb Loss Information Center, it is approximately 1.7 million people were living with limb loss in the United States [National Limb Loss Information Center, 2007], and this number will reach 3.6 million by 2050. In the year 2005, 41,000 persons in the United States were living with major upper-limb loss, and 62 percent of them had trauma-related injuries [K. Ziegler-Graham et al., 2008]. Prosthetic rehabilitation of function may be suboptimal in cases of devastating hand or forearm loss, but hand transplantation could restore the appearance, anatomy, and function of a native hand better than any other reconstruction. Upper extremity transplantation is an innovative reconstruction approach with the potential of immediate clinical application. For select amputees it allows the most near-term pay-off with reintegration into employment and society [V. Gorantla et al., 2011]. There are only a few centers in the US and abroad that practice this surgery. Up to now, more than 50 hand transplants in more than 37 recipients have been performed in centers in Europe, the United States, and in China, and all were technically

successful [P. Petruzzo et al., 2008]. The complex surgery procedure (which includes attachment between the donor and recipient of bones, tendons, nerves, vasculature and other tissues) represents an example in the rapidly growing field of Reconstructive Transplantation, which includes transplantation of the face, scalp, abdominal wall, femur, larynx and other vascularized composite organs. The early experience with upper extremity transplants around the world has confirmed that this procedure is not only feasible but that it also represents a valuable option to reconstruct limb loss secondary to devastating traumatic injuries as seen in military service members and veterans. However, several key issues still remain in urgent need of a solution.

1.2.1 Nerve Regeneration:

The nervous system is divided into the central nervous system, which consists of the brain and spinal cord, and the peripheral nervous system, which consists of cranial and spinal nerves along with their associated ganglia. There is currently no treatment for recovering human nerve function after injury to the central nervous system [J. Recknor et al., 2006]. However, evidence shows that the peripheral nervous system is capable of regeneration, which makes hand transplantation possible. In addition, although the peripheral nervous system has the capability for regeneration, much research still needs to be done to optimize the environment for maximum regrowth potential, which is the key to success in upper extremity transplantation. Unlike solid organ transplants such as the kidney, liver, or heart that are immediately functioning after revascularization, a composite tissue allograft is viable after revascularization of the graft but not

functional [G. Brandacher et al., 2010]. The recipient nerves/axons have to regrow and replace the donor nerves, which serve as a temporary scaffold. This growth occurs at a rate of about 1 mm per day, which corresponds to roughly 1 inch per month or 1 foot per year. Once the axons regrow back into the denervated muscles, the muscles will begin to function again. During the months or years it takes for the axons to regrow into the muscles, the muscles will be paralyzed and will atrophy. Unfortunately there is very limited data on nerve regeneration in this context since nerves cannot be biopsied during follow up. Therefore, non-invasive monitoring of neuroregeneration after transplantation is needed both for evaluation of the efficacy of reinnervation and for implementation of treatment strategies that may have important implications in recovery and outcome.

1.2.2 Chronic Rejection:

Chronic rejection is the major cause of graft loss in renal transplantation. It is less well defined than either hyperacute or acute rejection and is probably caused by multiple factors: antibodies as well as lymphocytes. Across the spectrum of transplanted organs, the phenotype of chronic rejection is varied. However, obliterative arteriopathy is a consistent and major finding, and chronic rejection is therefore better defined as graft vascular disease (GVD). The diagnosis has been confirmed by recognizing obliterative arterial changes, including luminal narrowing and concentric initimal hyperplasia. According to a previous study, medium sized arterioles that are nearly 50um will be most commonly affected, with a predilection for vessel bifurcation (branch) points [S. Hawke et al., 1990].

Under the microscopic view, the tunica intima is thickened while the tunica media is thinned. Standard techniques to monitor for graft arteriopathy, such as skin biopsy and angiography, have been shown to be ineffective in detecting the subtle changes associated with GVD. More invasive deep tissue and large vessel biopsies are costly, risk infection, and may potentially damage the blood supply to the graft.

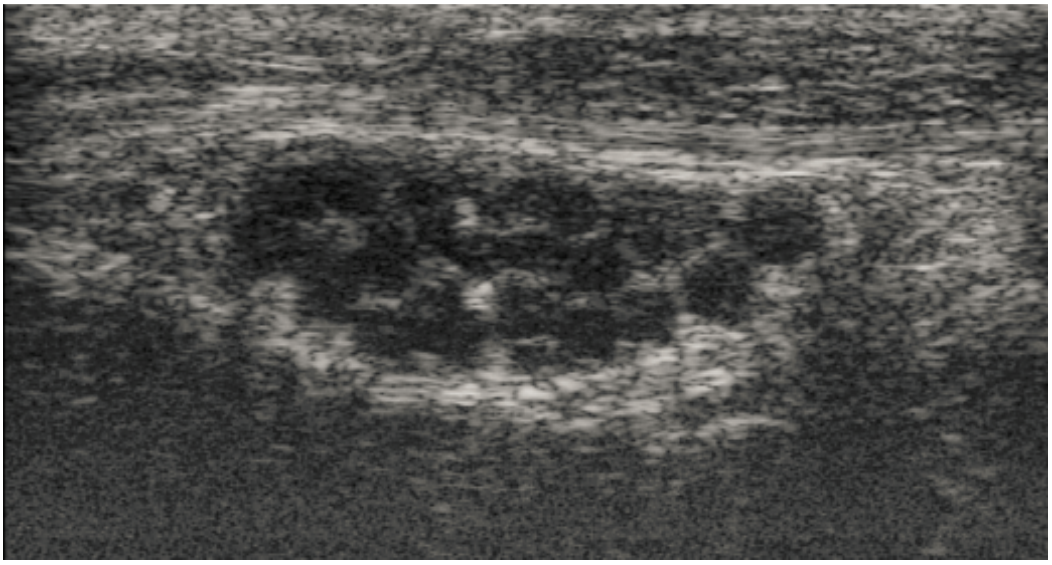


Figure 1.2: Nerve image with individual fascicles. (Scanned using VisualSonics Vevo 2100 system at 50 MHz)
(Courtesy of Ada Zhang and the University of Pittsburg Medical Center)

Recent advances in high frequency ultrasound have made possible the visualization of individual fascicles in nerve, as well as the changes in arterial walls associated with GVD. Figure 1.2 shows an example of a nerve image for human arms using state-of-the-art 70 MHz ultrasound. We can clearly see the fascicles along with small bundles of nerve fibers under the high resolution ultrasound scan. A 70 MHz Probe-Sight device could therefore initially be applied in upper extremity transplantation, potentially providing more information for

doctors to better localize and observe nerve regeneration and early signs of chronic rejection.

1.3 Our Approach

1.3.1 Get Surface Information using Stereopsis

Based on the challenges in upper extremity transplantation discussed in the previous context, it is important to give surgeons an efficient approach to monitor the condition of nerves in particular as well as the entire transplanted extremity in general. Establishing correspondence between the interior and exterior of the patient will, in general, provide clinicians with a more comprehensive understanding after treatment. Adding skin-surface information to the 2D ultrasound scans will provide more anatomical information to enable better localization and understanding of nerve regeneration. Anatomically localized nerve and vasculature images, combined with skin-surface images, may also provide an ideal dataset for detecting early-stage chronic rejection.

In this report, we propose a computer vision framework based on stereopsis for acquiring real-time disparity maps from stereo images. Matrix transformations then recover surface information, which is represented as three-dimensional point clouds. We will discuss the stereopsis approach and its implementation later in this report.

1.3.2 Registration of Ultrasound with Surface Information

Stereopsis can provide an accurate way to build a three-dimensional map from a disparity image. The aim of Probe-Sight is not only building the three-dimensional

surface information, but also registering the surface with the corresponding real-time ultrasound image at the proper location. As Figure 1.3 shows, when the ultrasound probe scans the patient, the two-dimensional ultrasound image is under the patient's skin in front of the probe. Since the depth and width of the ultrasound image is known, the ultrasound can be mapped at its actual size along with the surface information to the same ProbeSight coordinate system, in our case using the left camera as the origin. In this approach, different types of image modalities can be merged together to provide an anatomical context for the ultrasound data, in terms of the exterior of the patient.

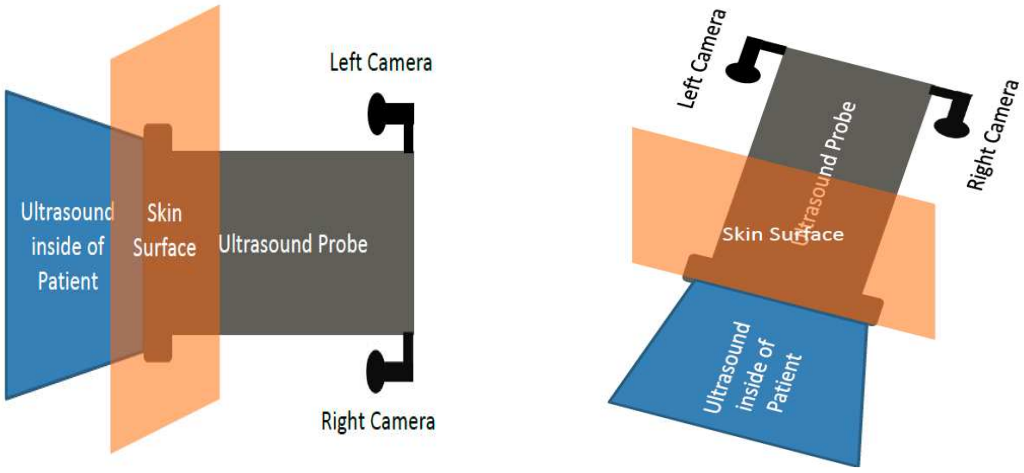


Figure 1.3: The geometricy relationship between ultrasound image, surface, and probe with cameras. (Left: Top view, Right: Side view)

1.3.3 Three-Dimensional Visualization of Registered Data

After properly registering and localizing two kinds of image modalities in the same coordinate system, the three-dimensional point clouds are then textured with the video information and visualized with OpenGL in a 3D graphical environment. Since the location of the ultrasound data is also known relative to

the cameras, the ultrasound image can also be rendered (using texture mapping) in the same 3D graphical environment in real time.

In sum, the computer vision framework combines three powerful ideas: stereopsis, image registration, and real-time image rendering. The combination of these three ideas yields numerous benefits. First, real-time (20-25 Hz frame rate) stereopsis detects the correspondences between each camera and builds a disparity map. Second, registration of both the 2D ultrasound image and the 3D surface image into a single coordinate system allows the 2D and 3D images to be combined together to provide comprehensive visualization for doctors. Finally, real-time visualization provides more comprehensive information for the doctors to effectively and efficiently make an optimal decision in a short diagnosis sessions.

1.4 Outline

Chapter 2 presents the background knowledge. It starts with the concept of augmented reality (AR) and several literature reviews about how augmented reality can be applied in clinical applications. Camera calibration is discussed next, since it is a pre-condition for most of the camera-based computer vision systems. An early version of Probe-Sight prototype based on lasers and a single camera is introduced after that. Finally, we present the basic background of stereopsis including calibration between stereo cameras, epipolar geometry, and the correspondence problem.

Chapter 3 describes our implementation for the Probe-Sight device. The GPU based computer vision module will be introduced first as the major contribution of our report. Both hardware and software design will be discussed after that in detail. The development of software tools used in the device and the flow chart of our system will also be introduced in this chapter.

Chapter 4 demonstrates an image that registers both the phantom 3D surface and 2D ultrasound information together at the proper location. The accuracy of our stereo vision system was tested by putting the phantom at known distances relative to the cameras.

Chapter 5 presents the results, limitations and recommendations for the future work.

Chapter 2

2. Background

2.1 Augmented Reality

Augmented reality (AR) may be considered as an extension of Virtual Reality (VR), instead of considering AR and VR to be opposite concepts [P. Milgram et al., 1994]. In the medical field, the term “augmented reality” has generally referred to augmenting the reality of the human operator by the introduction of information beyond the normal powers of human perception. Typically, operators can get and analyze the patient’s information by using different imaging modalities such as ultrasound, MR, or CT. The augmented reality system that includes video cameras can further help augment the experience of the human operator through analyzing the video data. Finally, the analysis of the video stream and the other imaging data can be combined for computer analysis, as it is in the mind of the operator. Thus the “reality” being augmented can include not only the human’s, but also that of the imaging scanner and associated computer algorithms.

In last 20 years, augmented reality has extensively utilized the medical imaging modalities, and some pioneering work has been done that gives operators better understanding of the patients. H. Fuchs et al. [1992] put forward the concept of incremental volume reconstruction and rendering for ultrasound. During the past 20 years, his group at the University of North Carolina at Chapel Hill has worked

to develop and operate a system that allows a physician to see directly inside a patient by combing computer graphics with images of the real world. The system displays live ultrasound data or laparoscopic range data in real time and properly registered to the part of the patient that is being scanned. Such a system could be a powerful and intuitive tool to assist and to guide the physician during various types of ultrasound-guided and laparoscopic procedures. A. State et al. [1996] presented a real-time stereoscopic AR system applied to the medical procedure of ultrasound-guided needle biopsy of the breast. They used a head-mounted display to show the merged images, which rendered live ultrasound data and geometric elements with stereo images of the patient acquired through head-mounted video cameras. The system was used by a physician during procedures on breast models and during non-invasive examinations of human subjects.

Since training is essential for developing the hand-eye coordination needed to choose the best needle trajectory, various types of augment reality devices have been developed in the 21st century to aid in both training and performance of needle insertion. For example, J. Kettenbach et al. [2004] and P. Wunderbaldinger et al. [2001] were part of a team that developed a robotic arm which can be used to hold the needle, and the motion of needle is controlled by the physician through computer. In this approach, the hand-eye coordination task is replaced with robotic operation, and such systems can also be combined with other medical imaging device such as fluoroscopy. C. Chan et al. [2005] proposed a novel tracking device for measuring the position and orientation of a

needle with respect to an ultrasound probe by using a pair of cameras which track the needle location so that a standard needle can be used without attaching a separate sensor to the needle.

Over the past decade, our laboratory has developed a method of fusing these two information streams in the mind of the operator using a device we call the Sonic Flashlight. It uses optical reflection of a real-time, tomographic ultrasound image to merge the visual outer surface of the patient with a simultaneous ultrasound scan of the patient's interior [G. Stetten et al., 2000]. This device combines a half-silvered mirror with a flat-panel monitor such that the image on the monitor is reflected precisely at the proper location within the patient. In this way, the ultrasound image is superimposed in real time into the view of the patient along with the operator's hands and any invasive tools all located in the proper field of view so that the clinician can avoid looking away at an ultrasound monitor, thus effectively solving the hand-eye coordination problem. Figure 2.1 (left) shows the Sonic Flashlight and (right) the operator's point of view using it to guide insertion of a needle into a vein in the upper arm of a cadaver. The virtual image is shown magnified in the white box. The needle tip is visible as a bright spot within the dark cross section of the vein [W. Chang et al., 2006].

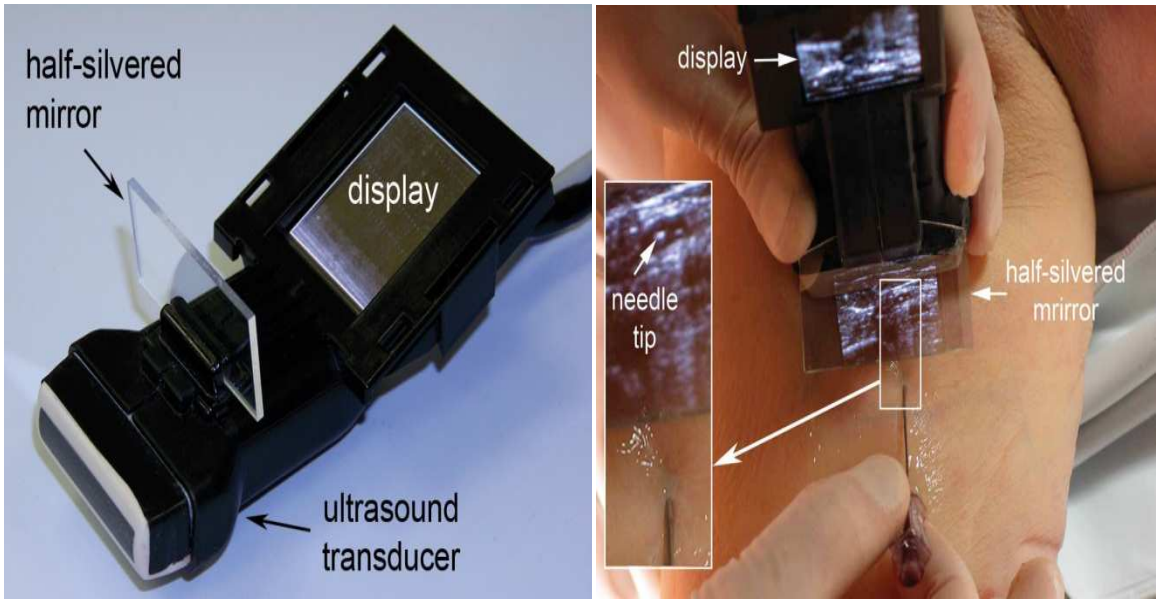


Figure 2.1: Sonic Flashlight device for viewing ultrasound in its actual location.
(Image from W. Chang's paper *The Sonic Flashlight is faster than conventional ultrasound guidance to learn and use for vascular access on phantoms.* 2006.)

2.2 Camera Calibration

Cameras play an important role in augmented reality, as they are a computer's interface to the visible world. In theory, it is possible to define a camera lens that will introduce no geometric distortions. In practice, however, no lens is perfect. This is mainly for reasons of manufacturing. It is also difficult to mechanically align the lens and imager exactly. Two main geometric lens distortions are described in this section. Radial distortions arise as a result of the shape of a lens, whereas tangential (decentering) distortions arise from the assembly process of the camera as a whole.

We start with radial distortion. The lenses of real cameras often distort the location of pixels near the edges of the imager. This phenomenon is the source of the "barrel" or "fish-eye" effect (Figure 2.2 Left).

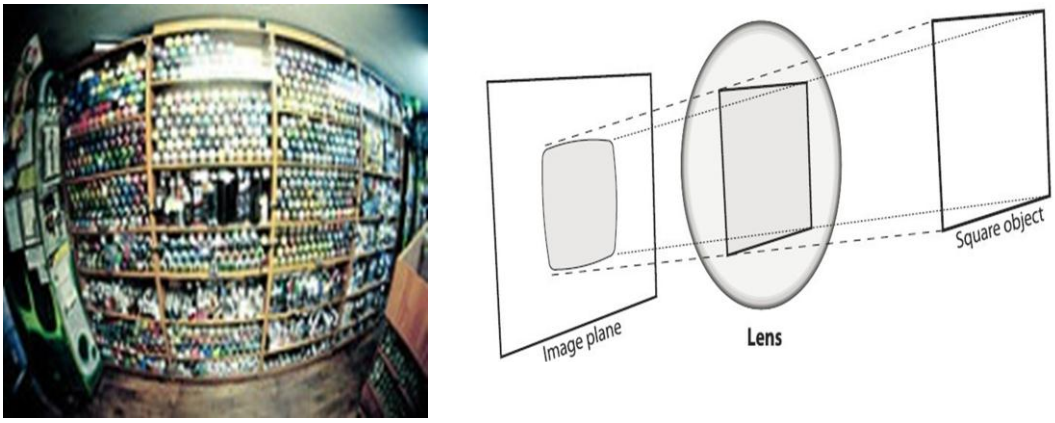


Figure 2.2: Fish-eye effect and distortion phenomenon.

Figure 2.2 Right shows the nature of radial distortion. With some lenses, rays farther from the center of the lens are bent more than those closer in. A typical inexpensive lens is, in effect, stronger than it should be as you get farther from the center. Barrel distortion is particularly noticeable in cheap cameras but may be less apparent in high-end cameras, where a lot of effort is put into complex lens systems that are typically engineered to minimize radial distortion.

For radial distortions, the distortion is 0 at the optical center of the imager and increases toward the periphery. In practice, this distortion is small and can be characterized by the first few terms of a Taylor series expansion around r = 0 [D. Brown 1971 and J. Fryer 1986]. For inexpensive optics, we generally use the first two such terms, which are conventionally termed k_1 and k_2. For highly distorted optics such as fish-eye lenses it may be necessary to use a third radial distortion term k_3. In general, the radial location of a point on the imager will be rescaled according to the following equations:

characterized by the first few terms of a Taylor series expansion around r = 0 [D. Brown 1971 and J. Fryer 1986]. For inexpensive optics, we generally use the first two such terms, which are conventionally termed k_1 and k_2. For highly distorted optics such as fish-eye lenses it may be necessary to use a third radial distortion term k_3. In general, the radial location of a point on the imager will be rescaled according to the following equations:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2.1)$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2.2)$$

where (x, y) is the original location (on the imager) of the distorted point and $(x_{corrected}, y_{corrected})$ is the new location as a result of correction for radial distortion. The second-largest common distortion is *tangential distortion*. This distortion is due to manufacturing imprecision resulting in the lens not being exactly parallel to the imaging plane. Tangential distortion is minimally characterized by two additional parameters, p_1 and p_2 [D. Brown, 1966], and the equations are as follows:

$$x_{corrected} = x + [2p_1 y + p_1(r^2 + 2x^2)] \quad (2.3)$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 x) \quad (2.4)$$

2.3 Early Version of the Probe-Sight Device Camera Calibration

Our lab's initial concept of Probe-Sight was designed to automatically determine the location of the surface of the patient being scanned and to aid analysis of the ultrasound data. In particular, the initial device determined the angle of the ultrasound probe relative to the surface of skin to disambiguate Doppler

phantom (Blue Phantom, Inc.), which was within the field of view of the video camera. The location in the video image of each of the red spots was determined in real time as the centroid of clusters of thresholded pixels, and these locations were used to determine the orientation of the phantom surface with respect to the ultrasound probe.

The system was tested by holding the ultr asound probe at known angles relative to the flat surface of the gel phantom, between -60 to +60 degrees (5 degrees interval) relative to perpendicular, and the result was sufficient accuracy for applying the device to disambiguate Doppler. They subsequently demonstrated holding the ultrasound probe at an angle towards the surface of the skin and getting the correct corresponding Doppler information of the carotid artery.

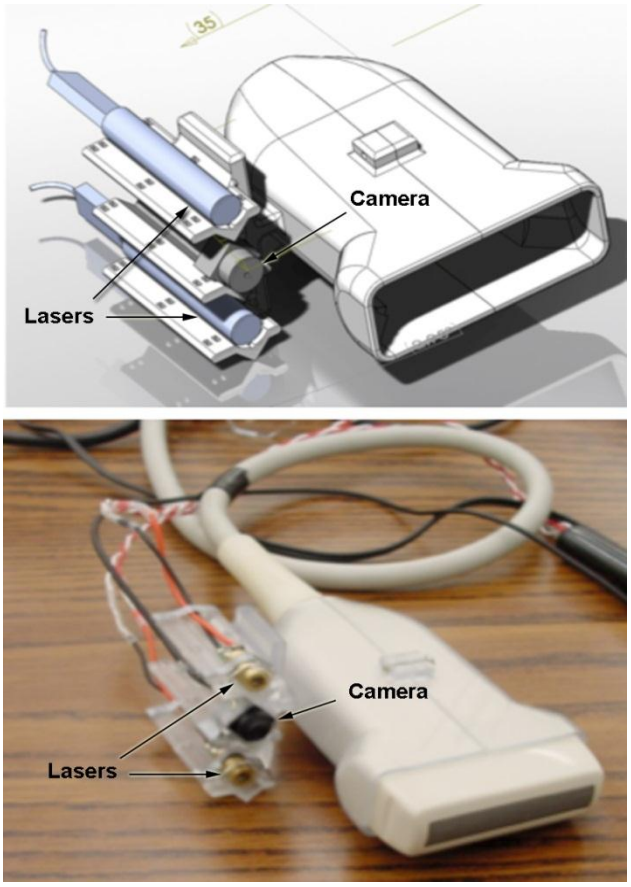


Figure 2.3: Design and actual device with camera and lasers mounted on probe.
(Image from S. Horvath's paper, *Towards an Ultrasound Probe with Vision: Structured Light to Determine Surface Orientation*, 2011.)

angle towards the surface of the skin and getting the correct corresponding Doppler information of the carotid artery.

2.4 Stereopsis

Due to the spacing between a human's left and right eyes, they will observe slightly different 2D images of the viewed scene. These differences, referred to as binocular disparity, provide information that the brain can use to calculate depth in the visual scene, providing a major means of depth perception. The brain uses the disparity image to recover a description of the 3-D structure of the environment. Since Wheatstone invented the concept of the stereoscope in 1838, the processes underlying primate stereo vision have been intensively studied, first with psychophysical techniques, and more recently in terms of the underlying physiological mechanisms in the visual cortex.

However, as with so many other visual tasks that humans perform easily and effortlessly, the development of computer systems implementing stereoscopic vision has proven surprisingly difficult. The problem of stereopsis is not only a problem in the area of psychophysics and physiology, but also a very complex computational problem involving both camera physics and information processing issues.

2.4.1 Stereo Calibration

Stereo calibration is the procedure of computing the geometrical relationship between the two cameras in space. Stereo rectification in turn is the process of

correcting the individual images so that they appear as if they had been taken by two cameras with row-aligned image planes as Figure 2.4 shows.

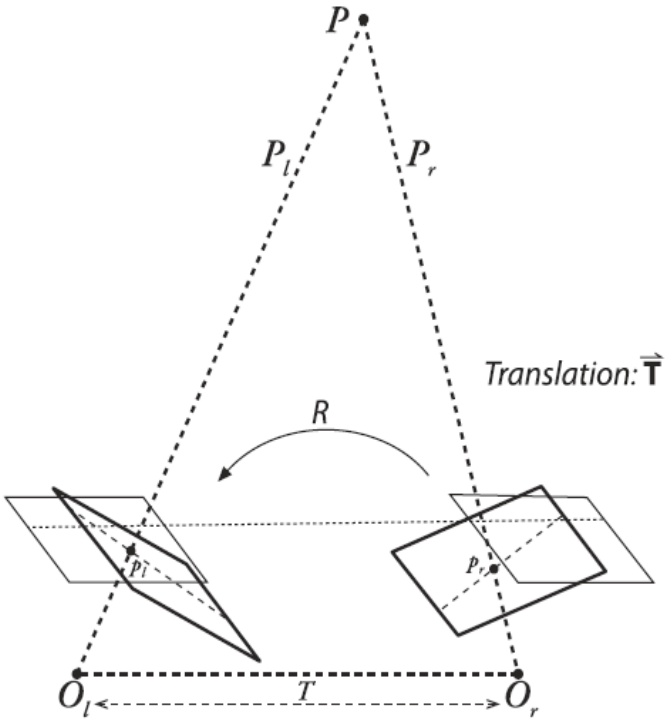


Figure 2.4: The procedure about align two cameras on viewing plane.

With such a rectification, the principal rays of the two cameras are parallel and so theoretically they intersect at infinity.

Stereo calibration depends on finding the rotation matrix R and translation vector T between the left and right cameras by using the distortion coefficients derived from individual camera calibration. For any given 3D point P in object coordinates, single-camera calibration can be used separately for each of the two cameras to put P in the camera coordinates so that it is easy to find the relationship between p_l and p_r that relates to both cameras as:

$$p_l = \mathrm{R}_l\mathrm{P} + \mathrm{T}_l \quad (2.5)$$

$$p_r = \mathrm{R}_r\mathrm{P} + \mathrm{T}_r \quad (2.6)$$

The two views of point P are also related by

$$p_l = R^T p_r + T \tag{2.7}$$

Taking the three equations above and solving for the rotation and translation separately yields the following relations:

$$R = \mathrm{R}_r(\mathrm{R}_\mathrm{l})^T \tag{2.8}$$

$$T = \mathrm{T}_r - R\mathrm{T}_l \tag{2.9}$$

Once the rotation and translation values (R, T) are derived, the results can be used to rectify the two stereo images so that the epipolar lines are arranged along image rows and the scan lines are the same across both images.

2.4.2 Epipolar Geometry and Rectification

Epipolar geometry is used in stereo vision to confine the search space when looking for matching points in both images (Figure 2.5). A point P in 3D space is described in the left view (which we will call the source image) as a point p, which will always be on the line between the left camera's focal point O_l and point P. This line can be seen in the right view (which we will call the search image), in which it will appears as a line. This is called an epipolar line (e_l and e_r respectively). Given both the cameras' internal and external matrices (determined by camera calibration) and a point p on source image, we can then generate an epipolar line corresponding to this point in the search image. This constrains the search space to this 1D line.

However, it is obvious that for each pixel in the source image, we have to calculate the corresponding epipolar line in the search image. It would be much more convenient if each epipolar line was on the same line in the search image as the pixel it corresponded to in the source image. It is feasible to transform the

images in such a way that the epipolar lines are parallel and horizontal, and that process is called rectification.

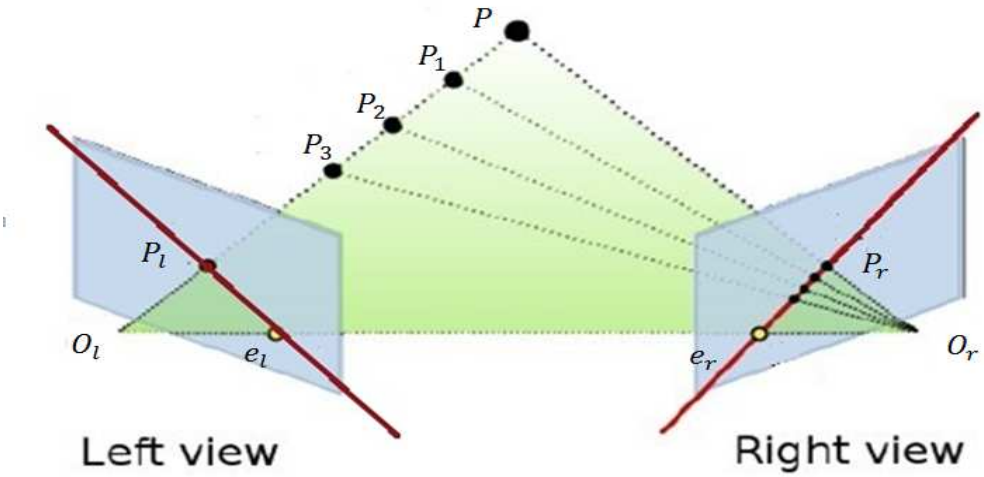


Figure 2.5: Epipolar geometry.

The basic idea of the rectification algorithm here is shown on Figure 2.6 where P represents the point in 3D space, O_l and O_r represent the camera focal points, p_l and p_r are the projections of P on the planes of the cameras. First both camera matrices are separately factorized into three parts: The internal camera matrix A, the rotational matrix R that gives the rotation between the camera's frame of reference and the world frame of reference, and the translation matrix T that gives the translation between the camera's frame of reference and the world frame of reference. In this way, the perspective projection matrix P is described as:

$$P = A[\mathrm{R}|\mathrm{T}] \tag{2.10}$$

A new rotational matrix R_n and new internal camera matrix A_n can be constructed such that in the new cameras reference systems, the x-axis is parallel to the baseline. The baseline is simply the line between the two optical

centers, which can be retrieved from P [A. Fusiello et al., 2000]. The new perspective projection matrix is now shifted as:

$$P_{nl} = A_{nl}[\mathrm{R}_l | -\mathrm{R}_{nl} O_l] \tag{2.11}$$

$$P_{nr} = A_{nr}[\mathrm{R}_r | -\mathrm{R}_{nr} O_r] \tag{2.12}$$

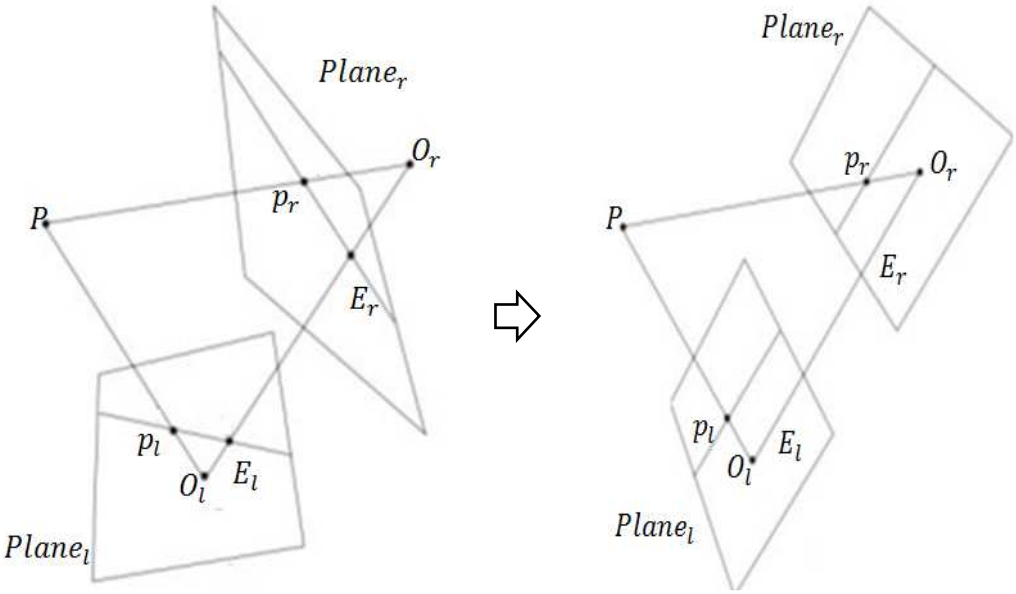


Figure 2.6: The procedure of camera rectification. (Left: unrectified cameras, right: rectified cameras)

(Image from A. Fusiello's paper *A compact algorithm for rectification of stereo pairs*, 2000.)

Taking the points E_1 and E_2 as the epipolar points of both images, all epipolar lines intersect the epipolar point of a given image. However, when the planes of both cameras lie on the same plane, the epipolar points move to infinity and the epipolar lines in one image become parallel to one another. The line between the optical centers of both planes is called the baseline. When the epipolar points lie at infinity, we can see that the epipolar lines are parallel to this baseline and that means the epipolar lines are horizontal.

2.4.3 Correspondence Problem

Because of the way human eyes are positioned and controlled, if two objects are separated in depth from the viewer, the relative positions of their images will differ in the two eyes. The brain can measure the disparity and use it to estimate the depth of the object. Two critical steps easily handled by human eyes and brains are: (1) in the first image, selecting unambiguously identifiable locations on a surface in the scene, and (2) in the other image, identifying the corresponding locations. However, existing computer vision systems struggle to solve this correspondence problem.

Typically, there are two reasons that some points in each image will have no corresponding points in the other image. One is that the cameras might have different fields of view, and another is due to occlusion as Figure 2.7 shows.

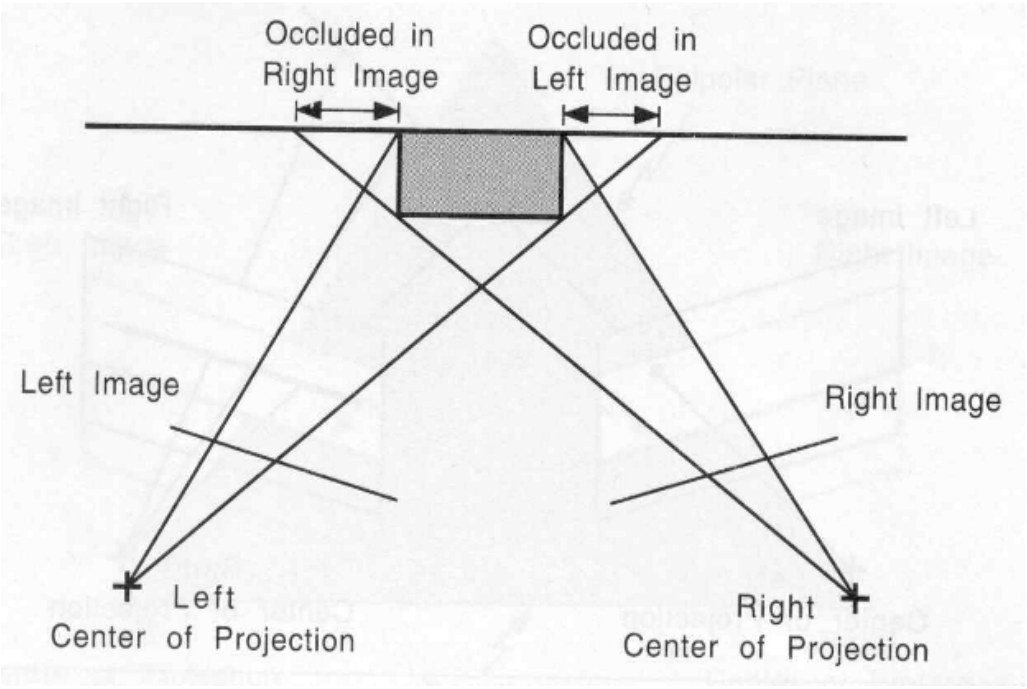


Figure 2.7: The occlusion which may cause camera mismatch.

In general, the techniques that build the disparity map by solving the correspondence problem can be divided into two categories. Local algorithms, such as either the Block Matching (BM) algorithm [K. Konolige, 1997] or the Semiglobal Block Matching (SGBM) algorithm [H. Hirschmuller, 2008] are based only on local pixel values, and do not take into account the computed disparity values of neighboring pixels. Local algorithms are typically an appropriate measure in order to quantify the similarity between a *template window* positioned at a location in one image and potential corresponding locations, i.e. *candidates* in the other image. Different terms can be used to measure similarity, such as sum of squared differences (SSD), the sum of absolute differences (SAD), and the normalized cross-correlation (NCC). Also, the correspondence problem can be considered as a global optimization problem which relies on the minimization of an energy function. This energy function is usually defined as the combination of data energy and smoothness energy. The data energy is often borrowed from the local matching algorithms, as it is often based upon matching metrics like SSD or SAD. The smoothness energy is used to penalize disparity solutions that are not smooth. Famous algorithms in this category include Graph Cut (GC) [Y. Boykov et al., 2001] and Belief Propagation (BP) [P. Felzenszwalb and D. Huttenlocher, 2006]. Compared to local algorithms, the performance (average percent of bad disparity pixels) of global algorithms is almost always better, but they need substantially more computation time.

Chapter 3

3. Implementation of a Stereoscopic Probe-Sight Device

3.1 Graphics Processing Units (GPU) Based Blocking Matching Algorithm

It has often been said that Computer Graphics and Computer Vision are inverses of each other: ordinarily GPUs are used to convert “numbers into pictures”. Modern graphics hardware is built around a specialized processor called the Graphics Processing Unit, or “GPU”. Modern GPUs utilize a highly parallel architecture that is optimal for performing the operations of computer graphics. As graphics algorithms became more sophisticated, there was a need to develop more flexible hardware and programming environments which led to the development of user-programmable hardware. The new flexibility combined with floating point capability and specialized parallel performance, led to work in General Purpose GPU (GPGPU) processing: using the graphics hardware to perform computations for tasks other than graphics.

Our reasons for choosing a GPU based approach are: (1) **Highly Parallel Floating Point Computation**. Current GPUs have hundreds of processors capable of concurrent floating point operation, which can accelerate parallelizable computation many times faster than the CPU alone is capable of [NVIDIA Corporation, 2007]. (2) **Increased Memory Bandwidth**. GPUs can be

useful for applications such as pattern matching where a set of reference data can be stored in the memory of each graphics card. On board graphics RAM is typically accessed by the GPU faster than the system RAM is accessed by the CPU, and current GPUs are available with up to 6 GB of RAM. (3) **Multi-GPU Computation.** Multiple graphics cards can be combined for even greater levels of parallelization [J. Fung et al., 2008].

For the present work we utilized a GPU based BM method that is publically available as part of the Open Computer-Vision Toolkit (OpenCV, a C++ based computer vision and real-time image processing toolkit that will be discussed in more detail later). OpenCV's GPU module provides GPU routines for both acquiring real-time disparity maps of the stereo images and for recovering three-dimensional point clouds from disparity maps. OpenCV's GPU code typically executes 5-10x faster than the corresponding CPU code on modern hardware, and as such the GPU implementation is typically necessary for real-time operation at modern resolutions. For example, the GPU BM method allows full HD, real-time operation when utilizing dual GPUs.

3.2 Hardware Design

In order to build a stereoscopic Probe-Sight device, two small video cameras were mounted to the back of an ultrasound probe. This apparatus, as well as a visually textured gel phantom that was used for testing, is shown in Figure 3.1. The photo is representative only, to show the constituent components. (The angle of the transducer would not actually produce an acceptable ultrasound

image.) The twin video cameras are Minocam SD-008, and the ultrasound probe they are mounted on is the Diasus 15-22 MHz from Dynamic Imaging. A lightweight aluminum frame holds everything together, so that the cameras have a clear and focused view of the surface of the ultrasound phantom, upon which a sheet of tracing paper has been laid and saturated with gel. A chessboard pattern has been printed on the tracing paper using an inkjet printer (a laserjet printer would melt the wax within the paper). The rea son for adding the chessboard pattern on the surface of phantom is that the phantom by itself is visually smooth and homogenous, and thus some visual texture must be added to enable any stereo matching algorithm. The saturated tracing paper does not interfere with the passage of ultrasound into the phantom.

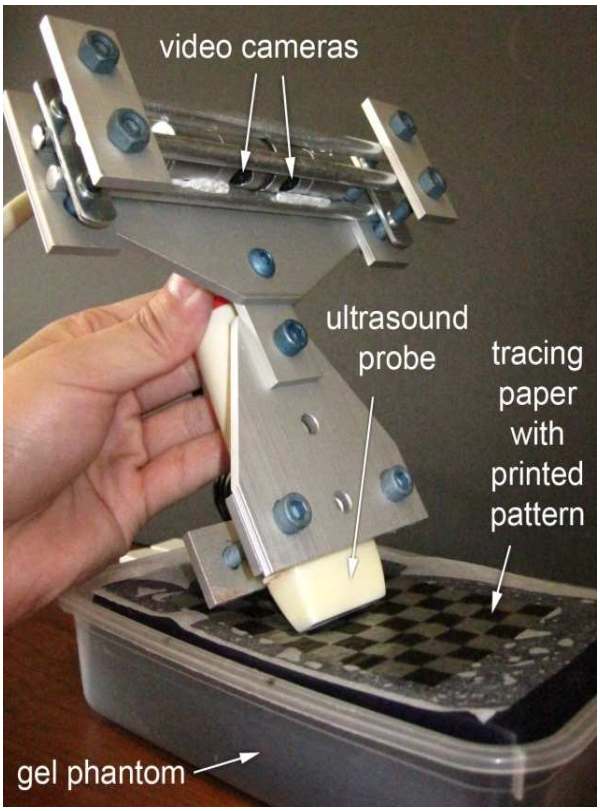


Figure 3.1: Apparatus and test phantom with printed surface.
(Image from J. Wang's paper *Real-Time Registration of Video with Ultrasound using Stereo Disparity*, 2012.)

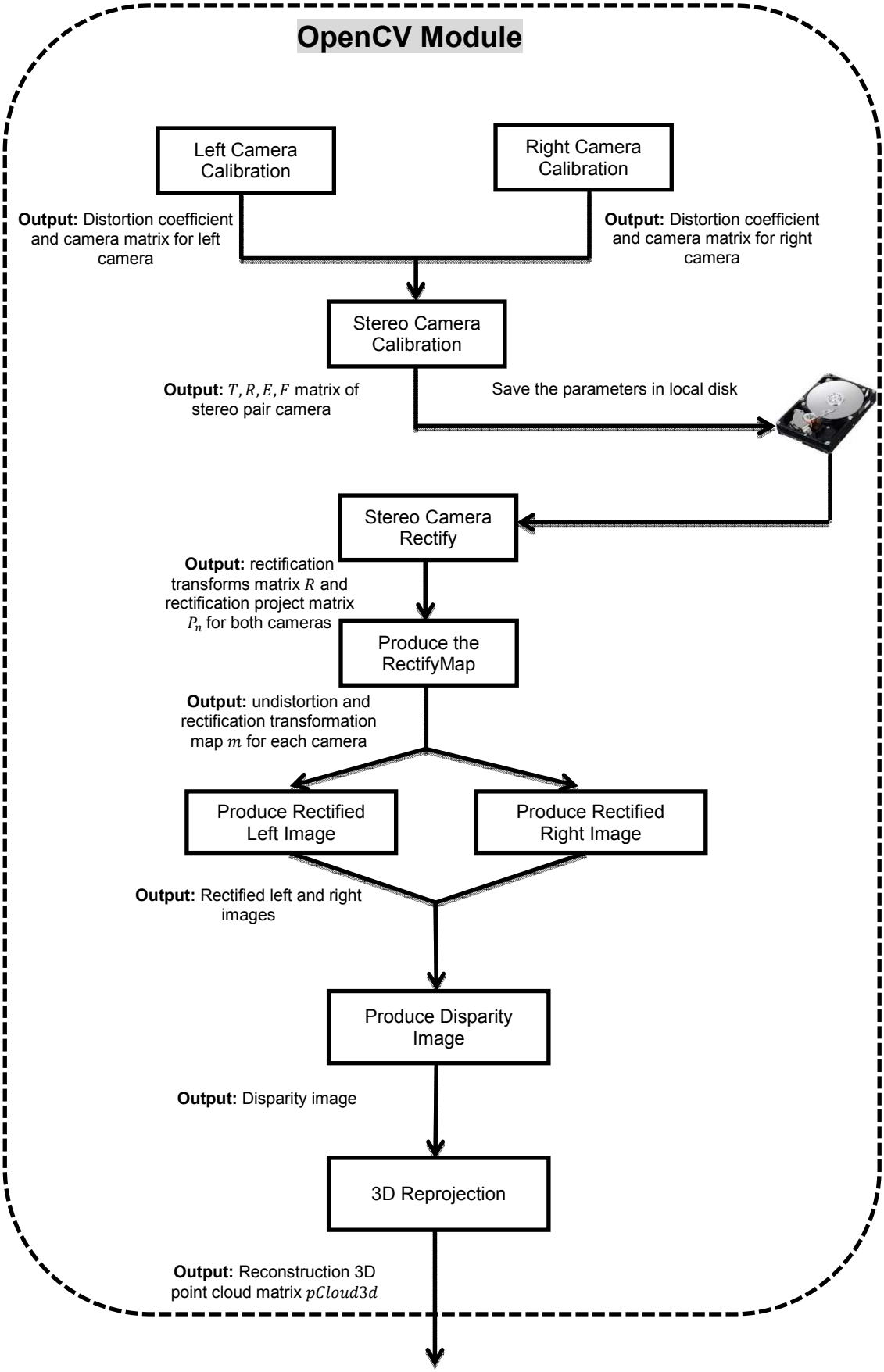
The position of the phantom relative to the cameras is then computed in real time by using stereo vision algorithms, wherein the separate viewpoints of cameras allow triangulation of the 3D coordinates of observed points. Each identifiable point on the phantom surface can be localized into the frame of reference

attached to the cameras and, by extension, to the ultrasound probe itself. It is therefore possible to know both the shape and position of the viewed portion of the surface relative to the ultrasound probe in 3D in real time.

3.3 Underlying Software Architecture

Three software development kits have been used in designing the Probe-Sight system. The computer vision module was developed using the Open-source Computer Vision library (OpenCV), which mainly focuses on real-time image processing and computer vision algorithms. It includes robust methods for both camera calibration and stereo matching, as described in more detail in section 3.4. This work represents one of the first usage cases of OpenCV version 2.2's new GPU-based methods for acquiring 3D point clouds, and this work has already helped others make use of these new capabilities [OpenCV Yahoo Group Message 80664]. OpenCV's new module of GPU-based methods provides developers with a convenient, high-performance computer-vision framework consisting of classes and functions that utilize the massively parallel computational capabilities of modern computer graphics processing units. It is implemented using NVidia's CUDA Runtime API, but it maintains conceptual consistency with OpenCV's current CPU functionality as well. Probe-Sight's real time rendering module was developed using Open Graphics Library (OpenGL), which is a standard specification defining a cross-platform API for writing applications that produce 2D and 3D computer graphics. The Graphical User Interface (GUI) module is developed using the Fast, Light Toolkit (FLTK), which is a light-weight GUI development environment. The general Probe-Sight design

flow chart is diagrammed in Figure 3.2, which will be discussed in more detail in the next section.


OpenCV Module
Left Camera Calibration
Right Camera Calibration
Output: Distortion coefficient and camera matrix for left camera
Output: Distortion coefficient and camera matrix for right camera
Stereo Camera Calibration
Output: T, R, E, F matrix of stereo pair camera
Save the parameters in local disk
Stereo Camera Rectify
Output: rectification transforms matrix R and rectification project matrix P_n for both cameras
Produce the RectifyMap
Output: undistortion and rectification transformation map m for each camera
Produce Rectified Left Image
Produce Rectified Right Image
Output: Rectified left and right images
Produce Disparity Image
Output: Disparity image
3D Reprojection
Output: Reconstruction 3D point cloud matrix $pCloud3d$

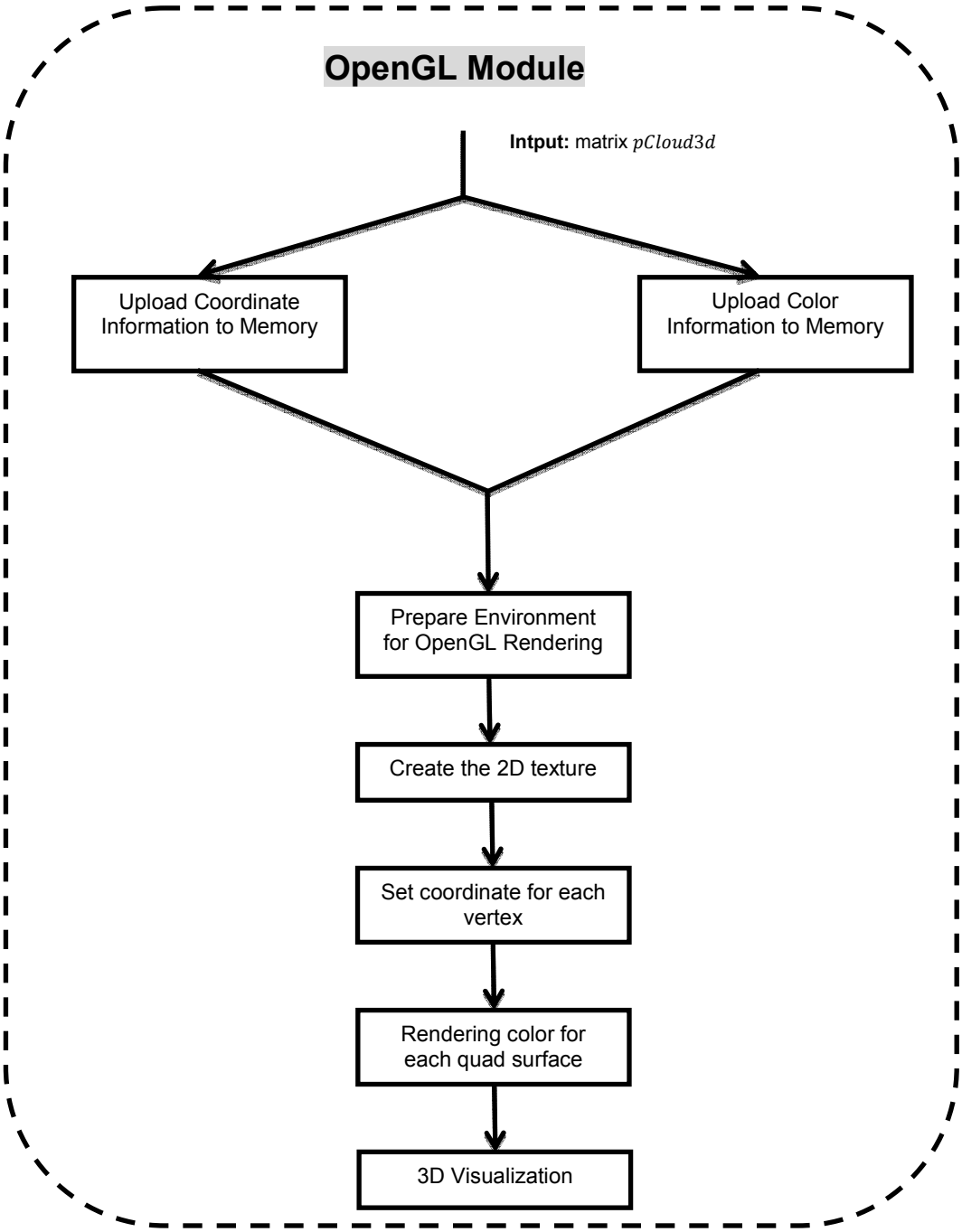


Figure 3.2: The flow chart for Probe-Sight.

3.4 Software Design

3.4.1 The Computer-Vision Module

The purpose of the computer-vision module was to calculate the real-time disparity map from the two views of the cameras. As discussed in section 2.4, each camera should be individually calibrated for imperfections in its lens system. For calibration we use Zhang's method [Z. Zhang, 2000], which targets the camera on a known structure that has many individual and identifiable points, such as the corner points of a chessboard. This method computes relative location and orientation of the camera for each image as well as the intrinsic parameters of the camera by viewing a chessboard pattern from a variety of angles. The chessboard is repeatedly rotated and translated in order to provide multiple viewpoints in the process of calibration, as Figure 3.3 shows.

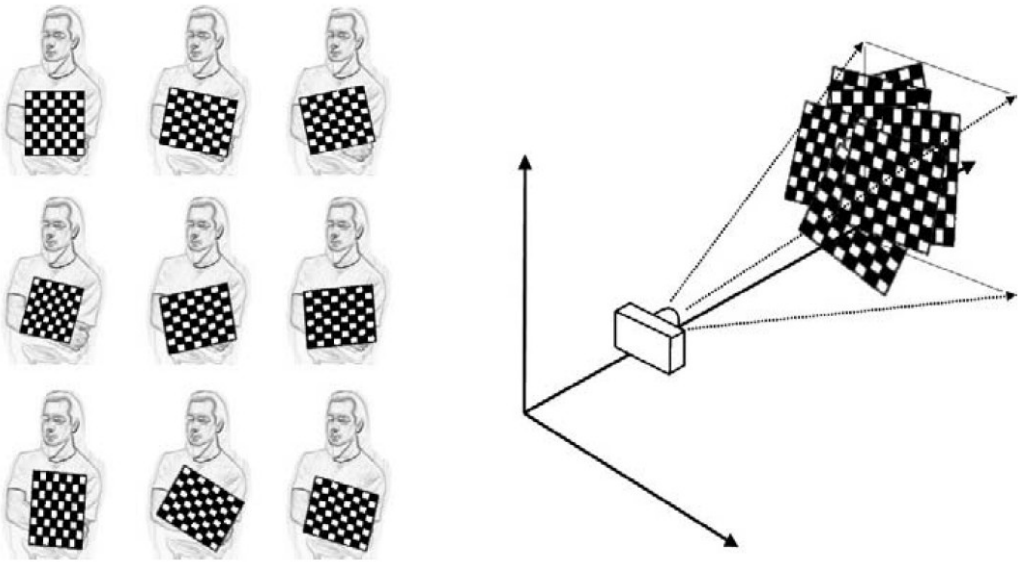


Figure 3.3: Images of a chessboard being held at various orientations. (This approach provides enough information to completely solve for the locations of those images in global coordinates (relative to the camera) and the camera intrinsic parameters)
(Image from G. Bradski's book *Learning OpenCV: Computer Vision with the OpenCV Library,* 2008)

Zhang's method is implemented in OpenCV's *CalibrateCamera()* function, which takes as input a list of 2D pixel-coordinates of points in the left image, the 2D

pixel-coordinates of the same physical points in the right image, and a set of 3D physical coordinates where these points actually lie in the real world.

OpenCV's function *FindChessboardCorners()* was used to locate the corners of the calibration chessboard and display where these corners were found in the original image as Figure 3.4 shows. The OpenCV function *CalibrateCamera()* finally calculates the distortion coefficients along with both intrinsic (camera matrix) and extrinsic parameters (translation/rotation matrix) for each camera [G. Bradski and A. Kaehler, 2008].

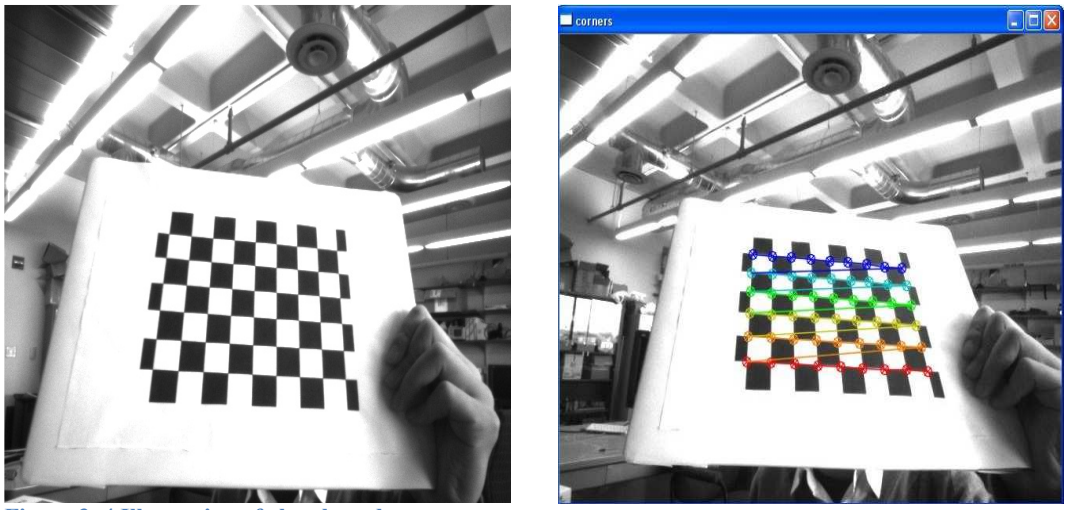


Figure 3.4: Illustration of chessboard pattern. (Left: chessboard pattern (original), right: chessboard pattern (corners detected))

After calibrating for the distortion coefficients and both intrinsic and extrinsic parameters for each individual camera, it is then possible to properly calibrate for the geometrical relationship between the two cameras in space. Given many joint views of chessboard corners, OpenCV's function *StereoCalibrate()* can be used to solve for the rotation and translation parameters of the chessboard views for each camera separately. The algorithm solves for the rotation and translation

parameters (R, T) based on equations (2.5) through (2.9). Because of image noise and environmental fluctuations such as changing illumination, each chessboard pair results in slightly different values for R and T, and so the median value of these is used as the initial approximation of the true solution after which Levenberg-Marquardt iterative optimization is used to find the local minimum of the reprojection error of the chessboard corners for both camera views. StereoCalibrate() then returns the minimizing R and T, which are used to derive the Essential matrix E and Fundamental matrix F for the camera pairs.

After calibration of all the camera pair's parameters, Bouguet's algorithm [R. Tsai, 1987; R. Zhang, 1999 and Z. Zhang, 2000] can be used to rectify the camera pairs by simply attempting to minimize the amount of change that reprojection produces for each of the two images while maximizing common viewing area. The basic idea of this method is to split the rotation matrix R in half between the two cameras, such that there are two coplanar rectified rotation matrixes r_l and r_r for the left and right cameras, respectively. In this approach each camera has its images rotated half of R, and each camera's principal rays each end up parallel to the vector sum of where their original principal rays had been pointing. This rotation will put cameras into coplanar alignment. In order to further make them row aligned, it is necessary to compute the collinear rectified matrix R_{rect} that moves the left camera's epipole to infinity and aligns the epipolar lines horizontally. To calculate the direction of epiople e_1 in the R_{rect} matrix (taking the principal point as the origin) e_1 is set coincident with the translation vector between the two camera's centers:

$$e_1 = \frac{T}{||T||} \tag{3.1}$$

Since the matrix should be orthogonal, e_2 and e_3 can be further computed as follows:

$$e_2 = \frac{[-T_y \; T_x \; 0]^T}{\sqrt{{T_x}^2 + {T_y}^2}} \tag{3.2}$$

$$e_3 = e_1 \times e_2 \tag{3.3}$$

Finally, the resulting rectification matrix is:

$$R_{rect} = [(e_1)^T, (e_2)^T, (e_3)^T]^T \tag{3.4}$$

This matrix rotates the left camera about the center of projection so that the epipolar lines become horizontal and the epipoles are at infinity. Combining with the matrix which pulls the two cameras on the coplanar, the row alignment rectified matrix of the two cameras is then achieved by:

$$R_l = R_{rect} r_l \tag{3.5}$$

$$R_r = R_{rect} r_r \tag{3.6}$$

Figure 3.5 summarizes the procedure above, where M_{rect} is the camera matrix after rectification.

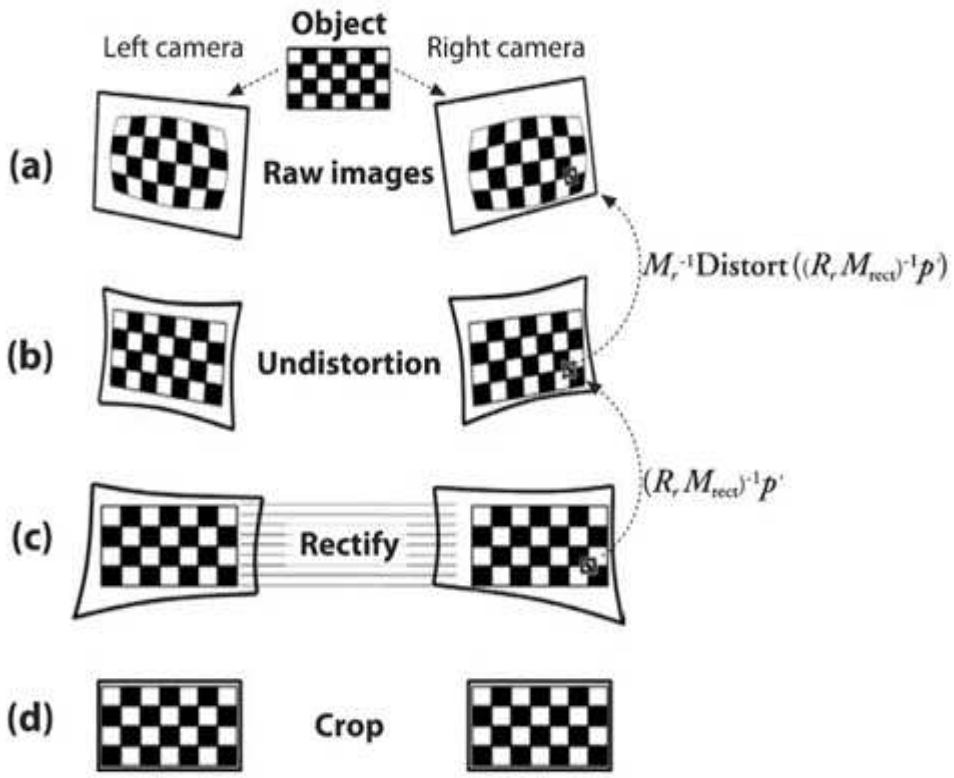


Figure 3.5: The procedure of stereo rectification. (For the left and right camera, the raw image (a) is undistorted (b) and rectified (c) and finally cropped (d) to focus on overlapping areas between the two cameras; the rectification computation actually works backward from (c) to (a))

(Image from G. Bradski's book Learning OpenCV: Computer Vision with the OpenCV Library, 2008)

After the steps above, the two stereo images will be undistorted, rectified, and have horizontal epipolar lines, making them ready for stereo matching. OpenCV implements several matching algorithms, including a fast and effective block matching (BM) method. BM uses small sum-absolute-difference (SAD) windows to find matching points between the left and right stereo rectified images. Because our cameras are necessarily close to the skin's surface (about 120 mm), disparities of 110 pixels are not uncommon, despite our minimizing the baseline distance between the cameras. Theoretically speaking, larger matching-window sizes result in more accuracy, but will also require more computation time,

potentially making real-time analysis impossible. We achieved real-time performance by using GPU acceleration as implemented in OpenCV's function *gpu::StereoBM_GPU()*. On modest hardware, relatively large 19-by-19 windows can be used on VGA (640x480) resolution images at 15-20 frames per second.

As previously discussed, 3D reprojection of an object requires t*riangulation* between the object and the two camera's centers (O_l and O_r). In this simplified case as shown in Figure 3.6, taking x_l and x_r to be the horizontal positions of the points in the left and right imager (respectively) leads to depth being inversely proportional to the disparity between these views, where the disparity d is defined simply by $\mathrm{d} = \mathrm{x}_l - \mathrm{x}_r$ [G. Bradski and A. Kaehler, 2008]. Depth Z can then be derived by using similar triangles as:

$$\frac{T-(\mathrm{x}_l - \mathrm{x}_r)}{Z-f} = \frac{T}{Z} \quad \Rightarrow \quad Z = \frac{fT}{d} \tag{3.7}$$

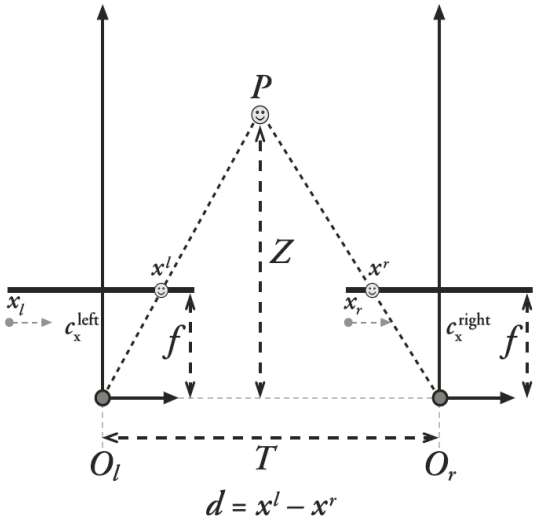


Figure 3.6: The triangulation between two camera centers and the object. (Image from G. Bradski's book Learning OpenCV: Computer Vision with the OpenCV Library, 2008)

The cameras' projection matrices take a 3D point in homogeneous coordinates to a 2D point in homogeneous coordinates as follows:

$$\mathrm{P}[\mathrm{X}\ \mathrm{Y}\ \mathrm{Z}\ 1]^T = [x\ y\ w] \tag{3.8}$$

where the coordinate shown in the screen can be calculated as $(\mathrm{x}/\mathrm{w}, \mathrm{y}/\mathrm{w})$. The two dimensions can also be reprojected back to three dimensions given the screen coordinates and the camera intrinsic matrix. The reprojection matrix is:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & 1/T_x & (c_x - c_x')/T \end{bmatrix} \tag{3.9}$$

Given a two-dimensional homogeneous point and its associated disparity d, the point can be projected into three dimensions using:

$$Q[\mathrm{x}\ \mathrm{y}\ \mathrm{d}\ 1]^T = [\mathrm{X}'\ \mathrm{Y}'\ \mathrm{Z}'\ \mathrm{W}]^T \tag{3.10}$$

The 3D coordinates are then $(\mathrm{X}'/\mathrm{W}, \mathrm{Y}'/\mathrm{W}, \mathrm{Z}'/\mathrm{W})$. We can prove the formulas 3.9 and 3.10 by multiplying the third and fourth row of Q with the third and fourth column of $[\mathrm{x}\ \mathrm{y}\ \mathrm{d}\ 1]^T$ separately. In that case $\mathrm{Z}' = f$ and $\mathrm{W} = \frac{d}{T}$, resulting in:

$$\mathrm{Z} = \frac{\mathrm{Z}'}{W} \quad \Rightarrow \quad \frac{fT}{d} \tag{3.11}$$

which is equal to equation 3.7 that was deducted from triangulation.

OpenCV provides a GPU-based method *gpu::reprojectImageTo3D()* to speed up this computation. It stores all the 3D point clouds in a giant n-by-6 matrix where columns 1-3 store coordinate information and columns 4-6 store color information as mapped using the rectified left image.

We know that real human arms are more complex than the flat surface of the phantom. Even so the probe position cannot always be disambiguated based only on the external shape of the local anatomy and the ultrasound data. This ambiguity is especially present for arms and legs, where different cross-sections may look identical. Tracking the visual surface feature (freckles, hair, markers, etc.) may provide a solution. We recently tried this by using 2D optical flow to establish spatial relationships between successive image frames. For each 2D point on the current video frame, since the 3D point sets calculated by stereo matching are known, it is possible to determine the correspondences between 2D image features and points on the 3D model based on the Iterative Closest Point algorithm [Z. Zhang, 1994]. Also, to further reduce ambiguity in probe position, we plan on pre-scanning each patient's arm in order to build an initial high-resolution 3D map. We have already used ThreeRivers 3D, Inc.'s scanner to scan a hand transplant patient's arm, and we got very accurate 3D visualization data as Figure 3.7 shows. The scanner uses laser-based structured light and a narrow-band camera which provides more accurate (but much slower) results compared to stereopsis.

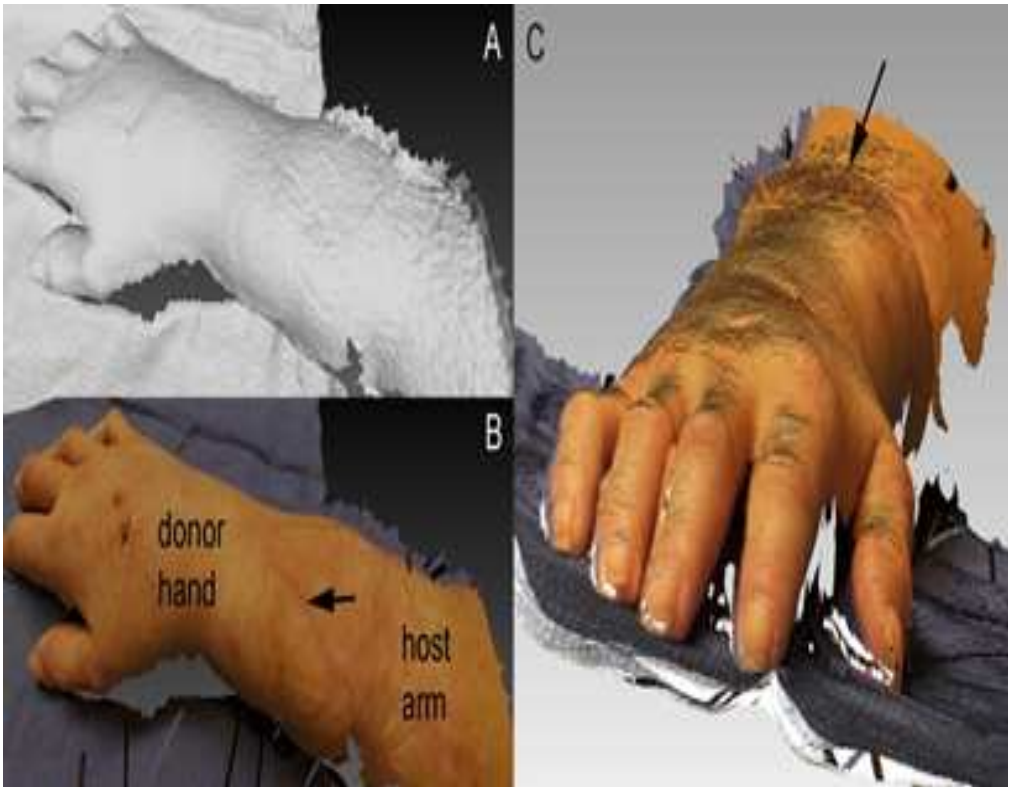


Figure 3.7: Surface reconstruction for hand transplant patient. (A. 3D surface reconstruction from ThreeRivers3D laser scanner. B. Same surface reconstruction textured with visible light information. C. Textured surface rotated to new viewpoint (arrow indicates suture line))

3.4.2 The Visualization Module

After getting all the 3D point clouds it is desirable to visualize those data. OpenGL provides an efficient texture-mapping approach to visualize the 3D point cloud along with ultrasound data. The first thing that must take place is to upload the texture (pixel color information) using a series of OpenGL functions, starting with *glBindTexture()*. This function tells OpenGL which texture "id" we will be working with, where a texture id is an integer that can be used to access the texture. Then both coordinate information and color information are uploaded to GPU memory. Then *glTexEnvf*() is called to set environment variables for the current texture, telling OpenGL how the texture will act when it is rendered into a scene. Since we want to register two images together and they are

approximately perpendicular with each other, sometimes the surface image will cover the ultrasound image when viewed from the front as Figure 1.3 shows. Accordingly, the surface rendering must be partially transparent for surgeons to be able to fully observe the ultrasound image behind the skin surface. Thus, the texture-combine state must be properly set so that transparency can then be defined. Finally, OpenGL's function *glColor4f() can be used* to render the color information on the meshed surface where the first 3 parameters are RGB color and the last one is transparency.

3D texture mapping in OpenGL requires that a mesh be created using one of several different kinds of shapes, such as triangles (3 points), quads (4 points) or polygons (more than 4 points). The present work utilized square mesh elements, which were textured as shown below. The following example code assumes that texturing has been enabled and that there has been a texture uploaded with the id X.

```
glBindTexture (GL_TEXTURE_2D, X);
glBegin (GL_QUADS);
glTexCoord2f (0.0, 0.0);
glVertex3f (0.0, 0.0, 0.0);
glTexCoord2f (1.0, 0.0);
glVertex3f (10.0, 0.0, 0.0);
glTexCoord2f (1.0, 1.0);
glVertex3f (10.0, 10.0, 0.0);
glTexCoord2f (0.0, 1.0);
glVertex3f (0.0, 10.0, 0.0);
glEnd ();
```

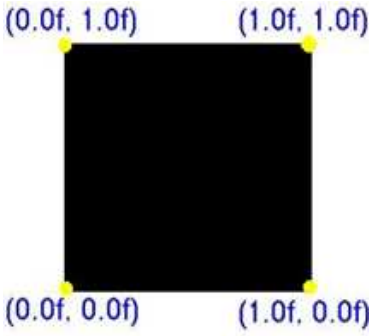


Figure 3.8: The quad shape texture.

Figure 3.8 shows the OpenGL texture coordinate system. In the code above the calls to glTexCoord2f are very important as to what the end result of the texture mapping will be. Calls to glTexCoord2f (x,y) place the texture coordinate at that place on the image. The function *glVertex3f()* can then be used to map all the 3D point cloud information to the provided quad surface. The present work then used *glColor3f()* to render color information on the surface. Figure 3.9 illustrates the processes of finding a disparity map using GPU based BM algorithm and subsequent 3D point rendering using 2D quad texture. The stereo images used are *Tsukuba.jpg* from the Middlebury Stereo Datasets (http://vision.middlebury.edu/stereo/data/)

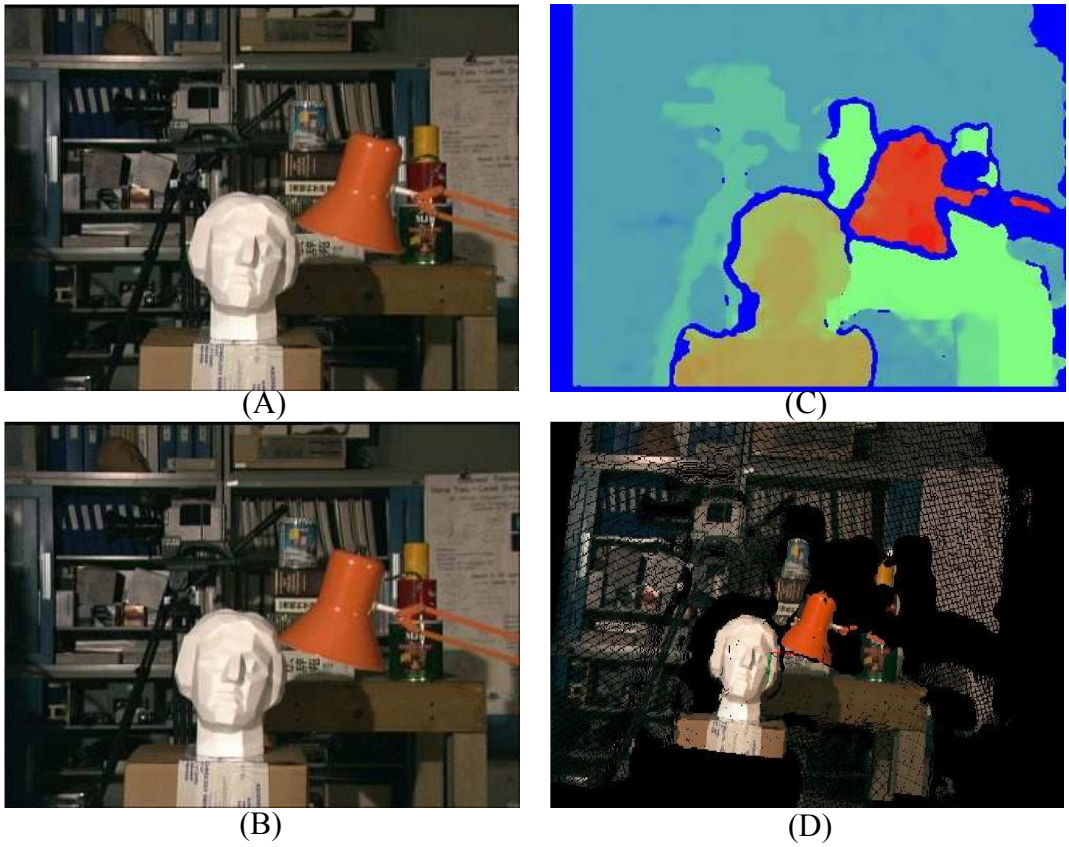


Figure 3.9: The disparity map and reconstruction rendering image. (A: the left image; B: the right image; C: the disparity image; D: the reconstruction rendering image)

3.4.3 The Graphic Users Interface (GUI) Module

A graphical user interface was also developed for the Probe-Sight device as Figure 3.10 shows below. It had five basic parts, which correspond with different functions of computer vision and image rendering. The *camera calibration* panel allowed the user to input chessboard features like row, column, and each grid's size. The *grab* button allowed users to take pictures by themselves until clicking the *finish* button. The *view rectify* option allowed users to always see the rectified image along with original image in order to determine whether some images may not have been taken well enough to get a proper calibration result. All the parameters (such as distortion coefficients) were shown in an *output window* panel. The *stereo matching* panel provided the choice of 3 methods, including GPU based BM algorithm, Constant Space Belief Propagation (CSBP) algorithm [Q. Yang et al., 2010] and CPU based semi-global BM (SGBM) algorithm. The BM algorithm had the highest speed but worst performance compared with the other 2 algorithms. The *real-time display* panel allowed the user to save disparity maps and 3D point clouds to the local disk. The user could also choose the image window size and load camera parameters from local disk. The *Control* panel allowed the user to

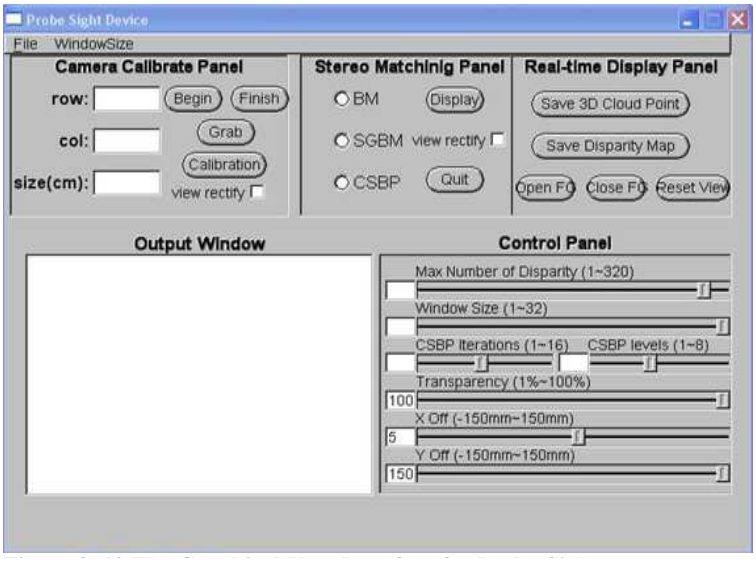


Figure 3.10: The Graphical User Interface for Probe-Sight

change the searching window's size, maximum disparity, and transparency in real-time using slider bars.

Chapter 4

4. Demonstration and Testing of a Stereoscopic Probe-Sight Device

The results for the apparatus and phantom discussed on Chapter 3 are shown in Figure 4.1. The reprojection of the chessboard pattern on the surface of the phantom has missing patches, since corresponding points in the stereo images were not found everywhere. However, a fairly consistent surface is evident. The ultrasound image is also projected in its correct location within the 3D coordinate space, and includes the cross section of a tube within the phantom. OpenGL was also used to create a simulated ultrasound probe which had the same size (height, width and length) as the physical probe in order to better describe the full 3D movement between the probe and the patient's exterior. The OpenGL window could be rotated to see the surface and ultrasound image from any point of view. The particular point of

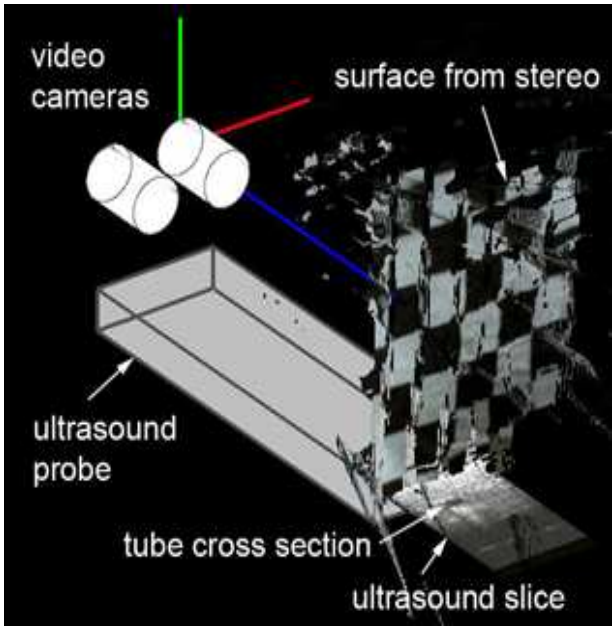


Figure 4.1 Real-time 3D simultaneous rendering of the gel phantom surface (from stereo), ultrasound, and probe/camera locations.

(Image from J. Wang's paper *Real-Time Registration of Video with Ultrasound using Stereo Disparity*, 2012.)

view of the rendering, as shown in Figure 4.1, is actually as seen from within the gel phantom.

We also tested the accuracy of our stereo vision system by putting the chessboard pattern phantom at known distances relative to camera pair over an appropriate range. The results are shown in Figure 4.2 for nine separate trials. The RMS error is ±1.12 mm [J. Wang, et al., 2012].

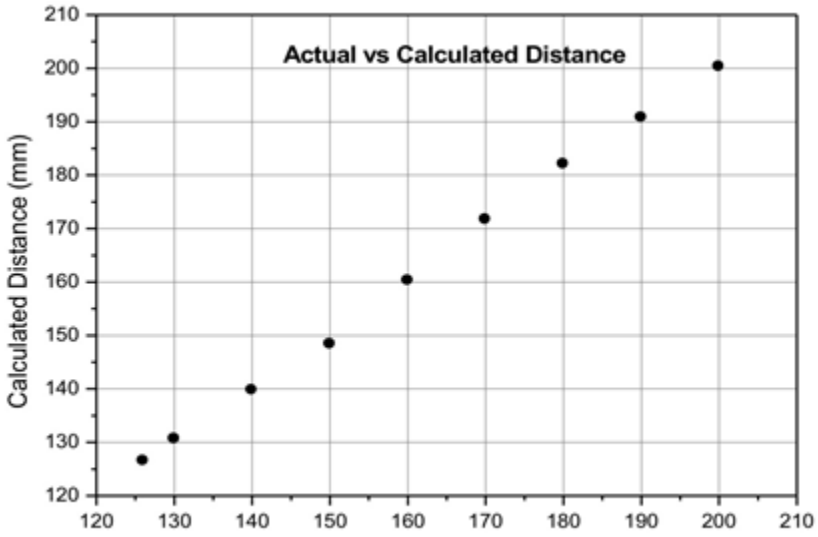


Figure 4.2: Validation of distance using stereo disparity.
(Image from J. Wang's paper *Real-Time Registration of Video with Ultrasound using Stereo Disparity*, 2012.)

Chapter 5

5. Conclusion

We believe this research represents important preliminary steps towards a clinically useful approach that merges visual surrounding information and ultrasound data in real time when the ultrasound probe is moved on the surface of a patient. Our initial application to hand transplants is just one of many clinical uses for Probe-Sight. The rapid reduction in cost and size of the cameras, and the steady development of computer vision into a practical and ubiquitous technology, practically demand that they be applied to ultrasound, where the added information about the exterior of the patient can provide anatomical context that is presently lost once the scan is recorded.

5.1 Limitations

Some problems and limitations were revealed during the design and experimentation. One problem is that the required ultrasound gel liquid can smudge the printed patterns, especially after repeated movements of the ultrasound probe. Accordingly, the chessboard paper had to be changed several times. Another important limitation is that a flat phantom is not a legitimate test for a clinically useful device. Human external anatomy consists of surfaces that not only are curved but also deform and move relative to each other due to elastic tissue and an articulated skeleton. The simple chessboard pattern is not a realistic representation of the natural markings on the surface of the skin. The matching algorithm we used finds only strongly matching (high-texture) points

between the two images. Thus, in a richly populated scene such as a freckled, the majority of pixels might have computed depth, while in a very low-textured scene, such as hairless skin with uniform complexion, very few points might register depth. Finally, the surface of a human arm is not as simple as the surface of the phantom, due to the presence of hair, and ultrasound gel further complicates the detection of the skin surface using stereo video.

5.2 Future Work

According to the limitations discussed in section 5.2, many things can be done to improve the current device. 2D optical tracking using the OpenCV method has many strict criteria for the outside environment, such as slow and steady movement and no occluding objects such as ultrasound gel. Therefore it is not suitable for tracking the ultrasound probe in real clinical applications. In the future, a more robust tracking method such as feature based tracking will be implemented. In this way, it may be possible to track the full 3D movement between the probe and the patient's exterior. The probe's position can then always be solved for if its position in the initial frame is known [J. Galeotti, et al., 2011]. New cameras and lenses will be used to improve the performance of both stereopsis and tracking. We have recently purchased four Prosilica GT cameras produced by Allied Vision Technologies. These cameras have 1.2 Megapixel 2/3-format sensors with a Gigabit Ethernet interface and offer precise iris lens control. We will synchronize these digital cameras using an Ethernet protocol and thus get higher quality stereo images from them. Also, as we mentioned in the end of section 3.4, we plan to use the result coming from ThreeRivers 3D, Inc.'s

scanner, or some other commercially available scanning system, to build the ground truth mapping first, and then use Probe-Sight to do real-time visualization based in part on the ground true data in order to get more accurate performance.

We believe this research represents important preliminary results towards a clinically useful approach in merging visual and ultrasound data in real time as the ultrasound probe is moved over the surface of the patient. Our immediate goal is to use Probe-Sight on normal human volunteers, scanning their arms, wrists, and hands. We will develop a brace with attached optical markers, to hold the upper extremity in constant position from one scan to the next for clinical trials. Then we will test the device on hand transplant patients and help surgeons to better understand the exterior and interior information, both alone and with CT and MRI data. Our ultimate goal is to merge these systems into our existing Sonic Flashlight as shown in Figure 5.1 to provide a system that can combine the ultrasound and visual data in both the mind of the operator and in the software of the computer. We then intend to explore how the human and the machine can help each other to understand what each is seeing, for the benefit of the patient [S. Horvath, et al., 2011].

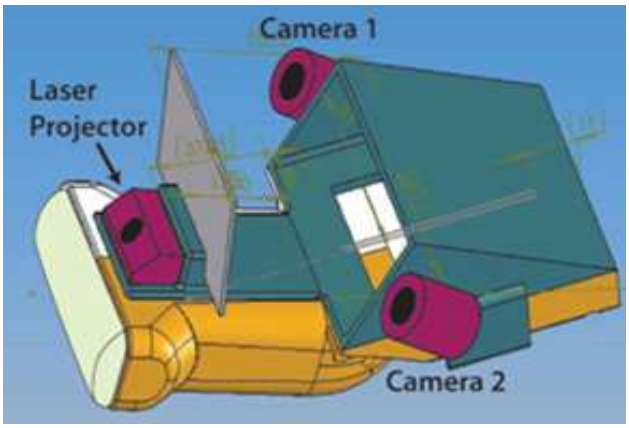


Figure 5.1: Plan for stereo cameras and a laser projector on the Sonic Flashlight.
(Image from S. Horvath's paper *Towards an Ultrasound Probe with Vision: Structured Light to Determine Surface Orientation*, 2011.)

Bibliography

M. Bajura, H. Fuchs, R. Ohbuchi. Merging virtual objects with the real world: seeing ultrasound imagery within the patient. In *Proceedings of the 19th Annual Conference on Computer graphics and Interactive Technique, ACM SIGGRAPH,* 26(2): 203-210, 1992.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11):1222–1239, 2001.

G. Bradski and A. Kaehler. Learning OpenCV: Com-puter Vision with the OpenCV Library. In O'Reilly. Media, Inc, 2008.

G. Brandacher, V. Gorantla, W.P. Lee. Hand Allotransplantation, *Seminars in Plastic Surgery*. 24(1): 011-017, 2010.

D. Brown, Close-range camera calibration. *Photogrammetric Engineering* 37: 855–866, 1971.

D. Brown, Decentering Distortion of Lenses. *Photometric Engineering* 32(3):444–462, 1966.

C. Chan, F. Lam, R. Rohling. A needle tracking device for ultrasound guided percutaneous procesdures. *Ultrasound in medicine & biology*. 31(11): 1469-1483, 2005.

W. Chang, N. Amesur., R. Klatzky. G. Stetten. The Sonic Flashlight Is Faster than Conventional Ultrasound Guidance to Learn and Use For Vascular Access on Phantoms. *Radiology*, 241(3), 771-779, 2006.

P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41-54, 2006.

J. Fryer and D. C. Brown, Lens distortion for close-range photogrammetry. *Photogrammetric Engineering and Remote Sensing* 52: 51–58, 1986.

J. Fung and S. Mann, Using graphics devices in reverse: GPU-based Image Processing and Computer Vision. *Multimedia and Expo*, 9-12, 2008.

A. Fusiello, E. Trucco and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Application* 12(1):16-22, 2000.

J. Galeotti, J. Wang, S. Horvath, M. Siegel, G. Stetten, ProbeSight: Video Cameras on an Ultrasound Probe for Computer Vision of the Patient's Exterior, Image Guided Therapy Workshop, Arlinghton, VA, 2011.

V. Gorantla, G.Brandacher, S. Schneeberger, X. Zheng, A. Donnenberg, J. Losee, W. Lee. Favoring the risk-benefit balance for upper extremity transplantation--the Pittsburgh Protocol. *Hand Clinics* 27(4):511-20, 2011.

K. Graham, E. MacKenzie, P. Ephraim, T. Travison, R. Brookmeyer. Estimating the prevalence of limb loss in the United States: 2005 to 2050. *Arch Phys Med Rehabil* 89(3):422–29, 2008

S. Hawke, L. Davies, R. Pamphlett, Y. Guo, J. Pollard, J. Mcleod. Vasculitic Neuropathy: A clinical and pathological study. *Oxford Journals (Brain).* 114:2175-90, 1991.

H. Hirschmuller. Stereo Processing by Semiglobal Matching and Mutual Information.In *IEEE Transactions on Pattern Analysisi and Machine Intelligence*, 30(2):328-341, 2008.

S. Horvath, J. Galeotti, B. Wang, M. Perich, J. Wang, M. Siegel, P. Vescovi, G. Stetten, Towards an Ultrasound Probe with Vision: Structured Light to Determine Surface Orientation, In *Augmented Environments for Computer Assisted Interventions, MICCAI*, 2011.

J. Kettenbach, G. Kronreif, M. Figl, et al. Robot-assisted biopsy using ultrasound guidance: Initial results from in vitro tests. *Eur Radiol*, 13(supp 1):576, 2004.

K. Konolige. Small vision system: Hardware and implementation. In *Proceedings of the International Symposium on Robotics Research* 111–116, Hayama, Japan, 1997.

P. Milgram, H. Takemura, A. Utsumi, F. Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. In *International Society for Optics and Photonics,* 2351:282-292, 1994.

I. Minin and O. Minin. *Ultrasound Imaging - Medical Applications*. InTech, Rijeka, Croatia, 2011.

National Limb Loss Information Center. Fact sheet: Limb loss in the United States. Knoxville (TN): Amputee Coali-tion of America; 2007 [updated 2008 Sep 18; cited 2009 Feb 25]; Available from: http://www.amputee-coalition.org/fact_sheets/limbloss_us.html.

NVIDIA Corporation, NVIDIA CUDA compute unified device architecture programming guide, Jan. 2007, http://developer.nvidia.com/cuda.

R. Ohbuchi, D. Chen and H. Fuchs. Incremental volume reconstruction and rendering for 3D ultrasound imaging. In *Proceedings of SPIE,* 1808: 312-23, 1992.

OpenCV Yahoo Group Message 80664, “BM method no accelaration in my laptop”, May, 2011, http://tech.groups.yahoo.com/group/OpenCV/message/80664.

P. Petruzzo, M. Lanzetta, JM. Dubernard, et al. In *The International Registry on Hand and Composite Tissue Transplantation*. 86:487-492, 2008.

L. Pezzin, T. Dillingham, E. Mackenzie, P. Ephraim, P. Rossbach. Use and satisfaction with prosthetic limb devices and related services. *Arch Phys Med Rehabil*. 85(5): 723-29, 2004.

G. Poggio and T. Poggio. The Analysis of Stereopsis. Annual Review of Neuroscience. 7:379-412 ,1984.

R. Prager, U. Ijaz, A. Gee. G. Treece. Three-dimensional ultrasound imaging. In *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine.* 224: 193-223, 2010.

A. State, M. Livingston, W. Garrett , G. Hirota, M. Whitton, H. Fuchs. Technologies for Augmented Reality Systems: Realizing Ultrasound-Guided Needle Biopsies. In *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, 1996.

G. Stetten, V. Chib, R. Tamburo. Tomographic Reflection to Merge Ultrasound Images with Direct Vision, In *IEEE Proceedings of the 2000 Applied Imagery Pattern Recognition (AIPR) Annual Workshop*, 200-205, 2000.

R. 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* 3: 323–344, 1987.

J. Wang, S. Horvath, G. Stetten, M. Siegel, J. Galeotti, Real-Time Registration of Video with Ultrasound using Stereo Disparity, SPIE Medical Imaging, San Diego, 2012.

P. Wunderbaldinger, TH. Helbich, B. Partik, K. Turetschek, G. Wolf. First experience with a new dedicated ultrasound system for computerguided large-core breast biopsy. *Eur Radiol* 11(12):2460–2464, 2001.

Q. Yang, L. Wang, and N. Ahuja. A constant-space belief propagation algorithm for stereo matching. In *Computer Vision and Pattern Recognition* (*CVPR)*, 1458-1465, 2010.

R. Zhang, P.-S. Tsi, J. E. Cryer, and M. Shah. Shape form shading: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:690 –706, 1999.

Z. Zhang, Iterative Point Matching for Registration of Free-Form Curves and Surfaces, *International Journal of Computer Vision*, 13:2, 119-152, 1994.

Z. Zhang. A flexible new technique for camera calibration.In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 2000

K. Ziegler-Graham, E. MacKenzie, P. Ephraim, T. Travison, R. Brookmeyer. Estimating the prevalence of limb loss in the United States: 2005 to 2050. *Arch Phys Med Rehabil*. 89(3):422-29, 2008.