Stony Brook University

Robot-Assisted 3D Mapping of Unknown Cluttered Environments

A Thesis Presented

by

Zhe Zhang

to

The Graduate School

in Partial fulfillment of the

Requirements

for the Degree of

Master of Science

in

Mechanical Engineering

Stony Brook University

December 2007

Stony Brook University

The Graduate School

Zhe Zhang

We, the thesis Committee for the above candidate for the

Master of Science degree,

hereby recommend the acceptance of this thesis.

Prof. Goldie Nejat, Ph.D., Advisor

Department of Mechanical Engineering, Stony Brook University

Prof. Q. Jeffrey Ge, Ph.D., Chair

Department of Mechanical Engineering, Stony Brook University

Prof. Wendy Tang, Ph.D. Member

Department of Electrical Engineering, Stony Brook University

This thesis is accepted by the Graduate School.

Lawrence Martin

Dean of the Graduate School

Abstract of the Thesis

Robot-Assisted 3D Mapping of Unknown Cluttered Environments

By

Zhe Zhang

Master of Science

in

Mechanical Engineering

Stony Brook University

2007

In this thesis, a unique landmark identification and matching method is proposed for identifying and matching distinguishable landmarks for 3D Visual Simultaneous Localization and Mapping (SLAM) in unknown cluttered Urban Search and Rescue (USAR) environments. The novelty of the method is the utilization of both 3D (i.e., depth images) and 2D images. By utilizing a Scale Invariant Feature Transform (SIFT) -based approach and incorporating 3D depth imagery, more reliable and robust recognition and

matching of landmarks from multiple images for 3D mapping of the environment is achieved. Landmarks are determined effectively within the images utilizing a combination of SIFT keypoints, depth segmentation, edge detection and morphological techniques and a convex hull algorithm. These landmarks are matched through out the scene and used by the proposed Visual SLAM methodology for 6 degrees-of-freedom robot localization and for creation of a 3D virtualized map of USAR environments with respect to a world frame. Experiments presented herein utilizing the proposed methodology verify: (i) its ability to identify clusters of SIFT keypoints in both 3D and 2D images for representation of potential landmarks in the scene, and (ii) the use of the identified landmarks in constructing a 3D map of unknown cluttered USAR environments. Furthermore, conclusions on the proposed methodology, highlighting the contributions and future work are presented.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to first thank my advisor Professor Goldie Nejat for her kind guidance for my research work and the thesis. I would also like to thank Professor Peisen Huang and his students Hong Guo and Xu Han for assisting in taking the sensory data for this work and the discussions on this work. Also, thank you to Alexander Reben for development of the robotic platform that was utilized as a test-bed in this work. Lastly, I would like to thank all of my labmates through out the two years of my study.

Chapter 1 Introduction

1.1 Motivation

The catastrophic earthquakes that hit northern and southern California in 1989 and 1994, Kobe, Japan in 1995 and the Izmit region in Turkey in 1999, and the terrorist attacks on the World Trade Centers in 2001 have clearly demonstrated the need for specially trained resources to respond to incidents of partial or complete structural collapse caused by these types of major disasters. Urban search and rescue (USAR) is defined to be the emergency response function which deals with the collapse of man-made structures [1]. It involves the location, rescue, and initial medical stabilization of victims trapped in confined spaces which are quite dusty and dark. Structural collapse by natural disasters and human-caused accidents is most often the cause of victims being trapped, but victims may also be trapped in transportation accidents, mines and collapsed trenches. In both human-caused and natural disasters, the fundamental tasks at hand are: (i) to find and rescue victims in the rubble or debris as efficiently and safely as possible, and (ii) not to further endanger the survivors or put human rescue workers' lives at great risk. With the advancement of robotic research in recent years, rescue robots have been developed to address these particular conundrums and to lessen the burden on the rescue

workers. Rescue robotics has been identified by both the National Research Council (NRC) [2] and the Computing Research Association [3] as a critical technology.

One of the first publicized use of USAR robots occurred after the World Trade Center (WTC) disaster, where six small robots were utilized to enter the scene through voids too small or deep for a person [4]. Furthermore, the robots were used to survey larger voids that people were not permitted to enter due to fire or structure instability. The various robots carried cameras and thermal imagers into the interior of a rubble pile. Due to environmental limitations, in particular the challenging terrain and the high heat sources within the rubble piles, the robots were very restricted in the tasks that they could accomplish. Nonetheless, this deployment demonstrated the potential of utilizing robots in USAR environments and the need for advanced technologies to aid in this robotic application.

1.2 Research Problem Statement

There are a number of challenges that roboticists must face in designing a USAR robot: (i) locomotion, (ii) sensing, (iii) power, and (iv) size [1]. The majority of USAR robots are far from being autonomous, they are tethered to a power supply and are tele-operated by humans with minimal sensory information. In particular, major advances are needed in sensor techniques and sensor information interpretation for two main tasks: (i) victim identification, and (ii) navigation of the robot. The objective of *this research work is to address the aforementioned sensory needs by proposing the development and integration of a sensory system for robot-assisted 3D mapping of USAR environments.*

Prior to a more detailed description of the research problem at hand, a brief review of the pertinent literature is provided.

1.3 Literature Review

The pertinent literature is reviewed herein in two main parts: (i) Rescue Robots and (ii) 3D Range Sensors.

1.3.1 Rescue Robots

Rescue robots for USAR environments have the ability to navigate through tightly confined spaces which people or dogs cannot access easily. They can be risked in searching for survivors in unstable structures and confined spaces which allow them to assess structural damage in remote locations. They can map the area and identify the location of victims to direct the rescue workers, guide the insertion of tools to aid extrication and shoring, and identify the location of limbs to prevent workers from damaging a victim's arm or leg with rescue equipment [4]. With their potential abilities, rescue robots can serve as a resource to the rescue workers and keep them out of harms way. Rescue robots have varying size, shape, weight, mobility, communication, power needs and sensing capabilities. This section provides a general introduction of rescue robots categorizing them based on their mobility techniques.

Wheeled-Robots

Figure 1-1(a) illustrates the Scout robot, a wheeled-robot developed by University of Minnesota, [5]. The Scout robot is a specialized robot capable of carrying out low-

level, usually parallel tasks. Scouts can include simple sensory units or varying locomotion units. The original Scout, Figure 1-1 (a) has a body approximately 11cm long, and 4cm in diameter (the special foam wheels can expand to 5cm in diameter). This body fits snugly inside a protective covering called a Sabot that absorbs much of the impact during the launch, and allows the Scout to even break through a glass window and land safely and ready to begin its mission. The Actuating Wheel Scout in Figure 1-1 (b) improved upon the original mechanical design to allow the robots actuators to range from 3.9 cm to 12 cm in diameter.

Fig. 1-1 (a): UMN Scout Robot [5].

Fig.1-1 (b): Actuating Wheel Scout [5].

This robot is one example of many wheeled robots that exist today. This vehicle-like robot is easy to design and implement, but may have problems to climb over big obstacles in cluttered environments.

Tracked-Robots

The Packbot is a wireless, suitcase-sized, tracked vehicle shown in Figure 1-2 [4]. The sensor suite consists of an 84-degree field of view low light camera, 118-degree field of view color CCD camera, and an optional Indigo Alpha FLIR. The Packbot is waterproof up to 3m depth and is self-right-able with fippers. The fippers also provide the ability to increase the height of the camera on top of the robot and provide better mobility.

Fig. 1-2: The Packbot by iRobot [4].

Another tracked robot is the Micro-VGTV (Variable Geometry Tracked Vehicle), shown in Figure 1-3, which can alter its shape during operation [4]. The tracks, in their lowered configuration, take the shape of conventional crawler tracks. When the geometry is varied to the point where the vehicle is in its raised configuration, the tracks take the shape of a triangle. The Micro-VGTV remains fully operational throughout these shape alterations and as a result, can continue to travel and maneuver while its configuration is being changed. This unique feature allows the vehicle to negotiate obstacles and operate in confined spaces and over rough terrain. The complete system is easily transported and managed by a single operator [4].

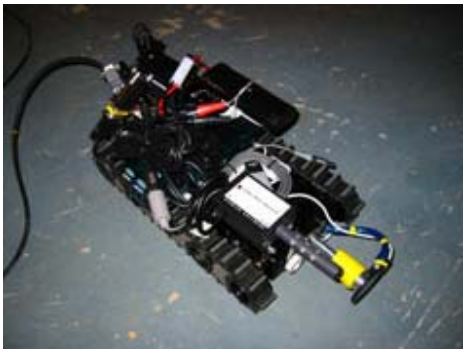

Fig. 1-3: Inuktun's Micro-VGTV [4].

However, the robots are not energy efficiency and are relatively large leading to mobility problems in cluttered environments.

Serpentine Robots

A serpentine robot is a relatively new kind of robotic mechanism, which may be able to provide a solution for inspections and surveillance in USAR environments. The University of Michigan has developed a snake-like robot “OmniTread” [6] shown in Figure 1-4 that conquers obstacles. It is composed of 5 segments of 8-inch diameter each and weighs 26 pounds. It is currently piloted by a human operator. And it can maneuver in extremely rugged terrain, climbing stairs and pipes. This kind of serpentine robot is slender, multi-segmented vehicles designed to provide greater mobility than conventional wheeled or tracked robots. The OmniTread serpentine robot was tested at the Small Robotic Vehicle Test Bed at Southwest Research Institute (SwRI).

Fig. 1-4: University of Michigan’s OmniTread Serpentine Robot [6].

Another wheeled-snake combined robot by Carnegie Mellon University [4] shown in Figure 1-5 has many more degrees of freedom than conventional robots and rescue machinery; while at the same time having a small cross-sectional area. These many degrees of freedom enable snake robots to thread through tightly packed volumes reaching locations otherwise inaccessible to conventional robots and people, while at the

same time, not disturbing the surrounding areas. This is critical in search and rescue operations where large pieces of debris become fragile.

Fig. 1-5: CMU's Snake Robot with wheels [4].

Snake-like robots have a lot of flexibility and easy to move in cluttered environments filled with many rocks and dusts. Their size is relatively small, which enables them to enter holes where wheeled-robots and tracked-robots cannot fit. However, all the segments of a snake robot have to carry their own computers and batteries. They are radio-controlled since addressing autonomous control is very difficult for these types robot.

1.3.2 Current Sensors for Providing 3D Range Information

Most robots' relationship to their environments is limited by sensor technologies and cost, where their location in the environment, the layout of the environment, and the presence of victims is usually extracted from a single 2D video camera [1]. Furthermore, all robots that operate in USAR environments do not have any a priori information about landmarks in the scene and due to the nature of the surroundings cannot employ GPS. A robot operator in USAR environments faces the important tasks of remembering,

recognizing and diagnosing a scene and how the robot and its camera are positioned and oriented within the scene merely from this camera. Often times, this leads to disorientation, the robot getting stuck, and not being able to identify victims that are present in the scene.

Since sensing is always a big issue for search and rescue robots, different kinds of sensory system using different types of sensors have been proposed to utilize in USAR environments. Stereovision is probably the most studied method [7] among all the existing sensory techniques; 3D cameras based on time-of-flight technologies are quite compact and real-time to use; laser scanner consists of using a laser light source that sweeps a thin laser stripe across a scene. However, it is slow and requires a lot of time for scanning in spite of the fact that it could be quite accurate. This section gives a brief literature review for current sensors for providing range information.

Stereovision

Among all the existing sensory techniques that can be potentially used for mapping, stereovision is probably the most studied method. A stereo camera is the prime example of a passive optical triangulation system. Traditional stereovision methods estimate shape by establishing spatial correspondence of pixels in a pair of stereo images. Determining the correspondences between left and right view by means of image matching, however, is a slow process. Furthermore, for 3D reconstruction, passive stereovision techniques depend heavily on cooperative surfaces, mainly on the presence of surface textures, and on ambient light [7]. Such texturing is absent in USAR environments where the surroundings are dark and covered in gray dust. Recently, Zhang et al., [8] developed a new concept called spacetime stereo, which extends the matching of stereo images into

the time domain. By using both spatial and temporal appearance variations, it was shown that matching ambiguity could be reduced and accuracy could be increased. As an application, Zhang et al. demonstrated the feasibility of using spacetime stereo to reconstruct the shapes of small dynamically changing objects. The shortcoming of spacetime stereo is again the requirement of the time-consuming task of matching of stereo images. Therefore, it is difficult to reconstruct high-resolution 3D shapes from stereo images in real-time.

3D Cameras

Recently, 3D cameras based on time-of-flight technologies have also been developed for the gathering of 3D data, [9-11]. The camera systems mainly consist of a modulated light source, in most cases infrared or near infrared [9,10] and a CMOS/CCD image sensor. However, the pixel array size of these systems are limited and hence the resolution of both the 3D depth and 2D grey-scale images can be low, where in particular: (i) in the 3D images (sporadic) noise can be easily detected and can also increase as the distance from the camera to the scene increases, and (ii) nonlinear distortions caused by lens effects can be present, making it difficult to distinguish different objects in the scene. In order to utilize 3D cameras in USAR environments, changes to the current hardware and software components of the system would be required to minimize image corruption and hence increase accuracy [12].

Laser Scanners

Laser scanning technology consists of using a laser light source that sweeps a thin laser stripe across a scene. Simultaneously, a light sensor, i.e. camera, acquires the scene,

where the surface of the scene is measured via triangulation, [i.e., 13] or time-of-flight, [i.e., 14]. The main disadvantage of using these systems for robotic 3D mapping of USAR environments is that they are slow and require a lot of time for scanning, due to the fact that the laser stripe has to be physically moved across the scene to digitize the surface, and hence cannot provide real-time range data acquisition. Other disadvantages of laser scanners are that they are expensive due to the high cost for production of their hardware components (i.e., costs are in the range of several tens of thousands), they are bulky and heavy for a small robot, and they can produce a variety of wrong points in the vicinity of edges due to the fact that when the laser hits an object edge, only a part of it will be reflected there and the rest may be reflected from adjacent or rear surfaces or may not reflect when no other object is present within the possible range of the scanner.

1.4 Proposed Methodology and Research Tasks

The overall proposed methodology comprises the following components with corresponding reference to the Dissertation Chapters:

1. *3D sensory system for robotic mapping and localization*

   In Chapter 2, a detailed literature review of the two main research topics of this work is presented: (i) current sensing technologies for USAR environments and (ii) 3D Simultaneous Localization and Mapping (SLAM) in cluttered unknown environments. The first application of using a 3D sensory system for sequential map building within 3D Visual SLAM framework will be described in detail.

2. *Landmark Identification and Matching*

When traveling in 3D cluttered environments, data association (i.e., landmark identification and matching) becomes a pertinent problem.. In Chapter 3, the developed SIFT-based landmark identification and matching techniques will be described in regards to: (i) 3D image conditioning, and (ii) SIFT-Based Recognition of Landmarks.

3. *3D Visual SLAM*

Visual SLAM is implemented for creation of a 3D virtualized map of USAR environments with respect to a world frame. In Chapter 4, both the proposed 3D SIFT-based: (i) 6 DOF ego-motion methodology for robotic localization, and the (ii) Iterative Closest Point (ICP) 3D mapping method utilized for stitching of 3D images of the USAR scene are described.

4. *Implementation*

Chapter 5 will present all the experimental results utilizing the proposed methods in USAR-like environments. The 2D and 3D images for this system are generated by a structured light based sensor.

Finally, Chapter 6 presents conclusions on this research work, highlighting its contributions and future work.

Chapter 2 Literature Review

2.1 Sensing for Urban Search and Rescue (USAR)

The majority of USAR robots are far from being autonomous for the reasons that they are tethered to a power supply and are tele-operated by humans with minimal sensory information. One of the biggest issues is sensing especially in our USAR environments. Major advances are needed in sensor techniques and sensor information interpretation for two main tasks: (i) victim identification, and (ii) navigation of the robot. Most robots' relationship to their environments is limited by sensor technologies and cost. Furthermore, all robots that operate in USAR environments do not have any a priori information about landmarks in the scene and due to the nature of the surroundings cannot employ GPS. While chapter 1 compares three general sensing techniques for 3D range information, this section presents the sensory systems that have been developed or applied to USAR applications.

2.1.1 2D Laser Range Finder

A hand full of research projects have been proposed for the development of laser range finders for cluttered USAR environments. In [15], Kurisu et al. proposed the use of two different laser range finders for 3D mapping of rubble: (i) a ring of laser beam

module and an omnivision CCD camera, (ii) and an infrared laser module with a CCD camera to capture the laser image and another camera for capturing the texture. The optimal range of this system was determined to be 300 mm. There are two main limitations to these types of sensors: (i) they do not address real-time range data acquisition, and (ii) their reliance on robot internal sensors for mapping, in particular they can only measure in the x,y plane, the z-direction measurement for the 3D information is based on the robot's inaccurate internal sensors.

The integration of sonar and 2D laser range finder has been utilized for robot mapping, collision avoidance and path planning in USAR environments, i.e., Aboshosha et al. [16]. In this work, an electrostatic sonar transducer was used because of its advantages such as low cost, small size, and low power consumption. In addition, a SICK LMS 200 laser scanner mounted on a robot has been used to gather the odometric data of an environment by 2D slice scanning. The 2D laser scanner could get consistent high precision maps with low memory requirements with the vector mapping algorithm to. The generated maps then were used as a base for an autonomous path planning algorithm depending on the straight line navigation (SLN) algorithm. The main objective of the integration of these sensors is to reinforce the robustness of the overall system, overcome the sensors' disadvantages, and improve the performance of the overall system. The algorithms have been verified by simulation and real experiments. The method has a small storage size and high precision compared with the traditional mapping algorithms. It improved the performance of the overall system compared with previous similar methods, but no real USAR environments have been utilized to verify the effectiveness of this sensory integration system.

2.1.2 3D Cameras

Recently, 3D range cameras based on the time-of-flight technologies have also been developed for the gathering of 3D data, [9-11]. The camera systems mainly consist of a modulated light source, in most cases infrared or near infrared [9,10] and a CMOS/CCD image sensor. These 3D cameras have been preliminary tested for USAR environments. In Murphy et al. [12], a Canesta EP200 series of range camera was used at the University of South Florida's robot test bed to obtain depth images of a USAR-like scene. The camera provides a 64-by-64 pixel range map and corresponding black and white image in real time with the size of 12.7cm wide × 5.08cm tall × 5.08cm deep. The maximum unambiguous range (resolvable distance) of the camera is 11.5m. The maximum depth resolution is approximately 5mm and is achieved with the minimum unambiguous range of 1.44m. Their results presented three main limitations for this type sensor: (i) over saturation of the CCD in direct sunlight, (ii) inability to detect certain materials, and (iii) sporadic noise when used in indirect sunlight. The third can be solved by using image processing methods, while the first two require hardware improvements. Ellekilde et al. [17] also utilized a 3D camera, the Swiss Ranger SR-2 range camera, to provide visual information of an indoor USAR environment. In general, current 3D cameras are found to be low resolution, too noisy and subject to substantial changes in illumination as the camera pose changes. Filters and image processing methods are needed for further application.

2.1.3 Integration of Various or Multiple Sensors

Various types of sensors can be implemented for robotic USAR environments for: (i) obstacle avoidance, (ii) location estimation, and (iii) victim detection. Pissokas et al. [18] present an overview of the sensing techniques that can be used for the aforementioned tasks. The feasibility of the sensors was tested at the RoboCup Rescue 2001 competition. In particular, ultrasonic range sensors and bumpers were used for obstacle avoidance; for location estimation wheel encoders and a magnetic compass were utilized to provide translational and orientation information; and a pyroelectric sensor was utilized for body heat detection and a microphone for voice detector. This preliminary work lacks large scale simulation or experiments in real USAR environments where noise and clutter are a major issue.

The utilization of multiple sensors for a team of robotic platforms and sensor agents for Robot-Sensor Networks has been explored, i.e., Reigh et al. [19]. In this work, a team of heterogeneous agents are considered in which a potentially very large number of small, simple, sensor agents with limited mobility are deployed by a smaller number of larger robotic agents with limited sensing capabilities but enhanced mobility. The sensor agents provide the robots with target information. The key challenge is to reconfigure the network automatically, as robots move around and sensors are deployed within a dynamic, potentially hazardous environment, while focusing on the two high-level goals: (i) to map the space in three dimensions using a local, relative coordinate frame of reference; and (ii) to identify targets within that space. Maintaining information flow throughout the robot-sensor network is vital. In addition, the size of sensors within the network and the processing rate of sensing information is an important factor for practical

use. A network routing scheme is utilized to route the system's communications and the movement of its mobile components via a Distributed Vector (DV) routing algorithm for 2D mapping of obstacles in the environment.

2.2 Simultaneous Localization and Mapping (SLAM) for USAR

In order to map its environment, the robot must be able to determine where it is in relation to its surroundings. Due to the increase in uncertainty over time, robot sensors such as odometers are not sufficient for such a task. In indoor environments, usually the robot is mapping scenes in which known landmarks exist; hence the location of these landmarks can be utilized in order to localize the robot. In outdoor environments, accurate sensors such as GPS can be utilized to determine the location of the robot. However, all robots that operate in USAR environments do not have any a priori information about landmarks in the scene and cannot employ GPS or radio positioning due to the nature of the surroundings (i.e., inside cluttered collapsed buildings). Furthermore, what make USAR environments even more unique are the characteristics of the uneven terrain. Hence, a localization algorithm is crucial while mapping the unknown site. The simultaneous localization and map building (SLAM) problem addresses the question: Is it possible for an autonomous vehicle to start in an unknown location in an unknown environment and then incrementally build a map of this environment while simultaneously using this map to compute absolute vehicle location. A number of different solutions have been developed in order to address the SLAM problem, (i)

Extended Kalman Filter (EKF) based methods [20], (ii) Particle Filter methods, such as FASTSLAM and DP SLAM [21], and (iii) Submap based methods [22].

Some attempts have been made to directly formulate SLAM for rescue environments, [23,24]. However, there are still a number of issues that need to be addressed in order to effectively implement these methods in 3D unknown cluttered environments. In [23], Ishida et al. utilized a 2D laser scan matching-based SLAM method. A robot using a rotating 2D laser range finder is assumed to explore an environment with flat ceilings. Sphere digital elevation maps (SDEM) are used to represent local maps from the sensor information. A global map is then created using several SDEMs and the relative locations among them. The orientation of the robot is determined from the relationship of the angles to the normal vector of the robot's altitude plane. The robot's yaw orientation is extremely difficult to estimate based on this normal vector, hence leading to errors in localization. Furthermore, the assumption of the environment having a flat ceiling limits the method's application. In [24], Yokokohji et al. have conducted some preliminary work on 3D SLAM assuming known data association. Two different laser range finders have been utilized for mapping rubbles. The first laser range finder consists of a ring of laser beam module and an omnivision CCD camera. The second sensor utilizes an infrared laser module with a CCD camera to capture the laser image and another camera for capturing the texture. There are two main limitations to these types of sensors: (i) they do not address real-time range data acquisition, and (ii) they rely on robot internal sensors for mapping, in particular they can only measure in the *x*,*y* plane, the *z*-direction measurement for the 3D information is based on the robot's inaccurate internal sensors. Based on robot accelerations and 2D range measurements

from the laser range finders, an EKF is utilized for system state estimation. Herein, due to the nature of the range sensor, only 2D positions of the landmarks can be measured, thus, relying on inaccurate robot sensory information for the third coordinate. In simulations, the utilized algorithm has proven to be quite sensitive to measurement and initial errors.

Only recently interest has increased in utilizing cameras for SLAM applications, known as Visual SLAM [25]. Cameras are more affordable and compact than their laser counterparts and can be used to provide 3D range information. Furthermore, they have a higher rate of acquisition and high angular resolution. The disadvantage of utilizing stereovision is that matching of stereo images is usually time-consuming, hence making it difficult to reconstruct a 3D map in real-time for USAR environments.

Recently, attempts have also been made in the literature to develop methods for identifying distinctive invariant features from images that can be used to perform matching of objects from different views. One particular method is Scale Invariant Feature Transform (SIFT) developed by Lowe in [26]. This approach transforms an image into a large collection of local feature vectors, each of which is invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. The resulting feature vectors are called SIFT keys. This method has been utilized effectively on 2D grayscale images to identify and match invariant features. Moreover, it works efficiently for object recognition problems where a training image of the object of interest is given. In [27], Se et al. have proposed a vision-based SLAM method by tracking SIFT keys on 2D images and building a 3D map simultaneously utilizing a trinocular stereo system for an indoor environment, where the robot moves in an approximate 2D planar motion. In [17], Miro et al. proposed a

stereovision based EKF SLAM algorithm utilizing features extracted on 2D images from SIFT in indoor scenes.

2.3 Human-Robot Interface

In difficult USAR environments, it is not yet possible to create a fully autonomous search and rescue robot to completely take the place of a human rescue worker. In fact, most USAR robots that are sent into disaster zones are tele-operated. For this application, operators must have a good understanding of their surroundings, yet it is difficult to obtain situation awareness due to disorientation, the robot getting stuck, and not being able to identify victims that are present in the scene. Many researchers have attempted to improve user interfaces for robot operators for effective human-robot interaction. An example of an interface developed by Baker et al. [28] is shown in Figure 2-1. This interaction can be utilized with robots containing numerous sensors such as described by Pissokas et al. [18]. Even though the utilization of these interfaces can assist in aiding the human operator, their use does not address the robot autonomy problem which is one of the main contributions of this thesis.

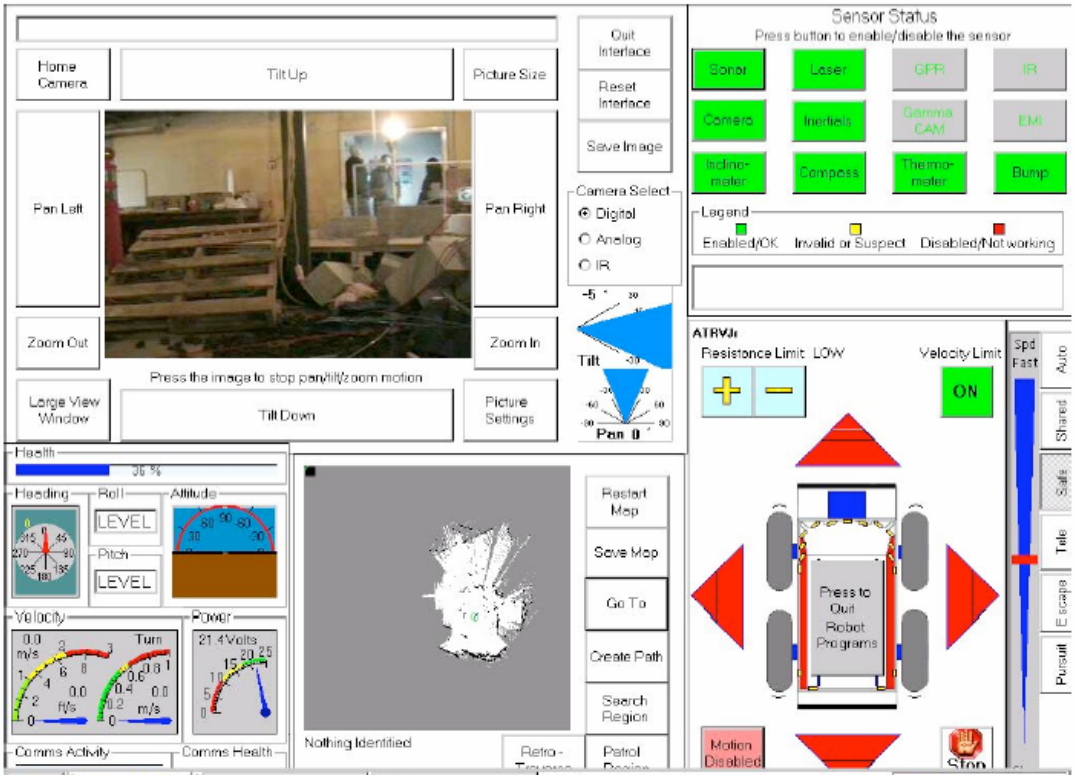

Fig. 2-1: Baker's interface for their robotic system.

2.4 USAR Simulated Environments

USAR researchers build simulated challenging and reproducible environments to evaluate mobile robot capabilities and behaviors. NIST's Reference Test Arenas for Autonomous Mobile Robots [29] is one of the most impressive USAR environments in the world. There are three separate indoor arenas, each labeled by a color, forming a continuum of difficulty for robots. The Yellow arena is the easiest to traverse (Figure 2-2): it consists of a plane maze with a variety of passages. It has doors, blinds, and simple collapses to block passages during missions, specifically mapping and localization algorithms. The Orange arena provides more difficult challenges for both sensing and agility (Figure 2-3): it consists of an elevated floor section where the only way to get to it

is via ramp, stairs, or ladder. Holes in the elevated flooring provide negative obstacles to avoid. The Red arena provides the least structure and the most challenge to robot agility (Figure 2-4): it consists of a rubble pile with assorted debris, i.e., steel wire, gravel, plastic bags, pipes, throughout the arena. The long-term plan or future work of this research is to potentially test our developed system in this arena. All the proposed methodologies of this thesis are developed keeping in mind the nature of these types of environments.

Fig. 2-2: The Yellow Arena.

Fig. 2-3: The Orange Arena.

Fig. 2-4: The Red Arena.

Chapter 3 Landmark Identification and Matching

When traveling in 3D cluttered environments, data association (i.e., landmark identification and matching) becomes a pertinent problem. In particular, there could exist many repetitive features. As the robot moves, it must be able to determine whether different sensor measurements correspond to the exact same landmark in its environment. In most cases presented in the literature, the SLAM problem has been addressed under known data association [30]. However, in most situations this is definitely not the case. Furthermore, incorrect data association can induce extreme errors in SLAM solutions. By incorporating 3D grayscale depth imagery, we will be able to use more reliable and robust recognition and matching between landmarks from different images, therefore minimizing false matches. If an object in the foreground of an image is similar in intensity to the background, it is difficult to determine its boundaries. The use of depth images solves this problem, since a foreground object will always be at a closer depth, and can therefore be easily detected and identified as a potential landmark. A real-time structured light 3D shape measurement system [31] based on a digital fringe projection and phase shifting technique is utilized to obtain the depth images shown in this chapter. A DLP projector projects fringe patterns with the frequency of 360Hz, and the B/W high

speed CCD camera synchronized with the DLP captures the fringe images at the frequency of 90Hz. Each frame of the 3D shape is reconstructed using three consecutive fringe images. Together with the fast 3D reconstruction algorithms and parallel processing software, high-resolution real-time 3D grayscale depth imagery is realized at a frame rate of up to 30 3D frames per second and a resolution of 532×500 points per frame. In this section the main components of the proposed landmark identification and matching are described.

3.1 Image Conditioning

3.1.1 Background Subtraction

The objective of background subtraction is to eliminate background noise in the images to better define edges of the foreground landmarks of the scene. As is noted in Figure 3-1(a), foreground objects in the 3D grey-scale depth image are presented in lighter gray shades, whereas background objects or noise is presented in darker shades. Our objective is to determine and extract the boundaries of the foreground objects. The advantage of using background subtraction over other boundary extraction methods such as edge detection is that continuous boundaries can be easily defined. Background subtraction regenerates the 3D grayscale image into a binary image in which 1 represents the foreground objects (defined as white objects in Figure 3-1 (b)) and the background is represented by 0 (defined as the black region in Figure 3-1 (b)) in black. [32].

In order to determine what objects in the scene should be considered as the required foreground information, a threshold depth value can be set to separate these

objects from the background in the 3D image. The depth value of the threshold is determined based on the environment of interest.

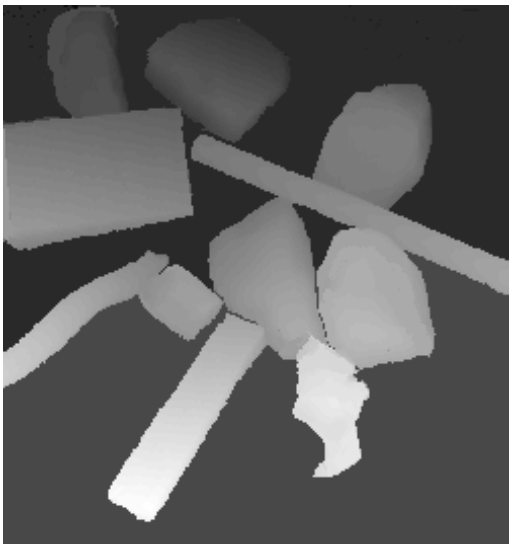

Fig. 3-1 (a): Grey-scale depth image

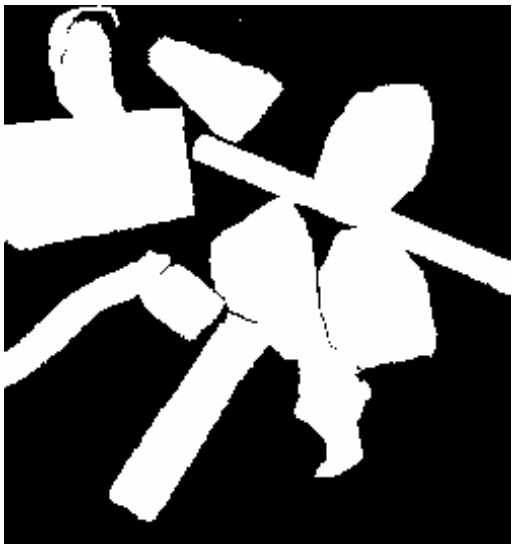

Fig. 3-1 (b): Binary image.

3.1.2 Edge Detection

There are many edge detection algorithms to determine potential boundaries of objects in the scene. Algorithms including Prewitt, Roberts, LoG and Canny-Deriche methods have been tried to select an optimal one for our application.

The Prewitt method finds edges using the Prewitt approximation to the derivative. It returns edges at those points where the gradient intensity of the input image is maximum [33]. The Roberts method finds edges using the Roberts approximation to the derivative. It returns edges at those points where the gradient intensity of the input image is maximum [33]. The Laplacian of Gaussian (LoG) method finds edges by looking for zero crossings after filtering the input image with a Laplacian of Gaussian filter [33].

The Canny-Deriche method finds edges by looking for local maxima of the gradient of the input image. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be "fooled" by noise and more likely to detect true weak edges [33]. Figure 3-2(a)-(d) shows the 3D image of a scene and its object boundaries obtained using different edge detection methods including Prewitt, Roberts, Log, and Canny-Deriche algorithms. In relation to other edge detection algorithms, the Canny-Deriche method has shown to be the most optimal for our work. Figure 3-3(a)-(b) shows the original 3D grey-scale depth image and the edge image after background subtraction and edge detection algorithms.

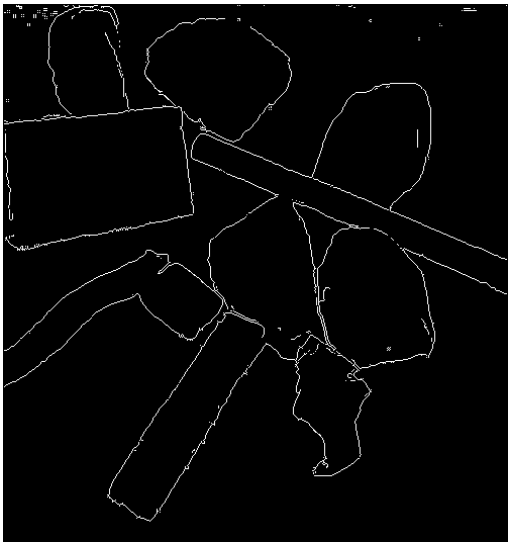

Fig. 3-2 (a): Prewitt method

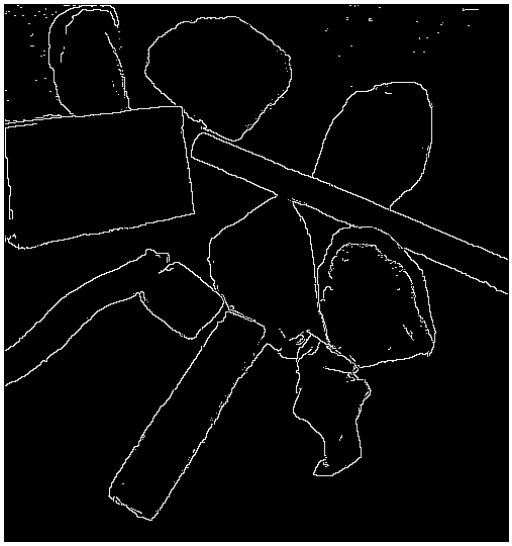

Fig. 3-2 (b): Roberts method

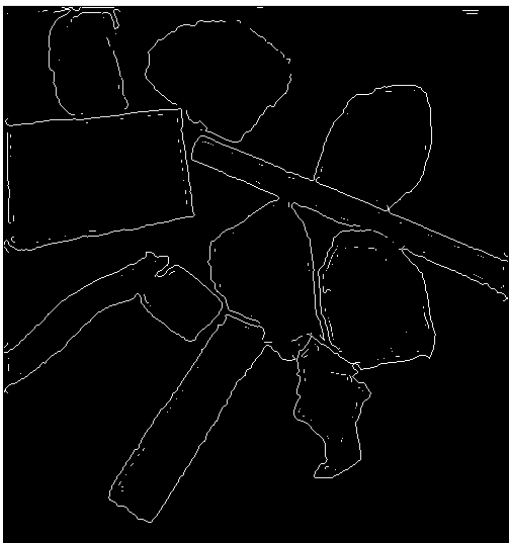

Fig. 3-2 (c): LoG method

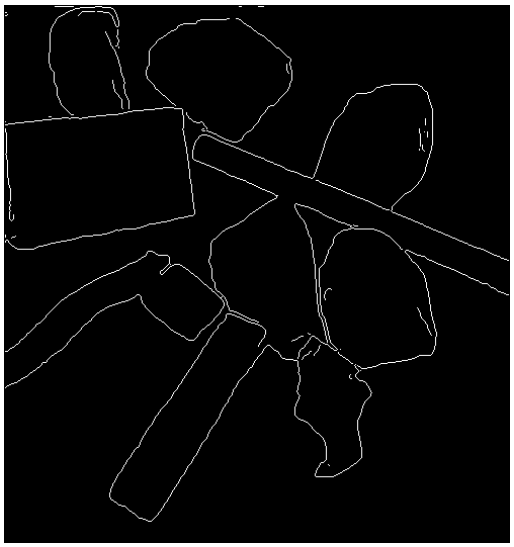

Fig. 3-2 (d): Canny-Deriche method

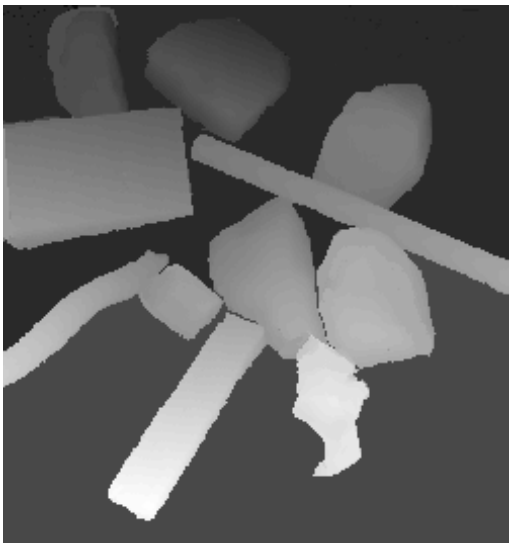

Fig. 3-3 (a): Grey-scale depth image

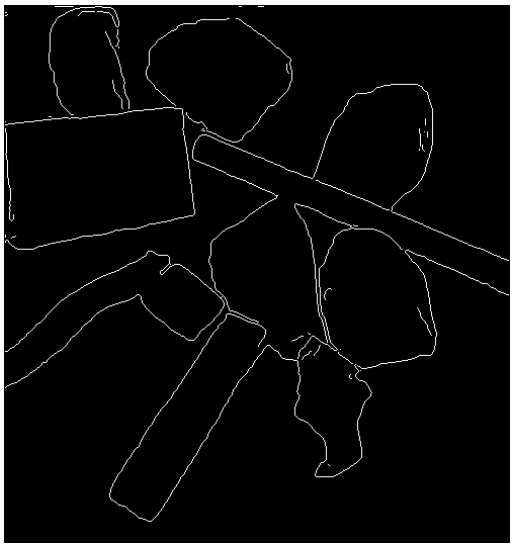

Fig. 3-3 (b): Edge image

3.1.3 Dilation and Thinning

Even though edge detection results can be quite good at determining edges of objects, one shortcoming is their inability at times to produce continuous edges. The two morphological methods of dilation and thinning are further utilized in order to address

this limitation. The dilation technique is a binary method utilized to enlarge the boundary region of an object, in order to make the detected broken edges continuous. The manner and extent of this "thickening" of the edge is achieved via a structuring element [32]. The set-theoretic relationship of dilation is presented as follows:

$$A \oplus B = \left\{ z \middle| (\hat{B})_z \cap A \neq \varnothing \right\}, \tag{1}$$

where A is the Canny-Deriche edge detection image, B is the structuring element, $\hat{B}$ is the reflection of all elements of B about the origin of this set, z is a set of points, $(\hat{B})_z$ is the translation of the origin of $\hat{B}$ to point z. In this work, a 3x3 square structuring element is utilized and repeated until all holes within a region of interest shrink via the thickening of the edge pixels. Figure 3-4 illustrates the principle of dilation with the 3x3 square structuring element. Edge pixels are represented as 1, and non-edge pixels are represented as 0 accordingly.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 3-4 (a): Edges before dilation.

1	1	1
1	1	1
1	1	1

Fig. 3-4 (b): The structuring element.

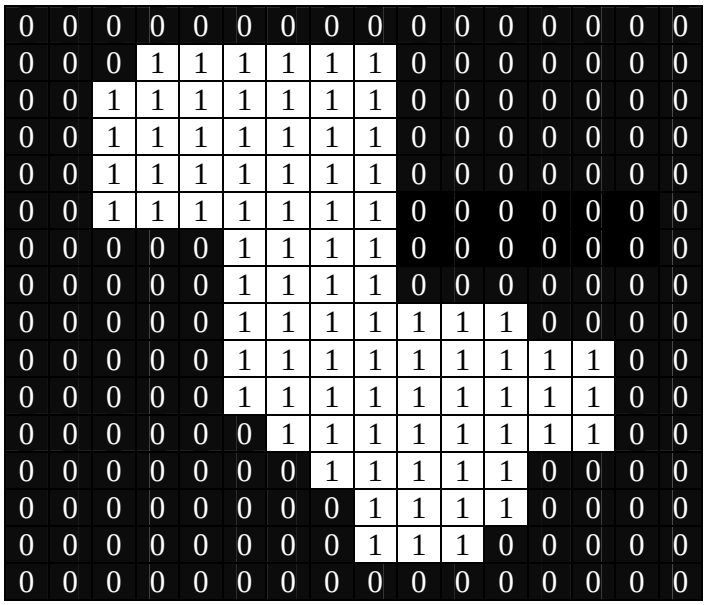

Fig. 3-4 (c): Edges after dilation.

After connecting the edges using the dilation technique, the boundaries of the objects are once again “thinned” back to their original one pixel width via the following thinning method: [32]

$$A \otimes \{B\} = ((...((A \otimes B^1) \otimes B^2)...) \otimes B^n) \quad , \qquad (2)$$

where $\{B\} = \{B^1, B^2, ..., B^n\}$ is a sequence of structuring elements. Figure 3-5 (b) shows the effect of the edges after dilation, and Figure 3-5 (c) shows the edges after the “thinning” operation.

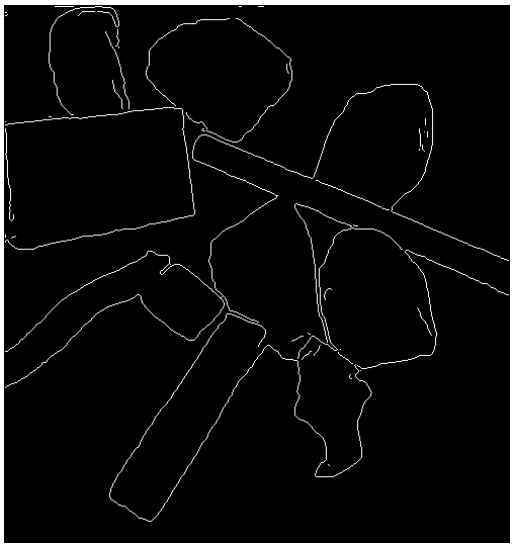

Fig. 3-5 (a): Original edge image.

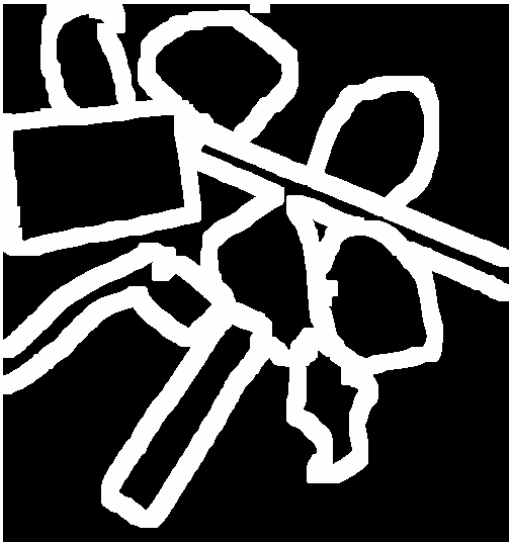

Fig. 3-5 (b): Edges after dilation.

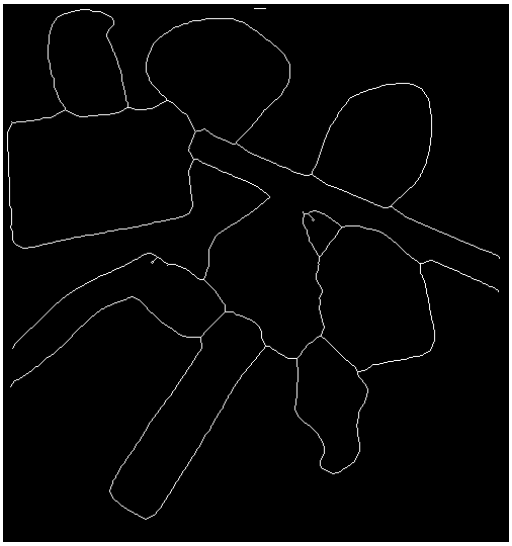

Fig. 3-5 (c): Edges after thinning.

3.1.4 Remove Non-Connecting

There exists such possibility that the iterations for dilation and thinning are not proper so that there are some little pieces of edges which locate inside some potential object instead of the real edges of potential landmarks. These are not helpful at all because in the following steps when we test the vector connecting two SIFT keypoints,

these useless edge pixels will block the way and conclude to wrong judgments. Thus, these “little pieces” which are called “Non-Connecting” have to be removed.

“Non-Connecting” refers to two kinds of edge pixels: one is an isolated edge pixel which has no edge pixel in its 8 neighbor pixels; the other is an edge pixel which has only one edge pixel in its neighbors. The first kind of “Non-Connecting” is easy to understand, since isolated edge pixels do not belong to any edge loops, so there is no reason for them to exist. Edge pixels which have only one neighbor edge pixel should also be deleted for the reason that if such kind of an edge pixel is part of some edge loop, then it should have at least two neighbor edge pixels in the sense of 8-connectivity. The following steps outline the remove non-connecting algorithm:

Step 1: For every edge pixel, calculate the number N of neighbor edge pixels. If N is equal to 0, delete the edge pixel, and change the property of the original pixel from “edge” to “non-edge”. If N is greater than 1, just keep the edge pixel, because it has at least two neighbor edge pixels.

Step 2: If N is equal to 1, figure out the location of the neighbor edge pixel, delete the edge pixel, change the property of the original pixel from “edge” to “non-edge”, and mark the location of this neighbor edge pixel.

Step 1 and Step 2 are repeated until there is no edge pixel which has 0 or 1 neighbor edge pixel.

Figure 3-6 shows the effect after removing Non-Connecting. As is seen, the ends of edge loops are deleted. Although there are still some edges which seem unnecessary, we strictly follow the results from the program. Experiments verify that the edge image and

corresponding matrix is good enough to be used for the following search and convex hull algorithms.

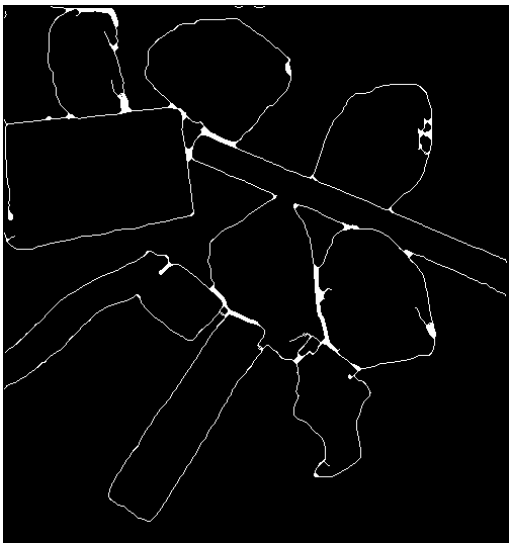

Fig. 3-6 (a): Edges before “remove”.

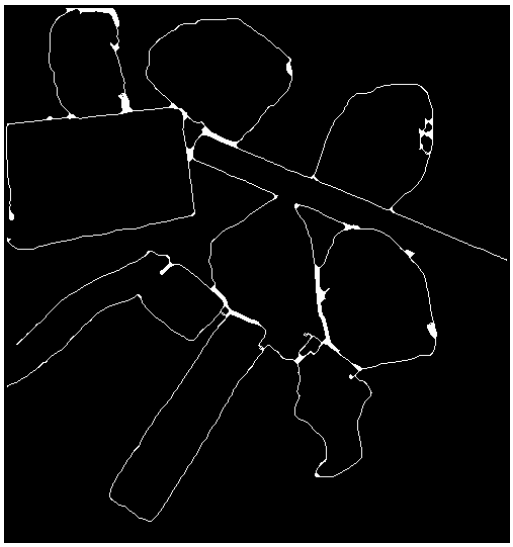

Fig. 3-6 (b): Edges after remove Non-Connecting.

3.2 SIFT-Based Recognition of Landmarks

For Visual SLAM in USAR environments we propose the utilization of Scale Invariant Feature Transform (SIFT) features for identifying and matching of non a priori landmarks. This SIFT-based approach transforms an image into a large collection of local feature vectors, each of which is invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. The resulting feature vectors are called SIFT keypoints. The SIFT keypoints provide information that is utilized to extract strong evidence in discontinuity between multiple objects detected in a scene in order to locate large distinguishable landmarks in a cluttered environment for 3D mapping of the environment. The first step of our landmark identification method consists of determining the keypoints of an image and their dimensional descriptors. In

our proposed work, this will consist of two stages, finding the keypoints and descriptors for the 2D image and corresponding 3D depth image utilizing the four stage procedure of the SIFT algorithm. Both keypoints and descriptors are then stored for the two images, Table 1. Figure 3-7 shows keypoints that have been found on 2D and 3D images of a rubble-like environment with same size objects, and a large distinguishable object.

Table 1: Step 1 of algorithm: Keypoint parameter matrix **A**.

Keypoint #	*x* position	*y* position	Depth	Scale	Orientation
1	80.13	259.74	162	27.14	-1.357
2	373.37	115.63	123	18.89	-1.751
3	316.39	528.41	97	21.67	-1.57
4	504.38	328.62	121	7.4	0.767
5	54.61	221.26	93	2.97	1.222
6	41.46	457.03	89	1.49	1.8
7	562.92	329.69	138	2.36	-0.007
8	264.1	493.26	121	9.21	-1.277
9	480.45	311.58	138	6.82	0.335

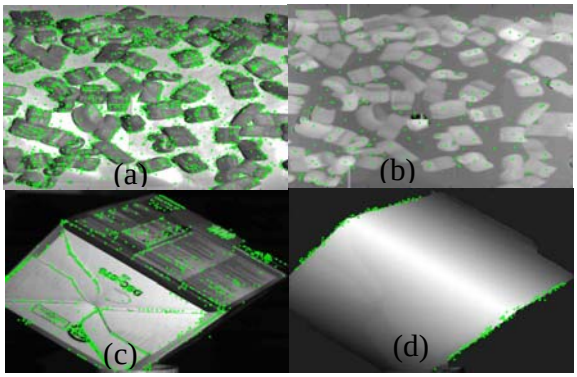

Fig. 3-7: (a) 2D image, (b) 3D image of landmarks.

The SIFT approach consists of four main stages [34]:

(i) Scale-Space Extrema Detection: The first stage of keypoint detection is to identify locations and scales that can be repeatedly assigned under differing views of the same object. Potential interest points are selected by scanning the image over location and scale, by constructing a Gaussian pyramid and searching for local peaks in a series of difference-of-Gaussian (DoG) images which are invariant to scale and orientation.

(ii) Keypoint Localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability, therefore, points are rejected if they are sensitive to noise or are poorly localized along an edge.

(iii) Orientation Assignment: Based on the local image gradient directions, one or more orientations are assigned to each keypoint location. From hereon all operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

(iv) Keypoint Descriptor: The descriptor for each keypoint is made based on the image gradient magnitudes and orientations that are determined in its surrounding region. Orientation histograms are then created over 4x4 sample regions based on this information. Furthermore, there are eight directions for each orientation histogram at 45 degree intervals. This leads to a $4\times4\times8=128$ dimensional descriptor vector also known as the SIFT key. After normalization, this feature vector is stored in a database with the keypoint for subsequent recognition.

As previously mentioned, SIFT features have several advantages, they are invariant to image scaling, translation, and rotation, hence, making SIFT descriptors robust to small geometric distortions and small errors in the detection region. Furthermore, when compared with other types of descriptors, SIFT-based descriptors were found to perform the best [35]. This makes them a strong candidate for landmark detection in USAR environments.

The overall proposed method will be discussed herein outlining its most pertinent stages: (i) identifying keypoints, (ii) identifying clusters, and (iii) matching of clusters.

3.2.1 Keypoint Identification

3D Image Analysis

Keypoints that are determined in the 3D image are grouped together based on grayscale depth information into depth clusters, where they represent the cluster boundaries for the keypoints in the 2D image. The depth grayscale is determined from 0 to 255.

Initially, each keypoint is specified by 5 parameters: x location, y location, depth, scale and orientation, and stored in the matrix A_{ln}, where l represents the number of keypoints and n represents the number of parameters, i.e., Table 1.

Utilizing the keypoint information matrix A, we check every keypoint: if the keypoint falls into the background area we generated in the step of background subtraction, it is discarded from the matrix A; if its location belongs to the object area, we continue keeping it. After this operation, only the keypoints in the potential object areas are prepared for the following steps.

3.2.2 Keypoint Clustering

The clustering of keypoints is not only important in defining landmarks but also in reducing the number of keypoints of interest. The keypoints are denoted as green circles. In general due to shadowing effects and texture changes, a number of keypoints can be identified in the 2D images. Fig. 3-6 (c) shows the keypoints (green circles) found on a box in an environment, with multiple keypoints on the flat surfaces of the box. In the 3D (i.e., depth) image, Fig. 3-6 (d), we can see that the keypoints on the flat surfaces are no

longer present due to the fact there is no significant change in depth on these surfaces. We can analyze and cluster the keypoints we found in the 2D image based on the keypoints found in the depth image in which for the latter image shadowing and texture effects are not present. The 2D and 3D images have a one-to-one correspondence. Mainly, if a keypoint does not exist in the same pixel in the 3D image, then the keypoint is assumed to be due to image shadowing and texture effects. Clusters are bound in the regions where a large number of keypoints in the 3D image do not exist, i.e., they have considerably the same depth information. These clusters can then be used to represent large distinguishable landmarks in the scene. Hence, we can identify a cluster of keypoints in the 2D image by bounding them by keypoints in the 3D image.

Search Region

A nearest neighbor search algorithm is proposed, herein, that defines regions in the 3D images containing keypoints that may potentially represent various landmarks in the scene. The search algorithm utilizes information from both the edge detection algorithm and the deformation matrix in order to estimate these regions of interest. The following steps outline the search algorithm:

Step 1: Random starting points, p_{ij} , for the algorithm is chosen on the image.

Step 2: A square of side length 2r is drawn symmetrically around p_{ij} to search for its nearest neighbor keypoints, Figure 3-8. If no keypoints are initially found, r is continuously incremented until keypoints are detected. Each detected keypoint and its 5 parameters are stored in a temporary matrix B_{pn} for evaluation, where p represents the

number of detected keypoints. For the initial point, only the portion of the square that encompasses the image is searched, i.e., the red square in Figure 3-8.

Step 3: A vector is drawn from p_{ij} to every keypoint in matrix B. The nearest neighbor keypoints are determined to be the keypoints for which the vector connecting them to p_{ij} does not cross edge pixels. All pixels for which the vector crosses are sampled for edge information.

Steps 2 and 3 are repeated until all vectors found for each starting point cross edge pixels.

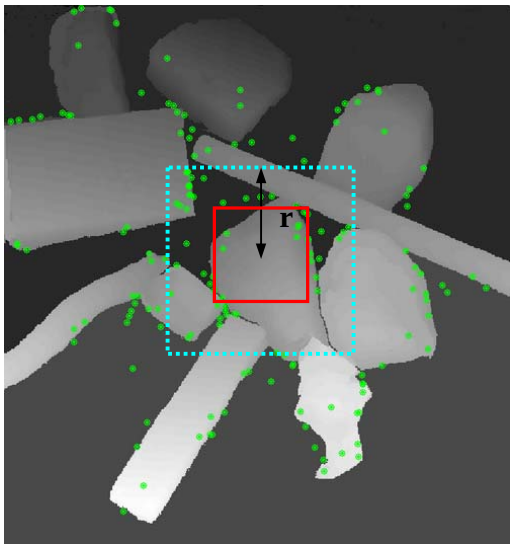

Fig. 3-8: The growing square region.

Clustering

Keypoints in each region of interest are ordered and connected to their nearest neighbors in order to define the respective boundaries for potential landmarks. Depth sampling method and convex hull method are utilized to obtain the clusters. Comparison between these two methods will be represented in the next section.

Depth Sampling Method

A vector is drawn from an initial keypoint, which is defined as the closest keypoint to the initial starting point, $keypoint_{11}$ to every keypoint in matrix **B**. *N* number of points on each vector are sampled for depth information, $samplepoint_i$, where i=1,2,…,N, Figure 3-9. The nearest neighbor keypoint, $keypoint_{12}$ is determined to be the keypoint with the minimum change in depth information from $keypoint_{11}$ and whose corresponding sample points have the smallest variation in depth from itself (i.e. $keypoint_q$, where q=1,…,p-1,p) and $keypoint_{11}$:

$$Minimum\_depth\_value = \min[\mathbf{A}(keypoint_{11},3),\mathbf{B}(keypoint_q,3)], \quad (3)$$

$$Maximum\_depth\_value = \max[\mathbf{A}(keypoint_{11},3),\mathbf{B}(keypoint_q,3)], \text{ and} \quad (4)$$

$$Minimum\_depth\_value - threshold < samplepoint_i\_depth < Maximum\_depth\_value + threshold. \quad (5)$$

The objective of sampling multiple points between the keypoints is to ensure that boundaries of objects are not crossed. If a situation arises where a vector path from a keypoint to its nearest neighbor must cross an already existing vector, this latter vector must follow a different path. The most optimal path for clustering would be to follow along the edge, in order to provide the maximum surface area.

Steps 1 to 3 for search region are repeated until all keypoints in the corresponding cluster are identified. The sample points from previous keypoints in the cluster are stored with their corresponding keypoints and this information is used along with sample points determined for the keypoint of interest in deciding whether the keypoint belongs to the cluster and its order within the cluster:

$$keypoint_{j(k+1)} = f[(samplepoint_i)_m, (samplepoint_i)_{k+1}, keypoint_{jm}]$$,

where m = 1,…, k-1, k. For every keypoint that is added to the cluster, its **A** matrix information is updated with the following additional parameters: order in the cluster, number of connections to other keypoints, depth information stored from sample points. In order for a keypoint to be considered a part of the cluster, it must have a minimum of two connections to other keypoints in the cluster.

Once all keypoints are determined in a particular cluster, a new matrix with all the corresponding keypoint information is defined for that cluster.

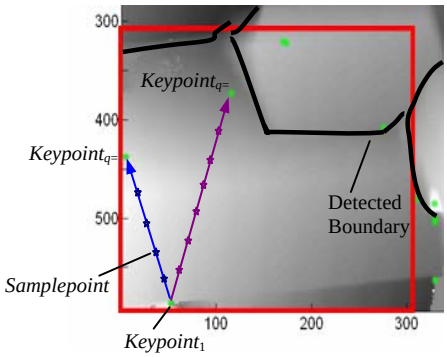

Fig. 3-9: Depth sampling method.

Depth sampling method works well when the shape of potential landmarks is like a circle or a rectangular. But in USAR environments, potential landmarks for rescue robots can be of any shape. Besides, it always happens that the vector between two SIFT keypoints in one cluster crosses edges. Therefore, Convex Hull method is proposed to overcome this kind of difficulties.

Convex Hull Method

A convex hull method is used, herein, to generate the boundaries of each of the defined clusters which represent potential landmarks. The convex hull of a geometric

object, which is defined as a point set or a polygon, is the smallest convex set containing that object. As is shown in Figure 3-10, a set is considered to be convex if whenever two points P and Q are inside the set, then the whole line segment PQ is also inside the set.

In our case, our point set is defined by a group of SIFT keypoints which will be utilized to define the boundaries of our clusters, which will in turn provide us with information about the landmarks. Given the SIFT keypoints in the 3D depth images, optimal cluster sizes need to be defined in order to represent the maximum surface area on the landmarks. This will allow for the inclusion of more SIFT keypoints in the corresponding 2D image for matching; A two dimensional finite set is utilized in this work, for which the convex hull can be defined as a convex polygon.

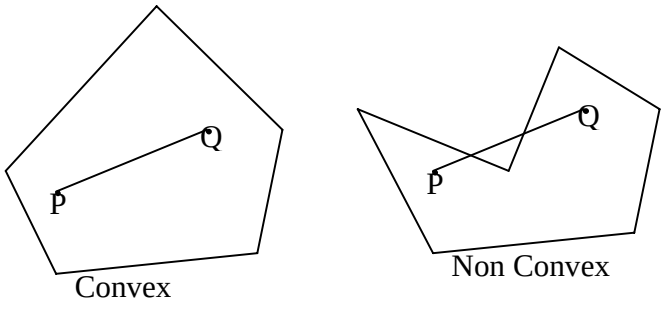

Fig. 3-10: Convex and Non Convex.

In this work, we will utilize the Gift Wrapping algorithm [36] due to its simplicity in implementation and its favorable computational complexity (i.e., $O(nh)$ [36]) for our purposes. n is the number of SIFT keypoints in the cluster and h is the number of sides defined by the line segments of the final generated convex polygon. The following steps were utilized to implement the Gift Wrapping convex hull algorithm shown in Figure 3-11:

Step 1: Set the index $i = 0$, where i = 0 to h, $h \in R$ and is defined as the number of iterations. Find a keypoint p_0 known to be on the convex hull, e.g., the leftmost keypoint in the cluster.

Step 2: Select the next keypoint p_{i+1} such that all remaining keypoints are to the right of the line $\overline{p_i p_{i+1}}$.

Step 3: Let $i = i + 1$, and repeat Step 2 until $p_h = p_0$ which yields the convex hull in h iterations.

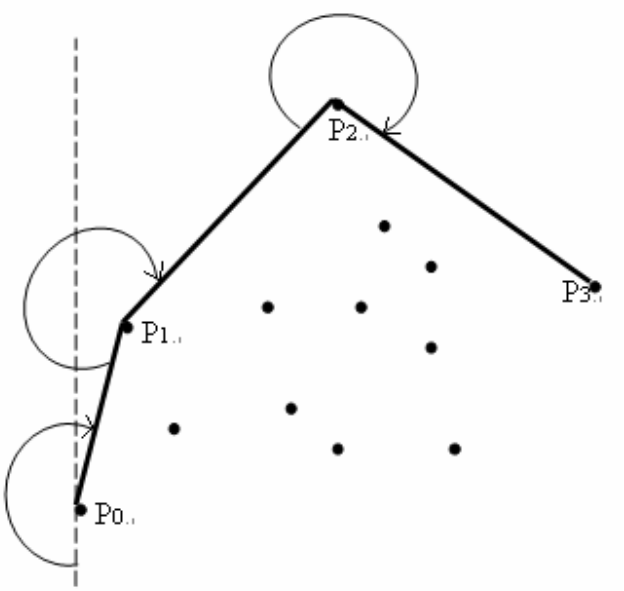

Figure 3-11: Gift Wrapping algorithm.

Several different cases have been considered in the proposed method. For example, when the convex hull has only one or two vertices, the method determines one line between the two vertices and reports that a cluster cannot be generated for this case. When three or more points are collinear, the method does not consider the middle points. The method has been successful implemented and tested for these different conditions. Experiments presented in the next section verify the method's robustness to such cases.

3.2.3 2D Image Analysis

Once all depth clusters in the 3D image have been identified, they can be used to identify their corresponding keypoints in the 2D image. Each depth cluster represents the boundary conditions for the 2D keypoints. Since there exists a one-to-one correspondence between the 3D and 2D images, the boundaries can be superimposed on the 2D image. Herein, cluster boundaries are represented by the connection vectors between the keypoints in the depth clusters. Based on the pixel occupancy of the boundaries, 2D keypoints that are located within these boundaries are identified and stored in the cluster matrix. Each cluster is defined to represent a landmark in the environment, Figure 3-12. It is important to note that this clustering method does not attempt to represent the shape of the landmark in the environment; it merely identifies detectable regions that can represent a portion of a true landmark and that can be matched in successive images with different viewpoints.

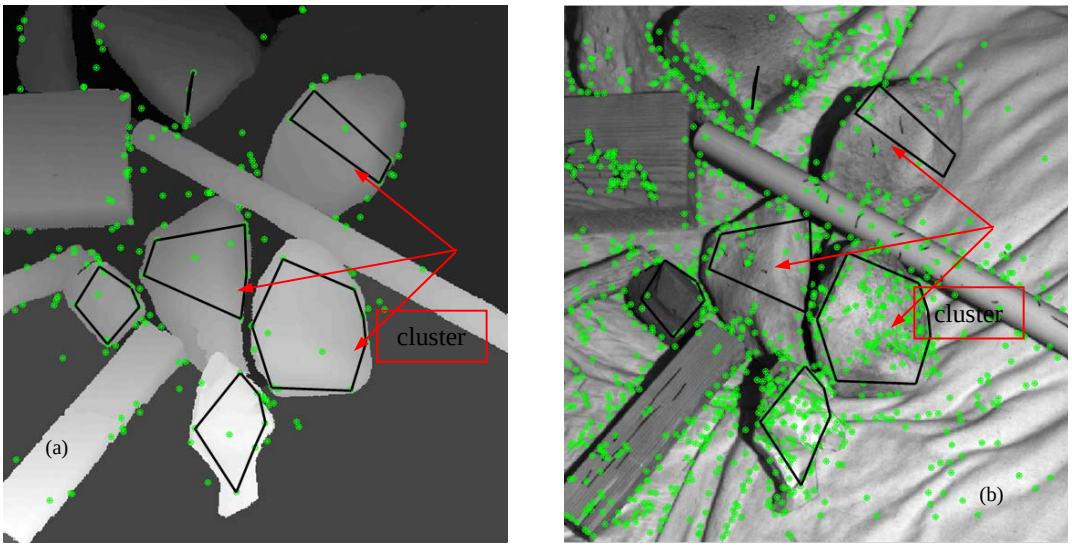

Fig. 3-12: Cluster results: (a) 3D image, and (b) 2D image.

3.2.4 Matching of Clusters

Matching of clusters relies on finding the same clusters in consecutive images by matching keypoints from the clusters from previous frames (as defined in our database) with ones in the new cluster of the current frame, we utilize the matching method proposed by Lowe, in [34], known as the Best-Bin-First (BBF) method. Herein, this can be achieved in terms of matching the key descriptors of the keypoints which can correspond to finding a set of nearest neighbors (NN) to a query point. The advantage of this method is its ability to handle high-dimensional spaces, i.e., the 128 dimensional descriptor vectors. Since individual SIFT keypoints are easily distinguishable, they can be matched correctly with an exception of a few false matches. Figure 3-13 illustrates matching between two clusters determined in two different viewpoints of a scene. The blue lines represent the keypoints that were matched within the two images. The effectiveness of the method is shown in Figure 3-13, where the majority of the matches made are correct matches.

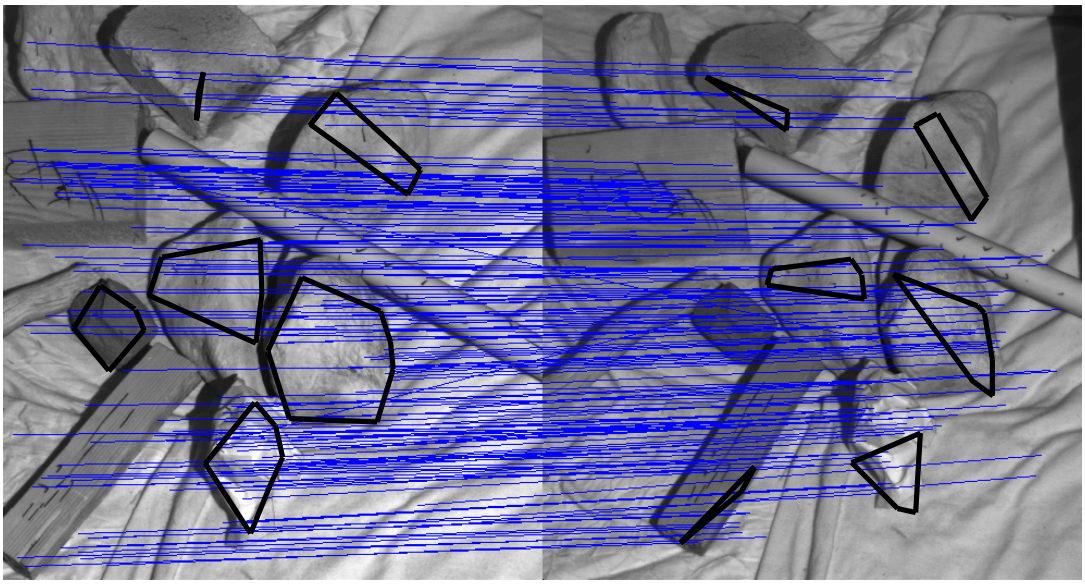

Fig. 3-13: Matching of clusters in different images.

Chapter 4 Simultaneous Localization and Mapping (SLAM)

The previous Chapter outlines the proposed SIFT-based procedure for identifying 3D distinguishable non a priori landmarks for a robot to detect as it moves in the environment in order to create a 3D global map of its environment. The matching stage of the SIFT method is able to bring consecutive images into close alignment for 3D reconstruction of the environment. However, in order to build the 3D *map* in world coordinates, the robot must be able to *localize* itself utilizing these determined landmarks. This can be achieved by stitching consecutive 3D range information corresponding to the landmarks. This chapter outlines the proposed SLAM-based techniques for creation of a 3D virtualized map of the disaster environment with respect to a world frame in which victims can be found. Section 4.1 describes the proposed 3D SIFT-based ego-motion methodology for robotic localization and Section 4.2 defines the Iterative Closest Point (ICP) –based method utilized for stitching of 3D images of the USAR scene in order to generate a 3D map of the environment.

4.1 Robot Localization

In order for the mobile robot to localize itself accurately and effectively within its environment, it must know its pose relative to a pre-defined world coordinate frame. In our proposed methodology, a SIFT-based ego-motion approach is proposed for robot localization.

4.1.1 3D Sensory System Calibration

In order to determine the ego-motion transformations effectively and accurately, the visual sensory system utilized in the application must be calibrated with respect to the environment. Herein, the calibration procedure utilized to identify the relationship between the 3D mapping sensor and the scene of interest is presented. Similar calibration techniques for other sensory systems that can be utilized, i.e., 3D cameras, can be implemented.

Calibration Procedure

A checkerboard placed on top of a sub-micron motion control system is utilized to determine the 3D transformation, ${}^{m}T_{c}$, between the camera coordinate frame, F_c, and the motion control system frame, F_m, Figure 4-1. The calibration procedure is outlined below:

Step 1: Identify N corner points on the checkerboard. The value of the sub-pixel coordinates of the N points in the 2D image taken by the sensory system is determined by using the Matlab camera calibration toolbox [37]. Locate the corresponding 3D coordinates of the N points, i.e., $[\Delta x_i \quad \Delta y_i \quad \Delta z_i]$ where i=1,2,3,…,N, in the point cloud, to determine their corresponding locations in the camera coordinate frame.

Step 2: Move the motion control system by $[\Delta X \quad \Delta Y \quad \Delta Z]$ while tracking the N points both in the 2D image and corresponding 3D point cloud. Repeat this step for P positions. After each new position, the motion control system should be homed.

Step 3: Determine ${}^{m}T_{c}$ utilizing the coordinate information of the N points:

$$[\Delta X \quad \Delta Y \quad \Delta Z \quad 1]^{T} = \underbrace{\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_9 & a_{10} & a_{11} & a_{12} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{m}T_{c}} \cdot [\overline{\Delta x_i} \quad \overline{\Delta y_i} \quad \overline{\Delta z_i} \quad 1]^{T} \tag{8}$$

where $\overline{\Delta x_i}, \overline{\Delta y_i}, \overline{\Delta z_i}$ are the averages of $\Delta x_i, \Delta y_i, \Delta z_i$.

For P=8 experiments, the optimized ${}^{m}T_{c}$ was determined to be:

$${}^{m}T_{c} = \begin{bmatrix} -0.0694 & 0.0573 & 1.0472 & -0.9070 \\ -0.8932 & -0.2782 & -0.1385 & -0.2937 \\ 0.0184 & -0.7313 & 0.0575 & -0.0151 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Since the surface of the flat board is very smooth, the measurement noise is mainly due to the sensory system itself, where the RMS (Root-Means-Squared) error of the 3D range data is determined to be approximately 0.05mm for a measurement area of 260 × 244 mm [38]. This RMS error is acceptable for our application, since we do not require high precision.

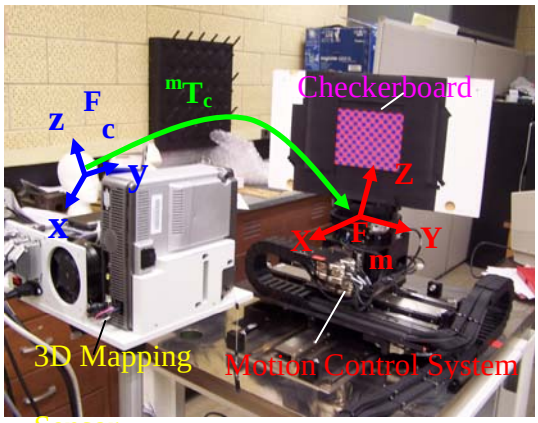

Fig. 4-1: Calibration Set-up.

4.1.2 Ego-motion

The 3D range information of the landmarks is provided by the 3D sensor via a point cloud with respect to the camera coordinate frame. This information corresponds to the pixels the landmark occupies in the 2D image. Hence, by identifying the location of the SIFT keypoints representing one landmark in the 2D image, its 3D range information in the camera coordinate frame can be determined. The 3D coordinates of the same SIFT keypoints (SIFT pairs) in different images can be utilized to solve for the 6 DOF ego-motion parameters (i.e., ΔX, ΔY, ΔZ, $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$). At least three pairs of SIFT keypoints are needed to estimate the ego-motion transformation ${}^{i}T_{j}$, Figure 4-2. Since the position of the camera relative to the robot's coordinate frame is known, the transformation ${}^{ri}T_{rj}$ between the robot at two different locations can be determined. By utilizing this information and the localization information from the previous position ${}^{w}T_{ri}$, the robot's location can be estimated. Furthermore, once the alignment of the same landmarks is determined between different visual sensor locations, the corresponding 3D range information of the scene can be stitched together for reconstruction of the USAR environment.

Once a potential ego-motion transformation has been calculated, it is verified by determining how many additional SIFT pairs support this particular transformation. The following ranking scheme is utilized: $r = l/m$, where r, l and m represent the rank, the number of matched SIFT pairs that confirm the transformation and the total number of SIFT pairs, respectively. If r is greater than a certain threshold b, then we assume we have the most accurate ego-motion transformation.

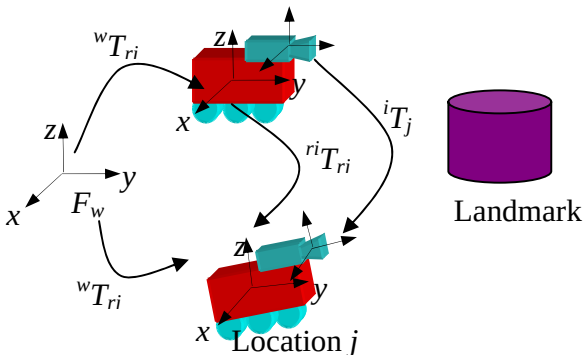

Fig. 4-2: Transformations for localization of the robot.

4.2 3D Mapping

3D map building in this research work will be achieved utilizing a 3D surface registration method. The most popular registration methods include Schwartz and Sharir's curvature extrema method [39], Kamgar-Parsi et al.'s method [40], Szeliski's method [41] to create a method for estimating the motion of the observer between two range image frames of the same terrain, and Besl and McKay's ICP algorithm [42].

Schwartz and Sharir developed a solution to the freeform space curve matching problem by utilizing a non-quaternion approach to compute the least squares rotation matrix. The method works well with reasonable quality curve data but has difficulty with very noisy curves because the method uses arclength sampling of the curves to obtain corresponding point sets.

Kamgar-Parsi et al. propose a method for the registration of multiple overlapping range images. Firstly the inputs used for local registration are contours of constant range represented by means of a modified chain code method. All best matches of pairs of contours are considered tentative until their geometrical implications are evaluated. Then a cost function is constructed and minimized to do global registration. Terms contributing

to the cost include violation of local matches as well as compression and bending in range images. In cases where there is no appreciable compression and bending in the images, the proposed global scheme could improve the quality of local registration by enforcing consistency among them. This method works very well using the level sets of 2.5-D range data but is essentially restricted to the three degrees of freedom in the plane since the work was addressed toward piecing together terrain map data.

Szeliski's method applied a smoothness assumption to create a smoothing spline approximation of the points given the set of points from one frame. Then, a conventional steepest descent algorithm is used to rotate and translate the second data set so that it minimizes the sum of the covariance-weighted z differences between the points and the surface. This work presents some interesting ideas, but the experimental results are unconvincing for practical applications. This is because his experiments did involve 6 DOF estimation, but the test object is a very simple shape.

The Iterative Closest Point (ICP) algorithm is a reliable and popular method utilized for point cloud registration. If a priori information about the point-to-point correspondence of two point clouds is provided, then the ICP can iteratively recover the relative transformation of the point clouds. It converges monotonically to a local minimum, which may or may not be the global minimum. The closest point of a point in a point cloud in terms of Euclidean distance is assumed to be its corresponding point. It was first developed by Besl and McKay [42], and modified by Chen and Medioni [43] and optimized by Zhang [44]. The concept and procedure were proposed and convergence theorem is proved in [41] and [42]. The optimized version of the ICP algorithm proposed by Zhang works well in registering two partly overlapping surfaces.

It converges to the closest local minimum efficiently with a complexity of $O(N \log N)$ in the most expensive computation. k-D trees are utilized to speed up computation time. Zhang's optimized algorithm is capable of dealing with gross outlines in the data, appearance and disappearance in which curves in one set do not appear in the other set, and occlusion.

In addition to the aforementioned advantages, ICP is utilized in this work due to its robustness to noise and outliers. It is also the basic algorithm on which a number of existing stitching methods have been based on to achieve fine alignment. Since ICP is a local optimization method, the initial parameters for the algorithm are provided from the 6 DOF transformations determined by the proposed ego-motion technique. These transformations bring two point clouds in close proximity and hence, assist in allowing ICP to converge to an optimal solution. By utilizing the ICP method and taking advantage of the redundancy from observing the same landmarks multiple times, the localization errors inherent to the vision system can be minimized.

4.2.1 ICP Algorithm

The ICP algorithm is a highly effective local optimization algorithm. However, the algorithm does not guarantee that it will achieve global alignment. We address this issue by utilizing the SIFT method proposed above to initially align two sets of 3D points of interest, Pi and Pj, before implementing the ICP algorithm for fine alignment. The ICP can then be used to align the data sets from this initial registration utilizing a nonlinear optimization procedure. The two sets of 3D points correspond to a single landmark expressed in the different reference frames. The objective is to find the 3D

transformation, ${}^{ci}T_{cj}$, which, when applied to Pj, minimizes the distance between the two point sets. It can be said that for each point pi from the set Pi, there exists at least one point, termed the closest point, on the surface of Pj that is closer to pi than all other points in Pi. The ICP algorithm repeatedly computes the closest points between data sets and computes the transformation to register the data, until a minimum tolerance on a mean square distance metric between the surfaces is obtained. The following simple procedure is implemented herein utilizing the SIFT keypoints as inputs into the ICP algorithm:

Step 1: Given two sets of 3D point clouds, three matched pairs of SIFT keypoints from the point clouds are chosen accordingly.

Step 2: The three pairs are utilized as the initial registration data into the ICP algorithm, which is then implemented.

Step 3: Once all point clouds have been stitched. Generate the mesh and surface model of the scene.

In Besl and McKay [42], ICP requires every point in one surface to have a corresponding point on the other surface. In our application, the surfaces to be registered are partly overlapped with each other, which means not all points in set Pi have their counterparts in set Pj. Only points in the overlapping area should have reasonable closest point in the other surface. Based on the fact that it is not reasonable for the distance between a point pair to be too large, a distance value threshold $D_{\max}$ can be set. If the distance between a point pair exceeds $D_{\max}$, this pair is discarded. Every time the transformation is applied, the average distance between point pairs is calculated, and $D_{\max}$ is updated. If the transformation draws the point pairs closer to each other, we set a smaller $D_{\max}$. In this way, $D_{\max}$ is dynamically updated based on the statistical

information of point pairs.

Monotonic convergence to a local minimum has been proven for the ICP algorithm based on two key ideas [44]: (i) Least squares registration utilized in the algorithm generically reduces the average distance between corresponding points during each iteration; and (ii) The closest point determination generically reduces the distance for each point individually. The optimized approach has increased the rate of convergence: in which a coarse estimation utilizing only one sample point for a group of five points is implemented during the first few iterations instead of all sample points, and only for the latter iterations are all sample points used to obtain a precise estimation. Zhang's experiments verify the effectiveness of the optimized approach for convergence.

4.2.2 Stitching of Point Clouds

Herein, the optimized ICP [44] algorithm is utilized as the registration algorithm. We implement the algorithm in two different approaches: utilizing Besl and McKay's approach [42] and by utilizing the ICP algorithm in GSI Studio [45] to implement 3D stitching. Figure 4-3 represents two sets of initial point clouds before stitching. The following simple procedure is implemented herein utilizing the SIFT keypoints as inputs into the ICP algorithm:

Step 1: Given two sets of 3D point clouds, three matched pairs of SIFT keypoints from the point clouds are chosen accordingly.

Step 2: The three pairs are utilized as the initial registration data into the ICP algorithm, which is then implemented.

Step 3: Once all point clouds have been stitched. Generate the mesh and surface model of the scene.

Figure 4-4 (a) shows the merging details of the two sets of point clouds after running ICP, and Figure 4-4 (b) gives the final result of the overall mapping.

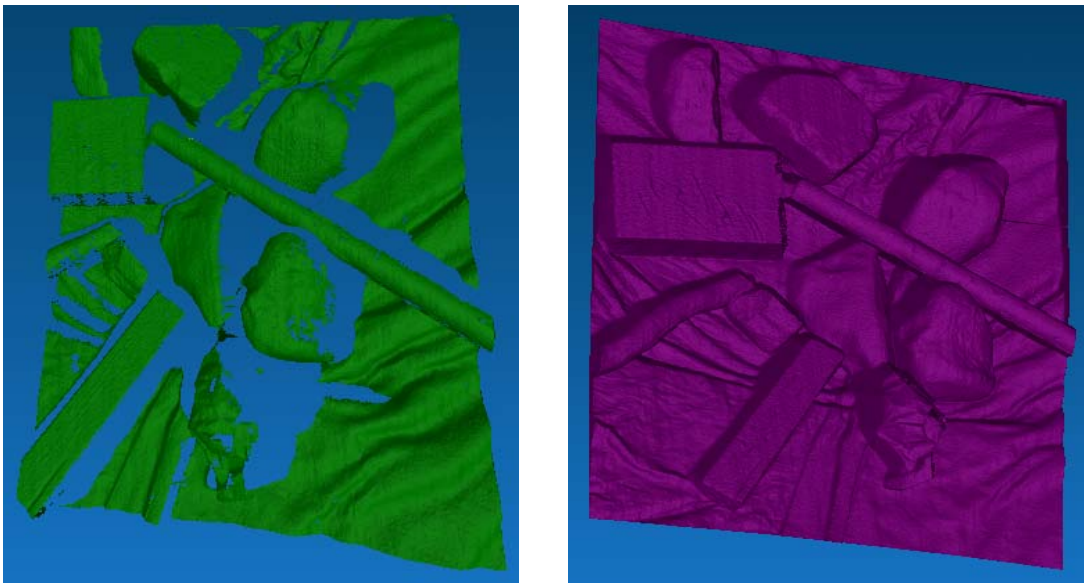

Fig.4-3: Two sets of point clouds before stitching.

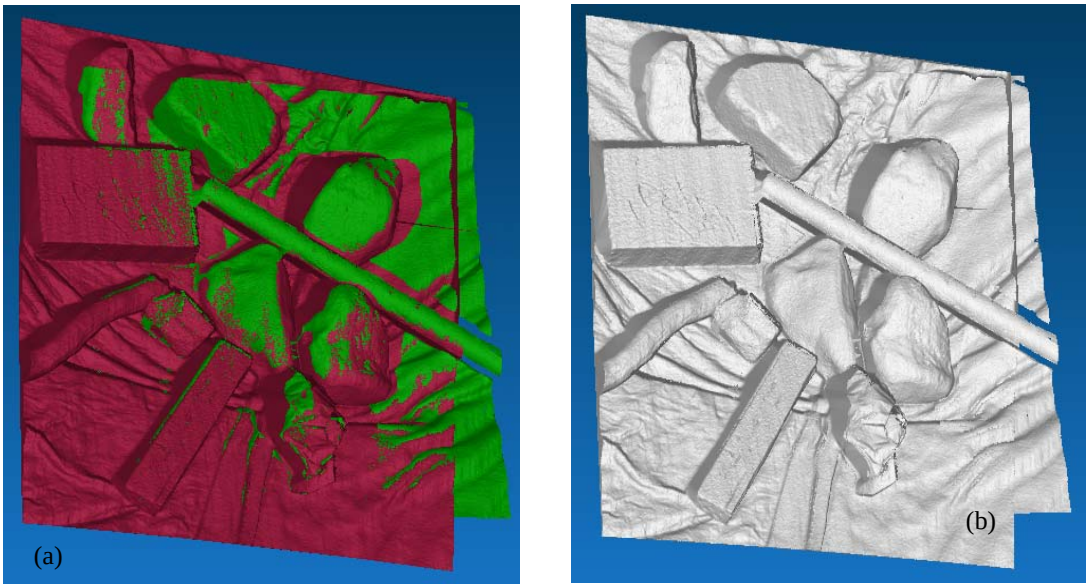

Fig.4-4: The final mapping result.

Chapter 5 Experiments

Several preliminary experiments were conducted to verify the proposed landmark identification and 3D Visual SLAM methods. Herein, the 3D and 2D images were provided by the structured light vision system proposed in [31]. This chapter outlines the experimental set-up, procedure and results.

5.1 System Components

5.1.1 System Hardware

The sensory system consists of a DLP projector, in particular the PLUS U5-632 Digital projector with 1024×768 resolution and 3000 lumens light output and the Dalsa CA-D6-0512 B/W high speed CCD camera (resolution 532×500), as shown in Figure 5-1. The effective range of measurement of the system has been determined to be 0.7~1.4m, with the current lens configuration of the camera and projector. Utilizing the sensory system, both 2D and 3D images can be provided in real-time. In addition to these types of images, 3D range information stored in point cloud formation is also provided by the sensor. This sensory information is utilized by the proposed landmark identification and matching, and Visual SLAM algorithms.

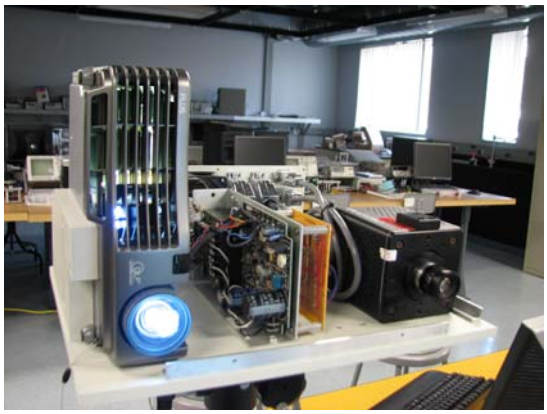

Fig. 5-1: The sensory system.

5.1.2 Software

The software for the proposed landmark identification and matching and 3D Visual SLAM methodology is written in Matlab 7 R14 and implemented on a Pentium IV 3.0 GHz 1.0G RAM personal computer. Figure 5-2 illustrates the different modules of the software.

5.2 Experiments

Preliminary experiments were implemented utilizing the aforementioned software and hardware components. Two types of experiments were implemented: (i) in a controlled setting, where the sensor remained static and the scene was moved using a high precision motion control system and (ii) where the sensory system was placed on top of a robotic platform and moved in a static scene. In general, the overall proposed methodology took at most 60 seconds of CPU time.

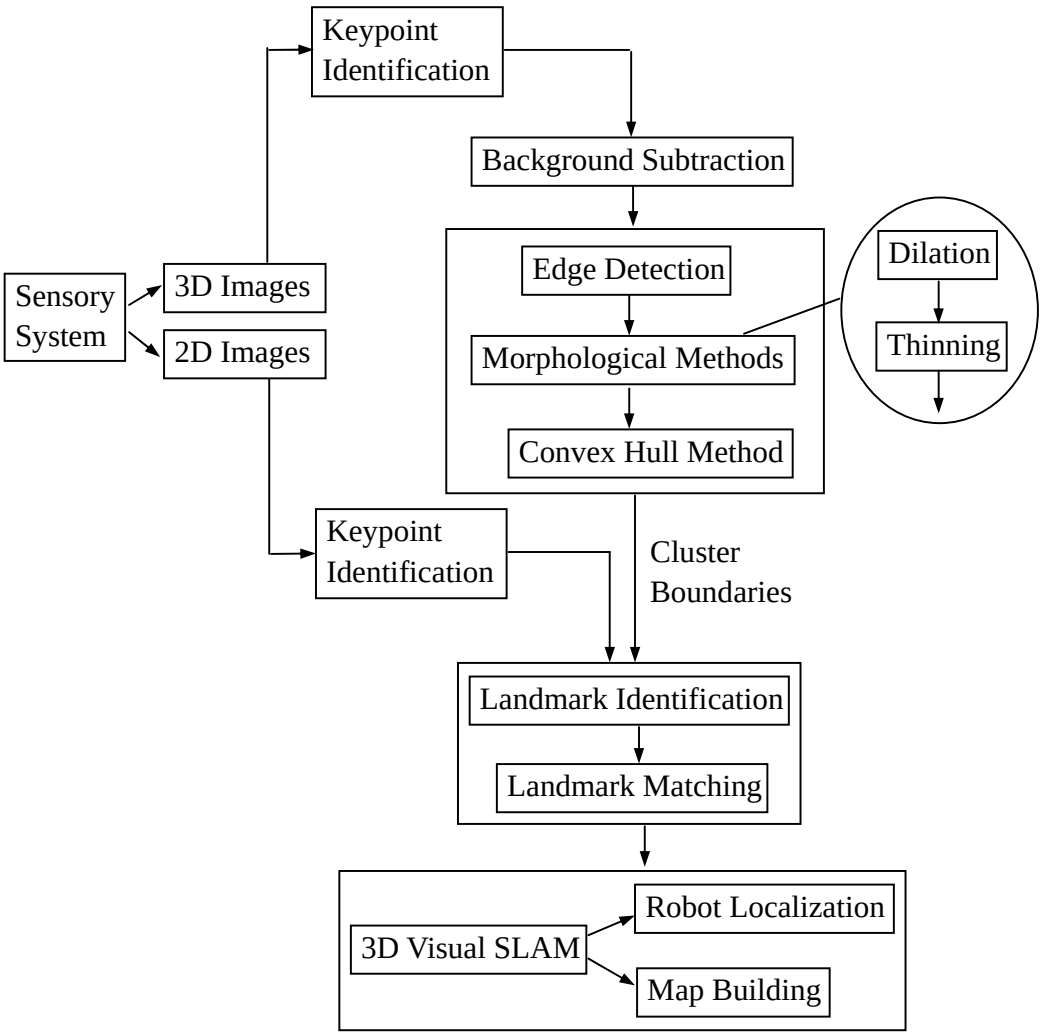

Fig. 5-2: The software architecture.

5.2.1 Experiments #1: A USAR Simulated Scene in a Controlled Environment

Experimental Set-up

For these experiments, several brown cardboard boxes, foam and a human mask were utilized to mimic a USAR environment in the sense that they represent different

shapes of objects and also the small variation in color of the scene. The objects were placed on top of a high precision motion control system as shown in Figure 5-3. Two sets of experiments were performed. The objective of these experiments is to utilize identified and matched landmarks in a controlled scene to localize the robot and generate a 3D map.

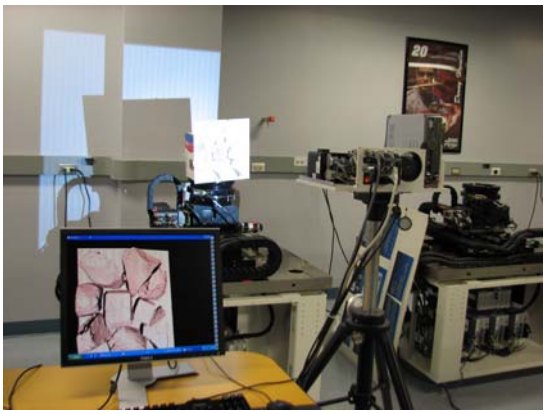

Fig. 5-3: The motion control system with the USAR simulated scene.

Experimental Procedure

1) 3D and 2D images are taken by the structured light sensor.
2) SIFT keypoints are found on the 2D and 3D image of the scene.
3) Image analysis on the 3D image is performed by eliminating keypoints in background using background subtraction.
4) Edges of the foreground objects are determined utilizing the Canny-Deriche edge detection method and the morphological methods of dilation and thinning.
5) Utilizing the edge information and SIFT keypoint locations within the 3D image, keypoints are grouped together.

6) These keypoints are then utilized via the convex hull Gift Wrapping algorithm to determine cluster boundaries.
7) These cluster boundaries are superimposed on the one-to-one correspondent 2D image to group 2D keypoints into clusters that can represent potential landmarks.
8) Matching using the BBF method is implemented using consecutive 2D images of the scene.
9) 3D Visual SLAM is performed, in which matched 2D SIFT pairs and the point cloud information provided by the sensor are utilized to determine 6 DOF ego-motion of the motion control system and generate a map of the scene.

Experimental Results and Discussions

Figure 5-4 presents the results for the USAR simulated scene with cardboard boxes and foam at two different robot poses. An average of 269 and 1074 keypoints were determined in the 3D and 2D images, respectively. 7 clusters were found and matched at two different robot poses. The clusters that had more than 3 correct keypoint matches were recognized to be the same landmark in the scene. 5 (1,3,4,5,7) of the 7 matched clusters were matched effectively in this experiment. The matched keypoint pairs of these clusters and their corresponding 3D range information were utilized to estimate the ego-motion parameters via the Levenberg-Marquadt nonlinear solver: i.e., ΔX=30.04 mm, ΔY=10.20 mm, ΔZ=-15.21 mm, $\Delta\alpha$=14.98^o, $\Delta\beta$=0.1^o, $\Delta\gamma$=0.01^o. The true ego-motion parameters determined by the high-precision motion control system are: ΔX=30.00 mm,

ΔY=10.00 mm, ΔZ=-15.00 mm, $\Delta\alpha$=15.00^o, $\Delta\beta$=0.00^o, $\Delta\gamma$=0.00^o.

Figure 5-5 presents the results for the USAR simulated scene in which, in addition to the boxes and foam, a human face mask potentially representing a victim is presented. An average of 177 and 766 keypoints were determined in the 3D and 2D images, respectively. 9 clusters were found and matched at two different robot poses. The clusters that had more than 3 correct keypoint matches were recognized to be the same landmark in the scene. 4 of the 9 matched clusters were found to represent the same landmarks.. The matched keypoint pairs of these clusters and their corresponding 3D range information were utilized to estimate the ego-motion parameters via the Levenberg-Marquadt nonlinear solver: i.e., ΔX=19.93 mm, ΔY=30.00 mm, ΔZ=20.17 mm, $\Delta\alpha$=-9.97^o, $\Delta\beta$=-0.66^o, $\Delta\gamma$=0.18^o. The true ego-motion parameters determined by the high-precision motion control system were: ΔX=20.00 mm, ΔY=30.00 mm, ΔZ=20.00 mm, $\Delta\alpha$=-10.00^o, $\Delta\beta$=0.00^o, $\Delta\gamma$=0.00^o. In a real setting, a thermal camera would be utilized to identify the landmark associated with the face mask as a potential victim in the scene.

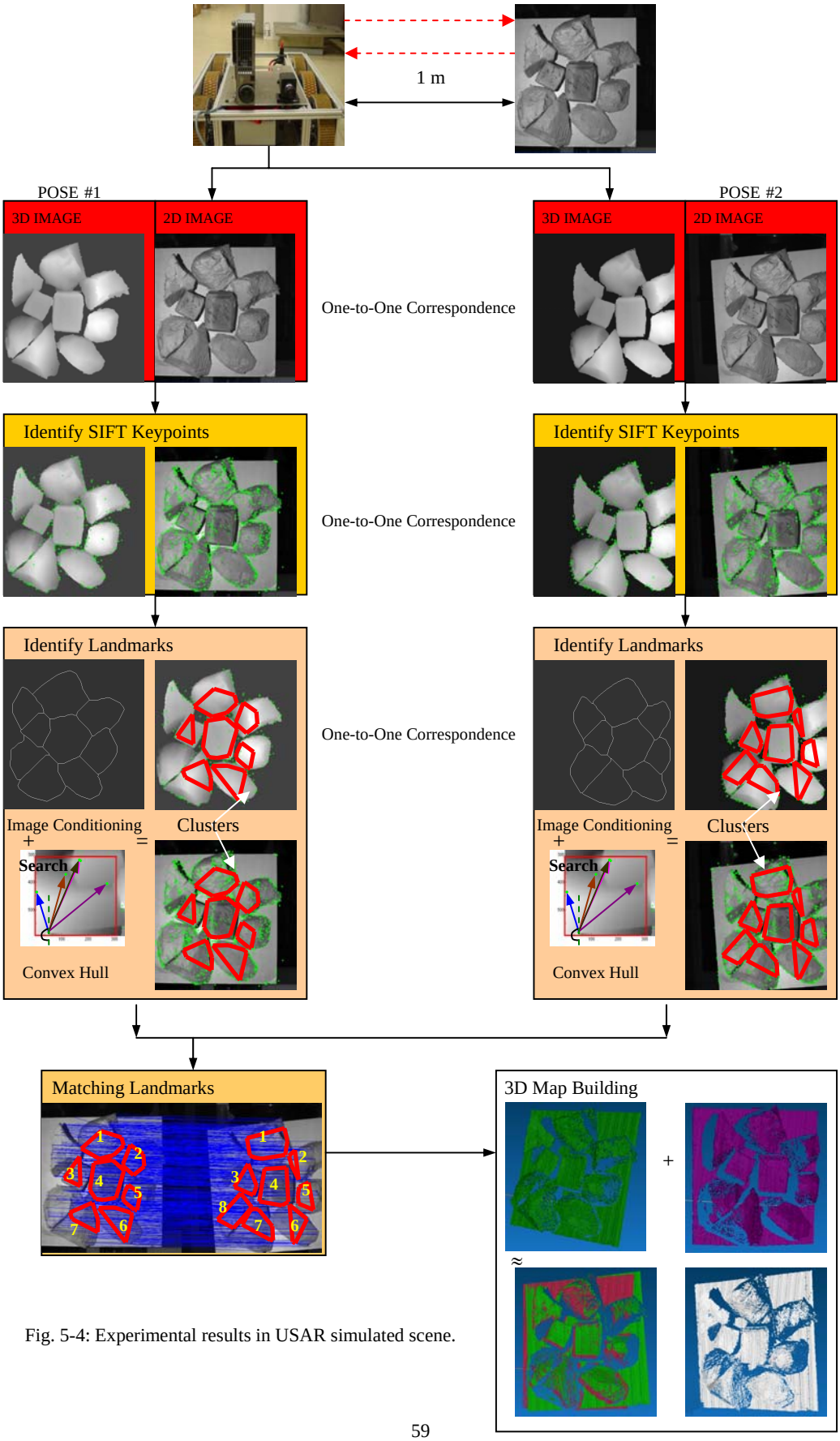

Fig. 5-4: Experimental results in USAR simulated scene.

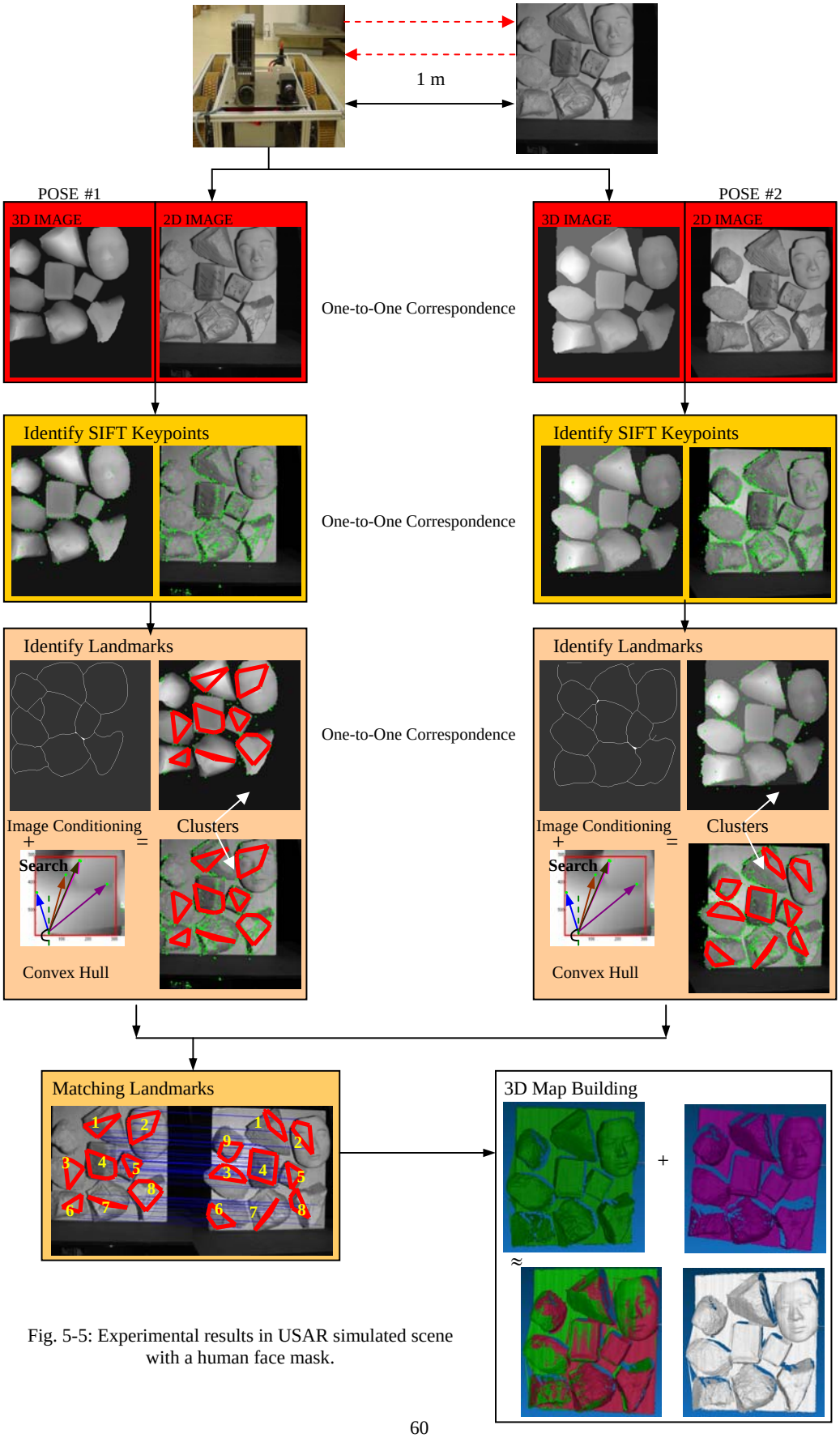

Fig. 5-5: Experimental results in USAR simulated scene with a human face mask.

5.2.2 Experiments #2: A More Natural Scene

Experimental Set-up

The experimental set-up in these sets of experiments consists of placing the 3D sensory system on top of an all terrain six-wheeled robot that is navigated through a scene, Figure 5-6. The robot is defined to navigate small-sized obstacles and carry heavy loads including people. The environment utilized in these experiments consists of minimal lighting and rubble piles containing pipes, wood, rocks and paper covered with a gray dust. In this particular experiment, the proposed methodology's robustness to identifying and matching clusters within these types of scenes is tested.

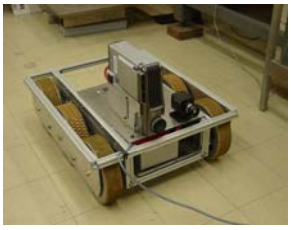

Fig. 5-6: The system on a mobile robot.

For these experiments, several brown cardboard boxes, foam and a human mask were utilized to mimic a USAR environment in the sense that they represent different shapes of objects and also the small variation in color of the scene. The objects were placed on top of a high precision motion control system as shown in Figure 5-3. Two sets of experiments were performed. The objective of these experiments is to utilize identified and matched landmarks in a controlled scene to localize the robot and generate a 3D map.

Experimental Procedure

1) The robot fist moves to one position, and 3D and 2D images are taken by the structured light sensor on the robot. Then the robot moves to another position and takes another set of 3D and 2D images. During this process, the robot's position changes in 6 DOF.
2) SIFT keypoints are found on the 2D and 3D image of the scene.
3) Image analysis on the 3D image is performed by eliminating keypoints in background using background subtraction.
4) Edges of the foreground objects are determined utilizing the Canny-Deriche edge detection method and the morphological methods of dilation and thinning.
5) Utilizing the edge information and SIFT keypoint locations within the 3D image, keypoints are grouped together.
6) These keypoints are then utilized via the convex hull Gift Wrapping algorithm to determine cluster boundaries.
7) These cluster boundaries are superimposed on the one-to-one correspondent 2D image to group 2D keypoints into clusters that can represent potential landmarks.
8) Matching using the BBF method is implemented using consecutive 2D images of the scene.
9) 3D Visual SLAM is performed, in which matched 2D SIFT pairs and the point cloud information provided by the sensor are utilized to determine 6 DOF ego-motion of the

robot and generate a map of the scene of the robot.

Experimental Results and Discussions

For this experiment, an average of 197 and 1124 keypoints were determined in the 3D and 2D images, respectively. Figure 5-7 depicts the 3D and 2D images provided by the structured light sensor at two different robot poses. 6 distinguishable clusters were found at each pose. The clusters that had more than 3 correct keypoint matches were recognized to be the same landmark in the scene. 4 (#2,3,4,5) out of the 6 clusters were matched effectively between the two poses. It is important to note that even though cluster #3 encompasses two objects in the scene, only keypoints on the lower object were matched. The ego-motion parameters were determined to be: ΔX=-25.03 mm, ΔY=-9.77 mm, ΔZ=51.21 mm, $\Delta\alpha$=-10.62^{o}, $\Delta\beta$=-11.68^{o}, $\Delta\gamma$=10.73^{o}, respectively.

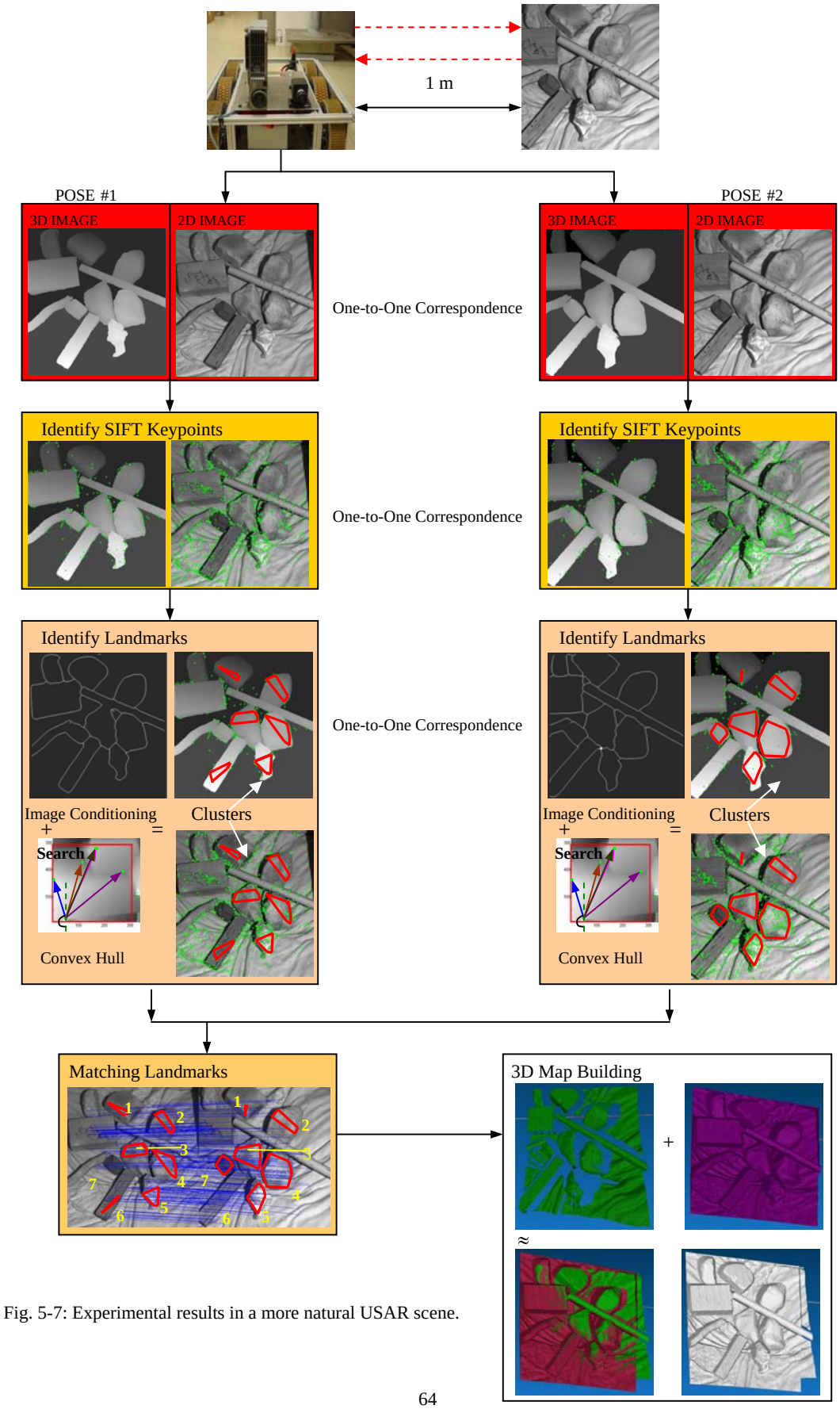

Fig. 5-7: Experimental results in a more natural USAR scene.

Chapter 6 Conclusions

6.1 Summary of Contributions

The primary contributions of this work are summarized below.

6.1.1 Landmark Identification and Matching

Chapter 3 presents a unique SIFT-based methodology to identify and match landmarks in unknown cluttered environments utilizing 3D and 2D images of a scene. Boundaries for landmarks are determined in the 3D images using the following techniques: Background subtraction, Edge Detection, the Morphological techniques of Dilation and Thinning and a Convex Hull geometric technique. The first application of using such a geometric technique for clustering of SIFT keypoints in the 3D images is proposed in this work. These boundaries are then superimposed on the 2D images for identification of large distinguishable landmarks as defined by clusters of 2D SIFT keypoints. Reliable matching of landmarks in the 2D images is proposed using the cluster information.

6.1.2 3D Visual SLAM

The identified landmarks are utilized for 3D Visual SLAM. In particular, an

ego-motion based method is proposed for localization of the robot using matched clusters of 2D keypoints and the 3D point cloud information provided by the sensor. In addition, an Iterative Closest Point (ICP) method is proposed for stitching of 3D information of the scene, where the initial input to the ICP algorithm is given by the clustered 2D keypoints and 3D information from the sensor. A map of the environment can be built by stitching numerous different views of the scene together.

6.1.3 Implementation

Several preliminary experiments were conducted to verify the overall methodology. Chapter 5 describes the sensory system in detail including a DLP projector and a high speed CCD camera. While the robot navigated through an environment, 2D and 3D images were taken in real-time. Utilizing the images taken by the sensor, the landmark identification and matching, and Visual SLAM algorithms were implemented. For these experiments, the clusters that had more than 3 correct keypoint matches were recognized by the algorithm to be the same landmark in the scene. Three sets of experiments were conducted. Experiment set #1 includes a USAR simulated scene in a controlled environment. Experiment set #2 contains a more USAR-like scene with the environment consisting of minimal lighting, rubble piles, pipes, wood, rocks and paper covered with a gray dust. These experiments have shown the efficiency of the proposed methodology for USAR use.

6.2 Discussion and Future Work

In the experiments presented, the effects of dusty environments have been considered and the sensory system was able to generate 2D and 3D images of the scene. However, the overall system will need to be tested in more harsh environments which include smoke and fire. Appropriate hardware changes will have to be considered.

The overall proposed methodology took at most 60 seconds of CPU time in Matlab on a Pentium IV 3.0 GHz 1.0G RAM system. Although it is efficient, it still can be optimized. The most time-consuming portion of the algorithms is the morphological processing step in image conditioning. The proposed algorithms can be implemented in a real-time programming language such as C++ and tested for optimization.

All three experiments are implemented in lab simulated USAR scenes. Future work consists of testing and evaluating the overall system in simulated or real USAR test environments, including the National Institute of Standards and Technology (NIST) Test Arenas.

6.3 Final Concluding Statement

In this thesis, the development of a unique SIFT-based landmark identification and matching method as well as a 3D Visual SLAM approach for cluttered USAR environments is proposed. The novelty of the method is the utilization of both 3D (i.e., depth images) and 2D images. Landmarks are determined effectively within the images

utilizing a combination of SIFT keypoints, depth segmentation, edge detection and morphological techniques and a convex hull algorithm for the construction of a 3D map of the environment. Experiments show the potential of the proposed methodology in USAR-like environments. Future work will include the adaptation of the system for harsh environments, time optimized implementation and testing in real USAR environments.

References

[1] R. R. Murphy, "Human-Robot Interaction in Rescue Robotics", *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, Vol. 34, No. 2, pp. 138-153, 2004.

[2] "Making the Nation Safer: The Role of Science and Technology in Countering Terrorism", Washington, D.C.: *National Academies Press*, 2002.

[3] "Grand Research Challenges in Information Systems", Washington, D.C.: *Computing Research Association*, 2003.

[4] B. Shah and H. Choset, "Survey on Urban Search and Rescue Robots", *Journal of the Robotics Society of Japan*, Vol. 22, pp. 582-586, 2004.

[5] A. Drenner, I. Burt, B. Kratochvil, B. Nelson, N. Papanikolopoulos, and K. B. Yesin, "Communication and Mobility Enhancements to the Scout Robot", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[6] G. Granosik, M. G. Hansen, and J. Borenstein, "The OmniTread Serpentine Robot for Industrial Inspection and Surveillance", *International Journal on Industrial Robots*, Special Issue on Mobile Robots, Vol. IR32-2, pp. 139-148, 2005.

[7] S. Se and P. Jasiobedzki, "Photo-realistic 3D Model Reconstruction", *IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 3076-3082, 2006.

[8] L. Zhang, B. Curless, and S. Seitz, "Spacetime Stereo: Shape Recovery for Dynamic

Senses", *Computer Vision and Pattern Recognition*, 2003.

[9] PMD Technologies, Available HTTP: http://www.pmdtec.com/.

[10] CSEM, "SwissRanger SR3000," Available HTTP: http://www.swissranger.ch.

[11] S. B. Gokturk, H. Yalcin, and C. Bamji, "A Time-Of-Flight Depth Sensor – System Description, Issues and Solutions", Available HTTP: http://www.canesta.com/assets/pdf/technicalpapers/CVPR_Submission_TOF.pdf.

[12] J. Craighead, B. Day, and R. Murphy, "Evaluation of Canesta's range sensor technology for urban search and rescue and robot navigation", *Technical Report: CRASAR-TR2006-2*, pp. 1-5, 2006.

[13] M. Rioux, "Laser range finder based on synchronized scanners", *Journal of Applied Optics*, Vol. 23, No. 21, pp. 3837-3844, 1983.

[14] J. Tripp, A. Ulitsky, S. Pashin, N. Mak and J. Hahn, "Development of a compact, high-resolution 3D laser range imaging system", *Proc. SPIE-The International Society for Optical Engineering*, Vol. 5088, pp. 112-122, 2003.

[15] M. Kurisu, Y. Yokokohji, and Y. Oosato, "Development of a Laser Range Finder for 3D Map-Building in Rubble the 2nd Report: Development of the 2nd Prototype", *IEEE International Conference on Mechatronics and Automation*, pp.1842-1847, 2005.

[16] A. Aboshosha and A. Zell, "Robust Mapping and Path Planning for Indoor Robots based on Sensor Integration of Sonar and a 2D Laser Range Finder", *IEEE 7th International Conference on Intelligent Engineering Systems*, 2003.

[17] L. Ellekilde, S. Huang, J. V. Miro, and G. Dissanayake, "Dense 3D Map Construction for Indoor Search and Rescue", *Journal of Field Robotics*, Vol. 24, No.

1-2, pp. 71-89, 2007.

[18] J. Pissokas and C. Malcolm, “Experiments with Sensors for Urban Search and Rescue Robots”, *International Symposium on Robotics and Automation*, 2002.

[19] J. Reich and E. Sklar, “Toward automatic reconfiguration of robot-sensor networks for urban search and rescue”, *Agent Technology for Disaster Management (ATDM) Workshop at Autonomous Agents and Multiagent Systems (AAMAS)*, 2006.

[20] M. W. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, “A Solution to the Simultaneous Localization and Map Building (SLAM) Problem”, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, pp. 229-241, 2001.

[21] M. Montemerlo, and S. Thrun, “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem”, *The National Conference on Artificial Intelligence*, pp. 593-598, 2002.

[22] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, “An Atlas Framework for Scalable Mapping”, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1899-1906, 2003.

[23] H. Ishida, K. Nagatani, and Y. Tanaska, “Three-Dimensional Localization and Mapping for a Crawler-type Mobile Robot in an Occluded Area Using the Scan Matching Method”, *IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, pp. 449-454, 2004.

[24] Y. Yokokohji, M. Kurisu, S. Takao, Y. Kudo, K. Hayashi, and T. Yoshikawa, “Constructing a 3-D Map of Rubble by Teleoperated Mobile Robots with a Motion Canceling Camera System”, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3118-3125, 2003.

[25] A. J. Davison, and D. W. Murray, “Simultaneous Localization and Map-Building Using Active Vision”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 865-880, 2002.

[26] D. G. Lowe, “Object Recognition from Local Scale-invariant Features”, *Int. Conference on Computer Vision*, pp. 1150-1157, 1999.

[27] S. Se, D. Lowe, and J. Little, “Vision-based Mobile Robot Localization and Mapping using Scale-invariant Features”, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2051–2058, 2001.

[28] M. Baker, R. Casey, B. Keyes and H. A. Yanco, “Improved interfaces for human-robot interaction in urban search and rescue”, *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, 2004.

[29] A. Jacoff, E. Messina, B. Weiss, S. Tadokoro, and Y. Nakagawa, “Test Arenas and Performance Metrics for Urban Search and Rescue Robots”, *Proceedings of the Intelligent and Robotic Systems Conference (IROS)*, 2003.

[30] R. Smith, M. Self, and P. Cheeseman, “Estimating Uncertain Spatial Relationships in Robotics”, *Autonomous Robot Vehicles*, pp. 167–193. Springer, 1990.

[31] Z. Zhang, H. Guo, G. Nejat, and P. Huang, “Finding Disaster Victims: A Sensory System for Robot-Assisted 3D Mapping of Urban Search and Rescue Environments”, *IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 3889-3894, 2007.

[32] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Upper Saddle River, New Jersey: Prentice-Hall, 2nd Edition, pp. 534-549, 2002.

[33] Image Processing Toolbox, Matlab 7.0.0 (R14), The MathWorks, Inc.

[34] D. G. Lowe, “Distinctive Image Features from Scale-invariant Keypoints”,

International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110, 2004.

[35] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp. 1615-1630, 2005.

[36] D. R. Chand and S. S. Kapur, "An algorithm for convex polytopes", *Journal of ACM*, pp. 17-78, 1970.

[37] K. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter, "Camera Calibration Toolbox for Matlab", Available http://www.vision.caltech.edu/bouguetj/calib_doc/, 2005.

[38] S. Zhang and P. Huang, "High-resolution, real-time 3-D shape acquisition", *IEEE Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2004.

[39] J. T. Schwartz and M. Sharir, "identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves", *Int. J. Robotics Rex*, Vol. 6, No. 2, pp. 29-44, 1987.

[40] B. Kamgar-Parsi, J. L. Jones, and A. Rosenfeld, "Registration of multiple overlapping range images: Scenes without distinctive features", *Proc. IEEE Computation & Pattern Recognition Conference*, 1989.

[41] R. Szeliski, "Estimating motion from sparse range data without correspondence", *2nd Int. Conf. Comput. Vision*, pp. 207-216, 1988.

[42] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes", *IEEE Trans. PAMI*, Vol. 14, pp. 239-256, 1992.

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

[44] Z. Zhang, “Iterative point matching for registration of free-form curves and surfaces”, *International Journal of Computer Vision*, 13(2):111-152, 1994.

[45] GSI Technologies, Available HTTP: http://www.geometrysystems.com.