CHARACTERISTICS OF INDOOR DISASTER ENVIRONMENTS AND THEIR IMPACT ON SIMULTANEOUS LOCALIZATION AND MAPPING FOR SMALL UNMANNED AERIAL SYSTEMS

A Thesis

by

SIDDHARTH AGARWAL

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Jean-Francois Chamberland
Co-Chair of Committee,	Robin R. Murphy
Committee Members,	Scott L. Miller
	Gregory H. Huff
Head of Department,	Miroslav M. Begovic

August 2015

Major Subject: Electrical Engineering

ABSTRACT

This thesis explores the use of small unmanned aerial systems (SUASs) for mapping of unknown disaster environments and investigates the impact of characteristics of such challenging environments on simultaneous localization and mapping (SLAM) algorithm. It provides a formal analysis of indoor disaster environments and identifies four characteristics of a region of space: scale, degree of deconstruction, location of obstacles, and tortuosity. The analysis compares the value of these characteristics for Prop 133 at Disaster City and develops computer simulated environments. Furthermore, a SLAM algorithm for SUAS flying in indoor disaster environments is developed and the system is tested in these virtual environments. Three different environments with increasing deconstruction are designed. For each type of environment, 10 different maps with a common floor plan are simulated with randomly placed obstacles. For each map, three trials with varying flight paths are run, thus conducting 90 trials of experimentation. As verified from the statistical testing, there is a convincing increase of 26.36% in the average value of RMSE as the deconstruction changes from Group 1 to Group 3. But, the change in value of error is not statistically convincing when Group 1 and 2 and, Group 2 and 3 are respectively compared. Hence, though the result suggest that the value of error increases between different groups, it cannot be claimed that the RMSE in localization will always increase with deconstruction. The tortuosity increases with deconstruction and this value is empirically calculated. The average RMSE in localization does not change as the Agent to Environment ratio changes. These results can help identify the remaining gaps in the state of the art indoor SUAS for disasters.

DEDICATION

This thesis is dedicated to my advisors - Dr. Murphy and Dr. Chamberland, my lab mates, my parents and my friends for their unconditional encouragement and support.

ACKNOWLEDGMENTS

All the credit for this work goes to my advisors - Dr. Murphy and Dr. Chamberland. Their unconditional support and continued belief made this endeavor successful. I am grateful to Dr. Adams for her contribution and advise in designing of the SLAM system. I thank the committee members - Dr. Miller and Dr. Huff for their expert evaluation and feedback. I heart-fully thank Dr. Suntzuff for his guidance and support that lead me to this field.

I am grateful to my lab mates - Brittany Duncan, Grant Wilde, Traci Sarmiento, Khuong Nguyen, Jesus Suarez and Pranay Kumar for their feedback and support in all phases of the project.

Portions of this work were supported by a SEC Travel grant and NSF Grant IIS-1143713 EAGER: Shared Visual Common Ground in Human-Robot Interaction for Small Unmanned Aerial Systems. I thank the Texas A&M Engineering Extension Service for access to their facilities and acknowledge the support of Electrical and Computer Engineering, and Computer Science departments at the Texas A&M University for making this joint research possible.

Lastly, I am thankful to my family, friends and colleagues for their moral support and love.

NOMENCLATURE

SUAS	Small Unmanned Aerial System
SLAM	Simultaneous Localization and Mapping
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
CD	Characteristic Dimension
US&R	Urban Search and Rescue
EM	Expectation Maximization
DOF	Degrees of Freedom
RMSE	Root Mean Squared Error

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

LIST OF TABLES

1. INTRODUCTION *

The possibility of using Small Unmanned Aerial Systems (SUASs) for surveying damage inside buildings and structures affected by a disaster is increasing. The successful flights by a University of Pennsylvania/Tohoku University team [41] in a multi-story building damaged by the 2011 Great Eastern Japan Earthquake and the multi-university NIFTi team's inspection of cathedrals in Mirandola collapsed by the Finale Emilia Earthquake [28] demonstrate the potential utility of SUAS for multi-story buildings and processing facilities, such as Fukushima Daiichi. However, the inability of SUAS to make progress in indoor flights inspecting buildings in Biloxi damaged by Hurricane Katrina in 2005 [48] and the Christchurch Catholic Basilica damaged by the 2011 Christchurch, New Zealand, Earthquake [45] act as reminders of remaining challenges.

Small UAS for flying indoors is fortunately an active area of investigation [41, 38, 33, 66, 20, 61, 2, 25, 3, 37, 18, 59] SUASs are being used to explore obstacle-filled indoor environments without relying on outside operators or sensors and GPS waypoints. To do so, the agent must first accurately localize itself and simultaneously map the unknown environment for obstacle avoidance. Over the past decade, researchers have made significant progress in solving this problem. However, advances in flying for normal, undamaged indoor environments may not be directly transferable to disaster response. Kinetic events, such as earthquakes, tornadoes, hurricanes, industrial accidents, or explosions often *deconstruct* interiors, while leav-

*Part of this section is reprinted with permission from S. Agarwal, R.R. Murphy, and J.A. Adams. Characteristics of indoor disaster environments for small UASs. In *Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on*, pages 1–6, Oct 2014. Copyright[2014] by IEEE.

ing the building compromised. A mild earthquake may rearrange office furniture, knock over bookcases, and cause ceiling fixtures to hang loose, while leaving the structural elements, such as walls, ceilings, floors, and pillars intact. A more severe event will have a higher degree of deconstruction, collapsing walls and ceilings, depositing debris, and changing the overall layout of the building. Substantial issues remain in practically realizing these solutions and efficiently building perceptually rich maps of unstructured environments. Most of these compelling algorithms rely on the assumptions of flying in a structured environment that is made up of vertical walls and horizontal planes that are all piecewise constant. Though this assumption is reasonable in most applications, it is easily violated in disaster environments. Indoor SUAS navigational algorithms for obstacle avoidance and SLAM depend upon these assumptions and may fail in disaster conditions. Therefore, having an accurate characterization of deconstructed indoor environments and understanding its impact on these algorithms is essential to developing indoor SUAS that can fly in realistic disaster conditions. Moreover, to our best knowledge, none of the existing systems have been tested in as rigorous an environment as that of an average disaster[1].

We start this chapter by understanding the characteristics of indoor disaster environments and then discuss the SLAM problem.

1.1 Characteristics of Indoor Disaster Environment

This thesis uses the definitions from Disaster Robotics [45] to describe the characteristics of Indoor Disaster Environment. While these definitions were originally developed for unmanned ground vehicles, they can be extended to the three dimensional environments SUAS operate within.

The operational envelope for a SUAS is defined as a collection of one or more

regions. For example, in a multi-story office building, a hallway is a distinct region from an office and a stairway. The environmental characteristics influencing the navigability of a region can be divided into three groups: the scale and degree of deconstruction, which captures the state of the structure; the severity of obstacles and tortuosity, which captures the impact of the deconstructed structure and damaged furnishings; and other characteristics that affect sensing.

1.1.1 Scale and Degree of Deconstruction

The scale of a region reflects the relationship of the size of the agent A to the size of the environment E [45]. A large environment, such as high bay provides more space for a SUAS than a narrow hallway. To quantify this, scale is given as the relative size of characteristic dimension CD of the agent and environment. The A_{cd} is the largest single dimension affecting SUAS navigation. For example, in Fig. 1.1 a SUAS has a platform size with a diameter of 0.5m in the horizontal plane with cameras and payloads protruding 0.2m, and a constraint that the SUAS is never allowed closer than 0.3m to an obstacle. Therefore, the maximum dimension i.e. the $A_{cd} = 1.1m$. Note that the A_{cd} is the equivalent of reducing a SUAS to a sphere.

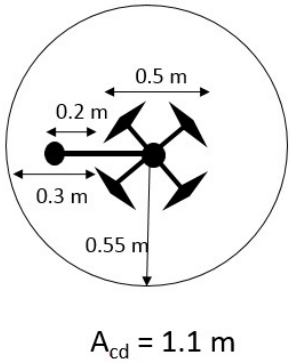
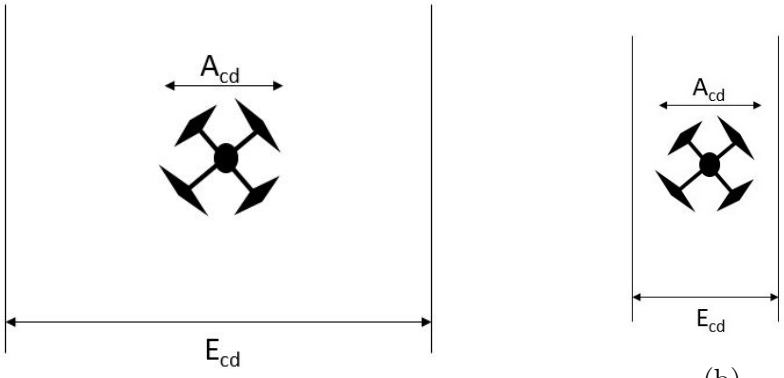


Figure 1.1: Characteristic dimensions of a SUAS.

The E_{cd} is the nominal minimum dimension of the environment affecting navigation. For a hallway, it is the average width, as obstacles intruding into the hallway are rare. An office may have a smaller E_{cd}, where the furniture is arranged to allow a human to walk through, but with less free space than in a hallway and a lower ceiling.

The intrinsic navigability of a region based on scale can be categorized as one of three indoor regimes in [45]. As shown in Fig. 1.2a, when $E_{cd} > 2A_{cd}$, the agent can move freely through the environment in the *habitable* regime. For a human, this regime represents "normal" interior spaces designed for people to work and live in that have not been altered by a kinetic disaster. For a SUAS with a A_{cd} about the width of a person, a human habitable space will be the same as a SUAS habitable space. A SUAS may be deployed into a habitable environment if there was a chemical, radiological, or biological incident where human movement was restricted by safety procedures or personal protection gear, such as the use of UGVs at the Fukushima Daiichi nuclear emergency. In the *restricted maneuverability* regime shown in Fig. 1.2b, $E_{cd} < 2A_{cd}$ - the agent can still move in the environment, but that movement is restricted by the much narrower spaces. The environment may be naturally small, such as a sewer pipe; however, the more interesting case for disasters are human habitable environments that have become deconstructed from normal dimensions. A partially collapsed building from a kinetic event, such as an earthquake or explosion that a responder can walk through, though perhaps have to bend over or squeeze through, is an example of a restricted maneuverability regime. Robots for surface entry into mine disasters or parking garage collapses function in this regime.

In the third indoor regime, the agent is burrowing into the environment and working at a granular level, $E_{cd} < A_{cd}$. It is not possible for an SUAS to displace material and create space for itself, so only the habitable and restricted maneuver-



(a) (b)

Figure 1.2: Types of regimes. a) Habitable - $E_{cd} > 2A_{cd}$ and b) Restricted maneuverability - $E_{cd} < 2A_{cd}$.

ability regimes are discussed.

The *degree of deconstruction* of an indoor region reflects the condition of the structural elements, essentially are the walls and ceilings still orthogonal and in place. More the damage to the structural elements, higher is the degree of deconstruction. For example, an environment affected by a kinetic disaster has a high degree of deconstruction while a normal environment has none.

1.1.2 Severity of Obstacles and Tortuosity

The *severity of obstacles* captures the number and size of obstacles that temporarily reduce the E_{cd} and may require obstacle avoidance. If the environment is essentially a path through nearly continuous obstacles, the free space between obstacles becomes E_{cd}. A normal habitable space will have very few navigational obstacles, as human spaces are designed for people to move and work in. A kinetic event may deconstruct the habitable space by creating debris and hanging obstacles. However, what is an obstacle for a human or a ground robot, may not be an obstacle for a

SUAS. Therefore, this research rates severity based on location:

- Obstacles on the ground, below the nominal flying zone (Fig. 1.3a).
- Obstacles on the ground, up to the nominal flying zone (Fig. 1.3b).
- Obstacles hanging from the ceiling, in the nominal flying zone (Fig. 1.3c).
- Obstacles hanging from the ceiling, above the nominal flying zone (Fig. 1.3d).

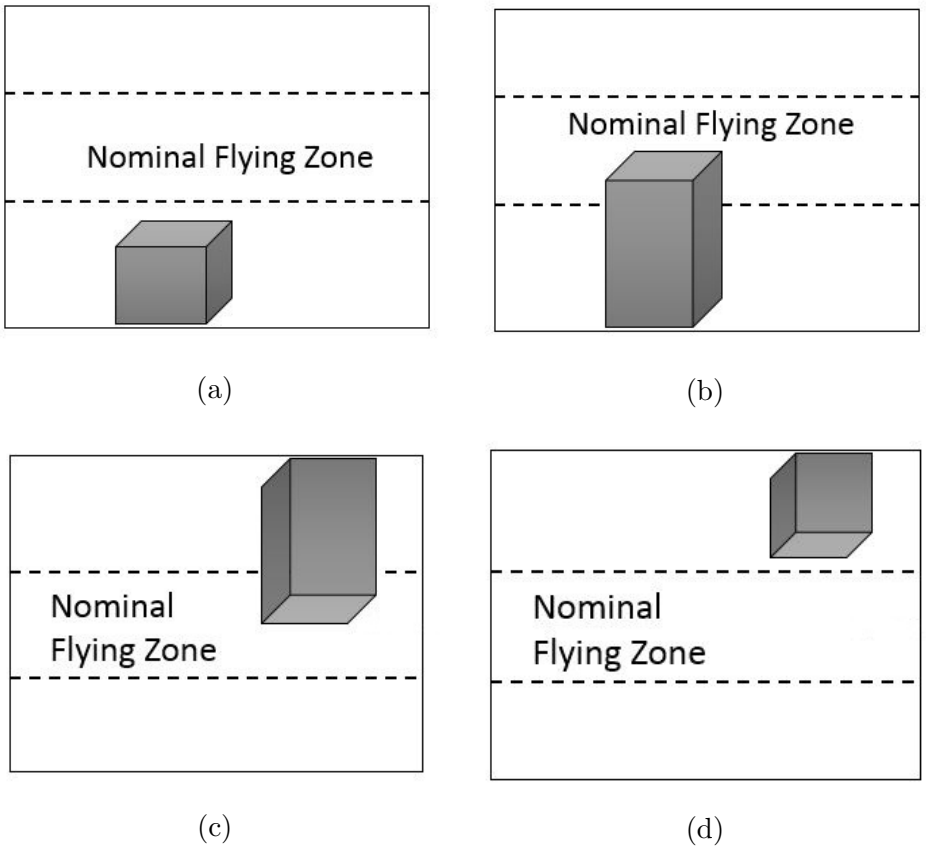


Figure 1.3: Severity of obstacles. a) Obstacles on the ground, below the nominal flying zone, b) Obstacles on the ground, up to the nominal flying zone, c) Obstacles hanging from the ceiling, in the nominal flying and d) Obstacles hanging from the ceiling, above the nominal flying zone.

The deconstruction and severity of obstacles in turn leads to the *tortuosity* of a region. Tortuosity represents the meters between turns, including changes in altitude, in the region for navigation; it does not include yawing to provide sensor views. Tortuosity is calculated as the number of turns taken by the SUAS per unit distance. For example in Fig. 1.4, if the SUAS takes 3 turns to avoid obstacles over a linear distance of 6m, then the tortuosity is $3/6 = 0.5$. The tortuosity at Prop 133 is estimated to be 1.0, i.e., 1 turn per meter. A low tortuosity indicates that the frequency of obstacle avoidance is low and the environment is comparatively easier to navigate, as opposed to one with higher tortuosity.

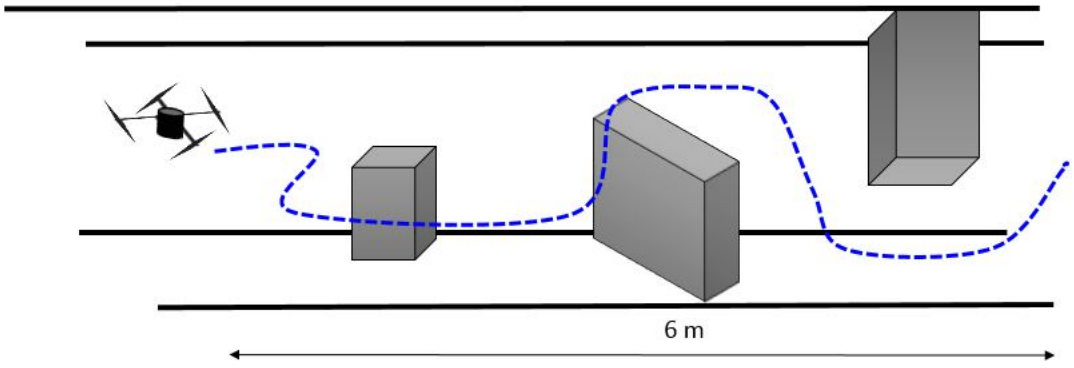


Figure 1.4: Calculation of tortuosity.

1.1.3 Other Environmental Characteristics

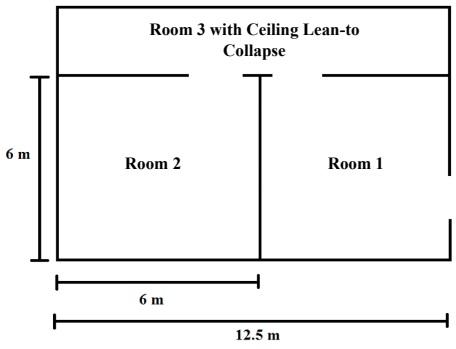
In addition, the performance of an indoor SUAS will also be influenced by other secondary components. Lighting conditions and surface properties are two characteristics that have been observed at disasters [45]. Cameras do not work well in dim lighting and may need an artificial light source, while the Kinect does not work in high luminescent conditions, as shown in [60]. The building materials including

metal, glass and sharp edges may scatter active sensors, such as LIDAR and ultrasound. Carpets, cloth, soundproof tiles and partitions typically found in office buildings may absorb sound signals. Furthermore, suspended dust due to debris and loose building materials may affect the visibility and make sensors less effective or even non-functional.

1.2 Average Indoor Disaster Environment

Since SUASs have been used only four times for surveying the interior of damaged buildings [45], those data sets are too limited to project the broad set of regions for indoor flight. However, Prop 133 at Disaster City® was designed to represent an average expected state of a damaged multi-story commercial building and is presented as a pictorial example of these characteristics for an average case. While Prop 133 stages only one possible scenario, the partial collapse of a multi-story office building, it is a realistic representation used for training responders and thus is helpful in visualizing the characteristics.

(a)



(b)

Figure 1.5: Prop 133 at Disaster City®. a) View of the federal building component and b) The floor plan for the first and second floors. Copyright [2014] IEEE [1].

Disaster City® is a complex of props designed by professional trainers, who are themselves responders, to accurately represent physical conditions that urban search and rescue (US&Rs) teams will experience for a range of disasters. It is owned by the Texas A&M Engineering Extension Service and is used to train over 80,000 humans and canines annually, including Federal Emergency Management Agency (FEMA) US&R teams. None of the spaces are specifically designed for robots. Prop 133, shown in Fig. 1.5 is an exemplar of realistic deconstructed human habitable or human restricted maneuverability indoor office building and thus is a projection of what a SUAS will encounter. Portions of the prop follow the floor plan and room size of a multi-story government office building, such as the standing portions of the Alfred P. Murrah Federal Building destroyed at the Oklahoma City bombing in 1995. The prop consists of six office-sized rooms on two floors, with four of the rooms structurally intact and two rooms part of a lean-to collapse. It should be noted that Prop 133 does not have the carpets, wallpaper, acoustic tiles, or other organic materials normally found in an office building, as those furnishings will mold in the outdoors; therefore, Prop 133 may be less challenging for robotic navigation and sensing than the partial collapse of an actual office building.

1.3 Simultaneous Localization and Mapping

Simultaneous localization and mapping (SLAM) has been one of the most challenging problems in robotics. SLAM is the problem of constructing a map of an unknown environment while simultaneously tracking the location of the agent within it. SLAM was originally developed by Hugh Durrant-Whyte and John J. Leonard [31] based on earlier work by Smith, Self and Cheeseman [56].

Mapping deals with the problem of representing the world. It answers the question "What does the environment of the agent look like?" It deals with the representation of the environment and the interpreting sensor data. On the other hand, localization involves estimating the pose of the robot with respect to the map. It answers the question, "Where is the agent in the environment?" In practice, these two problems are dependent on each other. To localize itself, the robot needs to know the map and to map an unknown place, the robot needs to know its location. Therefore, SLAM is often associated with the chicken and egg problem: An accurate map is needed for localization of the agent while an accurate pose estimate is needed for an agent to build a map.

There are several algorithms known for solving the SLAM problem. SLAM algorithms are customized according to the availability of resources and, are not aimed at perfection. Popular approaches are employed in self-driving cars, aerial vehicles, underwater vehicles, planetary rovers, and even inside the animal body.

The process of solving the problem begins with the agent itself. The type of agent used must have good odometry i.e. estimate of robot's own position. However, there is normally a significant margin of error with odometry readings and the agent might be off in its measurements. Consequently, the robot is not where it thinks it is in a given environment. This error is reduced by analyzing the observations from the environment when the robot moves around. A range measurement device is often used for observing the environment around the robot and create a map. The most common form of sensors that are used are laser scanner, sonar and imaging devices. A Kalman Filter is generally responsible for updating where the robot thinks it is based on observed features. The Kalman Filter keeps track of an estimate of the uncertainty in the robots position and the uncertainty in the observed features. Thus, the robot simultaneously corrects errors in its pose and maps the environment.

The SLAM problem can be mathematically represented as:

Given:

u_k - Control Signal applied at k-1 to drive the agent from x_{k-1} to x_k

z_k - Feature Observation (Sensor Measurements) at instant k

Estimate:

m_i - True location of ith feature

x_k - Agent Pose at instant k

m - Set of all features

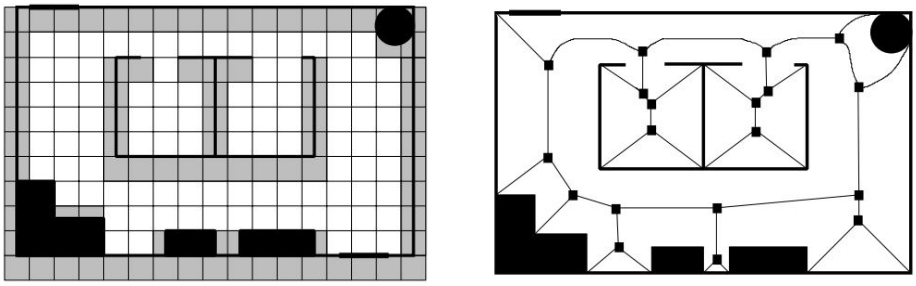
Therefore probabilistically, SLAM requires the distribution:

$P(x_k, m|Z_{0:k}, U_{0:k}, x_0)$ to be computed for all instants k.

This distribution describes the joint posterior density of the feature position and the agent's state based on the recorded environmental observations and control inputs up to and including the instant k together with the initial state of the vehicle [16]. A number of approaches to solve the SLAM problem are discussed in the next chapter.

As for as mapping is concerned, there are two major techniques [44]. Regular Grid based maps (Fig. 1.6a) are collection of discrete spaces/cells. If there is any object in the cell, that cell is marked occupied. Hence, regular grids are often referred to as occupancy grids. Their computational complexity depends on the grid size and its resolution. A variant on regular grid is quadtrees. It avoids wasting space by starting out with grid elements representing a large area. If an object falls into the cell, but not all of it, the algorithm divides the cell into four smaller cells. If the object does not fill a particular sub-cell, the algorithm does another recursive division of that cell into four more sub-cells, and so on. A 3D quadtree is called an octree. The second type technique is called Generalized voronoi graphs (Fig. 1.6b). They form a relational graph of nodes where the entire area is mapped as a graph. The voronoi graphs are sensitive to noise and require the robot to sense all boundaries.

On the other hand, the occupancy grids suffer from digitization bias where left over space is marked unoccupied.



(a) (b)

Figure 1.6: Mapping techniques. a) Regular grid map and b) Generalized voronoi graph. Copyright [2000] MIT Press [44].

2. LITERATURE REVIEW *

An extensive review of existing literature was carried out to understand the SLAM problem for indoor environments and formulate our approach. This chapter first discusses the systems that were identified to develop an analysis of indoor disaster environments and later describes the existing SLAM and Loop Closure approaches.

2.1 Indoor Small Unmanned Aerial Systems

Twelve systems were identified that have experimented with indoor SUASs and four systems were intended for application to search and rescue [33, 3, 41, 61]. The twelve systems were evaluated in one or more of three testbeds: computer simulation [3, 37, 18, 59, 25], physical - general indoor environments [33, 61, 38, 66, 20, 2, 37, 18, 59], or in a building that had experienced an actual disaster [41]. A summary of the specifications of the twelve robots, the scale of the testbed, the type of testbed, the type of regions represented in the testbed, and the nominal flight altitude is provided in Table 2.1.

2.1.1 Computer Simulation

Five SUASs were evaluated using computer simulated testbeds. Two of the five simulated environments emulated habitable scale open spaces. Jongho and Youdan simulated open spaces contained parallelepipeds, with sides of 1m and 3m [25]. Stow-

*Part of this section is reprinted with permission from S. Agarwal, R.R. Murphy, and J.A. Adams. Characteristics of indoor disaster environments for small UASs. In *Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on*, pages 1–6, Oct 2014. Copyright[2014] by IEEE.

No.	Author	SUAS Diameter (m)	Scale	Testbed	Region	Nominal Altitude (m)
1	Masanori et al., 2013	–	Habitable	Physical - General Indoor (Staged)	Open Space	0.7
2	Li et al., 2013	0.57	Habitable	Physical - General Indoor (Staged)	Office	–
3	MacAllister et al., 2013	–	Restricted Man.	Physical - General Indoor (Staged)	Collection of hallways, offices and open space	0.7
				Computer	Office	–
4	Jongho and Youdan, 2013	–	Habitable	Computer	Open Space	7.0
						2.0
5	Al Newaz et al., 2013	–	Habitable	Computer	Office	–
6	Fossel et al., 2013	0.73	Habitable	Physical - General Indoor (Natural)	Office	–
		–		Computer	Collection of lab and office	–
					Open Space	–
					Lab	–
7	Toratani et al., 2013	0.54	Habitable	Physical - General Indoor (Staged)	Open Space	0.8
8	Grzonka et al., 2013	–	Habitable	Physical - General Indoor (Natural)	Hallway	0.5
					Office	–
9	Michael et al., 2012	0.65	Restricted Man.	Actual Disaster Environment	Collection of Offices and Hallways	2.0
10	Stowers et al., 2011	–	Habitable	Physical - General Indoor (Staged)	Lab	1.5
				Computer	Open Space	–
11	Suzuki et al., 2010	1.0	Habitable	Physical - General Indoor (Staged)	Open Space	1.5
12	Ahrens et al., 2009	0.54	Habitable	Physical - General Indoor (Staged)	Open Space	0.5

Table 2.1: A summary of the reviewed systems, including SUAS size, environmental scale, space classification and nominal altitude. Copyright [2014] IEEE [1].

ers et al. used a large number of blocks with heights up to 3m [59], while Al Redwan Newaz et al. simulated an office space containing multiple objects positioned on the floor with varying heights [3]. Two others simulated a habitable scale office space. Fossel et al. simulated an office space containing orthogonal walls, an open space containing vertical pillars, and a collection of laboratory and office space with a tilted wall [18]. The fifth system simulated a restricted maneuverability office space [37]. The environments represented randomly generated areas ranging in size from (25 x 25 x 3)m to (50 x 50 x 3)m. 20% of the environmental area contained floor to ceiling walls and randomly placed obstacles, such as boxes projecting from the floor up to a random height, and fixed width beams mounted at random heights.

2.1.2 Physical - General Indoor Environment

Nine systems were evaluated in a physical - general indoor testbed, with eight systems flying in habitable scale spaces and only one in a restricted maneuverability.

Eight of the nine systems were evaluated in habitable scale environments [33, 61, 38, 66, 20, 2, 18, 59], while the remaining system was evaluated in a restricted maneuverability scale environment. Four of the environments are classified as open space [61, 38, 66, 2]. Masanori et al.'s open space environment contained three 1m cylinders and a horizontal cross section that protruded from one cylinder at a height of 0.8m from the ground [38].

Torantani et al. operated in a testbed with a rectangular obstacle 1m wide and 0.8m high, right at the nominal flight altitude [66]. Suzuki et al.'s SUAS flew at a nominal altitude of 1.5m, while avoiding a white board of similar height [61]. Ahrens et al.'s open space testbed contained a 1m long cylindrical pole and a cuboid, similar to a bar stool [2]. The SUAS flew at a nominal altitude of 0.5m, as inferred from

the paper, but flew up to 1.5m to avoid the obstacles. The second most common habitable scale evaluation testbed was an office space [33, 20, 18]. Li et al.'s SUAS planned paths to allow the vehicle to avoid tables by flying underneath them from one side to the other [33], while Fossel et al.'s environment contained "low lying" tables, cabinets, and benches [18]. Grzonka et al.'s SUAS flew at a maximum of 1.5m, while avoiding 48cm high chairs and 77cm high tables with other obstacles [20]. Stowers et al.'s laboratory environment contained two large benches with instruments on them, for a total height of 3m above the floor [59]. Their SUAS flew at a nominal altitude of 1.5m, as inferred from the paper. A 41m hallway provided a second environment in which Grzonka et al.'s SUAS flew at a nominal altitude of 0.5m, as inferred from the paper.

MacAllister et al.'s [37] SUAS was the only system evaluated for a restricted maneuverability scale environment. Their environment was a collection of hallways, offices and open spaces containing obstacles of random heights that were placed on the floor and a horizontal bar placed at 0.7m above the ground.

2.1.3 Search and Rescue

Four systems explicitly discussed the search and rescue applications. Michael et al. [41] conducted evaluations in a building on Tohoku University's campus containing hallways and offices. The building had been damaged by an earthquake, but was still accessible to humans and was at the habitable scale. Al Redwan Newaz et al. [3] considered a surveillance and recovery mission after nuclear disasters or severe accidents in industrial areas as an application, and tested a SUAS in computer simulated habitable office space. Two systems [33, 61] were evaluated in physical - general indoor staged office and open spaces, respectively.

2.2 Simultaneous Localization and Mapping

Smith, Self, and Cheeseman [57, 58] introduced a statistical framework for simultaneously solving the problem of mapping the environment and simultaneously localizing the agent relative to its growing map [63]. Since then, robotic mapping has commonly been referred to as SLAM [13, 17].

The SLAM approach is dependent on the type of sensors available. There are different sensors with each having its own advantages. At one end, laser scans and vision provide detailed representation of an area via range based point clouds. At the other end, tactile sensors are very sparse. Most SLAM approaches use a combination of sensors.

SLAM algorithms can also be divided into landmark-based and raw-data approaches. Landmarks are uniquely identifiable objects in the agent's environment. Raw-data approaches make no assumption about existence of landmarks and observe each point in the environment as a function of location. For most outdoor applications, SLAM algorithms usually rely on high precision differential GPS sensors. These may be viewed as location sensors with very sharp likelihoods. However GPS sensors may go down entirely due to shadowing from other objects, bad weather conditions or jamming in military applications.

A number of approaches can be used to solve the SLAM problem:

2.2.1 Kalman Filters

One family of probabilistic approaches [7, 9, 14, 22, 32, 46, 68] use Kalman filters to represent the map and estimate the robot's location. These maps generally describe the location of landmarks, or features in the environment. In some cases,

the environment is also represented through raw range sensor measurements [63]. These approaches describe the SLAM problem as:

Motion Model:

$x_{k+1} = F_k x_k + G_k u_k + v_k$

Sensor Model:

$y_k = H_k x_k + w_k$

where

x_k is the n - dimensional state vector.

u_k is the m - dimensional input vector.

y_k is the output vector.

F_k, G_k and H_k are system matrices.

v_k and w_k are zero mean white gaussian noise.

The Kalman Filter is a recursion that provides the best estimate of the state vector x. The advantage of this method is that its covariance matrix can converge strongly and it provides optimal estimates of the state. However, the assumption that that all noise processes are gaussian can restrict its use.

2.2.2 Particle Filters

Some methods use particle filters [15, 10, 43, 42, 55] that are inherently Bayes filter that efficiently represent non-Gaussian distributions. Their basic principle is to start with a set of particles and test the survival of the fittest as time passes. Thus, particle filters are models representing probability distribution as a set of particles which occupy the state space. Such SLAM techniques decouples the feature-map from pose. Each particle represents the robot's pose and correlated feature measurements. In the update step a new particle distribution, for a given motion model and

controls is generated. For each particle, the prediction of measurements are compared with actual measurements. Particles with higher prediction match are given a high weight. These methods can handle non-gaussian noise and non linearities but the complexity grows as new features are detected.

2.2.3 Expectation Maximization

Another family of algorithms [11, 51, 62] is based on the expectation maximization algorithm [12, 39]. Expectation Maximization (EM) estimation is a statistical algorithm which was developed in the context of maximum likelihood (ML) estimation and it offers an optimal solution. EM iterates in two steps: the first step is an expectation step, where the posterior over robot poses is calculated, and the second step is maximization step, in which the most likely map is calculated given these pose estimates. [64]. It is generally used for building maps when the poses are known. But the estimation cost grows exponentially with the map and the error is not restricted [63].

2.2.4 Recent Approaches

More recently, Grzonka et al.[20] developed a navigation system for indoor flying vehicles. Their system includes state estimation modules for localization, altitude estimation, and SLAM.

Shen et al.[52] improved upon this approach by addressing the problem of multi-floor mapping with loop closure, localization, planning, and autonomous control. To ensure that the robot is fully autonomous, they did all computation on the robot without any external communication, or human interaction beyond high-level commands.

New SLAM algorithms are an active research area, and are often driven by differing requirements and assumptions about the environment, sensors and models.

2.3 Loop Closure

Loop closure is deemed as one of the main challenges in developing, a real-time, large-scale SLAM system [23, 35]. During the SLAM process, the robot may come to a place that it has visited before. This problem of recognizing previously-visited locations and updating the poses accordingly is called Loop Closure. In both topological and metrical SLAM algorithms, loop closure is the key to building consistent maps. A successful loop closing prevents re-mapping of the same location and allows errors in a map to be corrected.

The problem has been approached in multiple ways:

2.3.1 Feature Matching

Geometric Features in the environment can be matched to detect loop closures. In [4, 19, 67], bag of words methods are used to perform loop-closure detection. Bag of words methods represent the acquired images as a set of elementary features taken from a dictionary. This dictionary is built by clustering similar visual descriptors extracted from the images. Using a given dictionary, classification of image is based on the frequencies of the words. A major problem with this technique is perceptual aliasing i.e. physically distinct locations may appear similar to robot sensors.

Liu and Zhang [34] proposed a method for visual loop closure detection in appearance based SLAM. Unlike the bag of words approach, their method uses direct feature matching to detect loop closures and avoids the perceptual aliasing problem.

Shen et al. [52] correct the global inconsistency by employing vision-based techniques to enable loop closure. The loop closure detection that does not depend on the error in pose estimation. A fixed size vocabulary is constructed by clustering a large number of SURF features. These features are converted into the vocabulary and matched with previously obtained images. This matching is accomplished via histogram voting. If any matches are found, the matched candidates are verified by using scan matching. They only close the new loops. Although the proposed method maintains a globally consistent map, it is an approximation.

Wu et al. [69] presents a novel method for loop closure detection with low resolution binary that has been shown to be suitable to handle an appearance map with as many as 20 million images in slightly over two seconds. The proposed method does not require off-line visual vocabulary construction, as do the popular approaches in visual loop-closure detection based on visual Bag of Words.

Lynen et al. [36] have presented a method for batch placeless place recognition using projected binary descriptors and a kNN voting scheme with a loop-candidate segmentation using statistical-tests . Instead of scoring individual images spaced by time, they formulate place recognition as a continuous 2D probability density estimate in the space of matches along path distance. This allows us them to handle different sizes of places, indoor and outdoor environments as well as perceptual aliasing in a continuous and placeless way.

2.3.2 Sensor Data Matching

Raw data from sensors such as Laser scanners can be used to recognize loops by finding the similarity between two readings (scan matching) [23]. The latest sensor data is matched to some previously acquired data.

Grzonka et al. [21] used scan matching to detect loop closures. the method worked by identifying all previous poses that were within the bounds of the pose uncertainty. This uncertainty is obtained by a Dijkstra projection of the node covariances. If a good match is found, the obtained graph was augmented by adding a new edge. But this method needs good guess of initial poses and may not work well with all kinds of sensors. Other difficulties occur in data association - the process of making this decision for each sensor observation. Some different parts of the world may appear the same to the sensor and the measurements could be noisy.

2.3.3 Hybrid Feature-Sensor Matching

Newman et al. [47] illustrated how visual features, used in conjunction with scanning laser data, can be used to a great advantage for loop closured. Their paper presented initial results concerning the use of salient image features in detecting possible loop closure events that are independent of estimated pose.

Latif et al. [29] developed an incremental algorithm for loop closure. The algorithm was used to detect if the recognition system has generated any false constraints and was responsible for removing them if required. This approach is based on the observation that correct loop closure and odometry measurements can help in the detection of false loop closures. The estimation process differentiates between correct and incorrect loop closures.

Kerl et al. [26] proposed a dense SLAM method for RGB-D cameras that uses keyframes and an entropy-based loop closure detection to eliminate drift. To reduce the search space, they use metrical nearest neighbor search and look for loop closure candidates in a sphere with predefined radius around the keyframe position.

2.3.4 Expectation Maximization

The Expectation Maximization algorithm can also be used to tackle the problem of loop closures. In a recent approach [30], the loop-closure problem was modeled as a Bayesian network and solved with the EM algorithm. The robot poses and constraints were latent and observed variables and, an additional set of latent variables were introduced as weights for the loop-constraints.

3. ANALYSIS AND DESIGN *

The novelty of our approach is that we identify the key characterization of indoor disaster environments to develop a realistic computer simulated test-bed. This will help us test our SLAM algorithm in conditions that are representative of an actual disaster. Such testing is essential to developing indoor SUAS that can fly in disaster conditions. We first present an analysis of indoor disaster environments that impact the design of SUASs and their navigational algorithms. We then develop the algorithms required for simultaneous localization and mapping. With the help of our analysis, we design indoor cluttered environments in computer simulation that can best represent an indoor space affected by a disaster. Finally, we test our algorithms in this virtual environment to understand the impact of the characteristics of indoor disaster environment on our system.

3.1 Environmental Characteristics

The environmental characteristics influencing the navigability of a region can be divided into three groups: the scale and degree of deconstruction, which captures the state of the structure; the severity of obstacles and tortuosity, which captures the impact of the deconstructed structure and damaged furnishings; and other characteristics that affect sensing. Table 2.1 provides an overview of the 12 surveyed SUASs and the environmental characteristics for which they were evaluated.

*Part of this section is reprinted with permission from S. Agarwal, R.R. Murphy, and J.A. Adams. Characteristics of indoor disaster environments for small UASs. In *Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on*, pages 1–6, Oct 2014. Copyright[2014] by IEEE.

No.	Author	Testbed	Severity: Obstacle Location				Tortuosity
			Ground to below nominal	Ground up to nominal	Ceiling into nominal	Ceiling to above nominal	
1	Masanori et al., 2013	Physical - General Indoor (Staged)		✓			0.5
2	Li et al., 2013	Physical - General Indoor (Staged)		✓			–
3	MacAllister et al., 2013	Physical - General Indoor (Staged)	✓	✓	✓		0.18
		Computer	✓	✓	✓	✓	0.4
4	Jongho and Youdan, 2013	Computer		✓			0.14
				✓			0.2
5	Al Newaz et al., 2013	Computer	✓	✓			–
			✓	✓			–
			✓	✓			–
6	Fossel et al., 2013	Physical - General Indoor (Natural)	✓	✓			–
		Computer	✓	✓			–
				✓			–
			✓	✓			–
7	Toratani et al., 2013	Physical - General Indoor (Staged)		✓			0.3
8	Grzonka et al., 2013	Physical - General Indoor (Natural)	✓	✓			0.1
			✓	✓			–
9	Michael et al., 2012	Actual Disaster Environment	✓	✓		✓	0.6
10	Stowers et al., 2010	Physical - General Indoor (Staged)	✓	✓			–
		Computer	✓	✓			0.57
11	Suzuki et al., 2010	Physical - General Indoor (Staged)		✓			–
12	Ahrens et al., 2009	Physical - General Indoor (Staged)		✓			0.5

Table 3.1: Summary of severity of obstacles and tortuosity. Copyright [2014] IEEE [1].

3.1.1 Scale and Degree of Deconstruction

The scale of the average size of the twelve SUASs, with respect to the interior of Prop 133 is in the restricted maneuverability range. If the floor plan in 4.2a is used to compute the E_{cd}, the scale represents the habitable range, as $E_{cd} > 2A_{cd}$, where the room is 6m and if the SUAS is 0.6m, thus $6 > 2(0.6)$. However, Fig. 3.2a and Fig. 3.2b show that the damage to fixtures and furnishings reduce the actual free space to $E_{cd}approximately 1.5A_{cd}$, which falls into the restricted maneuverability range of $E_{cd} < 2A_{cd}$.

As seen in the Fig. 3.1, only two of the twelve systems were deployed in a restricted maneuverability ($2A_{cd} > E_{cd} > 1.5A_{cd}$) environment, comparable to Prop 133. The remaining systems were evaluated or deployed in environments within the habitable scale.

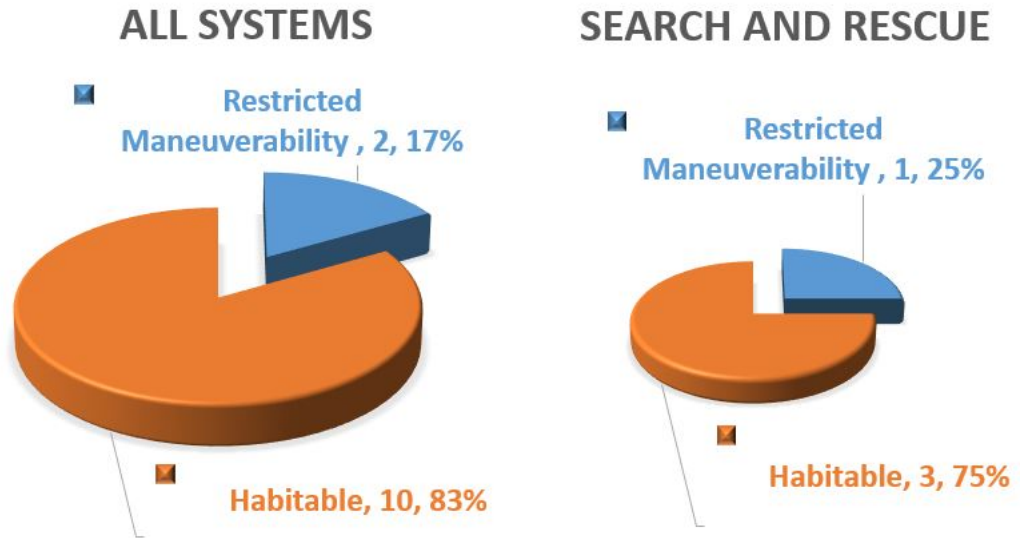


Figure 3.1: SUASs tested in habitable and restricted maneuverability regimes.

(a)

(b)

(c)

(d)

Figure 3.2: Interiors of Prop 133 at Disaster City®. a) Furniture up to heights of 1m to 2.5m scattered around the floor, b) Wires, open ventilators, metal frames hanging from the ceiling at 2m to 3m, c) Collapsed ceiling and wall, and d) Accumulated debris due to breaking of loose material. Copyright [2014] IEEE [1].

Prop 133 also illustrates different degrees of deconstruction to the structural elements. Rooms 1 and 2 on both floors, as seen in Fig. 3.2b and Fig. 3.2d have relatively minor deconstruction, given that the walls, ceiling, and floor are still orthogonal, though they exhibit holes or damage. Fig. 3.2c shows major deconstruction, where a ceiling has collapsed and the supporting pillars are clearly damaged and no longer uniform.

As seen in Fig. 3.3, only one of the twelve SUASs was evaluated in a deconstructed environment. Michael et al. deployed in a damaged building [41], but with only with a minor degree of deconstruction compared to Prop 133. The other three systems proposed for search and rescue missions [33, 3, 61] flew in regions with no visible deconstruction. The remaining eight general indoor SUASs operated in regions with no damage.

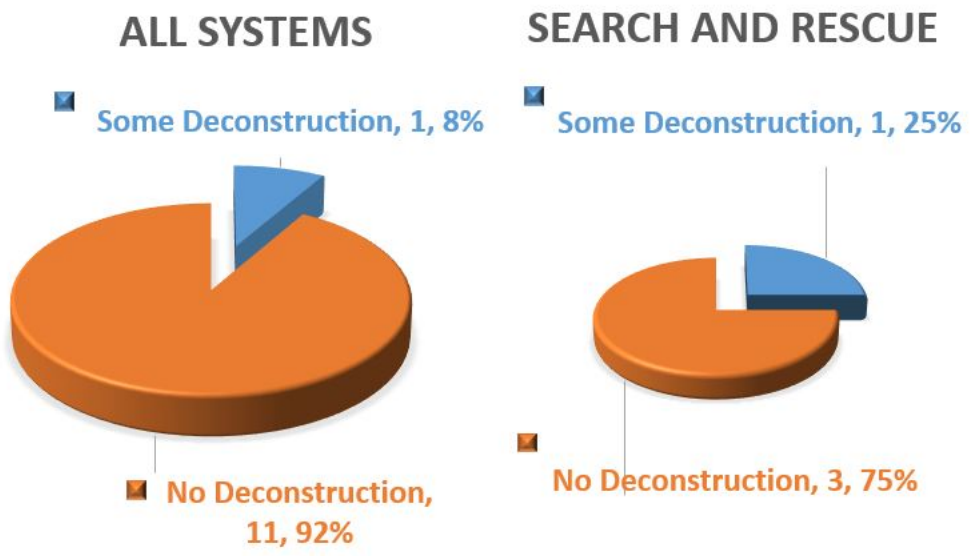


Figure 3.3: Degree of deconstruction for tested SUASs.

3.1.2 Severity of Obstacles and Tortuosity

All four categories of obstacle severity/locations exist in the environment at Prop 133. Altitude does not necessarily reduce the obstacles. If a SUAS were to fly in Prop 133 at an altitude of 1.17m (the average for flying in offices and hallways), it will encounter the same categories of obstacles if it flew at 2.08m (the average for in open spaces).

As seen in Fig. 3.4, only one surveyed system, MacAllister et al. [37], was evaluated in a testbed encompassing all four categories of obstacle severity/location as found at Prop 133, but only in computer simulation. The actual disaster deployment, Michael et al. [41], encountered three types of obstacles, but not those hanging from the ceiling into nominal flying zone. Al Redwan Newaz et al. [3] simulated two categories of obstacles, while the other two systems [33, 61] only tested with obstacles on the ground up to the nominal flying zone. This observation suggests that the obstacle placement in testbeds is not a good predictor of whether a SUAS will be able to fly indoors during a disaster.

As seen in Fig. 3.5, all three types of spaces in Table 3.1 have a tortuosity much lower than the tortuosity of Prop 133. The maximum tortuosity in computer simulation (0.5), physical - general indoor staged testbeds (0.31), physical - general indoor natural testbeds (0.1), and actual disasters (0.6), suggests that the evaluation testbeds are not sufficiently representative of actual disasters. A SUAS that performs well in these testbeds may not have the agility to make a higher frequency of turns and altitude changes.

A paper on this analysis of indoor disaster environments that impact the design of small unmanned aerial systems (SUASs) was recently published in IEEE SSRR 2014 [1].

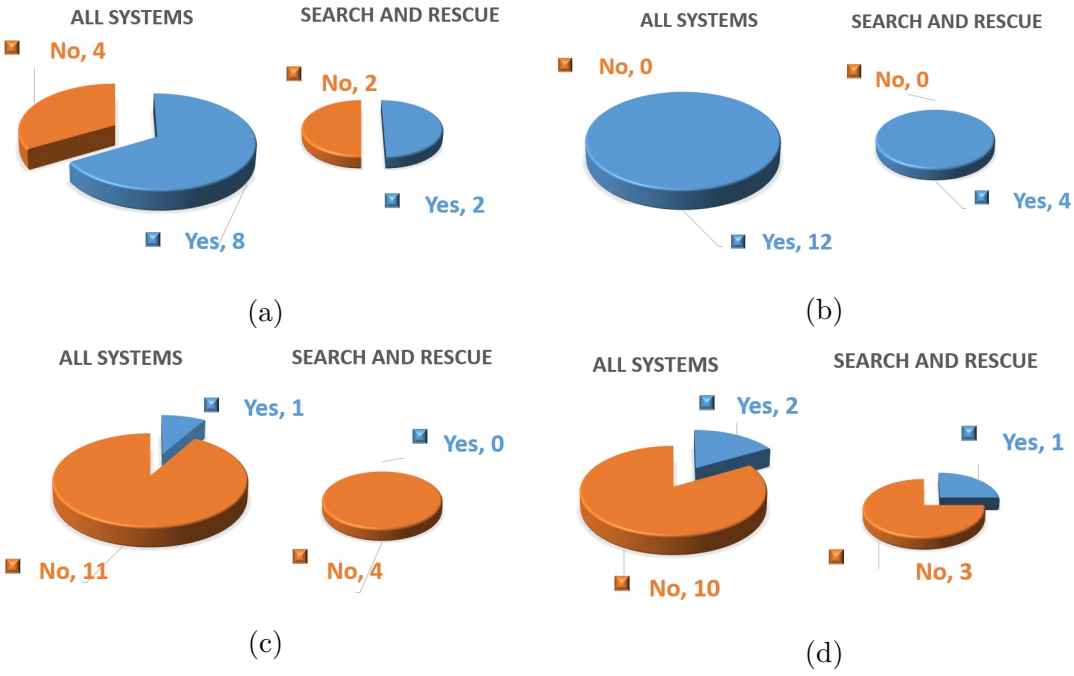


Figure 3.4: Severity of obstacles for twelve SUASs. a) Obstacles on the ground, below the nominal flying zone, b) Obstacles on the ground, up to the nominal flying zone, c) Obstacles hanging from the ceiling, in the nominal flying zone and d) Obstacles hanging from the ceiling, above the nominal flying zone.

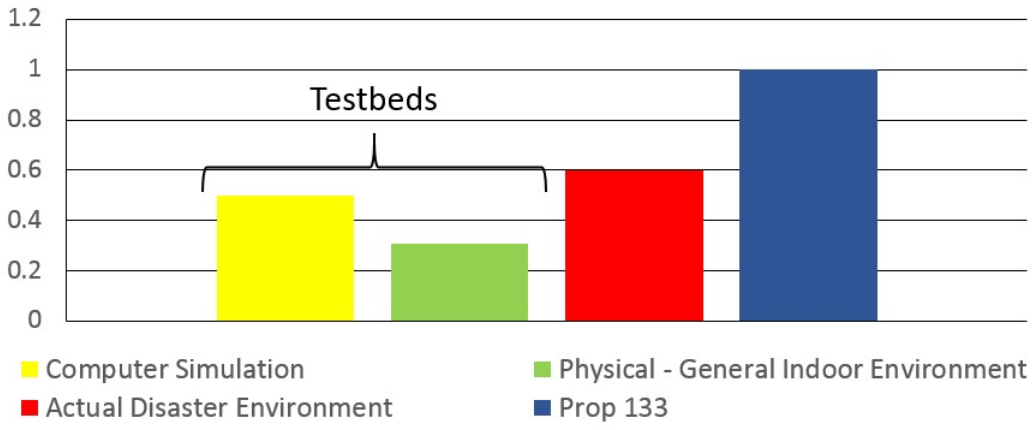


Figure 3.5: Comparison of tortuosity.

3.2 Algorithms and System Design

Fig. 3.6 below outlines our system design, which is based on the work done by Shen et al. [52, 53, 54, 41] and Grzonka et al. [20, 5].

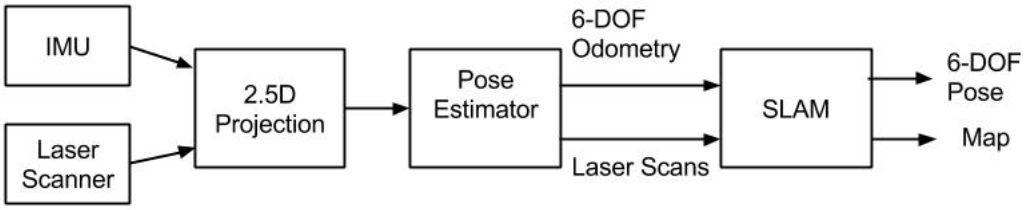


Figure 3.6: System design.

All the required algorithms were developed in C++ with the Robot Operating System (ROS) [49] framework used as a middleware. ROS is an open source, meta-operating system that can be modified as per requirement. It includes drivers, libraries and visualizers. The user can choose between modules in order to set up the required system. ROS provides visualization modes like RVIZ, a 3D visualization environment where the robot and map can be displayed in real-time. The user can control the robot manually or via path planners. ROS includes a runtime graph communication system, a peer-to-peer network of processes. ROS also supports Gazebo, a simulator that offers the ability to simulate robots in complex indoor and outdoor environments. The ROS framework seeks to support code reuse in robotics research and development [6].

3.2.1 Notion

For a SUAS, the six degree of freedom in the world frame is represented as (x, y, z, Φ, θ, Ψ), where Φ is the roll angle, θ is the pitch angle, and Ψ is the yaw angle as shown in Fig. 3.7. X_b is the forward direction and Z_b is the direction perpendicular to the plane of the rotors , when they are pointing vertically up. Translating from body frame to world frame requires:

- rotating about the Z_b axis by the yaw angle, Ψ
- rotating about the y axis by the pitch angle, θ
- rotating about the X_b axis by the roll angle, Φ

Figure 3.7: Notion.

The rotation matrix to transform from body frame to the world frame is:

$$\mathrm{R} = \mathrm{R}(Z_b, \Psi)\mathrm{R}\ (\mathrm{y}, \theta)\ \mathrm{R}(X_b, \Phi)$$

where the Rs are elementary rotations with respect to the x, y, and z

3.2.2 2.5D Projection

The first step of the algorithm is to compute the 2.5D projection, which uses the laser scans and IMU readings assuming 2.5D environment models formed by vertical walls and horizontal planes, all assumed to be piecewise constant [52]. As the 3D orientation of the laser scanner is know, we can project the scans onto a 2D plane, and then perform matching on sequences of projected scans instead of raw scans [24] as shown in figure 3.8.

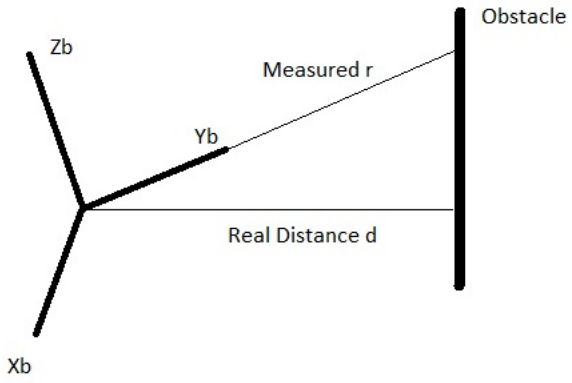


Figure 3.8: Projection of laser scans.

Suppose that the laser coordinates coincides with the body co-ordinates of the vehicle. IMU measures the attitude between body coordinates and world coordinates. The laser scanner measures a set of distances r and direction angles along the $x - y$ plane for each point. Each of these distances and angles α can be represented in terms of sines and cosines. The transformation matrix can be represented as:

$$T = \begin{pmatrix} cos\theta & 0 \\ sin\theta sin\phi & cos\phi \end{pmatrix}$$

Thus, we can compute the real position of laser endpoint from the following equation:

$$z_i = T \begin{pmatrix} r_i cos\alpha_i \\ r_i sin\alpha_i \end{pmatrix}$$

The scans that hit the floor or ceiling are eliminated. This technique simplifies the challenges of full 3D scan matching using only 2D laser scanners. But, the 2.5D environment assumption could be easily violated in highly cluttered environments such as one found at disasters.

Algorithm:

$[x_s, y_s, z_s]^T$: The laser scan endpoints in the body frame.

$[x_g, y_g, z_g]^T$: The 2.5 D laser scan projection to a horizontal plane.

z_{dflU} : Deflected laser scan pointing up

z_{dflD} : Deflected laser scan pointing down

If $((z_g == z_{dflU})(z_g == z_{dflD}))$

Eliminate the scan from consideration

Else

Calculate the projection for the scan to 2.5D space such that:

$[x_g, y_g, z_g]^T = RR[x_s, y_s, z_s]^T$

Return $[x_g, y_g, z_g]^T$

3.2.3 Pose Estimator

The pose estimator has two steps. Step one estimates (x, y, Ψ) using the ICP algorithm [50] and the second step determines the altitude using a modified version of Frieburgs algorithm [20].

3.2.3.1 Iterative Closest Point

ICP finds the transformation between two sets of data points - A reference point cloud and the new data point cloud, as shown in Fig. 3.9. It provides an estimate of (x, y, Ψ). The results of the ICP are combined with the IMU data in order to correct the error in the algorithm.

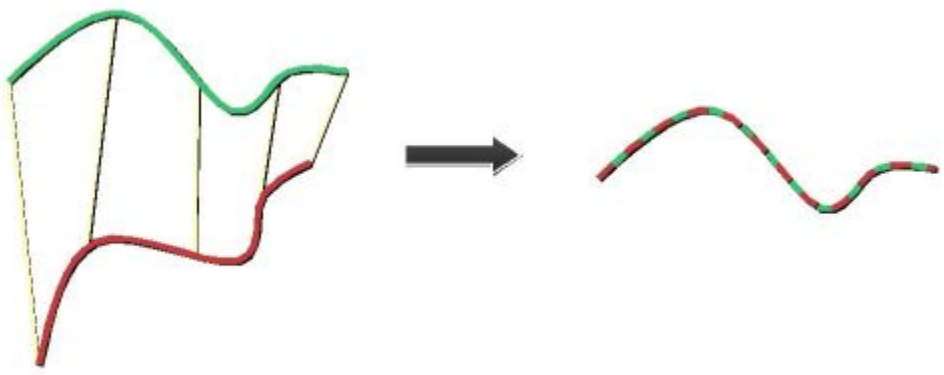


Figure 3.9: Iterative closest point.

The basic ICP algorithm proceeds in the following steps: Input: A reference point cloud and the new data point cloud. An estimate of (x, y, Ψ) is obtained using the following steps:

1. Preprocessing to clean the data.

2. Matching the associated points from the reference to the data using the neighbor search. It can use features to identify associated points.

3. Weighting changes the importance of some pairs.

4. Rejecting some of the pairs

5. Computing the error for each pair and where they should be located.

6. Finding the best transformation (minimization)

7. Loop back to step 2, unless there is convergence.

The results of the ICP steps prior to step 5 are combined with the IMU data in order to correct the error in the ICP algorithm Step 5. The correction is done by aligning the incoming scans with the map.

3.2.3.2 Altitude Estimation

The altitude sensing relies on the upward and downward deflected laser scans and the pressure sensor. The downward facing laser scan is used to measure variance in altitude and then uses this value to estimate the variance for the measurement update in the Kalman Filter. The upward facing scans are used when the downward facing scans contain too much variance. The pressure sensor is used when there is too much variance from both the upward and downward deflected laser scans.

3.2.3.3 Roll and Pitch Estimation

The roll and pitch angles (Φ, θ) are estimated solely from the IMU data.

3.2.4 SLAM

The goal of a SLAM algorithm is to estimate both the pose of the SUAS and generate a map of the environment from sensor measurements. As discussed in related work, we studied a number of approaches to solve the SLAM problem. Both feature based and particle filter-based methods work well in practice. However, these approaches are need high computational power. Because of the limited onboard processing capacity in most SUAS, the use of complex SLAM algorithms is not feasible. Furthermore, it is much more difficult to process 3D laser data than laser data acquired from two-dimensions [24].Hence, we use Shen et al.'s approach to address the problems of mapping and drift compensation via a simplified occupancy grid-based incremental SLAM.

Since roll and pitch angles are measured by the IMU with tolerable error, we directly use this information. This allows us to reduce the localization problem from 6 DOF to 4 DOF, consisting of the 3D position (x, y, z) and the yaw angle Ψ. Based on known current attitude, the endpoints of laser scan are projected into the global coordinate frame. Using these projection, estimates in (x, y, z, Ψ) are calculated.

To map and represent unknown environments, an occupancy grid map is used. The SUAS platform will exhibit 6 DOF motion. To create a map in 3D using a 2D laser scanner, the 2.5D projection is utilized and the scans are transformed into a local frame using the current estimated attitude. Each scan is then converted into a point cloud of endpoints that are used to estimate the occupancy of each cell in the grid. The point clouds are projected in space to create a 3D visualization of the map using RVIZ.

SLAM algorithm is prone to errors that increase over time and loop closure is often used to correct these errors. Loop closure is a huge area of study in itself and out of the scope of this work. Hence, our system does not embed loop closure functionality. The incremental motion of the SUAS is provided by scan matching and the IMU. The algorithm corrects the error in x, y and yaw by aligning the current laser scans against the previously obtained map. Thus the SLAM algorithm corrects the accumulated errors in the laser based pose estimator. Comparing a new scan to the global map provides more consistent pose estimation than comparing each scan only to the scan from previous instant. If the pose estimated by scan matching is fairly accurate, the posterior over the global map can be computed. However, there are no guarantees of obtaining the accurate pose at every step [24]. Loop closure is essential in correcting these errors with reasonable certainty.

4. EXPERIMENTS AND RESULTS

To evaluate our system and analyze the impact of the characteristics of indoor disaster environment on the SLAM algorithm, we tested our system in computer simulation. We used Gazebo [27], to develop rigorous computer simulated environments that are representative of realistic disaster conditions. Gazebo is a simulator capable of simulating articulated robots in three dimensional environments. It generates realistic sensor feedback and supports the required sensors.

The goals of our experiments are to test and quantify:

- The increase in localization error with increase in deconstruction in the environment
- The empirical value of tortuosity
- The decrease in localization error as the Agent to Environment ratio decreases

4.1 Environment Design

To run the experiment, three groups of environments with increasing deconstruction were designed - Group 1, Group 2 and Group 3. The amount of deconstruction increases as we move from Group 1 to Group 3. An environment in Group 1 (Fig. 4.1a) represents a normal undamaged indoor space. A similar environment was created in computer simulation by McAllister et al. Group 2 (Fig. 4.1b) will include an environment that has been affected by a mild earthquake. Such an event may rearrange office furniture, knock over bookcases, and cause ceiling fixtures to hang loose, while leaving the structural elements, such as walls, ceilings, floors, and pillars

intact. An environment in Group 3 (Fig. 4.1c) represents a space affected by a severe earthquake, with collapsed ceilings and deposited debris, as seen in our observations at Prop 133 and in Michael et al.'s work [41].

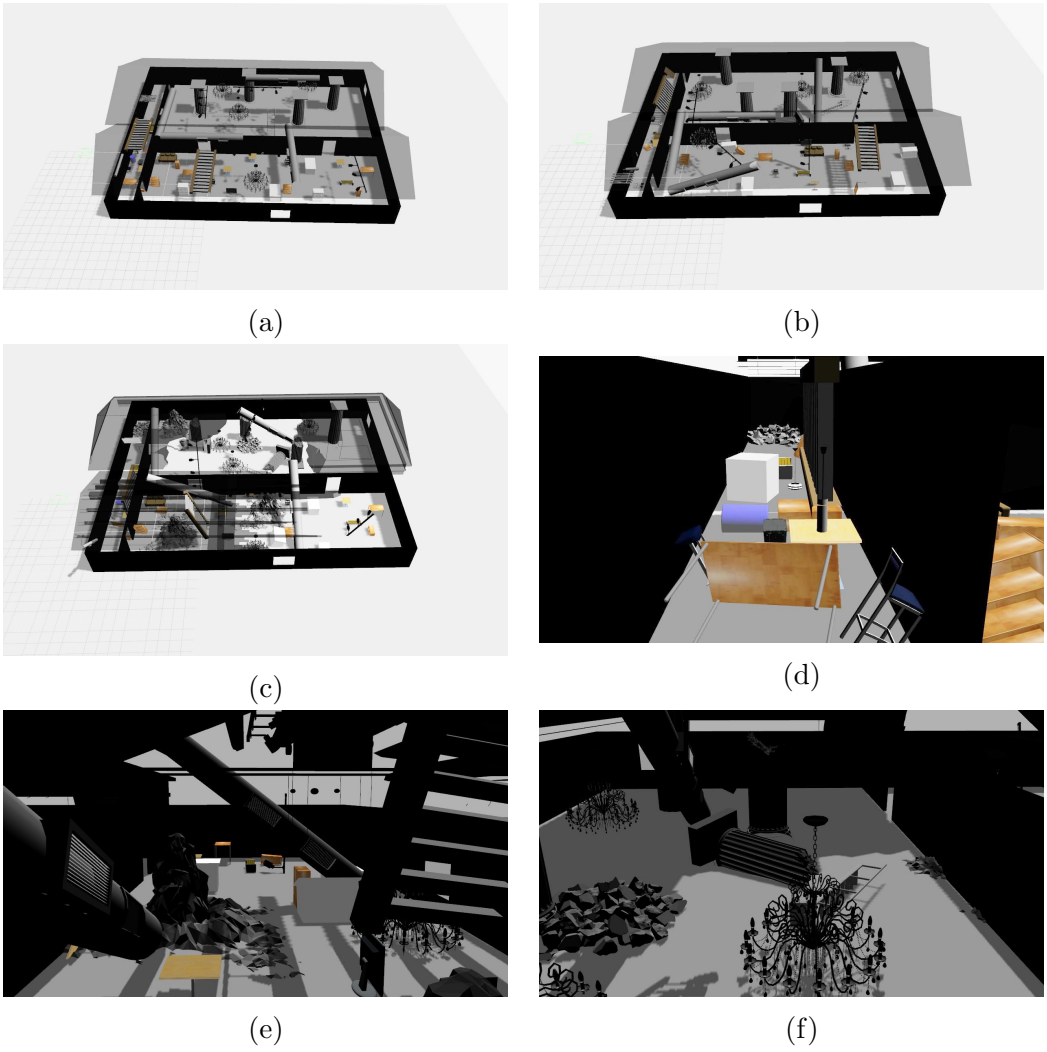


Figure 4.1: Computer simulated environments. a) Environment in Group 1, b) Environment in Group 2, c) Environment in Group 3, d) Hallway, e) Office space and f) Open space.

For each group, we simulate 10 different maps with a common floor plan as seen in Fig. 4.2a. Each map has a dimension of 30m x 24m. Every map comprises of a hallway (3m x 24m), an office space (27m x 14m) and an open space (27m x 14m) with obstacles randomly placed in each space. Fig. 4.1d, Fig. 4.1e and Fig. 4.1f illustrate a hallway, an office space and an open space with randomly placed obstacles.

For each map, we ran 3 trials with varying paths A,B and C. For each path we have goal points that helps map the entire area as shown in figure 4.2. The paths A.B and C will be random for each map according to the generated obstacles. Thus, we conducted 90 trials of experimentation in total. The maximum velocity for all trials was 1 m/s. Each trial is run only once to prevent the pilot from learning the environment.

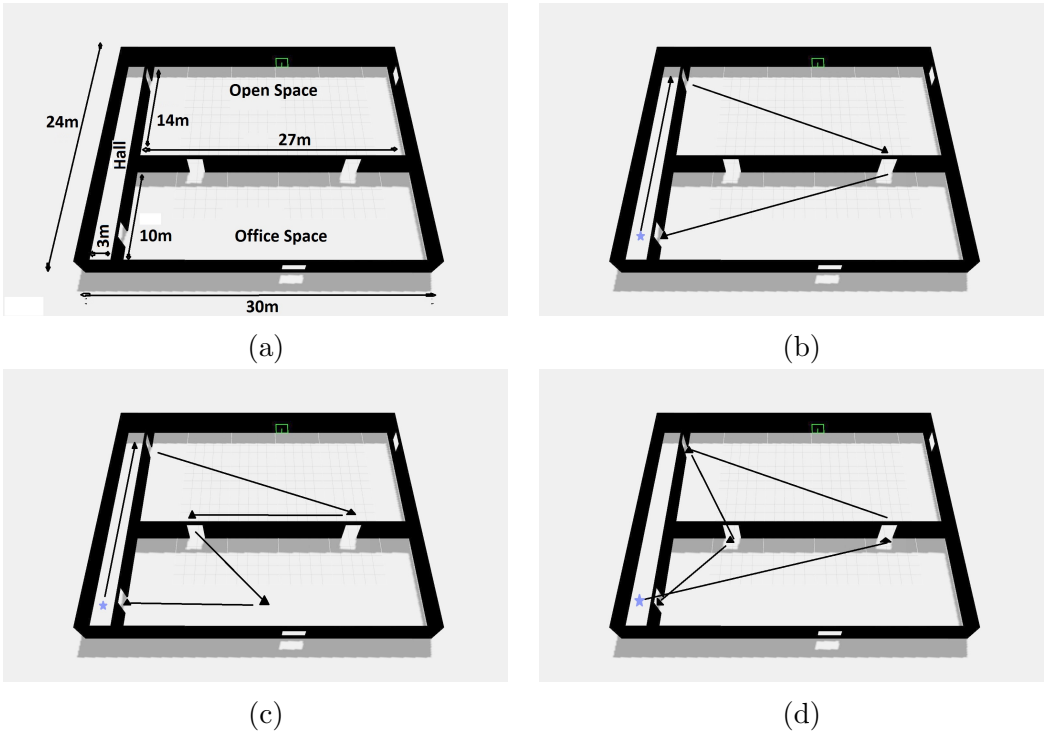


Figure 4.2: Floor plan and goal points. a) Floor plan, b) Goal points for path A, c) Goal points for path B and d) Goal points for path C.

4.2 Simulation of SUAS

To simulate an SUAS we used a custom UAV of 0.75 m diameter (average of minimum and maximum diameters of SUAS in the 12 studies) from the Hector Quadrotor stack [40]. The following sensors were added to provide input to the SLAM system. Gaussian noise was added to each sensor to try and replicate real world conditions:

- IMU
- Barometer
- Hokuyo UTM-30LX Laser Scanners
- Forward facing Camera

To reflect some of the laser scans upwards and downwards, we add two more laser scanners in each respective direction, with reduced field of views (See Fig. 4.3).

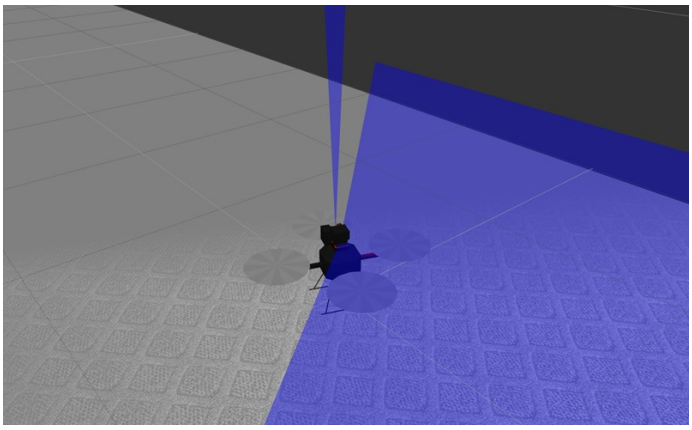
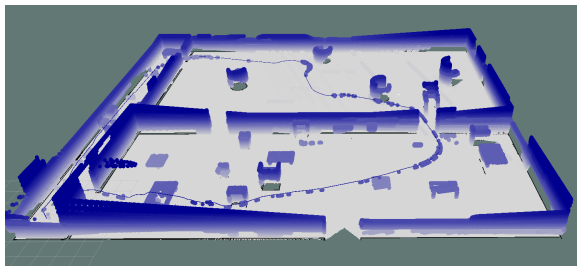


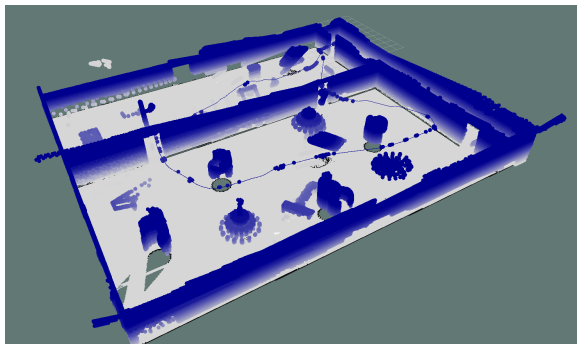
Figure 4.3: Simulated SUAS.

4.3 Results

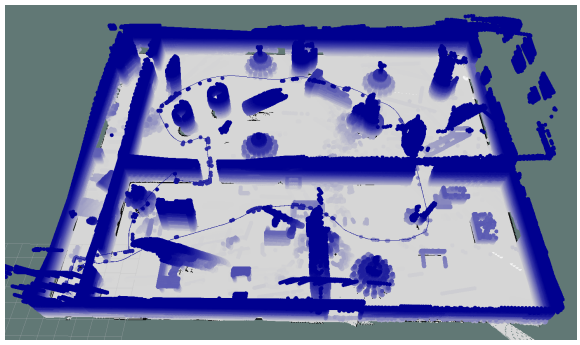
The SLAM algorithm ran successfully and we were able to produce maps and simultaneously localize the robot within them. Fig. 4.4 shows the maps generated for environments with increasing deconstruction for one particular map.



(a)



(b)



(c)

Figure 4.4: Map generated for environments in different groups. a) Group 1, b) Group 2 and c) Group 3.

For each trial, the errors in position, x, y, z, Ψ, average altitude and tortuosity are calculated. The error in localization can be calculated in a number of ways - Average Error, Mean Absolute Error (MAE) or Root Mean Square Error (RMSE). The average error is affected by the sign. In RMSE, since the values of error are squared and later averaged. Hence, the RMSE gives a higher weight to larger errors in comparison to MAE. This means the RMSE could be most useful when large errors are particularly undesirable. Also, RMSE has been popularly used by Shen et al.[52], Thrun et al. [65] and other researchers. Hence, the localization error is calculated in terms of RMSE values. The RMSE in position is calculated by using the euclidean distance between the ground truth and the actual pose of the SUAS in (x, y, z) at each instant of time as shown in Fig. 4.5.

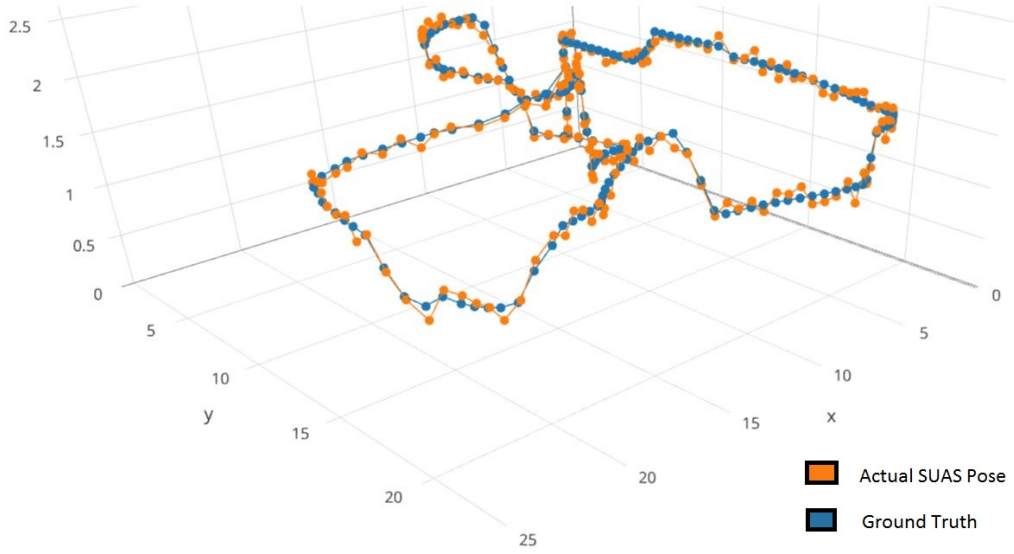


Figure 4.5: Actual SUAS position (x, y, z) vs ground truth.

Before calculating the results, the videos and images taken during the 90 trials were analyzed for any irregularities. It was found that 15 out of 90 trials were unsuccessful because of Operator's Error. The tele-operated SUAS either collided with an obstacle or swerved i.e. took a sudden turn. As seen in Fig. 4.6, to confirm this finding, the RMSE in localization for all the trials was plotted. The trials with operator error had a huge deviation of the RMSE in localization. These 15 trials are not included further in the calculations.

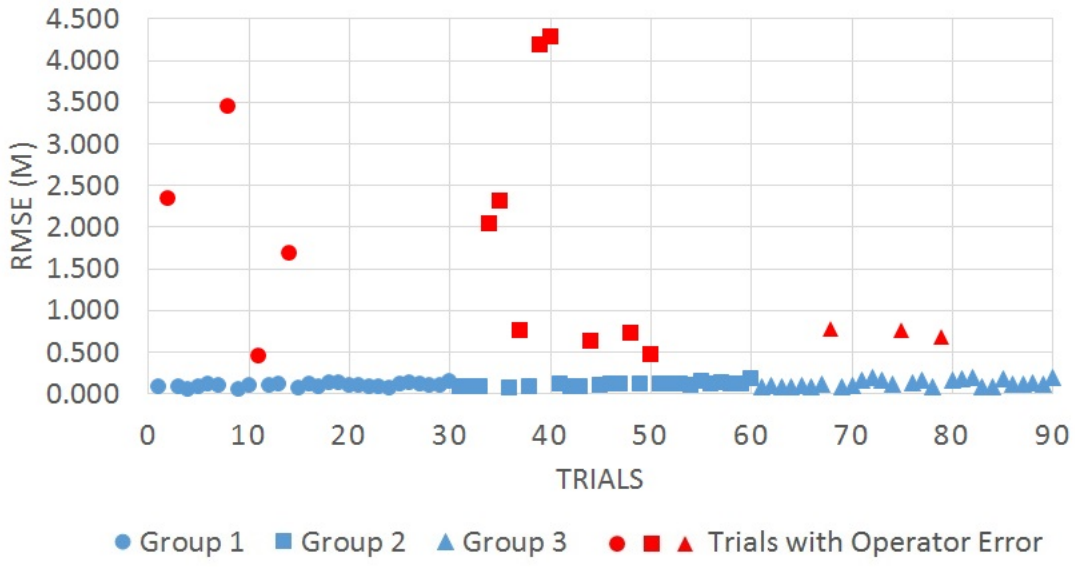


Figure 4.6: Operator error.

The results of the 90 trials along with the operator error are summarized in Table 4.1 and Table 4.2.

No.	Deconstruction	Path	RMSE					Average Altitude(m)	Tortuosity	Operator Error
			Pose	x (m)	y (m)	z (m)	Ψ (°)			
1	Group 1	A	0.085	0.050	0.057	0.039	0.137	1.215	0.15	-
		B	2.345	0.107	2.342	0.051	0.115	1.099	0.13	Swerve
		C	0.076	0.055	0.052	0.010	0.077	0.955	0.13	-
	Group 2	A	0.086	0.022	0.055	0.062	0.096	1.366	0.18	-
		B	0.086	0.052	0.047	0.051	0.223	1.550	0.17	-
		C	0.082	0.041	0.034	0.062	0.098	1.608	0.17	-
	Group 3	A	0.088	0.036	0.055	0.057	0.178	1.314	0.25	-
		B	0.096	0.052	0.052	0.062	0.146	1.530	0.25	-
		C	0.085	0.046	0.043	0.057	0.211	1.354	0.23	-
2	Group 1	A	0.054	0.016	0.019	0.048	0.057	1.008	0.15	-
		B	0.086	0.053	0.048	0.047	0.067	1.332	0.12	-
		C	0.117	0.102	0.027	0.049	0.066	1.233	0.13	-
	Group 2	A	2.032	0.051	2.031	0.048	0.221	1.289	0.18	Collision
		B	2.304	0.029	2.304	0.051	0.194	1.269	0.18	Swerve
		C	0.070	0.027	0.041	0.049	0.161	1.264	0.17	-
	Group 3	A	0.085	0.028	0.041	0.069	0.163	1.279	0.25	-
		B	0.105	0.061	0.050	0.069	0.107	1.332	0.27	-
		C	0.081	0.016	0.032	0.072	0.111	1.445	0.25	-
3	Group 1	A	0.096	0.058	0.062	0.045	0.200	1.410	0.15	-
		B	3.455	0.019	3.455	0.054	0.169	1.244	0.13	Swerve
		C	0.057	0.028	0.022	0.045	0.120	1.132	0.12	-
	Group 2	A	0.749	0.741	0.091	0.063	0.160	1.215	0.18	Swerve
		B	0.089	0.057	0.039	0.056	0.078	1.102	0.20	-
		C	4.189	2.930	2.994	0.063	4.141	1.522	0.17	Swerve
	Group 3	A	0.118	0.062	0.067	0.075	0.161	1.368	0.27	-
		B	0.776	0.279	0.721	0.072	0.825	1.414	0.27	Swerve
		C	0.090	0.042	0.025	0.075	0.201	1.373	0.25	-
4	Group 1	A	0.100	0.049	0.072	0.050	0.134	1.242	0.13	-
		B	0.447	0.310	0.318	0.051	0.735	1.401	0.13	Swerve
		C	0.102	0.044	0.078	0.049	0.220	1.018	0.12	-
	Group 2	A	4.282	0.089	4.280	0.061	0.121	1.292	0.17	Swerve
		B	0.116	0.087	0.047	0.061	0.086	1.546	0.20	-
		C	0.084	0.034	0.045	0.062	0.148	1.564	0.20	-
	Group 3	A	0.103	0.049	0.056	0.071	0.078	1.566	0.25	-
		B	0.170	0.131	0.092	0.058	0.986	1.436	0.27	-
		C	0.200	0.105	0.159	0.060	0.265	1.466	0.25	-
5	Group 1	A	0.122	0.095	0.058	0.048	0.138	1.090	0.13	-
		B	1.693	0.470	1.625	0.052	1.733	1.015	0.12	Swerve
		C	0.067	0.023	0.035	0.052	0.140	1.183	0.13	-
	Group 2	A	0.089	0.052	0.055	0.048	0.143	1.354	0.20	-
		B	0.630	0.531	0.333	0.055	0.710	1.563	0.18	Swerve
		C	0.091	0.050	0.054	0.054	0.521	1.507	0.18	-
	Group 3	A	0.157	0.074	0.128	0.053	0.287	1.537	0.30	-
		B	0.107	0.038	0.053	0.085	0.150	1.443	0.27	-
		C	0.758	0.652	0.372	0.102	0.704	1.659	0.25	Collision

Table 4.1: Summary of results - environments 1 to 5.

No.	Deconstruction	Path	RMSE					Average Altitude(m)	Tortuosity	Operator Error
			Pose	x (m)	y (m)	z (m)	Ψ (°)			
6	Group 1	A	0.113	0.091	0.056	0.035	0.109	1.359	0.13	-
		B	0.082	0.039	0.051	0.051	0.121	1.077	0.13	-
		C	0.125	0.050	0.103	0.050	0.891	1.014	0.12	-
	Group 2	A	0.114	0.031	0.040	0.102	0.113	1.210	0.20	-
		B	0.108	0.043	0.086	0.050	0.137	1.233	0.20	-
		C	0.717	0.306	0.646	0.048	1.129	1.175	0.18	Swerve
	Group 3	A	0.124	0.034	0.063	0.101	0.141	1.221	0.30	-
		B	0.161	0.115	0.101	0.051	0.365	1.477	0.30	-
		C	0.082	0.019	0.026	0.075	0.110	1.103	0.27	-
7	Group 1	A	0.126	0.099	0.056	0.054	0.085	1.311	0.15	-
		B	0.104	0.016	0.021	0.101	0.074	1.282	0.13	-
		C	0.101	0.080	0.034	0.052	0.540	1.414	0.12	-
	Group 2	A	0.115	0.069	0.059	0.071	0.497	1.661	0.20	-
		B	0.469	0.127	0.448	0.054	0.411	1.609	0.22	Collision
		C	0.109	0.046	0.079	0.060	0.162	1.600	0.18	-
	Group 3	A	0.682	0.602	0.317	0.044	0.626	1.775	0.27	Swerve
		B	0.160	0.069	0.127	0.068	0.293	1.595	0.30	-
		C	0.181	0.059	0.119	0.123	0.410	1.449	0.25	-
8	Group 1	A	0.082	0.054	0.053	0.032	0.071	1.169	0.15	-
		B	0.086	0.051	0.041	0.055	0.093	1.264	0.13	-
		C	0.074	0.020	0.015	0.070	0.068	1.239	0.12	-
	Group 2	A	0.109	0.046	0.079	0.060	0.162	1.600	0.22	-
		B	0.123	0.083	0.026	0.087	0.108	1.388	0.20	-
		C	0.096	0.020	0.052	0.078	0.108	1.274	0.20	-
	Group 3	A	0.197	0.052	0.173	0.079	0.093	1.369	0.27	-
		B	0.084	0.035	0.035	0.069	0.528	1.417	0.30	-
		C	0.085	0.047	0.017	0.069	0.490	1.386	0.25	-
9	Group 1	A	0.114	0.073	0.070	0.054	0.150	1.237	0.15	-
		B	0.138	0.089	0.041	0.097	0.141	1.392	0.12	-
		C	0.111	0.041	0.026	0.099	0.089	1.159	0.13	-
	Group 2	A	0.150	0.131	0.038	0.061	0.188	1.199	0.18	-
		B	0.116	0.031	0.035	0.106	0.182	1.510	0.20	-
		C	0.124	0.061	0.076	0.077	0.841	1.543	0.18	-
	Group 3	A	0.180	0.054	0.154	0.075	0.235	1.474	0.30	-
		B	0.114	0.089	0.044	0.056	0.091	1.414	0.30	-
		C	0.110	0.035	0.034	0.098	0.081	1.338	0.27	-
10	Group 1	A	0.093	0.048	0.053	0.059	0.125	1.295	0.15	-
		B	0.095	0.059	0.044	0.059	0.192	1.408	0.13	-
		C	0.141	0.041	0.024	0.133	0.089	1.254	0.12	-
	Group 2	A	0.114	0.079	0.066	0.048	0.299	1.322	0.22	-
		B	0.114	0.073	0.066	0.058	0.209	1.265	0.18	-
		C	0.172	0.122	0.104	0.063	0.691	1.243	0.23	-
	Group 3	A	0.125	0.024	0.057	0.108	0.084	1.340	0.27	-
		B	0.112	0.047	0.042	0.093	0.079	1.181	0.30	-
		C	0.187	0.131	0.035	0.128	0.329	1.507	0.30	-

Table 4.2: Summary of results - environments 6 to 10.

4.3.1 RMSE in Localization vs Deconstruction

We compare the average RMSE in position (x, y, z) for all trials in Group 1, Group 2 and Group 3 respectively. As shown in Fig. 4.7 and Table 4.3, we see that the value of average RMSE increases with increase in the amount of deconstruction in the environment from Group 1 to Group 3. The value of average RMSE changes by 9.46% when the deconstruction increases from Group 1 to Group 2 and the value increases further by 16.9% as the environment changes from Group 2 to Group 3. But this increase in value does not mean that the error always increases with deconstruction. To check this result, we run a statistical test.

Among all statistical tests, z-test and t-test are popularly used but z-test requires a prior value of deviation [8]. As the deviation itself needs to be estimated from the data, we use t-test. From Table 4.4, we see that when we compare Group 1 and 2 and, Group 2 and 3 respectively, the t-stat value is not lesser than negative of t-critical. Therefore, we do not reject the null hypothesis. The observed difference between the value of average RMSE in localization for each group is not convincing enough to claim if the error indeed increases from Group 1 to Group 2 and, Group 2 to Group 3 respectively. But the t-test result is considerable when comparing the value of average RMSE of Group 1 and Group 3. The t-stat value is lesser than negative of the t-critical value. Hence the observed difference is 95% convincing that RMSE is localization increases as the environment changes directly from Group 1 to Group 3. Though the individual change in error from Group 1 to Group 2 and, Group 2 to Group 3 is not significant, the total change in error from Group 1 to Group 3 is considerable. Hence, though the result suggests that the value of error increase between different groups, we cannot claim that the error will always increase with deconstruction.

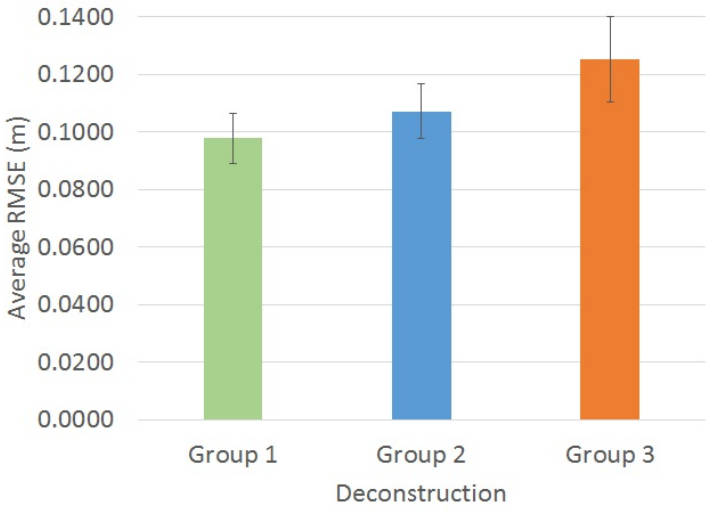


Figure 4.7: Average RMSE in localization vs deconstruction - Comparison for all trials

		Group 1	Group 2	Group 3
All Trials	Average RMSE	0.0979	0.1072	0.1254
	Standard Deviation	0.0226	0.0228	0.0394
	95% Confidence	0.0087	0.0095	0.0149

Table 4.3: Summary of results - average RMSE in localization vs deconstruction. All values are in meters.

	Group 1 and 2	Group 2 and 3	Group 1 and 3
t-stat	-1.382	-1.743	-3.042
t-critical	2.014	2.015	2.015

Table 4.4: Summary of t-test - average RMSE in localization vs deconstruction.

4.3.2 Tortuosity

The number of turns per unit distance increases with increase in deconstruction. The SUAS has to change direction in x, y and z planes to navigate through an obstacle filled environment. Hence, more the deconstruction, higher is the average tortuosity. We quantify the values of average tortuosity empirically for all trials in each group as shown in Fig. 4.8. The calculations are summarized in Table 4.5. The tortuosity of our computer simulated environments is at best 0.3 which is 30 % of that estimated at Prop 133 (1.0) as shown in our analysis.

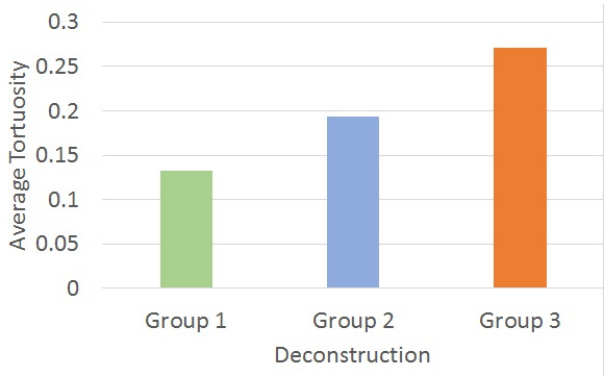


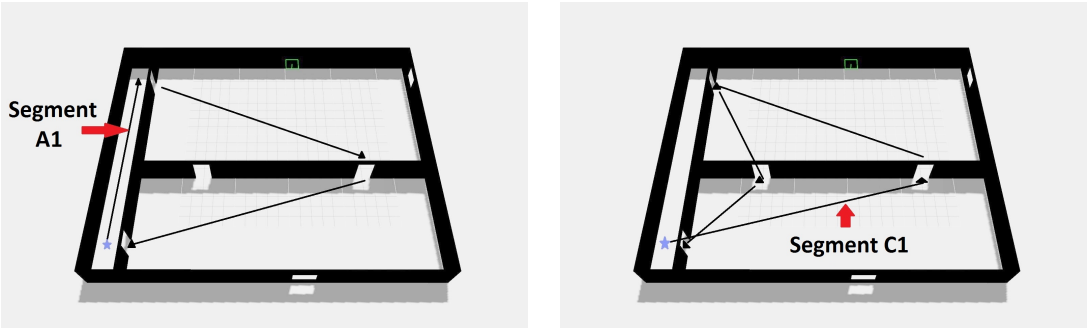
Figure 4.8: Average empirical tortuosity for all trials in each group.

Tortuosity	Group 1	Group 2	Group 3
Average	0.133	0.194	0.271
Minimum	0.117	0.167	0.233
Maximum	0.15	0.233	0.3

Table 4.5: Summary of results - empirical tortuosity.

4.3.3 RMSE in Localization vs Scale

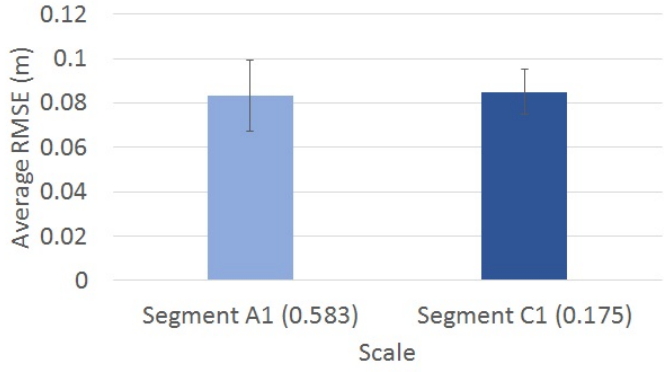
The simulated environments are made up of three types of spaces - a hallway, an office space and an open space. The SUAS used in these experiments has a diameter of 0.75m. The operator flies the SUAS so that the nearest obstacle is approximately 0.5m away. Hence the characteristic dimension of the SUAS comes out to be 1.75m. Paths A and C are divided into segments as seen in Fig. 4.9. As the error in SLAM increases incrementally, we consider the segments that are similar in length and connected to the starting point. Segment A1 for path A and segment C1 for path C meet these requirements. The nominal characteristic dimension of environment for Segment A1 and Segment C1 is 3m and 10m respectively. This space is further reduced by random arrangement of obstacles. Hence the nominal scale i.e. Agent to Environment ratio for segment A1 is 0.583 i.e. restricted maneuverability. Similarly, the scale for segment C1 is 0.175 i.e. habitable.



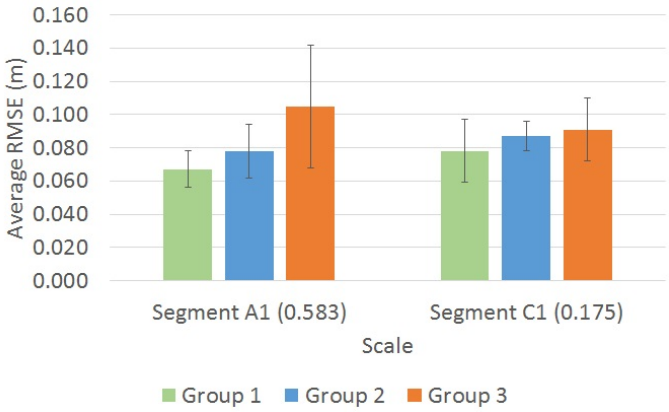
(a) (b)

Figure 4.9: Segments for each path. a) Path A and c) Path C.

As seen in Fig.4.10, the average RMSE for all trials hardly changes with scale. It changes by 2.4 % for all trials. Similarly there is a minor change in the average RMSE as the scale changes for each group. These value and their confidence intervals are too close to make any claims. The table 4.6 summarizes this result.



(a)



(b)

Figure 4.10: Average RMSE in localization vs scale. a) Comparison for all trials and b) Comparison for each group.

		A1 (0.583)	C1 (0.175)
All Trials	Average RMSE	0.083	0.085
	Standard Deviation	0.04	0.026
	95% Confidence	0.016	0.01
Group 1	Average RMSE	0.067	0.078
	Standard Deviation	0.019	0.03
	95% Confidence	0.011	0.019
Group 2	Average RMSE	0.078	0.087
	Standard Deviation	0.021	0.012
	95% Confidence	0.016	0.009
Group 3	Average RMSE	0.105	0.091
	Standard Deviation	0.056	0.028
	95% Confidence	0.037	0.019

Table 4.6: Summary of results - average RMSE in localization vs scale.

4.3.4 Summary

Using 90 trials of experimentation in computer simulation, we test and quantify the error in localization. Using statistical testing, we see that the average value of RMSE in localization increases convincingly by 26.36% as the environment changes from Group 1 to Group 3. But, the change in value of error is not statistically convincing when we compare Group 1 and 2 and, Group 2 and 3 respectively. Hence, though the result suggest that the value of error increases between different groups, it cannot be claimed that the RMSE in localization will always increase with deconstruction. We empirically quantify the values of tortuosity for all trials. The average RMSE in localization does not change with the scale of the environment.

5. CONCLUSION

The interest in using SUAS technology by urban search and rescue teams continues to grow; however, the unique situations in which SUASs will be considered a valuable tool place constraints on the system and algorithm design. This work focuses on identifying the key characteristics of indoor disaster environments and understanding their impact on SUAS SLAM algorithms. This research is critical for purposes of developing and evaluating SUAS technology for and within representative environments that will lead to transferring the technology to disaster response personnel. The formal definitions characterize the environment's scale, space type, obstacle severity and tortuosity as well as the SUAS' nominal flight altitude. These definitions were used to analyze and classify twelve existing SUASs evaluated for deployment in simulation or in actual indoor environments. These results were compared to a representative environment used to train urban search and rescue teams, Prop 133 at Disaster City®.

Using the lessons learned from analysis, computer simulated environments with increasing deconstruction were designed with randomly placed obstacles. A SLAM system was developed and tested in these environments using 90 trails. Using statistical testing, we see a convincing increase of 26.36% in the value of average RMSE in localization as the deconstruction in the environment changes from Group 1 to Group 3. But, the change in value of error is not statistically convincing when we compare Group 1 and 2 and, Group 2 and 3 respectively. Hence, though the result suggest that the value of error increases between different groups, it cannot be claimed that the RMSE in localization will always increase with deconstruction. The tortuosity increases with deconstruction and we empirically calculate this value for all trials.

The average RMSE for all trials does not change as the Agent to Environment ratio changes.

In spite of our best efforts, the computer simulated environments may not be as challenging as an actual disaster. Researchers working in similar areas could build upon our study to create better algorithms and test their systems using our computer simulated disaster environment. From this work, we have gained a better understanding of indoor disaster environments. Furthermore, we have developed the SLAM algorithms that could be used to develop better SUAS for exploration and mapping of such challenging environments in the future.

6. FUTURE WORK

There are a number of possible extensions to this work:

6.1 Hardware Implementation

The SLAM system has been implemented and tested in computer simulation. Something that works in simulation may not work well in the real world. In the next step, we could implement these algorithms on a real SUAS and run trials at Disaster City®. This will give us a chance to see if our results hold true in real world scenarios.

6.2 Loop Closure

Loop Closure is extremely important to correct errors in SLAM. Our system does not include loop closure functionality. Implementing loop closure will help us reduce the errors in localization. We could also test the effect of characteristics of indoor disaster environments on loop closure.

6.3 Beyond SLAM

SLAM is not the only hurdle when it comes to flying in damaged buildings and structures. One has to account for other factors like Human - Robot Interaction, autonomy, number of agents, their formations and path planning. Better situational awareness and improvement in controls could help the pilot tele-operate the robot effectively.

REFERENCES

[1] S. Agarwal, R.R. Murphy, and J.A. Adams. Characteristics of indoor disaster environments for small uass. In *Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on*, pages 1–6, Oct 2014.

[2] S. Ahrens, D. Levine, G. Andrews, and J.P. How. Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments. In *Robotics and Automation (ICRA), 2009 IEEE International Conference on*, pages 2643–2648, May 2009.

[3] A. Al Redwan Newaz, F.A. Pratama, and Nak Young Chong. Exploration priority based heuristic approach to uav path planning. In *RO-MAN, 2013 IEEE*, pages 521–526, Aug 2013.

[4] Adrien Angeli, David Filliat, Stéphane Doncieux, and J-A Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *Robotics, IEEE Transactions on*, 24(5):1027–1037, 2008.

[5] R. Bajcsy, R. Enciso, G. Kamberova, L. Nocera, and R. Sara. 3d reconstruction of environments for virtual collaboration. In *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, pages 160–167, Oct 1998.

[6] Mikael Berg. Navigation with simultaneous localization and mapping: For indoor mobile robot. Master's thesis, Institutt for Teknisk Kybernetikk, 2013.

[7] Jose A. Castellanos and Juan D. Tardos. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[8] Paul R Cohen. Empirical methods for artificial intelligence. *IEEE Intelligent Systems*, (6):88, 1996.

[9] Michael Csorba. *Simultaneous localisation and map building*. PhD thesis, University of Oxford, 1997.

[10] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999.

[11] Frank Dellaert, StevenM. Seitz, CharlesE. Thorpe, and Sebastian Thrun. Em, mcmc, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 50(1-2):45–71, 2003.

[12] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[13] Gamini Dissanayake, Hugh Durrant-Whyte, and Tim Bailey. A computationally efficient solution to the simultaneous localisation and map building (slam) problem. In *Robotics and Automation (ICRA), 2000 IEEE International Conference on*, volume 2, pages 1009–1014. IEEE, 2000.

[14] MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (slam) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241, 2001.

[15] Arnaud Doucet, Nando De Freitas, Kevin Murphy, and Stuart Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings*

of the Sixteenth Conference on Uncertainty in Artificial Intelligence, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[16] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

[17] Hugh Durrant-Whyte, Somajyoti Majumder, Sebastian Thrun, Marc De Battista, and Steve Scheding. A bayesian algorithm for simultaneous localisation and map building. In *Robotics Research*, pages 49–60. Springer, 2003.

[18] J. Fossel, D. Hennes, D. Claes, S. Alers, and K. Tuyls. Octoslam: A 3d mapping approach to situational awareness of unmanned aerial vehicles. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 179–188, May 2013.

[19] Dorian Galvez-Lopez and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *Robotics, IEEE Transactions on*, 28(5):1188–1197, 2012.

[20] S. Grzonka, G. Grisetti, and W. Burgard. A fully autonomous indoor quadrotor. *Robotics, IEEE Transactions on*, 28(1):90–100, Feb 2012.

[21] Slawomir Grzonka, Giorgio Grisetti, and Wolfram Burgard. Towards a navigation system for autonomous indoor flying. In *Robotics and Automation (ICRA), 2009 IEEE International Conference on*, pages 2878–2883. IEEE, 2009.

[22] Jose E Guivant and Eduardo Mario Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *Robotics and Automation, IEEE Transactions on*, 17(3):242–257, 2001.

[23] J-S Gutmann and Kurt Konolige. Incremental mapping of large cyclic environments. In *Computational Intelligence in Robotics and Automation, 1999.*

CIRA'99. Proceedings. 1999 IEEE International Symposium on, pages 318–325. IEEE, 1999.

[24] Dongze Huang, Zhihao Cai, Yingxun Wang, and Xiang He. A real-time fast incremental slam method for indoor navigation. In *Chinese Automation Congress (CAC), 2013*, pages 171–176, Nov 2013.

[25] Park Jongho and Kim Youdan. Obstacle detection and collision avoidance of quadrotor uav using depth map of stereo vision. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2013. doi:10.2514/6.2013-4994.

[26] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2100–2106. IEEE, 2013.

[27] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems (IROS), 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154 vol.3, Sept 2004.

[28] G.-J.M. Kruijff, V. Tretyakov, T. Linder, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, E. Pianese, S. Corrao, F. Priori, S. Febrini, and S. Angeletti. Rescue robots at earthquake-hit mirandola, italy: A field report. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pages 1–8, Nov 2012.

[29] Yasir Latif, César Cadena, and José Neira. Realizing, reversing, recovering: Incremental robust loop closing over time using the irrr algorithm. In *Intelli-*

gent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 4211–4217. IEEE, 2012.

[30] Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. Robust pose-graph loop-closures with expectation-maximization. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 556–563. IEEE, 2013.

[31] John J Leonard and Hugh F Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *Robotics and Automation, IEEE Transactions on*, 7(3):376–382, 1991.

[32] John J Leonard, Hugh F Durrant-Whyte, and Ingemar J Cox. Dynamic map building for an autonomous mobile robot. *The International Journal of Robotics Research*, 11(4):286–298, 1992.

[33] Qing Li, Da-Chuan Li, Qin-fan Wu, Liang-wen Tang, Yan Huo, Yi-xuan Zhang, and Nong Cheng. Autonomous navigation and environment modeling for mavs in 3-d enclosed industrial environments. *Computers in Industry*, 64(9):1161–1177, 2013.

[34] Yang Liu and Hong Zhang. Indexing visual features: Real-time loop closure detection using a tree structure. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3613–3618. IEEE, 2012.

[35] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[36] S. Lynen, M. Bosse, P. Furgale, and R. Siegwart. Placeless place-recognition. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 303–310, Dec 2014.

[37] Brian MacAllister, Jonathan Butzke, Alex Kushleyev, Harsh Pandey, and Maxim Likhachev. Path planning for non-circular micro aerial vehicles in constrained environments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3933–3940. IEEE, 2013.

[38] Harada Masanori, Nagata Hideyuki, Simond Johan, and Bollino Kevin. Optimal trajectory generation and tracking control of a single coaxial rotor uav. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2013. doi:10.2514/6.2013-4531.

[39] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[40] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar von Stryk. Comprehensive simulation of quadrotor uavs using ros and gazebo. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 400–411. Springer, 2012.

[41] Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, Kazunori Ohno, Eijiro Takeuchi, and Satoshi Tadokoro. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841, 2012.

[42] Michael Montemerlo and Sebastian Thrun. Fastslam 2.0. *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*, pages 63–90, 2007.

[43] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping prob-

lem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2002.

[44] Robin Murphy. *Introduction to AI robotics*. MIT Press, 2000.

[45] Robin R. Murphy. *Disaster Robotics*. MIT Press, 2014.

[46] Paul Newman. *On the structure and solution of the simultaneous localisation and map building problem*. PhD thesis, University of Sydney, 1999.

[47] Paul Newman and Kin Ho. Slam-loop closing with visually salient features. In *Robotics and Automation (ICRA), 2005 IEEE International Conference on*, pages 635–642. IEEE, 2005.

[48] Kevin S. Pratt, Robin Murphy, Sam Stover, and Chandler Griffin. Conops and autonomy recommendations for vtol small unmanned aerial system based on hurricane katrina operations. *Journal of Field Robotics*, 26(8):636–650, 2009.

[49] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. *Robotics and Automation (ICRA) Workshop on Open Source Software*, 3(3.2):5, 2009.

[50] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[51] Hagit Shatkay and Leslie Pack Kaelbling. Learning topological maps with weak local odometric information. In *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence - Volume 2*, IJCAI'97, pages 920–927, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[52] Shaojie Shen, Nathan Michael, and Vijay Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *Robotics and Automation (ICRA), 2011 IEEE international conference on*, pages 20–25. IEEE, 2011.

[53] Shaojie Shen, Nathan Michael, and Vijay Kumar. Obtaining liftoff indoors: Autonomous navigation in confined indoor environments. *Robotics & Automation Magazine, IEEE*, 20(4):40–48, 2013.

[54] Shaojie Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4974–4981, May 2014.

[55] Robert Sim, Pantelis Elinas, Matt Griffin, James J Little, et al. Vision-based slam using the rao-blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, volume 14, pages 9–16, 2005.

[56] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer, 1990.

[57] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer, 1990.

[58] Randall C Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986.

[59] John Stowers, Michael Hayes, and Andrew Bainbridge-Smith. Beyond optical flowbiomimetic uav altitude control using horizontal edge information. In *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, pages 372–377. IEEE, 2011.

[60] J. Suarez and R.R. Murphy. Using the kinect for search and rescue robotics. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pages 1–2, Nov 2012.

[61] R. Suzuki, T. Matsumoto, A. Konno, Y. Hoshino, K. Go, A. Oosedo, and M. Uchiyama. Teleoperation of a tail-sitter vtol uav. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1618–1623, Oct 2010.

[62] Sebastian Thrun. A probabilistic on-line mapping algorithm for teams of mobile robots. *The International Journal of Robotics Research*, 20(5):335–363, 2001.

[63] Sebastian Thrun. Robotic mapping: A survey. *Exploring Artificial Intelligence in the New Millennium*, pages 1–35, 2002.

[64] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5(3-4):253–271, 1998.

[65] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[66] Daichi Toratani, Takehiro Higuchi, and Seiya Ueno. Terrain following flight of uav using information amount feedback. In *SICE Annual Conference (SICE), 2013 Proceedings of*, pages 1503–1508, Sept 2013.

[67] Junqiu Wang, Hongbin Zha, and Roberto Cipolla. Coarse-to-fine vision-based localization by indexing scale-invariant features. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(2):413–422, 2006.

[68] Stefan Williams, Gamini Dissanayake, and Hugh Durrant-Whyte. Towards terrain-aided navigation for underwater robotics. *Advanced Robotics*, 15(5):533–549, 2001.

[69] Junjun Wu, Hong Zhang, and Yisheng Guan. An efficient visual loop closure detection method in a map of 20 million key locations. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 861–866. IEEE, 2014.