2012-03-13

Smart Phone-based Indoor Guidance System for the Visually Impaired

Brandon Lee Taylor
Brigham Young University - Provo

Follow this and additional works at: http://scholarsarchive.byu.edu/etd
Part of the Electrical and Computer Engineering Commons

Recommended Citation
Smart Phone-based Indoor Guidance System
for the Visually Impaired

Brandon Taylor

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

D. J. Lee, Chair
Doran K. Wilde
James K. Archibald

Department of Electrical Engineering
Brigham Young University
April 2012

Copyright © 2012 Brandon Taylor
All Rights Reserved
ABSTRACT

Smart Phone-based Guidance System for the Visually Impaired

Brandon Taylor
Department of Electrical Engineering, BYU
Master of Science

A smart phone camera based indoor guidance system to aid the visually impaired is presented. Most proposed systems for aiding the visually impaired with indoor navigation are not feasible for widespread use due to cost, usability, or portability. We use a smart phone vision based system to create an indoor guidance system that is simple, accessible, inexpensive, and discrete to aid the visually impaired to navigate unfamiliar environments such as public buildings. The system consists of a smart phone and a server. The smart phone transmits pictures of the user’s location to the server. The server processes the images and matches them to a database of stored images of the building. After matching features, the location and orientation of the person is calculated using 3D location correspondence data stored for features of each image. Positional information is then transmitted back to the smart phone and communicated to the user via text-to-speech. This thesis focuses on developing the vision technology for this unique application rather than building the complete system. Experimental results demonstrate the ability of the system to quickly and accurately determine the pose of the user in a university building.

Keywords: POSE, visually impaired, fundamental matrix, direct linear transform, SURF
I would like to express my appreciation and gratitude for my advisor, Dr. Dah Jye Lee, for giving me the opportunity to work with him and lending all of his help, guidance, and support. I would also like to give my thanks to my committee members, Dr. Doran Wilde and Dr. James Archibald. Furthermore, my family has always been there for me and I am forever grateful for their love and encouragement. Most of all I thank my wife for her love, support, and sacrifice for helping me achieve my goals.
# TABLE OF CONTENTS

**LIST OF TABLES** ................................................................................................................................. vi

**LIST OF FIGURES** ................................................................................................................................. vii

1 **Introduction** ......................................................................................................................................... 1

1.1 Motivation............................................................................................................................................. 2

1.2 Recent Developments for the Visually Impaired.................................................................................. 4

1.2.1 Obstacle Avoidance ....................................................................................................................... 4

1.2.2 Navigation Assistance .................................................................................................................. 5

1.3 System Requirements ......................................................................................................................... 8

2 **System** ............................................................................................................................................... 10

2.1 System Overview ............................................................................................................................... 10

2.2 Stored Image Map with 3D Correlation ............................................................................................ 11

2.2.1 Image Map Global Data .............................................................................................................. 11

2.2.2 Image Data .................................................................................................................................. 12

2.2.3 Collecting the Image Data .......................................................................................................... 15

2.2.4 Floor Plans and Supplemental Location Data ............................................................................. 15

2.3 Hand-held Device ............................................................................................................................... 15

2.4 Server ............................................................................................................................................... 16

3 **Vision Processing** ........................................................................................................................... 18

3.1 Background ....................................................................................................................................... 18

3.2 Algorithm Overview .......................................................................................................................... 20

3.3 Feature Detection, Description and Matching .................................................................................. 24

3.3.1 Common Methods ...................................................................................................................... 24

3.3.2 Speeded Up Robust Features ................................................................................................... 24
3.3.3 SURF Feature Descriptors ................................................................. 25
3.3.4 Scale Space ......................................................................................... 27
3.3.5 Matching Images .................................................................................. 28
3.4 Removing Outliers .................................................................................. 29
  3.4.1 Epipolar Geometry .............................................................................. 29
  3.4.2 RANSAC .............................................................................................. 31
  3.4.3 8-Point Algorithm with RANSAC ....................................................... 32
3.5 Pose Calculation ....................................................................................... 35

4 Testing and Results ..................................................................................... 41
  4.1 Test Objectives ....................................................................................... 41
  4.2 Location Information and Image Map ..................................................... 41
  4.3 Testing Hardware ................................................................................... 42
  4.4 Test Procedure ....................................................................................... 43
  4.5 Test Results ............................................................................................ 43
    4.5.1 Image Frequency ............................................................................... 43
    4.5.2 Algorithm Robustness ....................................................................... 43
    4.5.3 Pose Estimation Accuracy ................................................................. 44
    4.5.4 Insufficient Features .......................................................................... 47

5 Conclusion .................................................................................................... 48
  5.1 Summary ................................................................................................. 48
  5.2 Contributions .......................................................................................... 49
  5.3 Future Work ............................................................................................. 49

REFERENCES .................................................................................................... 51
LIST OF TABLES

Table 4-1: Estimated vs. Measured Positions ........................................................................46
LIST OF FIGURES

Figure 2-1: System Overview .......................................................... 11
Figure 2-2: Local Coordinate System ............................................. 12
Figure 2-3: Sample Map Image Data ............................................... 13
Figure 2-4: 2D and 3D Point Correspondence for Pose ................. 14
Figure 3-1: Vision and Pose Estimation Algorithm Overview ......... 23
Figure 3-2: Integral Image Creation and Use ................................. 25
Figure 3-3: Haar Wavelets ............................................................. 26
Figure 3-4: SURF Feature Descriptor Grid .................................... 27
Figure 3-5: Matching Distant Image .............................................. 28
Figure 3-6: Epipolar Geometry ...................................................... 30
Figure 3-7: Epipolar Lines .............................................................. 31
Figure 3-8: Epipolar Line RANSAC Outlier ................................. 33
Figure 3-9: Initial Matched Feature Points ................................. 34
Figure 3-10: Matched Feature Points after RANSAC with Homography .... 34
Figure 3-11: False Inlier after Fundamental Matrix RANSAC ........ 35
Figure 3-12: Point Correspondence Relation ............................... 37
Figure 3-13: Reprojection Outlier ................................................. 39
Figure 3-14: Final Pose Estimate .................................................. 40
Figure 3-15: Floor Plan with Pose ................................................ 40
Figure 4-1: Clyde Building 4th Floor with Local Coordinate System .. 42
Figure 4-2: Map Progression Images ........................................... 45
Figure 4-3: Insufficient Unique Features ................................. 47
1 INTRODUCTION

Visually impaired people need assistance to find their way in unfamiliar places. The current systems in place for helping the visually impaired are not sufficient for navigating unknown environments. There are two important requirements for navigation. The first is localization, or figuring out the person’s current location. The second is path finding, or determining the route a person needs to take to get from one place to another. Neither of these needs is sufficiently met by Braille markings in buildings. More recently some indoor guidance systems have been developed with the visually impaired in mind. However, several problems keep these systems from being a feasible commercial solution. In this thesis we present a unique 3-D vision algorithm that is suitable for building an indoor guidance system for the visually impaired that overcomes these obstacles.

In order to create an indoor guidance system that is simple, accessible, inexpensive, and discrete, we use a smart phone vision based system to help the visually impaired as they navigate unfamiliar environments such as public buildings. The system consists of a smart phone and a server. The smart phone transmits pictures of the user’s location to the server. The server processes the images and matches them to a database of stored images of the building. The camera pose is calculated from the matched image, and the current location and navigation instructions are given to the user.
1.1 Motivation

Navigating unknown environments is a necessity in life that has always proven difficult for individuals with vision impairments. Two primary problems impede a person’s navigation ability. The first problem is obstacle avoidance. Obstacle avoidance deals with the objects and terrain in the immediate area around the individual such as people, stairs, tables, and walls. Tools have been developed and are actively used among visually impaired to aid in obstacle avoidance such as guide canes and Seeing Eye dogs. The second problem is way finding or route navigation. Trying to get from point A to point B proves difficult without visual clues of building layouts or terrain mappings. No widespread system has been implemented that aids in route navigation. Studies show that the visually impaired feel that they have an especially difficult time learning new routes [1]. Being unable to handle these types of situations themselves also negatively affects the feeling of independence of the visually impaired.

Visually impaired people have few options when it comes to finding their way in an unfamiliar place. Traditionally the visually impaired must either be guided by or get directions from another person. Having someone to guide is useful but not always an option and leaves the impaired feeling quite dependent on other people. Asking for directions is common but has several challenges. Some individuals find it hard to translate directions into effective information for the visually impaired person. The difficulty arises because most people without visual impairments navigate differently, typically relying on visual landmarks. In addition, the visually impaired person must remember the directions, since writing them down is not feasible. If the visually impaired becomes lost, the only option is to find someone to help.

The only widespread solution for helping the visually impaired is Braille signs. A major problem with this system is that it is not adequate for navigation. Most public buildings include
Braille signs on doors, elevators and other various locations. Although Braille signs provide some benefit, they are only helpful for learning where the person is. Braille signs do not offer directions. Finding the Braille signs is also an issue as one must feel around to locate the sign. Lastly, the largest problem with Braille is that most visually impaired people do not know Braille [2], [3]. In the US only about 10 percent of legally blind people know Braille.

GPS has proved to be an indispensible tool as an outdoor navigational aid for everyone including the visually impaired. Programs have been developed for smart phones that utilize text-to-speech and voice commands to allow the visually impaired to take advantage of GPS guidance systems. However, GPS is inadequate for indoor navigation as the signal encounters too much interference to be usable.

Currently, there is no feasible solution to indoor navigation for the visually impaired. The traditional methods just discussed (Braille, GPS, spoken directions) are clearly not an optimal means for finding one’s way in an unfamiliar indoor location. More recently, several systems have been tested for indoor navigation. Studies have shown that these technological aids improve the ability for the visually impaired to find their way [4]. Unfortunately, many of these systems have complicated setups that cause the user to stand out or are awkward to use [4]. Other systems require cumbersome sensor rigs that hamper normal movement and are not suitable for people to carry around and use on a normal basis [4], [5]. Moreover, studies show that visually impaired people would prefer to not attract unnecessary attention to themselves while using a navigation system [1]. Another challenge to indoor navigation tools is the cost. The average user cannot afford a system which may cost over $1,000. Additionally, the financial burden placed on a company’s building to change infrastructure and accommodate a system is a factor that must be considered. A feasible, commercial solution requires a system to be compact, portable, light
weight, discrete, and easy to use. The system must also be cost effective for both the user and provider. Lastly, the system should offer hands-free operation to allow users to hold personal belongings such as a guide cane or harness for a Seeing Eye dog, as these belongings are essential for obstacle avoidance.

In this thesis we propose a novel solution for indoor route navigation that overcomes the limitations of other systems. Our proposed solution, dubbed the Seeing Eye Phone, uses a smart phone that can be worn around the user’s neck that takes pictures in regular intervals of the user’s location. The pictures are then sent to a server for image processing. The captured input image is compared against an image database of the building being navigated. The position and orientation of the user is estimated from the images, and then details are sent to the user about their current location as well as instructions to the desired location. This database is constructed offline with periodic images along with their stored feature points, 3D location, and other pertinent map information such as room numbers that one would look for on an indoor map.

1.2 Recent Developments for the Visually Impaired

Due to a large number of the population suffering from some sort of visual impairment, a significant amount of work has been done recently to find methods to aid the visually impaired. Most of the research typically focuses in the areas of reading, obstacle avoidance, and navigation or wayfinding. With our ever expanding technologies, new solutions are constantly being discovered to better aid the visually impaired.

1.2.1 Obstacle Avoidance

Recently, several methods have focused on obstacle avoidance for the visually impaired. Sonar has been used to provide guidance assistance in detecting obstacles and for navigation
Hesch used a foot-mounted pedometer and a white cane-mounted sensing package. The sensors were comprised of a three-axis gyroscope and a two-dimensional laser scanner [9]. Wong used stereo vision to aid in obstacle detection and communicated the object distance and details through structured sound as well as verbal sounds [10]. The NavBelt is a belt worn by the user with an array of ultrasonic sensors to assist in obstacle avoidance by sending sounds to headphones. The guide cane uses a similar sensor array, but is attached to a cane with a motorized wheel. This wheel can then steer to avoid objects much like a guide dog would [5]. The tongue vision system takes things to another level of detail. The system is comprised of a head-mounted camera that is attached to an electronic stimulator that is fed into the mouth. Multiple electrodes stimulate the tongue to represent what the camera sees [4].

1.2.2 **Navigation Assistance**

Many different technologies have been used for navigation and localization. With GPS devices quite prevalent, several systems have been implemented for outdoor guidance for the visually impaired [11], [12], [13]. Moulton created an outdoor talking navigation system with speech recognition using GPS [13]. GPS has proven to be an accurate and reliable method for outdoor navigation but fails when used indoors.

Only more recently have people been researching better indoor navigation systems. Kalia designed the Building Navigator system which provides information about the spatial layout of rooms, hallways, and other features. It uses synthetic speech output to communicate with the user and demonstrated improvement in the user’s ability to find destinations [14]. Several have used Radio Frequency Identification (RFID) systems as an indoor navigational method [15], [16], [17], [18]. Na expanded on the RFID by using a “smart floor” where every tile has a unique RFID tag to provide more accurate and frequent localization [19]. Tjan created a digital sign
system of patterned retro reflective tags that could be read by a handheld camera and vision system [20]. Coughlan used a mobile phone camera system that reads barcodes and conveys the barcode message with speech [21]. Several styles of RF based localization have also been used [22], [23], [24]. Biswas used the signal strength of known Wi-Fi locations for robotic localization and navigation indoors [23]. Inoue et al. tested a solution similar to the Wi-Fi system where several license free band radios were installed in locations around the building [24]. These radios would communicate with a handheld receiver which would determine location by the signal strength of the transmitters. The handheld receiver was in turn communicating with a smart phone via Bluetooth. One system requires a robot to accompany the user [25].

While all of these methods have proved useful for assisting the visually impaired, they have some notable drawbacks. Sonar provides a cheap method for locating obstacles but only gives rough measurements and requires dedicated hardware. It also does not help with navigation. Hesch’s pedometer and cane works well for obstacle detection but does nothing as far as navigation. Although the tongue vision system gives more special detail than others, it is extremely awkward to use. It is very unappealing with the headgear plugging into the mouth. Even with the high amount of detail it provides, this type of system is not preferred because of how much unwanted attention it attracts to the user.

GPS is reliable for use in outdoor navigation but is less accurate with smaller handheld devices such as smart phones and is unusable in an indoor environment because the signal encounters too much interference. RFID has proven to work quite well in indoor navigation and localization. One disadvantage of RFID is that it is a discrete system that can only provide periodic information when close enough to an installed RFID tag in a building. Increased map density and accuracy means increasing the number of RFID tags installed. The number of RFID sensors used by Na, where there is a separate sensor for every floor panel, effectively makes the
navigation continuous but greatly increases the infrastructure cost as the number of RFID tags would be in the thousands to tens of thousands to outfit a building. For new buildings it would not be as much of an issue, but installing it into currently existing buildings poses a real problem due to the required renovating. RFID also requires a dedicated sensor that is not readily available for the user. Other sign systems used can provide extra information about the location but also have the disadvantage of not providing continuous navigational information. Sign based systems are also more noticeable as they take up wall space and must be large enough for a camera to see. The more signs used, the more wall space occupied, and the signs are not particularly attractive and can clutter up a building using them.

Coughlan’s system takes advantage of the usefulness of a smart phone camera system, but again uses discrete tags that contain unique information for the given location. Every sign must be specially created for the given location installed. The radio beacon and WiFi systems have inaccuracies due to the signals not being constant. Buildings must be calibrated since the signal strength changes with walls and other objects in the way. Even after calibration, moving people, temperature, and other factors can drastically change the signal strength leading to inaccurate positioning. The radio beacon system has similar drawbacks to the Wi-Fi systems, but also requires a separate radio to be carried. Using robots for guidance allows more sensors to be used for increased accuracy, but it is not feasible in environments with other people moving around and would draw a lot of unwanted attention to the user. Robotic systems would be quite expensive with the amount of sensors used and the battery required for it to run. The user would also need to keep track of the robot.
1.3 **System Requirements**

There is a need for an indoor guidance system that the visually impaired can use in which all the basic requirements discussed earlier are fulfilled. Those basic requirements are: low cost, pose accuracy, simple, discrete, and easy to use. There are several systems that have been developed for the visually impaired. They function well and help the visually impaired navigate. However, each of these systems lacks one or more of the criteria. The basic requirements are discussed in further detail below. In order to create a system that is usable, desirable, and feasible for commercial use the following criteria are important:

- **Inexpensive for user:** Cost always proves a big concern for the average user. If the system costs too much for most people to afford then it is hard to justify making it a widespread solution. One technology specialist for the visually impaired suggested that any system costing more than $1,000 could succeed only if its benefits are enormous and obvious [4].

- **Inexpensive infrastructure setup:** Another of the main drawbacks to a commercial solution is the upfront costs for building the infrastructure. The ability to install the system in existing buildings at a reasonable expense and with minimal renovation is another key problem that must be overcome.

- **Accurate indoor pose estimation:** Outdoor systems usually only need to be accurate to a few meters to tens of meters to be usable. For an indoor system accuracy is very important. Inaccuracies of only several meters can mean the difference between being at the restroom and being in a completely different hallway.

- **Portable:** The user’s device must be easy to carry around and easy to operate.
• **Not cumbersome:** Ease of use is crucial to a good system. If the user thinks that the system is too much of a hassle to use then he may prefer not to use it. The system should also allow the person free use of his hands and hearing.

• **Discrete:** Although some users would still use a system that draws unwanted attention to themselves, a discrete system is still preferred, and for some a necessary requirement.
2 SYSTEM

2.1 System Overview

The Seeing Eye Phone system consists primarily of a server and a smart phone. The smart phone receives input from the camera and from the user. The phone takes pictures at regular intervals and sends them to the server along with a time stamp and its most recently known position. The user can also speak a command, such as asking for their current location or how to get to their desired destination. The commands are interpreted by the use of voice recognition software and passed on to the server. The server matches the input images to the map images in the database. Once a match is found the server calculates the camera pose in the 3D real world coordinate system and then uses the floor plan to find a route to the desired destination. It sends the location and directions back to the phone which uses a text-to-speech function to direct the user to his desired destination. Figure 2-1 shows an overview of the system.

The map uses a local coordinate system with sections of the coordinate system given labels of desirable locations or useful navigation information. These include things like hallways, doors, restrooms, room numbers, floor levels, and other useful information one may desire to find or need to navigate.
2.2 Stored Image Map with 3D Correlation

The “image map” of a building is a collection of data consisting of images, image features with corresponding feature descriptors and their 3D coordinates, floor plans, and other location data. This image map is created offline as a part of installing the Seeing Eye Phone system in the building. It is then used to match images from the phone and find directions and destinations.

2.2.1 Image Map Global Data

The image map contains several pieces of key data that all tie together. At the top level each building will use a local coordinate system as shown in Figure 2-2. The latitude and longitude of the origin are stored along with the rotation relative to a global coordinate system.
Having the global data allows the last position of the user coming into the building to be used as a starting point for image searching.

![Figure 2-2: Local Coordinate System](Image courtesy of Google.com)

2.2.2 Image Data

As part of the image map building process, map images are taken at regularly spaced intervals of hallways and other navigable areas of the building. The map images must be spaced frequently enough to ensure the phone images can always match to a map image that is not too much different from the phone images. Using images that are taken closer together improves the location estimation accuracy because of the smaller disparity between feature points. Matching
to distant images leads to higher inaccuracies in 3D pose estimation due to the image resolution. However, if the images are spaced too close together the map will take up too much memory and slow down image matching. Images taken every 10-15 feet prove to be a good balance between accuracy, map size, and processing speed. The images are processed with a feature detector to find quality 2D feature points. Feature descriptors are then created using pixel data in the proximity of the feature point. A collection of several different feature descriptors make up the feature vector. The feature vectors are typically made up of 64 or 128 feature descriptors. This feature vector can then be matched against the input image feature vectors to find the closest matches. Figure 2-3 shows a sample of image data that would be included in a building’s image map.

<table>
<thead>
<tr>
<th>Image</th>
<th>Floor</th>
<th>Relative Location (m)</th>
<th>Keypoints List</th>
<th>3D location List</th>
<th>Descriptor List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>(5,10)</td>
<td>Keypoints 1</td>
<td>3D 1</td>
<td>Descriptors 1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>(5,20)</td>
<td>Keypoints 2</td>
<td>3D 2</td>
<td>Descriptors 2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>(5,30)</td>
<td>Keypoints 3</td>
<td>3D 3</td>
<td>Descriptors 3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>(20,30)</td>
<td>Keypoints 4</td>
<td>3D 4</td>
<td>Descriptors 4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>67</td>
<td>4</td>
<td>(20,30)</td>
<td>Keypoints 87</td>
<td>3D 67</td>
<td>Descriptors 67</td>
</tr>
<tr>
<td>68</td>
<td>4</td>
<td>(55,15)</td>
<td>Keypoints 88</td>
<td>3D 68</td>
<td>Descriptors 68</td>
</tr>
</tbody>
</table>

Figure 2-3: Sample Map Image Data
In order to speed up real time processing of the Seeing Eye Phone system, the runtime version of the map uses preprocessed images. Once the map is created from the images, only the features, descriptors, and points need to be stored for use by the system. The entire image is not needed once processed. Using preprocessed images speeds up real time processing in the final system by saving redundant computations.

The main purpose of matching images is to determine the pose of the camera and hence the location and orientation of the user. The image feature points alone, however, are not enough to give an accurate pose measurement. Each feature point must be paired with a corresponding real world 3D position in the local coordinate system as shown in Figure 2-4. When the input image is matched to the map image, the 3D coordinates are used in conjunction with the 2D feature points in the input image and the camera parameters in order to calculate an accurate pose estimate of the phone.

Figure 2-4: 2D and 3D Point Correspondence for Pose
2.2.3 Collecting the Image Data

The most efficient way to create the map would be the use of a robotic system which would take pictures at regular intervals. It would use a SLAM algorithm, or Simultaneous Localization and Mapping, along with stereo cameras, dead reckoning, and other sensors or algorithms to generate the map images along with their corresponding 3D points. Fiducials (easy to detect markers) placed at key locations around the building to be mapped as an absolute reference can help increase the accuracy of the 3D points it finds. The development of the map-creating system is not discussed in this thesis but left to future research.

2.2.4 Floor Plans and Supplemental Location Data

In order to effectively navigate, floor plans and other location data must be added to the map. Some of this data may include key locations one may wish to find. These locations could be information desks, room numbers, rest rooms, and other key locations that would be included on a printed map or commonly requested destinations. Floor plans are described in the local coordinate system and assigned a floor number. The combination of the floor plan with the coordinate system allows the path finding algorithm to determine the position of the key locations as well as obstacles and pathways.

2.3 Hand-held Device

The hand held device only has a few requirements. It must be able to use Wi-Fi, run custom programs, and have a camera. Most smart phones have all of the required features and are very common in today’s society. Along with being portable, user friendly, and discrete, the smart phone makes an ideal platform for the guidance system. The user wears the phone around his neck and interacts with the phone using voice recognition software which is readily available.
for these types of phones. The phone first connects to the server over Wi-Fi. The phone then starts taking images at a constant rate and sends them to the server. Simple commands can be giving to the phone like “Where am I?” or “Find room 496.” The server processes the images and returns the location or directions to the desired location. Text-to-speech programs, also readily available, convey the location and directions to the user to guide him to the destination.

2.4 **Server**

The server performs the image matching, pose calculation, and navigation planning with the use of the stored image map. Once the server receives an image, it first checks the additional information sent with the image including the most recent location. It uses this initial location to determine the part of the building to start searching images. Usually the phone would connect to the building’s local network which would provide a good starting point. If entering from outside the building, it can pull the last known GPS- or network-based position if available. Once the initial starting point is found, the server only needs to look in the vicinity of the previous spot, thus improving image matching speed.

The server uses a feature detector to find feature points in the smart phone image. It then starts matching these features with the features stored in the image map database. After determining the initial matched features, the camera geometry is used to remove incorrect features. Once the server finds the correctly matched image of the user’s location, it associates the 3D points of the map features with the smart phone image features. With the associated points and the camera’s intrinsic parameters, the pose of the user within the building is estimated. A route is computed to the desired destination and directions are transmitted back to the user. The vision processing done by the server is discussed in greater detail in chapter 3.
The server does not have stringent hardware requirements. An average quad core computer is all that is needed to process the images. However, the building does need enough Wi-Fi coverage for the smart phone to stay constantly in contact with the server.
3 VISION PROCESSING

The core of this thesis focuses on the vision processing used to calculate the camera pose and hence the user’s location and orientation using only one camera with no added features to the scene. While some systems have used smart phone cameras to determine location they have all used specialized markers. The Seeing Eye Phone system uses the natural scenery for processing instead of filling up the walls with information markers. Needing to recognize features of the natural indoor scenes requires more processing power, but provides an accurate continuous estimate of the user’s position instead of recognizing periodic individual markers. This system also leaves the location in its original form instead of cluttering up the walls with odd markers.

3.1 Background

There exist several systems for finding pose using image processing. Many robotic systems are designed around SLAM, or simultaneous localization and mapping. The unique problem of SLAM is creating a map of one’s surroundings while at the same time localizing oneself within the map. SLAM proves to be a difficult problem since each question requires knowledge of the other. SLAM is primarily used for systems navigation in completely unknown environments and requires the system to travel the environment more than once in order to build up the map. In our application, localization of the user is being done in an already known environment. The map must be made available in order to determine the location as the user
moves. In addition, SLAM needs a way to find distance to obstacles on the fly, such as stereo vision.

Stereo vision is very popular for vision based systems that need to find the distance to features because it can compute distance to objects seen by both cameras. Stereo vision works using the same principles that our eyes do. There are two cameras set up close together side by side with their focal axis parallel to each other and facing the same direction. Each camera sees a slightly different view of the same scene. That difference allows the distance from the cameras to be computed or triangulated. In order to accurately compute the distance, the relative pose of the cameras must be known along with their individual focal lengths and other intrinsic camera values. Careful set up and calibration of the two cameras is required in order to accurately measure distances to objects and in turn localize the cameras in a map. Because of these requirements stereo vision systems tend to be quite large and cumbersome.

MonoSLAM has been developed to work with only a single camera for simultaneous localization and mapping; however, it uses an assumed movement model for the camera in order to compare frame to frame images and is less accurate [26]. This movement model would not work as well for a person who is walking as it would on a flying or rolling robot. In general, MonoSLAM requires a smooth and predictable movement of the camera. People have less predictable movement as they can start, stop, turn or change direction at any time. There is also camera bob due to the stepping motion a person makes that would be very difficult to account for.

Many robotic vehicles rely on other supplemental sensors like odometers, LIDAR, and sonar to accompany vision algorithms and help with localization within a map. Most of these systems are expensive or not readily available for use on small commonly used devices like
smart phones. LIDAR for example can easily cost several thousand dollars. Many methods using these sensors also require a more defined movement model. People do not have a predictable enough movement model to use. Because smart phones are being used as the guidance device, the system must work with the commonly used hardware available on the majority of smart phones. Although most smart phones have some sensors like accelerometers and compasses, it is assumed for this work that only one camera is available to capture images with no secondary sensors to aid in localization. The addition of supplemental sensors to improve accuracy is left to future research.

The purpose of this system is to both give the user knowledge about his current location and also guide the user to a specified known location. The proposed Seeing Eye Phone system takes advantage of the fact that the user is found within a known environment and so a-priori information can be used instead of building up a map on the fly like SLAM systems. However, because stereo vision or single camera systems like MonoSLAM are not available for depth information, another method is used to provide 3D information. In order to accomplish this, the image map is created beforehand containing known features with their 3D locations. Having the 3D locations available allows the pose of the user to be calculated without the need for stereo vision, specific movement models, and expensive or unavailable sensors.

3.2 Algorithm Overview

The vision algorithm uses the following key parts: feature detection, description and matching; outlier elimination; and pose estimation. A feature detector is used to find features in the smart phone image and match them to the map images. The phone images go through the exact same algorithm that the map images do, only the map images are processed prior to the system being used instead of in real time. Once features are determined, unique feature
descriptors are created for feature point and collected together to form a feature vector. These
feature vectors can then be compared like any vectors, where a very small distance error between
them constitutes a match. During the matching process, when an image has too few matched
features it is discarded. Once the number of matched features exceeds a threshold a potential
match to the image location is found. These matched features are referred to as inliers in the first
decision box in Figure 3-1. Only images in the near vicinity of the most recent known location
are compared to the phone image in order to speed up processing time.

Some of the remaining matched features from the first step will inevitably be incorrect
and cause errors in the pose estimation algorithm; therefore, they must be removed prior to pose
estimation. These incorrectly matched features are known as outliers. RANSAC, or RAndom
SAmple Consensus, is employed to help remove the outliers. RANSAC attempts to match a data
set to a model while simultaneously discarding outliers. The fundamental matrix relates
corresponding points of a pair of images, so it is used as the model for RANSAC in order to
accomplish outlier removal. Even though the phone image and map image are not set up like a
true stereo vision pair with a known relationship between the two cameras, the fundamental
matrix still proves useful for getting rid of outliers. A random sample of point correspondences,
or matched feature point pairs, is used to create the parameters of the fundamental matrix. The
point pairs that do not fit the model are discarded as outliers, and the model parameters are
recomputed using only the inliers. Only models with a minimum number of inliers are
considered for a possible solution. The process of choosing random data sets, computing model
parameters, and finding outliers is repeated until a model with the least amount of inlier error is
found. After performing RANSAC, as long as there are still enough matches above the threshold,
the algorithm continues on that image, otherwise it returns to the image matching stage to find a new image.

The remaining inliers give a feature correlation between the two images that allows the algorithm to obtain the correspondence between the phone image’s 2D features and the 3D real world location. The pose is then determined using the camera’s intrinsic parameters with the 2D to 3D point correspondences. The Direct Linear Transform runs on a subset of correspondences to find a rough initial pose estimate. The Levenberg-Marquardt algorithm, a minimization algorithm, further refines the pose estimate to find a more optimal solution. Although outliers were discarded previously, due to the nature of fundamental matrices some outliers may still remain in this step of the algorithm. Once again we employ RANSAC to remove any remaining outliers. This time the pose is used for the model. After a pose estimate is found from a random subset of data, the 3D points are reprojected onto the image using the pose estimate and the camera’s intrinsic parameters. The algorithm compares the reprojected points to the original matched points. A large error results in an outlier. As before, the RANSAC iterations continue until a solution with the lowest error is found. Assuming the number of inliers remaining after using RANSAC is still above a certain threshold the algorithm finally declares a true image match. The resulting pose is transmitted back to the smart phone and also passed on to the navigation algorithm. Figure 3-1 gives the general outline of the vision processing used to find the pose.
Figure 3-1: Vision and Pose Estimation Algorithm Overview
3.3 Feature Detection, Description and Matching

3.3.1 Common Methods

Algorithms for finding, describing, and matching features have been studied extensively. One popular detector is the Harris corner detector [27]. It uses eigenvalues of image derivatives to detect corners. The Harris corner detector is not scale invariant though. Scale Invariant Feature Transform (SIFT) is a more recently developed algorithm that is widely used for its proven performance [28]. In SIFT, features are found using Difference of Gaussians to approximate Laplacian of Gaussians. To find features SIFT, searches at multiple scales and Gaussian blur levels of the image. These scale and blur levels are called octaves. The use of octaves provides robust scale invariant features. The gradient magnitudes and orientations are computed to give each feature an overall orientation which provides rotation invariance. A feature descriptor is then computed finding gradient magnitudes and orientations in subsections of a window around the feature to create a 128 element feature vector. SIFT produces very good results in terms of finding and matching features, but proves too slow to use in near real time applications.

3.3.2 Speeded Up Robust Features

Speeded Up Robust Features (SURF) proved to be the best choice for this application [29]. SURF is another method of finding and describing features. SURF has the advantages of scale and rotation invariant features and runs much faster than SIFT. SURF uses a Hessian-matrix approximation for feature detection on integral images. The use of the integral images, or summed area tables, allows for fast feature detection using the Hessian-matrix approximation as a box filter since it only requires four additions to calculate. Since integral images are used, the Hessian filter is scaled instead of the image in order to handle scale invariance. By only changing
the scale of the filter instead of having the computational burden of scaling and blurring the image multiple times the speed at which SURF handles scale invariance is greatly increased.

The integral image is created by summing all pixel values to the left and above the current pixel as demonstrated in Figure 3-2. In order to sum the pixels in the red box with the original image we simply sum each pixel value which in this case uses 9 additions resulting in the sum of 578. With the integral image, we only need 4 additions: \( \text{Sum} = 1731 - 665 - 564 + 76 = 578 \). The number of additions stays the same regardless of the area summed over when using the integral image. Having the number of additions constant greatly increases speed, especially when scaling up the filters.

![Figure 3-2: Integral Image Creation and Use](image)

### 3.3.3 SURF Feature Descriptors

Both the orientation and the feature vector make heavy use of the Haar-wavelet filters. Haar-wavelets are popularly used for spatial filtering and are shown in Figure 3-3. The black part
has a value of -1 and the white a value of +1. This makes the filter ideal for detecting high contrast areas like edges and corners that make for good features.

The feature descriptors consist of a collection of image data around the feature point as well as an overall feature orientation to describe the feature. The feature orientation gives the feature rotation invariance when it comes to matching. The orientation given is based on the sum of Haar-wavelet responses in an angle around the feature point. Then the feature vector is constructed of Haar-wavelet responses in each of the 16 subsections of a 4x4 window of 25 points for each subsection around the feature point. A sample window is illustrated in Figure 3-4 where the feature point would be in the center of the grid. The response is found once with the horizontal filter and once with the vertical filter resulting in 32 data points. The response is stored along with orientation in both the x and y direction, yielding a 64 element feature vector. Having a size of only 64 elements, this feature vector decreases the computational time of the algorithm in comparison to SIFT that uses a 128 element feature vector. The reader is referred to the SURF paper by Bay et al. for further details [29].

![Figure 3-3: Haar Wavelets](image-url)
3.3.4 Scale Space

The scale space gives SURF scale invariance to allow features to be detected at different distances. More scale octaves result in higher scale invariance. The recommended number of octaves is usually 4 or 5. For this application, however, only 2 octaves are needed. In fact, making the features themselves too scale invariant can actually be detrimental to pose estimation for this application. Usually more scale invariance is needed for object detection because the object could be at any number of scale ranges in the image that contains it. However, in this indoor location finding application it is undesirable. Searching for an image of the end of a hallway provides a good example. The left image of the end of the hallway in Figure 3-5 will be the “object” to be found in the smart phone image on the right. If 5 octaves are used for the
feature detection, the camera image on the right taken from one end of the hallway could be matched to the map image on the left taken from the other end of the hallway. The matched image portion is marked by a black rectangle in the image on the right. This small matched area results in feature points that are too close together and at low resolutions, which will cause a large error in the pose estimation. Therefore, it is desirable to have the features the phone image maps closer to the phone rather than farther away. Using frequent map images allows images to be matched to nearby images, and also essentially creates several levels of scale invariance that are built into the map. Using 2 octaves for the features proves more than enough to handle the smaller scale changes in between map images.

![Figure 3-5: Matching Distant Image](image)

3.3.5 Matching Images

SURF is run on the map images offline and on the phone images in real time. Using SURF on both images gives feature vectors that can be directly compared to each other. The way the feature vectors are set up allows for fast and easy comparison. The orientation of the feature points are used to compensate for rotation in the images. Applying the rotation to the feature
vector simply shifts the elements in the vector so the correct elements in the feature vectors can be directly compared to each other. Then the vectors can be subtracted from each other to obtain an error figure. If the error is under a certain threshold, the feature pair is considered a match.

3.4 Removing Outliers

Using too high of a threshold for the feature matching distance reduces the number of incorrectly matched features, or outliers, but also tends to eliminate a good number of correctly matched features. In order to improve the accuracy of the pose estimation it is desirable to keep as many good features as possible and find another way to eliminate bad features. As a result, the threshold on what is considered a positive match for a feature is relaxed in order to maximize the number of correct features for further testing. Camera geometry and RANSAC are used to remove the extra false features introduced by the low threshold.

3.4.1 Epipolar Geometry

Assuming the image features remain static, the phone image and map image are related by a transformation. Using the point correspondences for the two images, the transformation between the two images can be approximated as the fundamental matrix $F$. This matrix describes the epipolar geometry between the two images. The fundamental matrix $F$ is defined as

$$0 = p_r^T F p_i,$$  \hspace{1cm} (3-1)

where $F p_i$ is the epipolar line in the image containing point $p_i$. The point $p_r$ must lie upon the epipolar line. See Figure 3-6.
Every feature point in one image must lie upon the epipolar line created by the corresponding feature point in the other image. Figure 3-7 demonstrates the features lying on the corresponding epipolar lines created using the fundamental matrix and feature points of the other image. If the corresponding feature point lies too far away from the epipolar line, it is considered an outlier and removed from the list of matched features. However, in order to remove these bad features the fundamental matrix must first be calculated using the features list which may contain bad features and lead to an inaccurate fundamental matrix. In order to find a valid fundamental matrix and remove outliers, RANSAC is used with the 8-Point algorithm [30].
3.4.2 RANSAC

Random sample consensus, or RANSAC, is a non-linear iterative technique used to estimate model parameters from a set of data points while at the same time eliminating bad data points, or outliers. RANSAC has proven itself as a robust parameter estimation method. Although the algorithm is iterative, typically only a few iterations are needed for good results thus allowing the algorithm to run quickly.

RANSAC starts by taking a random smaller sample set from the large data set to get an initial parameter estimate. The elements of the sample set are automatically considered inliers for the iteration they are used. After the initial estimate is made, every other element of the complete data set is then tested against the model. A distance measure is determined for the
amount of error and a threshold established to classify points as inliers or outliers. The inliers are
added to the initial sample set. If the resulting number of inliers is above required inlier
threshold, the model parameters are recalculated with the all of the inliers. The model parameters
and the total error of the inliers are recorded as the current best guess. This process is repeated on
subsequent iterations with new random data sets. The resulting error on each iteration is
compared against the recorded best error. If the new parameters produce less error, than the new
error and parameters are recorded as the new best estimate. After a set number of iterations the
points that match the best estimate are considered inliers.

3.4.3 8-Point Algorithm with RANSAC

For the 8-point algorithm, using \( p_r = (x', y', l)^T \) and \( p_t = (x, y, l)^T \) we multiply out
Equation 3-1 and rearrange the elements to get a vector multiplication of
\[
(x', x, y, x', y', x', y, y', x, y, x, y, l)(F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}) = \mathbf{a}^T \mathbf{f} = 0. \tag{3-2}
\]

Equation 3-2 holds for any given valid set of point correspondences in \( \mathbf{a} \). Matrix \( \mathbf{A} \) is
then created by stacking the row vectors of 8 different point correspondences chosen to get the
matrix multiplication of \( \mathbf{A} \mathbf{f}^T = 0 \). In order to obtain \( \mathbf{f}' \) as an initial estimate of \( \mathbf{f} \) the Singular
Value Decomposition (SVD) of \( \mathbf{A} \) is performed. The column of \( \mathbf{V} \) that corresponds to the
smallest singular value of the SVD is the estimate \( \mathbf{f}' \). After rearranging this column of \( \mathbf{V} \) back
into matrix form to create an initial fundamental matrix \( \mathbf{F}' \), the SVD of \( \mathbf{F}' \) is then taken. Setting
the lowest singular value of the decomposition to 0 and then multiplying the decomposition back
together results in a valid fundamental matrix. This second decomposition is done to account for
the fact that the system has noise and \( F' \) does not initially satisfy the condition that the fundamental matrix is only rank 2.

The RANSAC algorithm starts by choosing 8 random points from the set of matched features in one image that may contain outliers. The parameters of the fundamental matrix \( F \) are then computed with these 8 point correspondences. The model is then applied to all matched points and the error (distance from the epipolar line) computed as \( e = p_r^T F p_l \). Any points that are above a certain threshold distance from the epipolar line are marked as outliers shown by the red dot in the left image of Figure 3-8.

**Figure 3-8: Epipolar Line RANSAC Outlier**

Once all outliers are eliminated for the iteration, if the number of inliers is above a certain threshold the total error of the inliers is stored along with the fundamental matrix model \( F \) as the current best guess. The process is then repeated with a new set of 8 random points. After each iteration the total error is compared to the current best error. If the new fundamental matrix has less error, it replaces the current one. The algorithm stops after a certain number of iterations or after a certain error threshold is reached. Once the final fundamental matrix is determined, the
outliers are removed from the set of matched points and the remaining points can be used for the pose estimation. For this application, the resulting fundamental matrix is not needed. The reason for using RANSAC was simply to help eliminate outliers. Figure 3-9 shows an initial matched image pair with many outliers. Figure 3-10 shows the final result after running RANSAC.

Figure 3-9: Initial Matched Feature Points

Figure 3-10: Matched Feature Points after RANSAC with Homography

There is a chance that some incorrect features still remain after this step. Because the fundamental matrix multiplied by one point defines a line in the other image, if an erroneous
matched point happens to lie upon that epipolar line it will be considered a valid point after performing the RANSAC. An example of this can be seen in Figure 3-11, where the outlier clearly lies on the epipolar line of the point in the other image. With the addition of 3D information, during pose estimation we can further filter out these erroneous points to increase our accuracy.

![Figure 3-11: False Inlier after Fundamental Matrix RANSAC](image)

3.5 Pose Calculation

Once the majority of outliers are gone, the algorithm proceeds with finding the smartphone pose. Using the 2D to 3D point correspondences along with the camera’s intrinsic
parameters, the smart phone camera pose can be found with respect to the defined map coordinates. Several methods exist with different levels of speed, ease of implementation, and accuracy for solving for the pose given a set of correspondences. Several methods focus on using only 3 or 4 points for pose estimation, but more points results in greater accuracy when the points are found in a noisy environment. For ease of implementation and accuracy, the pose is estimated using the direct linear transformation (DLT) [31] for the initial starting point further optimized by the Levenberg–Marquardt algorithm (LMA) [32]. The LMA requires an initial starting point in order to converge on a minimized optimal solution. The DLT provides the starting point while the LMA minimizes the geometric error of the pose estimate. Using 6 or more points for the DLT results in an overdetermined system, so as before, singular value decomposition is used to solve the DLT [33].

All data points are represented by homogeneous coordinates. The 2D image points \( p_i = (x_i, y_i, w_i)^T \) relate to the 3D points \( P_i = (X_i, Y_i, Z_i, W_i)^T \) by

\[
p_i \sim K[R \mid t]P_i.
\]

(3-3)

The rotation and translation matrix \([R \mid t]\), or the extrinsic camera parameter matrix, bring the 3D points into the camera’s coordinate system, while the intrinsic parameter matrix \( K \) projects the points onto the image plane. This relationship between the 2D and 3D points is demonstrated in Figure 3-12.
Due to using homogeneous points the two sides of Equation 3-3 are equal to a scale factor. Combining the intrinsic and extrinsic matrices \( H = K[R|t] \), the relationship between our points can be expressed as \( p_i \times H p_i = 0 \). By rewriting \( H \) as its three transposed row vectors \( H = [h^{1T}, h^{2T}, h^{3T}]^T \) the cross product is expanded out to obtain

\[
p_i \times H p_i = \begin{pmatrix}
x_i h^{3T} P_i - w_i h^{2T} P_i \\
x_i h^{1T} P_i - w_i h^{3T} P_i \\
x_i h^{2T} P_i - w_i h^{1T} P_i
\end{pmatrix}.
\] (3-4)

Separating out the \( h^i \) vectors on the right side of the equation and leaving the points multiplied by the projection elements on the left results in the matrix multiplication

Figure 3-12: Point Correspondence Relation
Because the left matrix is linearly dependent, one equation can be removed leaving

\[
\begin{pmatrix}
0^T & -w_i^3P_i^T & y_i^3P_i^T \\
w_i^2P_i^T & 0^T & -x_i^2P_i^T \\
-w_i^1P_i^T & x_i^1P_i^T & 0^T
\end{pmatrix}
\begin{pmatrix}
h^1 \\
h^2 \\
h^3
\end{pmatrix} = 0. \quad (3-5)
\]

Equation 3-6 is for any given 2D to 3D point correspondence. Stacking up 6 different correspondences on the left side of Equation 3-6 creates the overdetermined equation. Calling this stacked matrix \( A \) the norm of \( \|Ah\| \) is minimized with the constraint that \( \|h\| = 1 \) by using the SVD in the same manner as discussed in section 3.4.3. After using the SVD to find an estimate of \( H \), the components of \( R \) and \( t \) are then easily extracted from the solution after multiplying \( K^{-1} \) on the left side.

Once the initial pose is calculated using the DLT, the LMA refines the pose while minimizing the reprojection error. LMA uses a method of gradient descent combined with Gauss-Newton iteration to minimize the sum of the squares of the errors between the data points and the function. The LMA takes the initial pose as the starting point for the algorithm.

Although the algorithm can take into account all points to solve for the pose, there may still be erroneous points so RANSAC is once again used to get rid of the few remaining outliers. Six points are chosen for the random sample on which to run the pose estimate. Using the resulting rotation and translation vector, the known 3D points are reprojected onto the image plane and the distance between the feature point and the projected point are measured for the error figure. See Figure 3-13.
Figure 3-13: Reprojection Outlier

If the error is above a threshold these points are discarded as outliers and removed from the inliers set. The purple line in Figure 3-13 represents a large error that exceeds the set threshold. The remaining points fall within acceptable limits. With the remaining inliers the algorithm is run again to refine the pose and measure the total reprojected error. The pose and error are saved and the algorithm continues the RANSAC iterations until the most accurate pose estimate is found. Figure 3-14 and Figure 3-15 show the final result of the pose estimation algorithm (location and orientation) overlaid on the phone image and the floor plan. If the pose estimate is extremely different than the previous known position, it is discarded as an error and the next phone image is used.
Figure 3-14: Final Pose Estimate

Figure 3-15: Floor Plan with Pose
4 TESTING AND RESULTS

4.1 Test Objectives

The main objective of the test was to determine the ability and robustness of the vision algorithms to accurately find the pose of the camera in a university building. The frequency of map images was also tested to determine the optimal spacing distance between images. Many text-to-speech, voice recognition and route planning algorithms have been tested and used in commercial applications. Testing and comparison of these different methods and algorithms will not be made. For the purpose of this thesis, we are evaluating the feasibility of using computer vision to analyze smart phone images to determine a user’s location and orientation in an indoor scenario from an image-based mapping system. Therefore, the testing focuses on the vision algorithms for indoor pose estimation.

4.2 Location Information and Image Map

The test was setup in the hallways of the 4th floor of the Brigham Young University Clyde building. Pictures were taken of the hallways in 10 foot increments with a point and shoot camera. The close image spacing allowed for testing several different increments for map images. These images were then run through the SURF algorithm to find the 20 best features in each image. These features were in turn measured by hand in the hallways in order to find the three-dimensional coordinates with respect to a predetermined world coordinate origin. Future measurements for implementation of the system would be done in conjunction with capturing the
images by a robot running SLAM for instance, but to establish ground truth the features were measured by hand. For ease of measurement and analysis the North-West corner of the hallway was chosen as the origin as shown in Figure 4-1. The feature keypoints along with their 3D coordinates and feature descriptors were stored in YML format as the image map dataset.

![Clyde Building 4th Floor with Local Coordinate System](image)

**Figure 4-1: Clyde Building 4th Floor with Local Coordinate System**

### 4.3 Testing Hardware

The smart phone used for the experiment is an LG Optimus S. The Optimus S is a lower tier smart phone to show that higher end phones are not required for the use of this system. The phone only uses a single core 600MHz processor running Android 2.2 Froyo. The computer used for the server is an AMD quad core running at 3.4GHz with 8GB of RAM and a 1TB 7200 rpm hard drive running Windows 7. The program was created with Microsoft Visual Studio 2010.
4.4 **Test Procedure**

Calibration pictures were taken on the smartphone camera to find the camera's intrinsic parameters and distortion coefficients. These parameters were saved to be used by the pose estimation algorithm. Pictures were then taken with a smartphone camera while walking down the mapped-out hallways and then fed to the server for processing. The image size of both the map and camera pictures was set at 640x480.

4.5 **Test Results**

4.5.1 **Image Frequency**

To begin our testing we checked usability at different distances from a map image. Map images too far away proved difficult in feature matching even with the scale space at several octaves. Most features show up on the walls which are quite visible up close, but at great distances are undetectable due to the skew created by the narrow angle of the camera with the feature point. Even at some distances where a match might be made despite the long distance to the map image, the pose estimation suffered from the increased error due to pixelation. We found intervals of 10 feet to be a good compromise of accuracy and speed.

4.5.2 **Algorithm Robustness**

In the Clyde building lighting caused problems with feature matching since the floor is very reflective. The use of the fundamental matrix with RANSAC proved robust enough to eliminate most of these erroneous features. For the images that still had outliers after the previous step, the remaining outliers were eliminated by the second RANSAC stage while calculating the pose.
There were a couple of phone pictures taken that were unable to map to an image. However, another picture in the near vicinity was able to match to the same location that the other image was unable to map to. These images that fail to map are not very frequent, around 1 in 15 or 20 images, and do not negatively affect pose estimation. When an image is unable to find a match in the near vicinity of the previous image it is discarded and the next phone image is used. During testing this next image was always able to find a match.

4.5.3 **Pose Estimation Accuracy**

Overall the Seeing Eye Phone system was able to accurately localize the phone within the mapped out hallways. Several images of the pose estimation are shown in Figure 4-2. The images show the picture taken by the smart phone with the map position overlaid for reference. Positions were tested where the user was walking in the middle of the hall as well as walking at each edge of the hall. The pictures on the edge of the hall effectively rendered most of the features on that side of the hall useless due to the angle. With the frequency of images used, the system was still able to handle the different angles without much difficulty. Pictures that were taken at extreme angles in relation to the map image did have problems finding matches. An example would be facing a wall in the hallway instead of facing the open hallway. In order to find a match, the user would need to turn enough to look down the hallway. Pictures were also used that were taken closer to map images and farther away in order to test the scale invariance of the system. As expected, some smart phone images would map to multiple map images since many of the hallway images overlap each other. Usually the closer of the images will have the strongest and most matches, but if not, matching to the next image further down is still valid for pose estimation.
The accuracy with only 20 points used in the map images is very good as shown in Table 4-1. The estimated position was almost always within a few feet of the measured position. Accuracy to a few feet is good enough for indoor localization as higher resolution is not needed. Two or three feet of potential inaccuracy will still allow a person to reach a doorway or other desired location. This vision based method has more detail in positional resolution than other methods such as periodic RFID tags or signs. It is also much more consistent than Wi-Fi or other RF based location methods that can vary widely with temperature, people moving around, and obstructions that affect the signal strength.

Figure 4-2: Map Progression Images
The algorithm is fast enough for real time use with a small number of images to search. Because the search looks in the area around the last known position only 5 to 15 images may need to be searched. The search for the initial starting location may take longer, but it is not unreasonable in time. It can be likened to the time it takes for GPS systems to initially acquire satellite positions before being able to find its position. The system can start finding the current location while the user gives the command for the desired destination. GPS can also be used when first entering a building to determine the entrance to begin the image search on and increase the initial location detection speed.

<table>
<thead>
<tr>
<th>Real (x,y) (m)</th>
<th>Estimated (x,y) (m)</th>
<th>Difference (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0.3048</td>
<td>1.2192</td>
<td>0.283464</td>
</tr>
<tr>
<td>3.048</td>
<td>1.8288</td>
<td>2.974848</td>
</tr>
<tr>
<td>7.62</td>
<td>0.6096</td>
<td>7.22376</td>
</tr>
<tr>
<td>10.668</td>
<td>0.9144</td>
<td>10.63752</td>
</tr>
<tr>
<td>12.8016</td>
<td>2.4384</td>
<td>12.3444</td>
</tr>
<tr>
<td>15.24</td>
<td>1.524</td>
<td>15.60576</td>
</tr>
<tr>
<td>24.384</td>
<td>1.8288</td>
<td>24.56688</td>
</tr>
<tr>
<td>27.432</td>
<td>2.1336</td>
<td>28.98648</td>
</tr>
<tr>
<td>30.48</td>
<td>0.9144</td>
<td>31.18104</td>
</tr>
<tr>
<td>35.6616</td>
<td>1.2192</td>
<td>35.32632</td>
</tr>
<tr>
<td>40.2336</td>
<td>1.524</td>
<td>40.05072</td>
</tr>
<tr>
<td>45.1104</td>
<td>0.9144</td>
<td>43.25112</td>
</tr>
<tr>
<td>49.6824</td>
<td>1.2192</td>
<td>49.46904</td>
</tr>
<tr>
<td>29.2608</td>
<td>6.096</td>
<td>29.01696</td>
</tr>
<tr>
<td>25.908</td>
<td>5.4864</td>
<td>26.85288</td>
</tr>
</tbody>
</table>
4.5.4 **Insufficient Features**

This method of using existing features assumes there are sufficient features available to recognize locations. Some areas have null spots where there are not enough features to make a map image such as end of some hallways or when facing a corner, as can be seen in Figure 4-3. Some of these areas are just intermittent positions which do not require a good pose estimate for wayfinding. When turning for instance, it is understood to turn 90 degrees to a new hallway or when entering a door. Matching an image halfway through the turn is unnecessary. Once the turn is completed the matching can continue using the new hallway’s features. Future research may be able to utilize the phone’s extra sensors to help handle areas that do not contain enough features to find the pose.

![Figure 4-3: Insufficient Unique Features](image-url)
5 CONCLUSION

5.1 Summary

The tested prototype vision system proved to be a reliable method for indoor pose estimation for navigation. The system was able to accurately and quickly locate the smart phone position and orientation within the building. It satisfies our main requirements of being inexpensive for the user and infrastructure, portable, not cumbersome, discrete and having accurate pose estimation. Meeting these requirements results in a practical navigation system for the average visually impaired user. The system is low cost both for the user and infrastructure. The vast majority of people use smart phones. The system requires a simple download of the program onto their existing smart phone. No additional sensors or equipment need to be purchased. Because it uses Wi-Fi, other handheld devices may be used that don’t require data plans for those that do now own smart phones. The infrastructure for the system requires only a computer for the server and wireless routers which many buildings already have in place. The system is socially acceptable in that it does not draw unwanted attention to the visually impaired user by using odd gadgets for sensing devises. It is also small, portable, and easy to use. The user can maintain complete mobility and function of his hands for other uses. Lastly, the hearing of the visually impaired remains unhampered because the instructions are given periodically. Overall, this system benefits the visually impaired user for navigating unknown indoor locations.
5.2 Contributions

This thesis makes the following contributions:

1. Guidance system
   a. Low cost – The designed system is inexpensive for both the user and the infrastructure for implementing the system.
   b. Discrete – The Seeing Eye Phone does not draw unwanted attention to the visually impaired user.
   c. Portable – The Seeing Eye Phone is easy to carry around.
   d. Easy to use – The Seeing Eye Phone does not hamper movement or hearing and allows for hands free operation.

2. Software algorithms
   a. Accurate single camera pose estimation – Algorithm is able to accurately find pose using a single camera in an indoor environment while eliminating outliers to improve results.

3. Experimental results
   a. Pose calculation – The system was able to accurately and quickly determine the pose of the smart phone.

5.3 Future Work

Efficient and accurate image map making are a key factor to the success of the Seeing Eye Phone system. A possible solution to fast and efficient map making would be to use a robot running SLAM with stereo vision. In order to improve accuracy, visual location markers can be placed in key corners of buildings and their real world position can be entered manually. These
markers will help increase accuracy for all other relative measurements. Map organization is another aspect that can easily be improved upon and have a large impact on searching speed.

A lot of future research can be done in the algorithms for the system, such as improved feature detectors and path prediction. Feature detectors taking advantage of color can provide more robust features, and even find features undetectable by grayscale feature detectors. These color features may help in areas of buildings where there are not many features to detect. Other algorithms can be developed to predict the movement of the user. The predictions can then be used to more intelligently decide which image to try to match to next and thus greatly increase processing speed.

Since only images are being processed, interfacing with dedicated hardware and sensors as many other systems do is unnecessary; however, other areas of further research may include taking advantage of extra smart phone sensors when they are available. Utilizing a phone’s built in sensors such as the compass, gyroscopes and accelerometers can give estimates on movement and add extra validation to the calculated results. They may be used to help with intermittent locations that lack sufficient features for image matching and may also be used for next image prediction to decrease search time.
REFERENCES


