Palestine Polytechnic University
Collage of Administrative Sciences and Informatics
Department of Information Technology

Visual Localization Aid for the Blind

Project team:

Fatinah Khader Sinnoqrot

Iman Fayez Younes

Supervisor:

Dr. Hashem Tamimi

A final project submitted in partial fulfillment of the requirements for the degree of B.Sc. in Information Technology

2007

Abstract

This project describes a computer-based location determination system for helping blind or visual impaired persons to navigate easily in an indoor environment. Unlike the existing systems, the proposed system requires only a light weight personal computer and a small camera fixed on their walking cane. The camera captures images from the place where the blind exists and sends it to the computer where it will be analyzed. As a response, a voice message is played to the person explaining where she/he is. The method adopted here is based on a Content-Based Image Retrieval (CBIR) approach which involves retrieving images from a large image database that are similar to a query image based on features taken from the content of the image. Image-based features are extracted using the well known SIFT algorithm. These extracted features are used as the basis for the similarity check between images. This approach is combined with Markov localization to enhance the localization rate. Experimental results in a public indoor environment show that the proposed approach can lead to a high localization rate.

Abstract

This project describes a computer-based location determination system for helping blind or visual impaired persons to navigate easily in an indoor environment. Unlike the existing systems, the proposed system requires only a light weight personal computer and a small camera fixed on their walking cane. The camera captures images from the place where the blind exists and sends it to the computer where it will be analyzed. As a response, a voice message is played to the person explaining where she/he is. The method adopted here is based on a Content-Based Image Retrieval (CBIR) approach which involves retrieving images from a large image database that are similar to a query image based on features taken from the content of the image. Image-based features are extracted using the well known SIFT algorithm. These extracted features are used as the basis for the similarity check between images. This approach is combined with Markov localization to enhance the localization rate. Experimental results in a public indoor environment show that the proposed approach can lead to a high localization rate.

Dedication

To all those who love Palestine and those who died for it or still waiting…

To our beloved parents whose heart beat with every notion flashed in our mind and every single letter we wrote down...

To our supervisor Dr. Hashem Tamimi for his intelligent editorial supervision and attentive production; for teaching us how to be creative…

To the candle which lighted and guide us through darkness and frustrating moments; To our siblings who inspired and propped…

And to all who are not cited and whose names may be inadvertently not mentioned…

Fatinah and Iman

Acknowledgment

At this stage when the project came into light, many names in mind, supervisor Dr. Hashem Tamimi for his time, efforts and his intelligent ideas.

We will never forget our university, information technology department and FFKITCE center for their support.

Many thanks to all our friends and teachers for the times and efforts they spend in this project.

Table of content

List of Tables

List of Figures

Declaration

The researchers their by declare that this project has been written by them, and all material not their own work has been identified by real and true references, and they are responsible for anything written inside this project.

Fatinah Khader Sinnoqrot
Eman Fayez Younes

CHAPTER ONE

INTRODUCTION

1.1 Overview

1.2 Project Hypothesis

1.3 The Objectives

1.4 Project problem

1.5 Project domain

1.6 The project pioneer points

1.1 Overview

Blind people are at tremendous disadvantage when they arrive new indoor environment like university, school, supermarket, and others [1]. They should have freedom and ability to locate where they are. Along the path to a certain destination, blind people need exact information about locations, dangers, and appropriate paths [2].

From this point we aim to provide a facility to make the mobility idea in an indoor environment easier using computer vision alone. In addition the proposed system does not require any artificial markers in the environment and relies on the natural decorations of the existing furniture which makes the establishment of such system inexpensive.

In our system we are focusing on indoor environments. The reason for this is that localization within indoor environments is more difficult, since global positioning system (GPS)[1] signals are normally unavailable. So, if we can solve the localization problem indoors, it should also be possible to solve it outdoors by combining our system with the GPS. Our project is the first that uses only image processing in helping blind people to navigate in an indoor environment.

1.2 Project hypothesis

- The study assumes that all hardware parts are functioning without any problem or lack of power.
- The study presumes that there are enough clues in the environment which will help the software to distinguish one place from another.
- The project will be tested on a static environment, where no moving persons exist.
- The study presumes that the database will be updated whenever there are changes in the building.

[1] (GPS) is a satellite-based navigation system made up of a network of 24 medium earth orbit satellites that transmit precise microwave signals. The system enables a GPS receiver to determine its location, speed and direction. GPS developed by the U.S. Department of Defense for military applications, but in 1980s, the government made the system available for civilian use.

1.3 Project objectives

The minimum objectives which the system will cover are addressed by the following characteristics:

1. *Size and space for project environment*: One floor in a building (indoor environment)
2. *Type of interaction (voice, text, or GUI)*: The system will interact with the blind user through voice messages which considered to be best way for interaction, since that the blind user can not benefit from Graphical User Interface (GUI).
3. *Ease of use*: The equipments that are necessary to make this project works effectively are easy to use and understand regardless of the users experience and knowledge. Also, it is useful and marketable to people who have diverse financial abilities and wide range of individual preferences and abilities.
4. *Physical effort*: The equipment that must be used to implement this project could be used efficiently and comfortably and with minimum of fatigue.
5. *Robustness and reliability*: We will try to minimize the number of negative messages that may back to the blind user by avoiding the software failure (error rate) and guide the user with instructions through voice messages.

1.4 Project problems

The problems in the user location detection for the blind are complicated by the challenges of both the software and the blind person. Our system addresses the following challenges:

<u>*The software:*</u>

1. The Building:
   - It is hard to determine where the person exactly exists in buildings that have the same design for each floor or side. This problem occurs particularly when there are few decorations in the indoor environment.
   - No GPS or sensor help, since GPS signals are normally unavailable at indoor environment.

- This service is limited within the map of the specific location.

2. The Speed:
   One challenge is to provide a fast feedback to the blind person because of the processing time that the software needs to process the inputs.

The user (blind person):

Shooting good picture is very hard in many cases due to:

- The quality of the captured image would affect the final result.
- Lightning effects.
- Camera quality.
- Camera orientation.

1.5 Project domain

This project is oriented for blind or partially sighted people, who are willing to navigate in an indoor environment. It could be implemented in any indoor and featured environment. For simulation purpose the project was applied at Friends of Fawzi Kawash Information Technology Center of Excellence (FFKITCE)1.

1.6 The project pioneer points

- The project under consideration is the first that uses *image processing only* in analyzing input pictures without using GPS or sensors, so installing the project in a new indoor environment does not require any especial infrastructure like other projects which use RFID tags and other sensors.
- This project is the first that measures the result in a quantitative approach.

[1] FFKITCE is one of few purpose-built IT development-training centers in Palestine; it was established at Palestine Polytechnic University-Hebron in 2005. It aims to encourage and support innovative IT research and projects that have significant potential value, and to provide exemplary dissemination, training and support in IT for the University, the region and the country. The IT Business Incubation Service at the Center will enable people with good ideas in IT to find a place to work and develop their ideas.

- This service is limited within the map of the specific location.

2. The Speed:

   One challenge is to provide a fast feedback to the blind person because of the processing time that the software needs to process the inputs.

The user (blind person):

Shooting good picture is very hard in many cases due to:

- The quality of the captured image would affect the final result.
- Lightning effects.
- Camera quality.
- Camera orientation.

1.5 Project domain

This project is oriented for blind or partially sighted people, who are willing to navigate in an indoor environment. It could be implemented in any indoor and featured environment. For simulation purpose the project was applied at Friends of Fawzi Kawash Information Technology Center of Excellence (FFKITCE)1.

1.6 The project pioneer points

- The project under consideration is the first that uses *image processing only* in analyzing input pictures without using GPS or sensors, so installing the project in a new indoor environment does not require any especial infrastructure like other projects which use RFID tags and other sensors.
- This project is the first that measures the result in a quantitative approach.

[1] FFKITCE is one of few purpose-built IT development-training centers in Palestine; it was established at Palestine Polytechnic University-Hebron in 2005. It aims to encourage and support innovative IT research and projects that have significant potential value, and to provide exemplary dissemination, training and support in IT for the University, the region and the country. The IT Business Incubation Service at the Center will enable people with good ideas in IT to find a place to work and develop their ideas.

CHAPTER TWO

REVIEW

2.1 Introduction

This field, navigation and localization assistance for the blind or visual impaired persons has become an active research area. There are a lot of researches made in this domain that can be classified into a few general categories. In this chapter we will review the previous works that were made for both indoor and out door navigation aid for the blind.

2.2 Indoor navigation

Many researches have focused on the indoor navigation aid for the blind because it is hard for both, the blind and the developer. For the blind, indoor environments are new and unknown most of the time because they vary very much in design and space. For the developer, it is a hard job to develop systems that can aid blind persons in navigating efficiently in indoor environment.

Researches done in this domain could be categorized into two categories, one category can be termed as the usage of sensors only; the other is researches depend on using sensors in a line with intelligent image processing.

2.2.1 Blind navigation aid using sensors only

There are a lot of researches that have been dedicated in this domain like Robotic guide which called RoboCart from Kulyukin [3]. This system aims to aid the blind and visually impaired shoppers to find their way around grocery store. For localization, RoboCart relies on (RFID) tags[1] which are deployed at various locations in the store. For navigation, RoboCart relies on laser range finding. Similar to this approach is the system that was developed by Willis and Hilal [1], who find a solution to the problem of way finding for the blind user in a university campus by using RFID information grid and wearable computing system. Each RFID tag is programmed

---

[1] Radio-frequency identification (RFID) is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders. An RFID tag is an object that can be attached to or incorporated into a product, animal, or person for the purpose of identification using radio waves.

upon installation with spatial coordinates and information describing the surrounding. This allows for self describing with no dependency on centralized database or wireless infrastructure for communications. Sonnenblick [5] evaluated and implemented a large-scale indoor navigation system for blind individuals at Jerusalem Center for Multi-Handicapped Blind Children. It relies on specially installed safely infrared beacons for orienting blind users and on a custom-built end-user device.

Although the mentioned researches work very well in new, unknown indoor environments they still rely on certain provided infrastructure for example RFID tags and infrared beacons which must be installed in many locations at the indoor environment because it is one dimensional signals.

Another research is the NavBelt from Shoval et al. [4], which produces a 120 degree wide view ahead from the current location of the blind. This information is translated into a stereophonic acoustical sound that allows the user to notice if a certain direction is blocked. This research dedicated in obstacles avoidance information rather than orientation and localization information.

2.2.2 Blind navigation aid using sensors in a line with intelligent image processing

Many researches dedicated in using sensors in a line with intelligent image processing to provide a solution for indoor navigation. Hub et al. [6] proposed a new system that assists blind users in orienting themselves in indoor environment using a sensor module that can be handled like a flashlight by a blind user and can be used for searching tasks within the three-dimensional environment. By pressing keys, the user's inquiries concerning their environment are acoustically answered through a text-to-speech engine. The system's limitation is that it is necessary to hit an object precisely using a picking ray within a 3D model in order to allow proper object identification [2]. In a later work Hub et al, developed a new prototype [2] which includes an option to receive augmented navigation hints automatically just by walking in virtual corresponding navigating areas, this orientation assistant consists of two parts – a sensor module and a small portable computer that can be carried in a small backpack. The portable computer has access to 3D environment models. The sensor module itself consists of direction sensors and a stereo camera. The orientation assistant works as follows: The location of the user respectively the orientation assistant within buildings can be determined over a conventional WiFi[1] system via the differences in the signal strength transmitted by the access points [2]. Hub et al. also presented a new method for interactive tracking of various types of moving objects [7]. For example the state of fixed or moving doors can be recognized by comparing the distance sensor data and 3D model. For the identification and model based tracking of free movable objects, like chairs. Further, using a common face detection algorithm, the system could inform the user of the presence of people. And enable the localization of a real person based on interactive tracking of virtual models of humans.

Although these systems work very well in new, unknown indoor environments, they still need certain provided infrastructure. And for location detection they still rely on the use of sensors.

In addition to the previous work Hub et al. proposed another prototype of a navigation assistant system for the blind that consists of a sensor module and a portable computer [8]. This prototype does not require infrastructure installation. Once the initial position is entered, e.g. the main entrance, it determines the user's current position using a 3D compass, 3D gyroscope and an acceleration-based step recognition algorithm. Related positional data, such as room numbers, architectural details, or distance covered along a specific route can be presented acoustically for blind or visually for sighted people, on a touch screen integrated into a small portable tablet PC. By moving one's finger on the map, the user gets a spatial impression of the current environment, or of any other area where adequate digital mapping has been done. Stored text information can be accessed via tactile-acoustical swhiches, allowing the user to address navigational tasks and virtually explore alternative routes.

Although the system does not require infrastructure installation, it depends on the initial value that must be entered first which in many cases hard job to do because the indoor environments are new most of the time for the blind.

---

1 (Wi-Fi) Wireless Fidelity is a brand originally licensed by the Wi-Fi Alliance to describe the embedded technology of wireless local area networks (WLAN) based on the IEEE 802.11 specifications. Wi-Fi was developed to be used for mobile computing devices, such as laptops in LANs, but is now increasingly used for more services, including Internet, and basic connectivity of consumer electronics such as televisions, DVD players, and digital cameras

2.3 Outdoor navigation

For outdoor navigation several systems have been proposed. Although they share common characteristics with indoor navigation systems; we will not go into further details because they depend on technologies that are not available at indoor navigation such as using GPS for tracking. Many researches also mentions the possibility of emerging outdoor navigation with indoor navigation, for example the system that is proposed by May [9] for Seamless outdoor\indoor navigation for blind and visually impaired individuals, which benefits from The Global Positioning System (GPS) and related indoor navigation technologies, combined with ever-growing location databases, present the opportunity for the blind to have an audible representation of the environment indoors and outdoors.

2.3 Outdoor navigation

For outdoor navigation several systems have been proposed. Although they share common characteristics with indoor navigation systems; we will not go into further details because they depend on technologies that are not available at indoor navigation such as using GPS for tracking. Many researches also mentions the possibility of emerging outdoor navigation with indoor navigation, for example the system that is proposed by May [9] for Seamless outdoor\indoor navigation for blind and visually impaired individuals, which benefits from The Global Positioning System (GPS) and related indoor navigation technologies, combined with ever-growing location databases, present the opportunity for the blind to have an audible representation of the environment indoors and outdoors.

CHAPTER THREE

BACKGROUND

3.1 Introduction to Content Based Image Retrieval (CBIR).

3.2 Introduction to Scale Invariant Features Transforms (SIFT).

3.3 Using SIFT approach for image Matching.

3.1 Introduction to Content Based Image Retrieval (CBIR)

As the impact of computers on our lives is becoming more and more significant, much of the information, including pictures, is being digitized. This fact was the trigger of the need for techniques that enable us to access and retrieve pictures from huge image databases. As processors become increasingly powerful, and memories become increasingly cheaper, the employment of large image databases for a diversity of applications have now become realizable. Effective, efficient, and convenient access to a desired image from large image databases is now a necessity as the application of image retrieval attracting more and more users from various professional fields [12].

Research in image retrieval domain has, so far, been divided into two parts, those who concentrate on text-based image retrieval, which focuses on using words to retrieve images (e.g. title, subject heading, keyword, and caption), while the other concentrates on content-based image retrieval or CBIR, which focuses on the visual features of the image (e.g. colors, textures, etc) [10].

In many large image databases, traditional methods of image indexing, ranging from storing an image in the database and associating it with a keyword or number, have proven to be insufficient, and extremely time consuming because of the following reasons [11]:

- These systems require too much effort and time for manual image annotation; this problem becomes more sever as the image collections grow.
- The description of the image content is subjective to human perception; different of people may end up with different description for the content of the image in hand.
- Any image information that the annotator forgets, ignores or considers as unimportant at the time of annotation can not be retrieved later.

Another problem is that manual text annotation is valid only for the languages used for the purpose of annotation. Other people that do not have a background in the used language(s) are not able to use text based retrieval systems [12].

In CBIR each image that is stored in the database has its features which are extracted and compared to the features of the query image.

3.1.1 What is CBIR?

Content-based image retrieval (CBIR), also known content-based visual information retrieval (CBVIR) which is an application of computer vision to the image retrieval problem of searching for digital images in large databases [13].

Content-based means that the search will analyze the actual contents of the image, which might refer to color, shape, texture, or any other information that can be derived form the image itself [13].

CBIR involves two main steps as shown in Figure 3.1 [14]:

1. ***Feature extraction***: In which the image features are extracted to a distinguishable extent.
2. ***Matching:*** Which involves matching these features to yield a result that is visually similar.

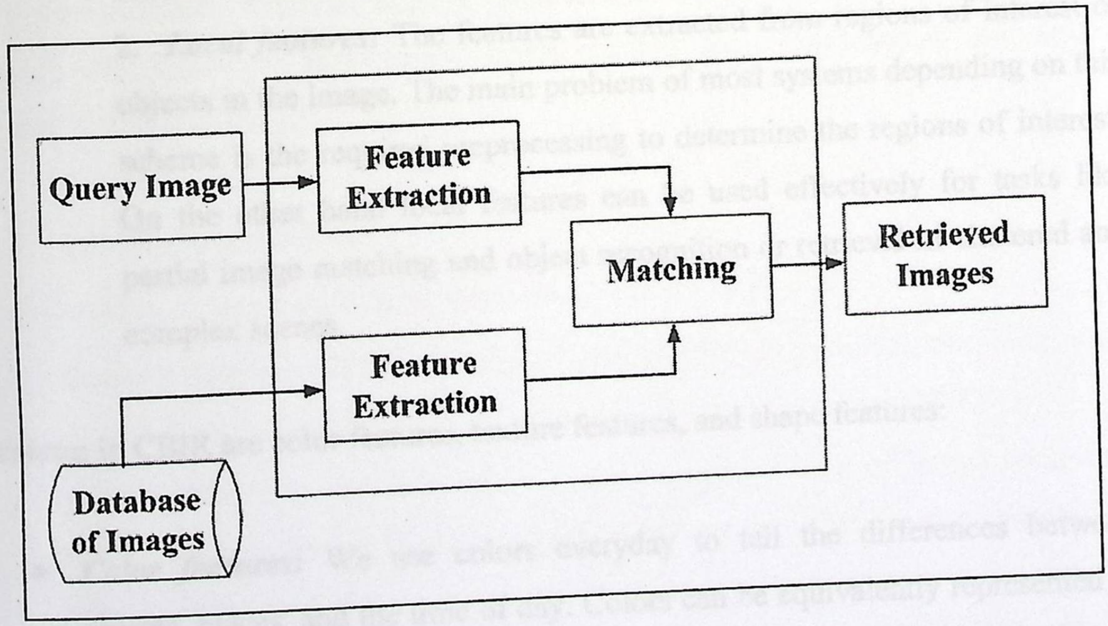


Figure 3.1 CBIR steps

3.1.2 CBIR features

Feature extraction is the core of the CBIR. Instead of using the whole image, only an expressive representation is extracted. The process of finding the expressive representation is known as feature extraction and the resulting representation is called the feature vector [14]. The features that are extracted from the picture should be small size, unique, stable, efficient to generate, and invariant to translation, rotation, scale, partial occlusion or changes, noise, and illumination. The task of finding good features that adequately represent an image is still a challenging task.

In general, image features can be either local or global [12]:

1. ***Global features:*** The features are extracted from the visual content of the entire image. These features have been used successfully for image retrieval. The easiest and most famous example is the global color histogram. The main problem of global features is that the resulting description can not differentiate between different image parts like the object of interest and the background. Therefore, they are usually not suitable for tasks like partial image matching and object recognition or retrieval in cluttered and complex scenes.
2. ***Local features:*** The features are extracted from regions of interest or objects in the image. The main problem of most systems depending on this scheme is the required preprocessing to determine the regions of interest. On the other hand local features can be used effectively for tasks like partial image matching and object recognition or retrieval in cluttered and complex scenes.

Features in CBIR are color features, texture features, and shape features:

- ***Color features:*** We use colors everyday to tell the differences between objects, places, and the time of day. Colors can be equivalently represented in different color spaces, which are [12]:
  - RGB (Red, Green, and Blue).
  - HSV (Hue, Saturation, and Value) or HSB (Hue, Saturation, and Brightness).

- CIE color spaces (the human perception of hue, saturation, and brightness).

The main method of representing color information of images in CBIR systems is through color histograms. A color histogram can be presented by bar graph, where each bar represents a particular color of the color space being used.

- ***Texture features*** [15]: Texture is that instinctive property of all surfaces that describes visual patterns, each having properties of homogeneity. It contains important information about the structural arrangement of the surface, such as; clouds, leaves, bricks, fabric, etc. It also describes the relationship of the surface to the surrounding environment.
- ***Shape features*** [16]: Shape may be defined as the characteristic surface configuration of an object; an outline or contour. It permits an object to be distinguished from its surroundings by its outline.

David Lowe [17] has introduced one of the most successful feature extraction of local features which he called SIFT (Scale Invariant Feature Transforms). The SIFT features are invariant against geometric transformation (e.g. rotation and scaling) and robust to photometric transformation (e.g. illumination, and small changes in viewpoint) [12].

3.1.3 CBIR applications

The following are some application domains of CBIR [12]:

1. ***General Purpose CBIR:*** The aim of general CBIR is to find similar images in a large diverse database such as image search engine based on CBIR.
2. ***Object Matching and Detection:*** This is a special case of image retrieval, where the principle is limited to finding images that contain some particular objects rather than object classes are considered, such as searching for images that contain Mazda 2007 car rather than searching for the general class cars. A good example that works under this domain is robot vision, where the robot is responsible of object identification and manipulation.

3.2 Introduction to Scale Invariant Feature Transforms (SIFT)

SIFT is an approach for extracting distinctive (local) features from images, that can be used to perform reliable matching between different views of an object or scene. It takes an image and transforms it into a large collection of local feature vectors extracted around "keypoints". Each of these features is not affected by many of the complications experienced in other methods, such as object scale changes (zoom), illumination, image noise, rotation, small changes in 3D viewpoint, and small distortion [18].

Figure 3.2 shows an example of matching produced by the SIFT matching algorithm. Features are extracted from each of the two images, and lines are drawn between features that are similar. In this example, many matches are found and only small fractions are incorrect.

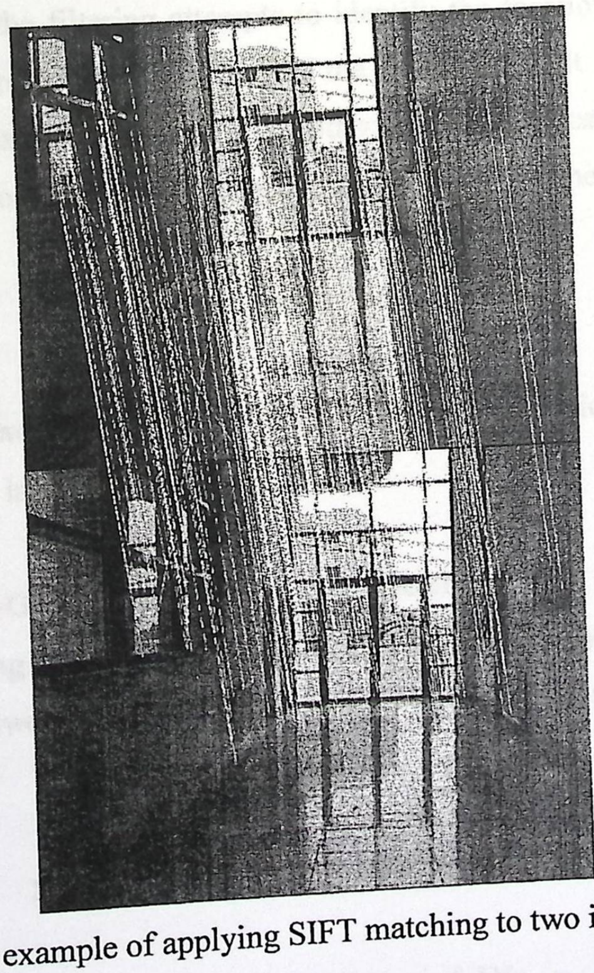


Figure 3. 2: An example of applying SIFT matching to two images.

3.2 Introduction to Scale Invariant Feature Transforms (SIFT)

SIFT is an approach for extracting distinctive (local) features from images, that can be used to perform reliable matching between different views of an object or scene. It takes an image and transforms it into a large collection of local feature vectors extracted around "keypoints". Each of these features is not affected by many of the complications experienced in other methods, such as object scale changes (zoom), illumination, image noise, rotation, small changes in 3D viewpoint, and small distortion [18].

Figure 3.2 shows an example of matching produced by the SIFT matching algorithm. Features are extracted from each of the two images, and lines are drawn between features that are similar. In this example, many matches are found and only small fractions are incorrect.

Figure 3. 2: An example of applying SIFT matching to two images.

Figure 3.3 shows the outline for SIFT algorithm which includes feature extraction and feature detection.

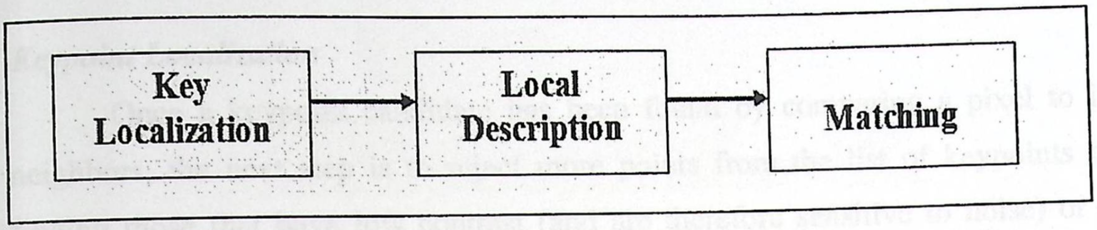


Figure 3.3: SIFT algorithm outline

3.2.1 Feature extraction in SIFT algorithm

To aid the extraction of these features, SIFT algorithm applies a four-stages filtering approach [17]:

1. *Scale-Space Extrema Detection*

This stage of the filtering attempts to identify the locations of keypoints that are identifiable from different views of the same object. It is implemented efficiently by using Gaussian function to identify potential interest points that are invariant to scale and orientation. The scale space is defined by the function:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad [17]$$

Where:

$G(x, y, \sigma) = 1 / (2\pi\sigma^2) \exp^{-(x2 + y2)/\sigma2}$ [18], *: is the convolution operation in x and y, and $I(x, y)$: is the input image.

Difference-of-Gaussians filter is one technique for detecting stable keypoints by locating scale-space extrema of $D(x, y, \sigma)$, by computing the difference between two Gaussian filtered images at scales σ and k times of σ, given by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad [17]$$

To detect the local maxima and minima of $D(x, y, \sigma)$ each point is compared with its 8 neighbors at the same scale, and its 9 neighbors up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

2. ***Keypoint Localization***

Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to reject more points from the list of keypoints by finding those that have low contrast (and are therefore sensitive to noise) or is poorly localized along an edge.

To remove keypoint with low contrast found in stage 1, if its value is below a specific threshold then this point is rejected and eliminated. To eliminate extrema based on poor localization it is noted that there is a large carvings across the edge but a small curving in the vertical direction in the difference-of-Gaussian function. If this difference is below the ratio of largest to smallest eigenvector, the keypoint is rejected.

3. ***Orientation Assignment***

This step aims to assign a consistent orientation to the keypoints based on local image properties. One or more orientations are assigned to each keypoint location based on local image gradient directions. The keypoint descriptor can then be represented relative to the assigned orientation, achieving invariance to rotation.

To determine the keypoint orientation:

- Gradient magnitude, *m*, orientation is computed in the neighborhood of the keypoint

$$m(x, y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad [17]$$

- Compute orientation, θ

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1,y) - L(x-1,y))) \quad [17]$$

- Locate the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation. Figure 3.4 illustrate an example for a canonical orientation assignment for each key location after computing the gradient magnitude and orientation at each keypoint.
- Fit a parabola to the 3 histogram values closest to each peak to determine the peaks position.

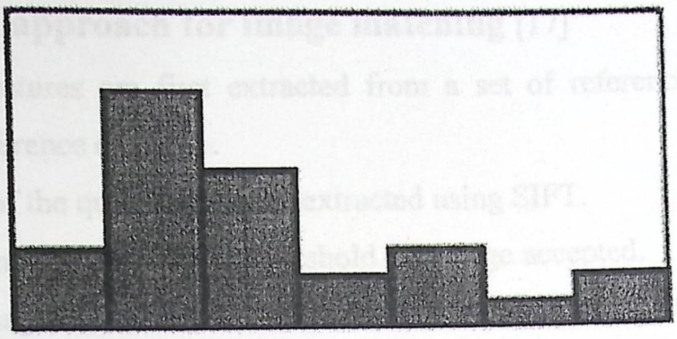


Figure 3.4 Canonical orientation assignments for each key location

4. ***Keypoint Descriptor***

The local gradient data, used above, is also used to create keypoint descriptors. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of 1.5 * keypoint scale (σ * 1.5). As shown in Figure 3.5 keypoint descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. These results in a feature vector containing 128 elements (4*4*8). This victor normalized to representation that allows for significant levels of local shape distortion and change in illumination.

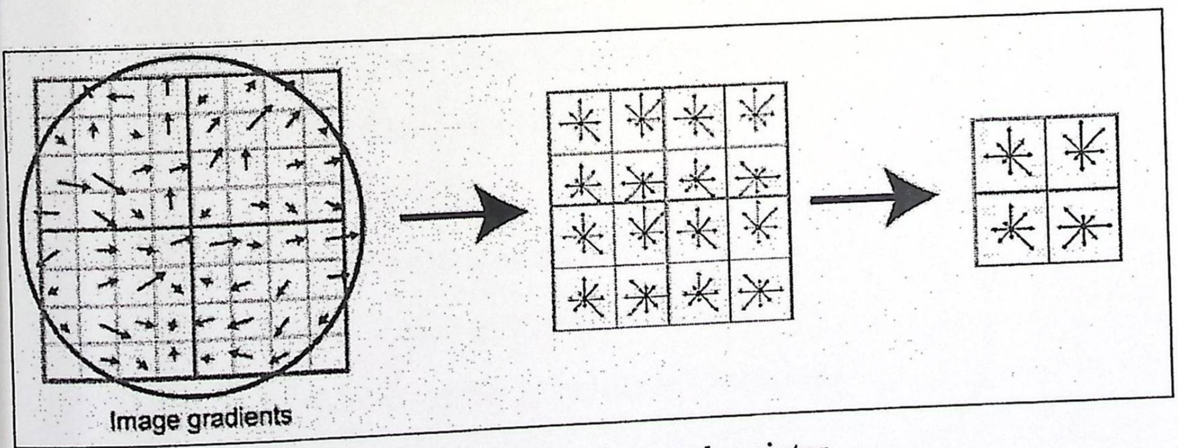


Figure 3.5 SIFT feature descriptor

3.2.2 Using SIFT approach for image matching [17]

1. SIFT local features are first extracted from a set of reference images and stored in a reference database.
2. The features of the query image are extracted using SIFT.
3. If number of matches above the threshold the image accepted.
4. The query image is matched by individually comparing each feature from the new image to this previous database and finding candidate matches.
5. The candidate images are sorted for each location depending on the maximum number of features.

CHAPTER FOUR

METHODOLOGY

4.1 Introduction

In this chapter the system will be analyzed and the relations between the system components will be cleared. Also the system will be divided into many sections, and the main operations will be cleared through analyzing the input and the output of every operation, with addition to the flow charts that describe the flow of the operations.

The project has two phases, a training phase, where the database is established, and a testing phase which simulates the execution of the system by the blind person. These two phases will be explained through out this chapter.

4.2 Project methodology description

These are the steps in which the system goes through:

1. As the Blind user turn the system on the camera will start capturing sequence of images.
2. The software converts the captured images to PGM image format (binary gray scale image format), and extracts their features (using SIFT algorithm).
3. The system will discard images with number of features less than a specified threshold (threshold is15 in our system).
4. The software compares the extracted features for each query image with local features in the reference database.
5. For each query image the system will find the highest number of matches in each location.
6. The system will find the probability of each location, which equals the highest number of matches for this location over the number of keypoints of the query image.

   P [Location] = best number of matches/ query image keypoints number
7. Then the system will find locations belief depending on Markov algorithm [19] which has two phases , as the following:
   a. *Prediction Phase* : where we use the motion model to predict the current position of the blind user in the form of predictive density for each location which is obtained by integration:

For each location do

$$P(L_t \backslash X^{t-1}) = \int P(L_t \backslash L^{t-1}) * PreBel(L_{t-1} \backslash X^{t-1}) dL_{t-1}$$

Where:

- $P(L_t \backslash X^{t-1})$ is predictive density.
- $P(L_t \backslash L^{t-1})$ is motion model which means that the current state L_t depends only on the previous state L^{t-1} (Markov) the values are obtained from the lookup table shown in table 4.3.
- $PreBel(L_{t-1} \backslash X^{t-1})$ is the previous location beliefe.

b. *Update Phase:* depends on sensor reading (camera images) and use measurement model to obtain Probability Density Function (PDF) $P(L_{t\backslash} X^t)$.

For each location do (Perception model)

$$P(L_{t\backslash} X^t) = P(L) * P(L_t \backslash X^{t-1}).$$

$$theta = theta + P(L_{t\backslash} X^t)$$

Where: P (L) is the location Probability.

For each location do (Normalizing)

$$P(L_{t\backslash} X^t) = P(L_{t\backslash} X^t) \backslash theta$$

$$Bel[Location] = P(L_{t\backslash} X^t)$$

8.Search for the location whose belief is higher than .80, and then play the voice message that is associated with this location. Otherwise go to step number 4.

4.3 Data Flow diagrams [1]

4.3.1 Context diagram

Figure 4.1 shows the context diagram for the Visual Localization Aid for the Blind (VLAB) system, the system interacts with two external entities which are the camera and the speaker. The camera is fixed on the blind cane captures sequence of images, and sends them to the system. The system will analyze the images to determine the current location and sends the appropriate voice message to the speaker to be played.

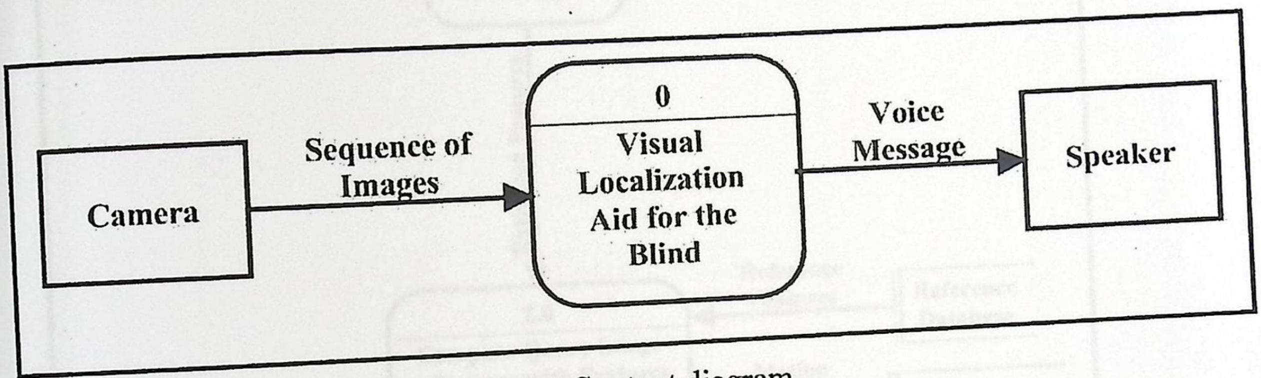


Figure 4.1 Context diagram

[1] The diagrams based on Gane and Sarson Symbols

4.3.2 Level One of Data Flow Diagram

Figure 4.2 shows level one of data flow diagram, which contains the main processes, databases, and the flow of data between them. The system will receive the sequence of images, extracts their features using SIFT algorithm, then the extracted features will be compared with reference database features to find the best number of matches for each location, then the locations belief is computed, depending on Markov algorithm [19], to determine the blind user location. Finally an appropriate voice message is played.

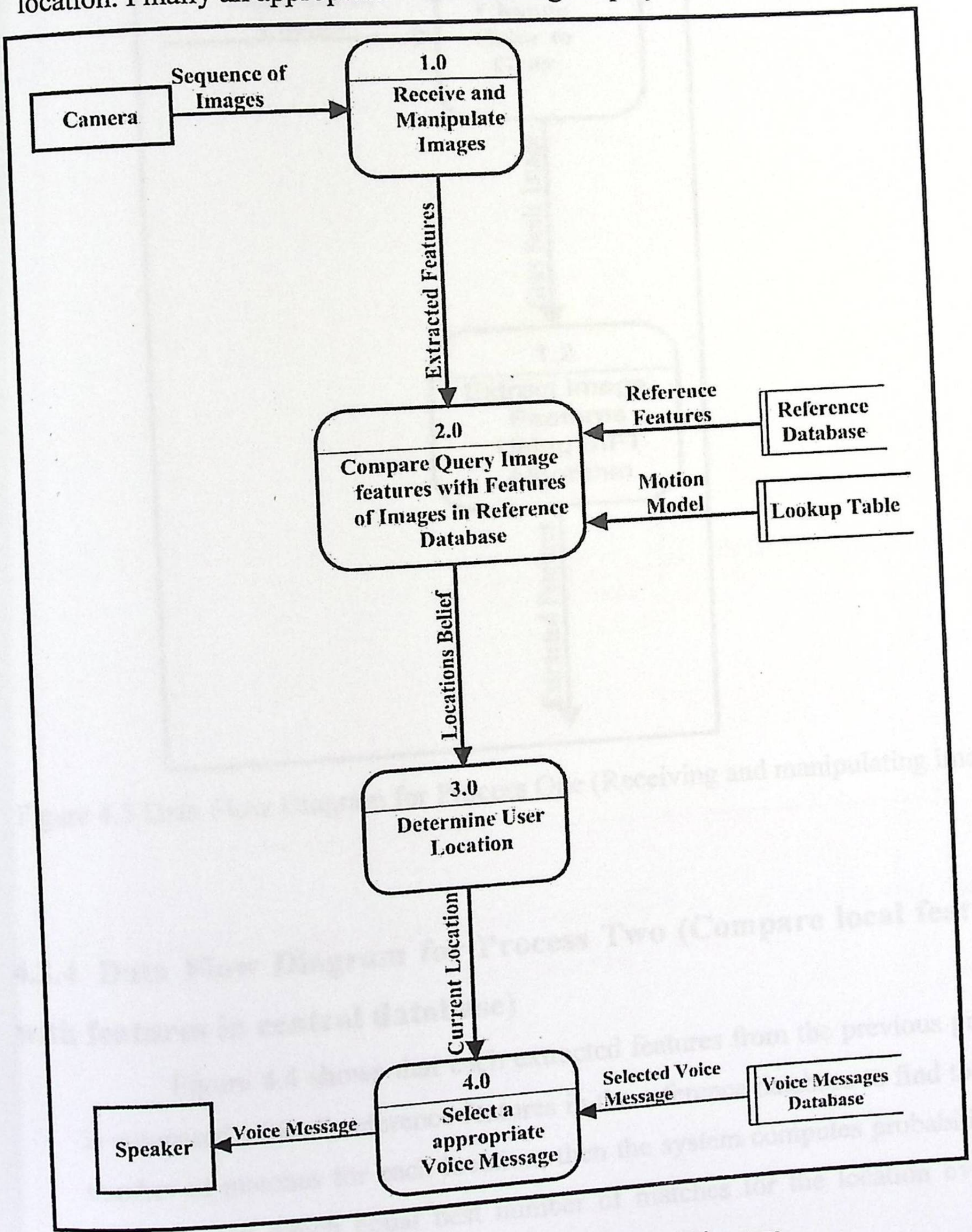


Figure 4.2 Level One of Data Flow Diagram

4.3.3 Data Flow Diagram for Process One (Receiving and manipulating new Images)

Figure 4.3 shows how each image is manipulated before extracting its features. Colored image is converted to gray scale (PGM format). Then SIFT receives the gray images and extracts their features. Extracted features are delivered to the next process.

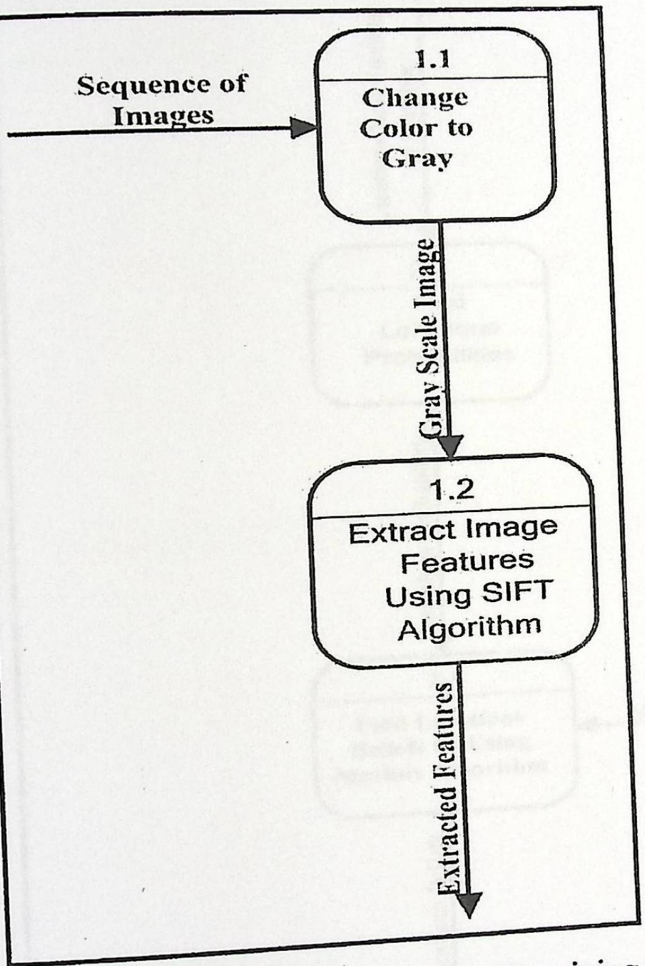


Figure 4.3 Data Flow Diagram for Process One (Receiving and manipulating image)

4.3.4 Data Flow Diagram for Process Two (Compare local features with features in central database)

Figure 4.4 shows that each extracted features from the previous process is compared with all reference features in the reference database to find the best number of matches for each location, then the system computes probability for each location which equal best number of matches for the location over the number of query image keypoints. Then locations probabilities and motion model readings are used in Markov algorithm [19] to find locations Beliefs.

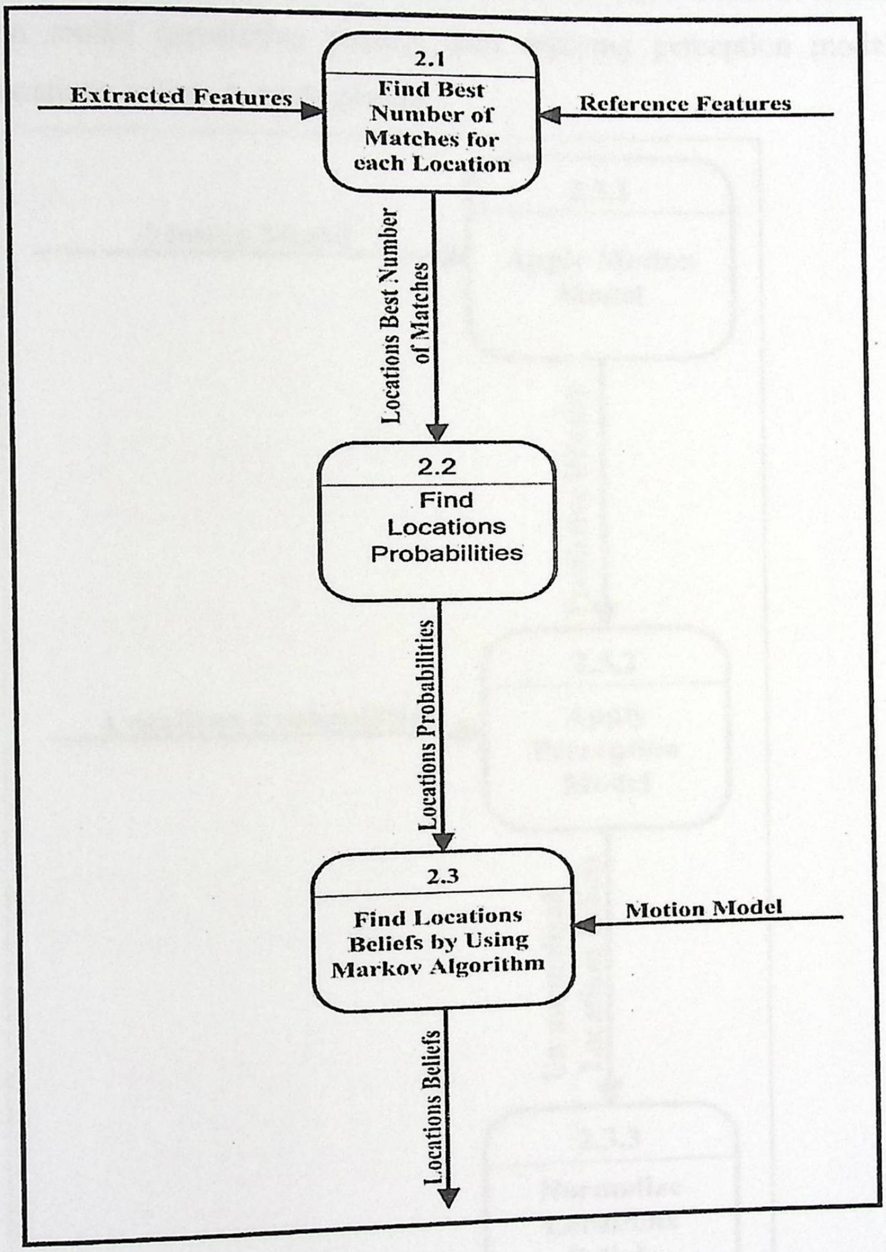


Figure 4.4 Data Flow Diagram for Process Two (Compare local features with features in central database)

4.3.5 Data Flow Diagram for Markov algorithm

After finding the probability for each location, Figure 4.5 shows how the system uses Markov algorithm [19] to find the belief for each location. It starts by applying motion model (predictive phase), then applying perception model and normalize the locations beliefs (update phase).

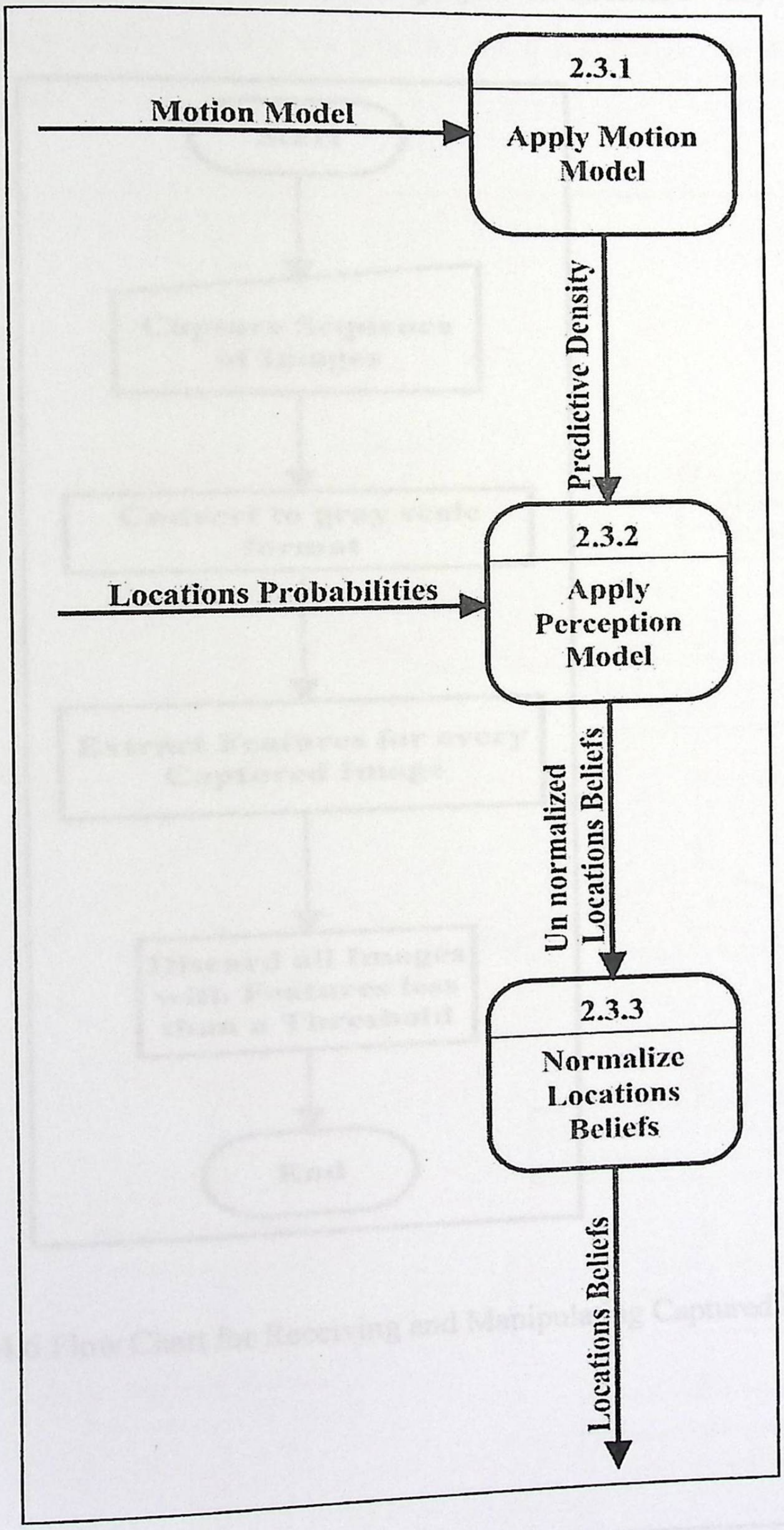


Figure 4.5 Data Flow Diagram for Markov algorithm

4.4 Flow Charts [1]

4.4.1 Flow Chart for Receiving and Manipulating Captured Images

In Figure 4.6 each captured image is converted to gray scale format then its features are extracted using SIFT algorithm. Any image with number of features less than a specified threshold is discarded (in our project the threshold = 15)

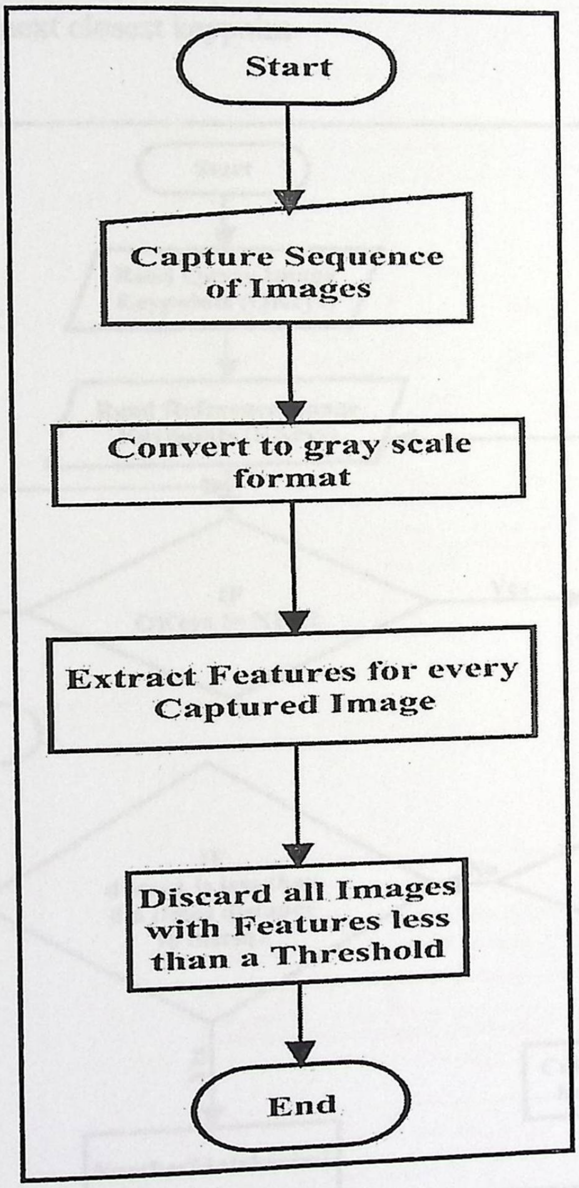


Figure 4.6 Flow Chart for Receiving and Manipulating Captured Images

[1] The diagrams based on Gane and Sarson Symbols

4.4.2 Flow Chart for Finding Number of Matches between Two Images

Figure 4.7 shows how number of matches between two images is found. For each query image the system finds its closest match in the reference images keypoints, by computing the distance between query keypoint and each keypoint in the reference image. The closest keypoint is the one whose square distance is less than 0.60 times distance of the next closest keypoint.

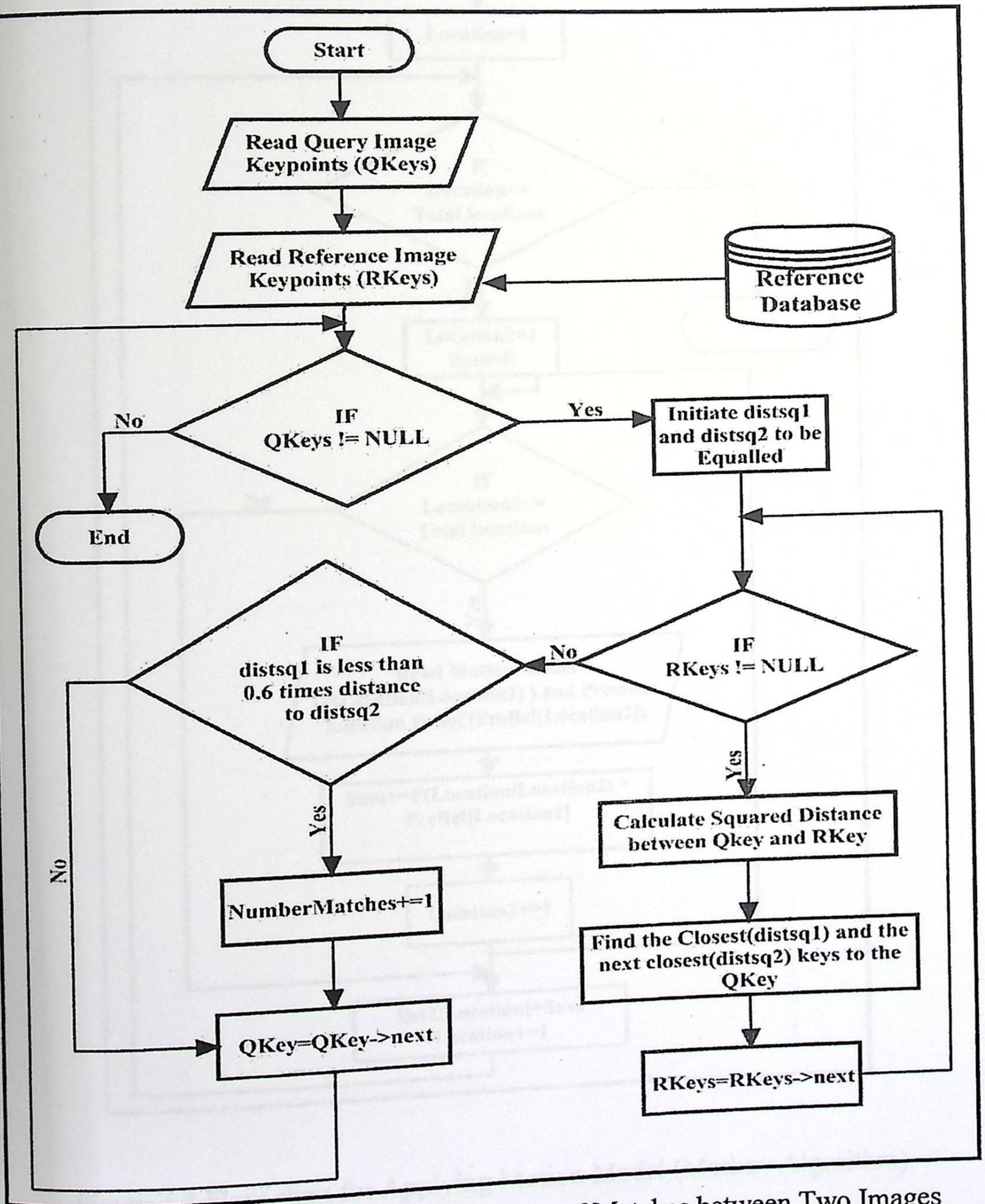


Figure 4.7 Flow Chart for Finding Number of Matches between Two Images

4.4.2 Flow Chart for Finding Number of Matches between Two Images

Figure 4.7 shows how number of matches between two images is found. For each query image the system finds its closest match in the reference images keypoints, by computing the distance between query keypoint and each keypoint in the reference image. The closest keypoint is the one whose square distance is less than 0.60 times distance of the next closest keypoint.

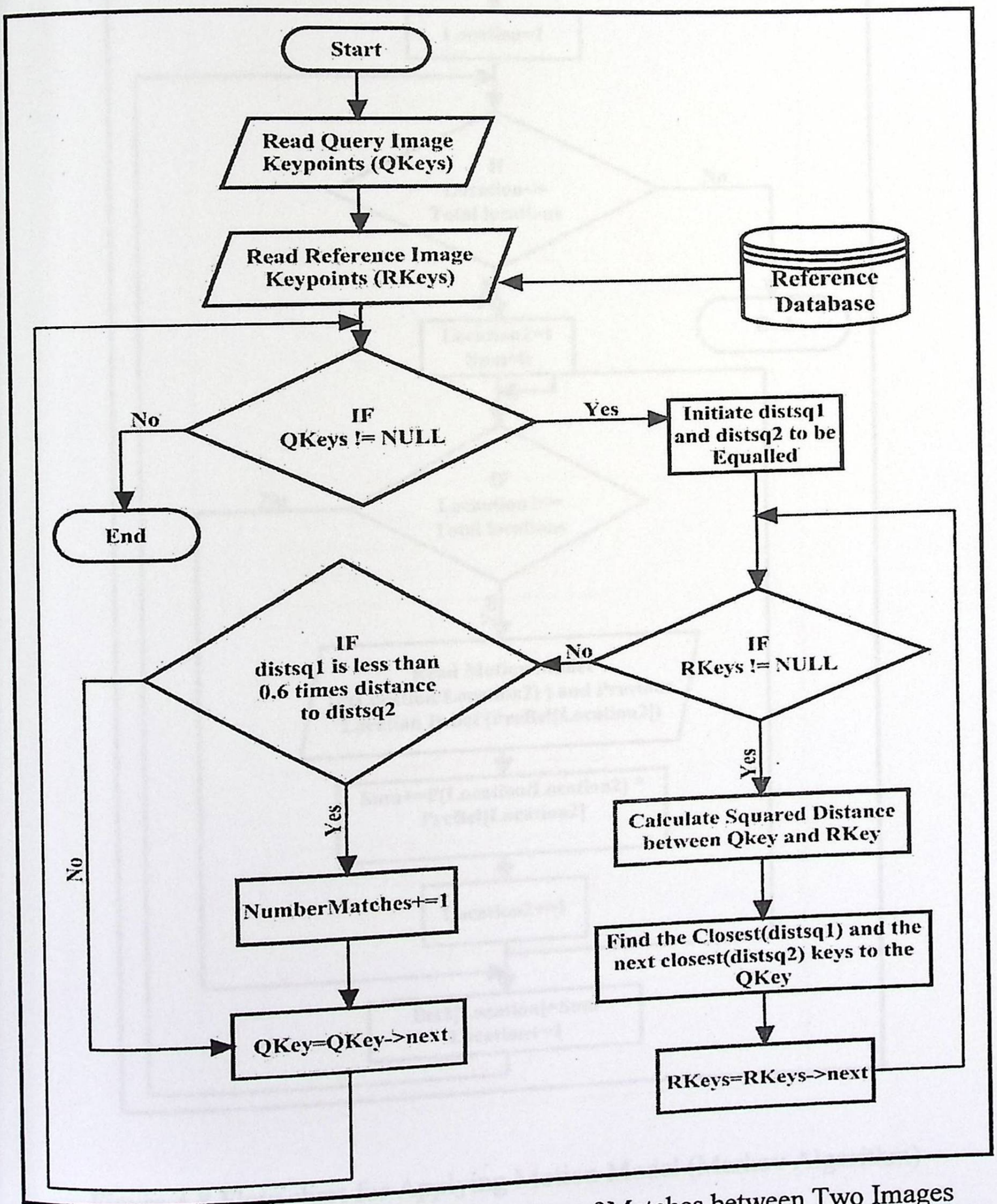


Figure 4.7 Flow Chart for Finding Number of Matches between Two Images

4.4.3 Flow chart for Applying Motion Model (Markov Algorithm)

The system depends on the motion model to predict the current position of the user. The motion model is specified as conditional density for moving from one location to another location (see Table 4.3). The predictive density for each location is obtained by integration as shown in Figure 4.8.

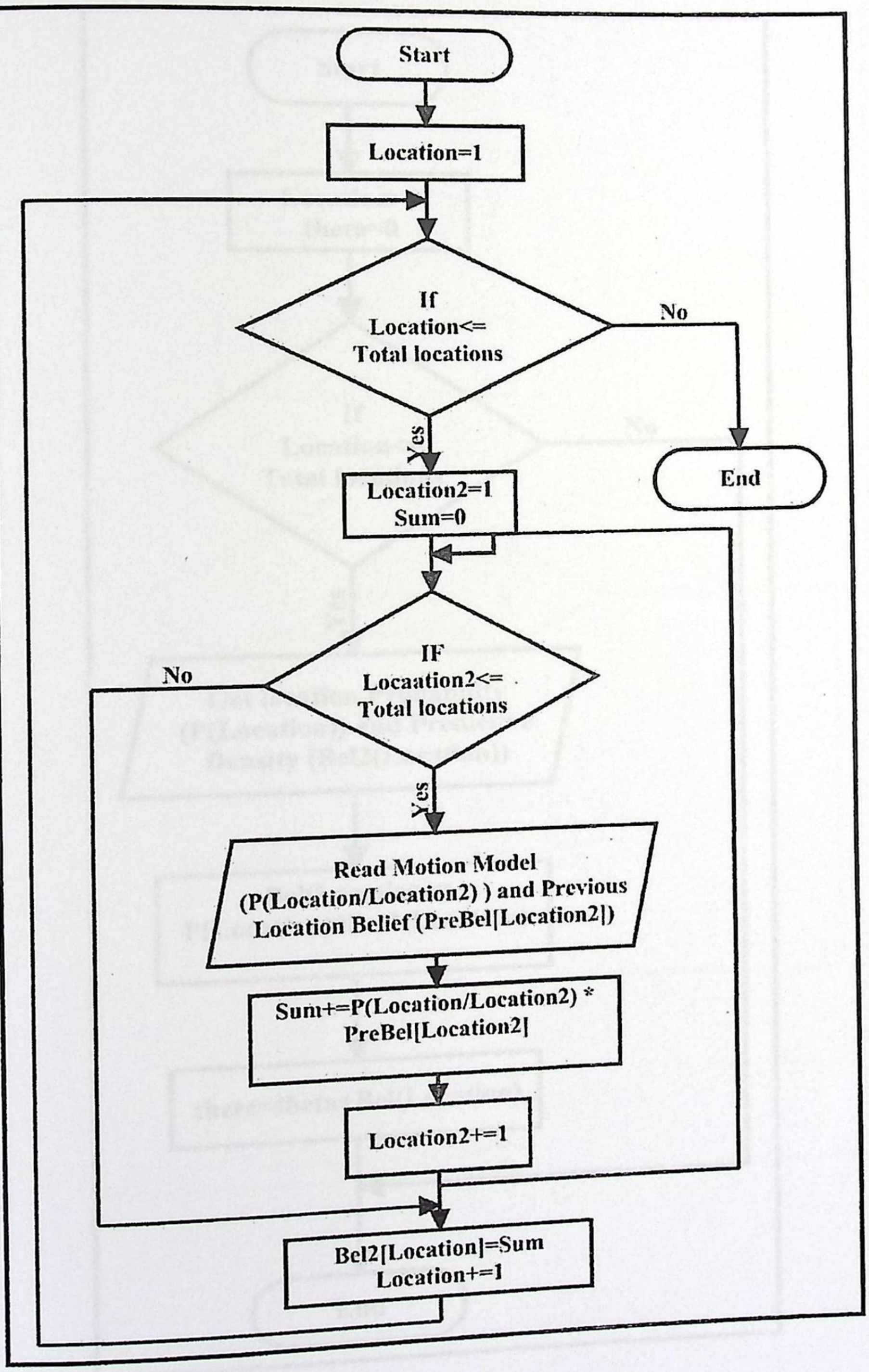


Figure 4.8 Flow chart for Applying Motion Model (Markov Algorithm)

4.4.4 Flow chart for Applying Perception Model (Markov Algorithm)

After computing the predictive density for each location the system computes unnormalized belief for each location by manipulating location probability with predictive density also a variable named theta is computed to be used later in normalization step. See Figure 4.9.

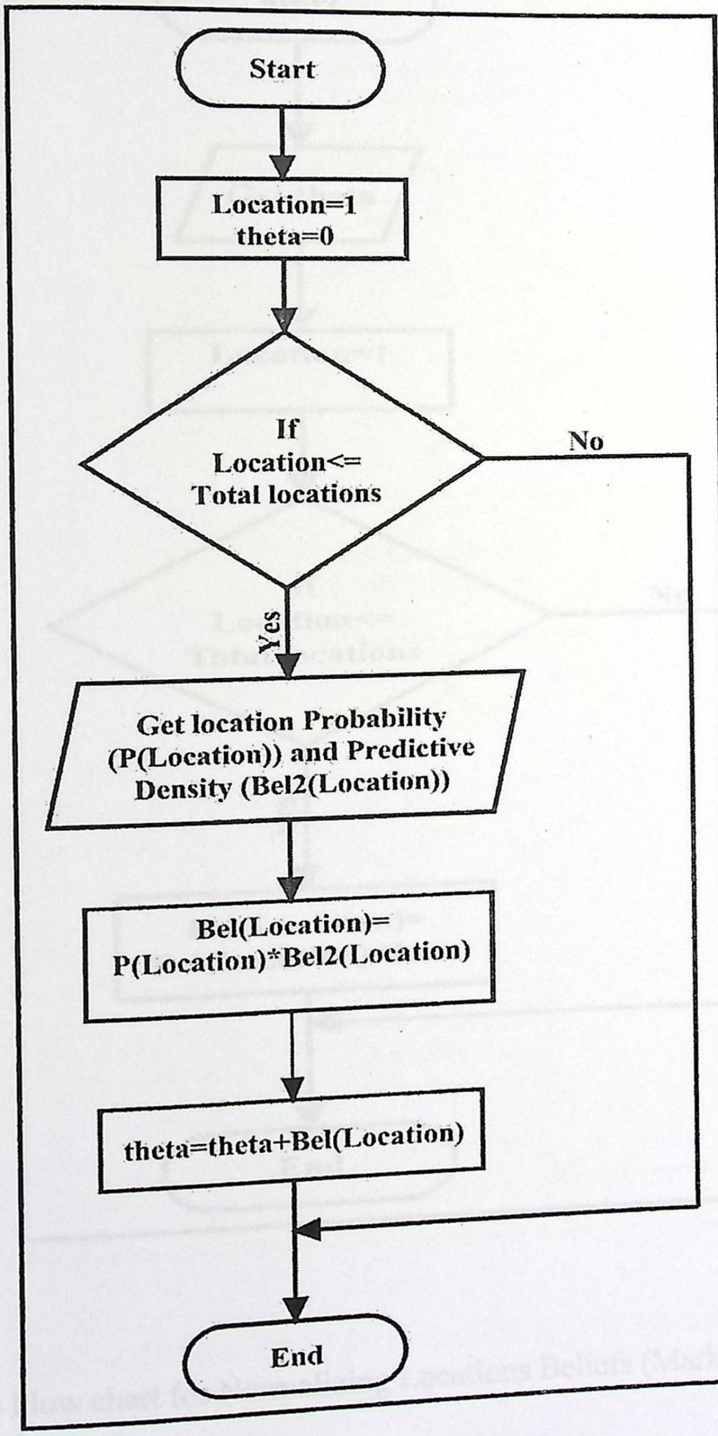


Figure 4.9 Flow chart for Applying Perception Model (Markov Algorithm)

4.4.5 Flow chart for Normalizing Locations Beliefs (Markov Algorithm)

Figure 4.10 shows the normalization process where each un-normalized location belief is divided on theta to obtain the locations belief.

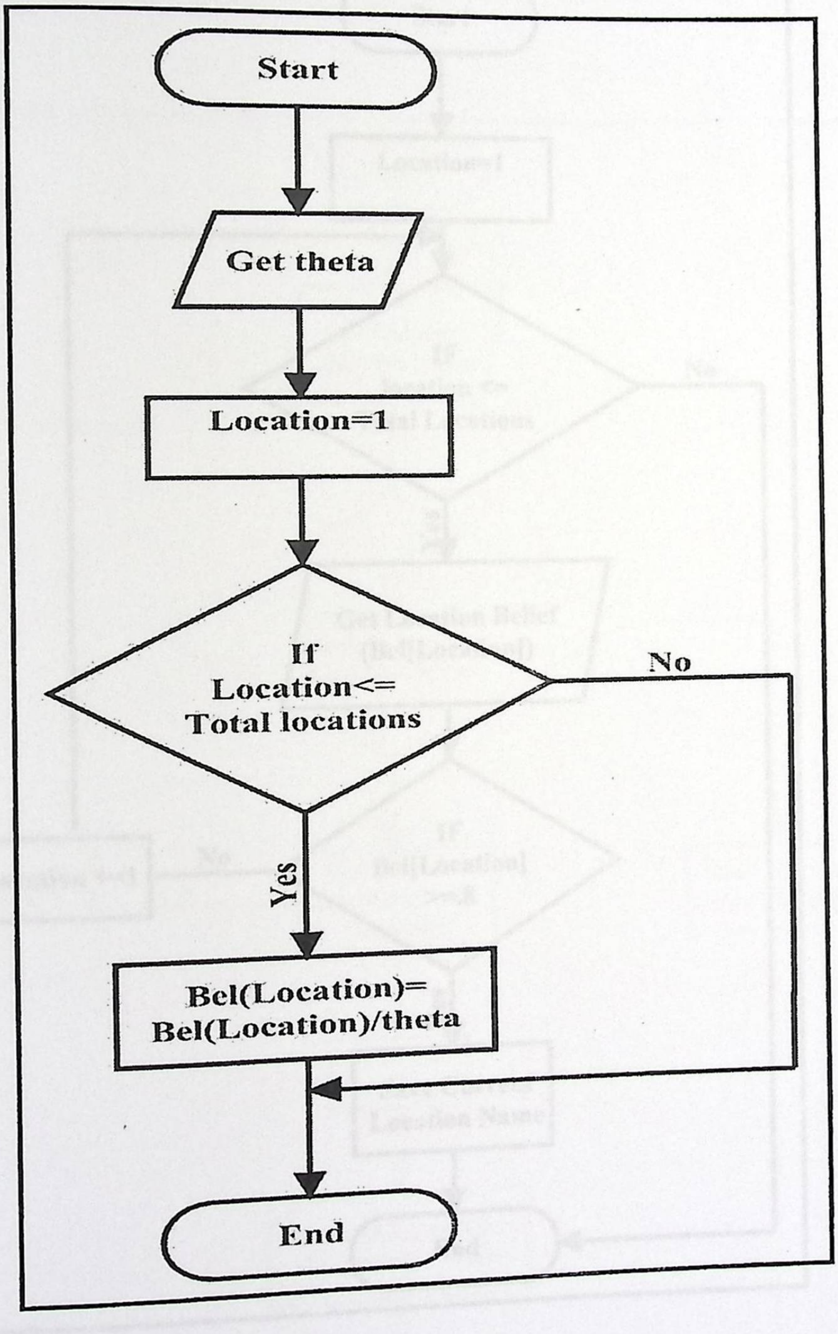


Figure 4.10 Flow chart for Normalizing Locations Beliefs (Markov Algorithm)

4.4.6 Flow Chart for Determining User Location

After computing belief for each location, the system will select the location which its belief is more than .80 to be the current location. The name of the current location is saved in the log file. See Figure 4.11.

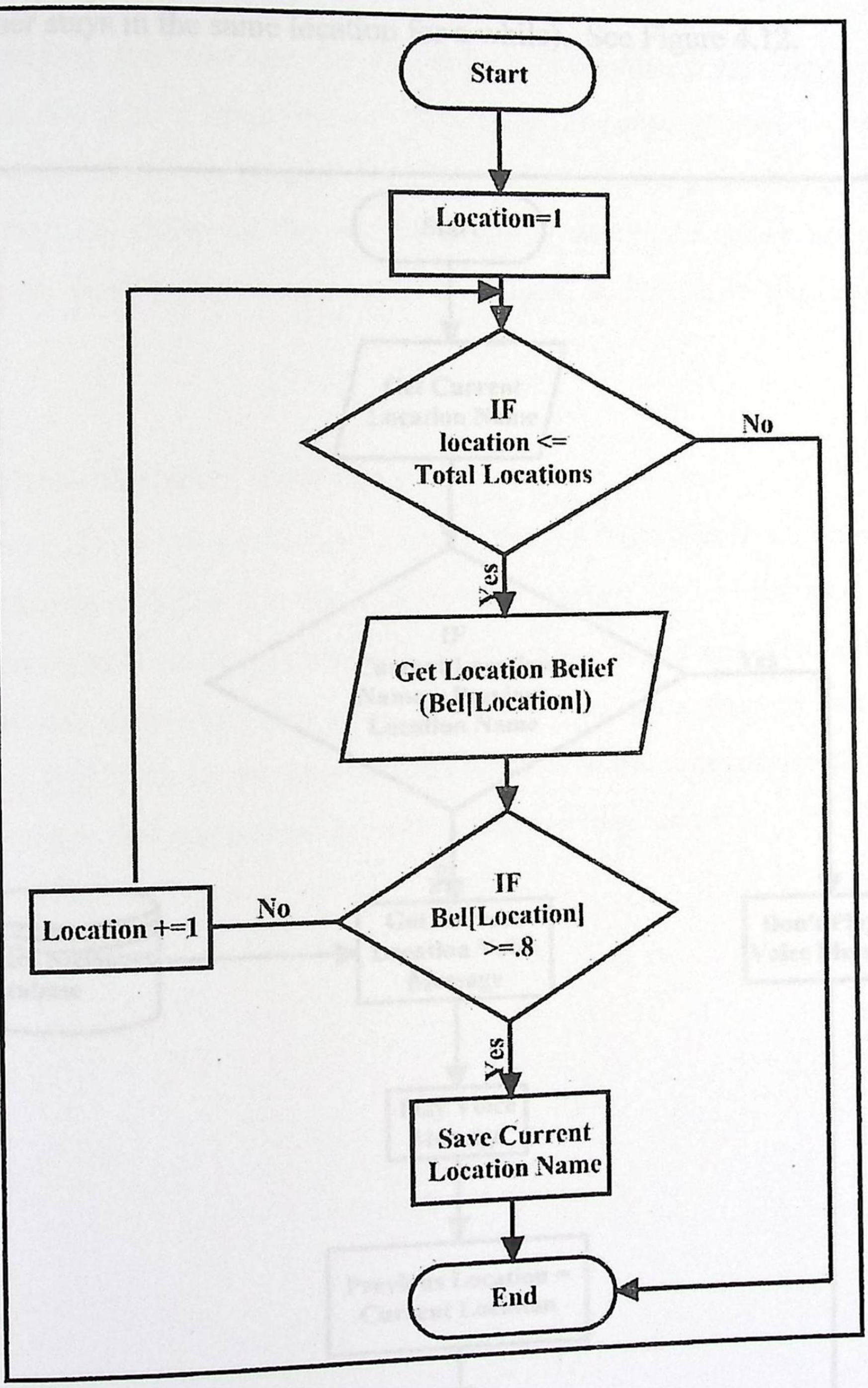


Figure 4.11 Flow Chart for Determining User Location

4.4.7 Flow Chart for Selecting Appropriate Voice Message.

The system will play voice message only when the user moves from one location to a new location. (This prevents the same message to be played more than once if the user stays in the same location for a while). See Figure 4.12.

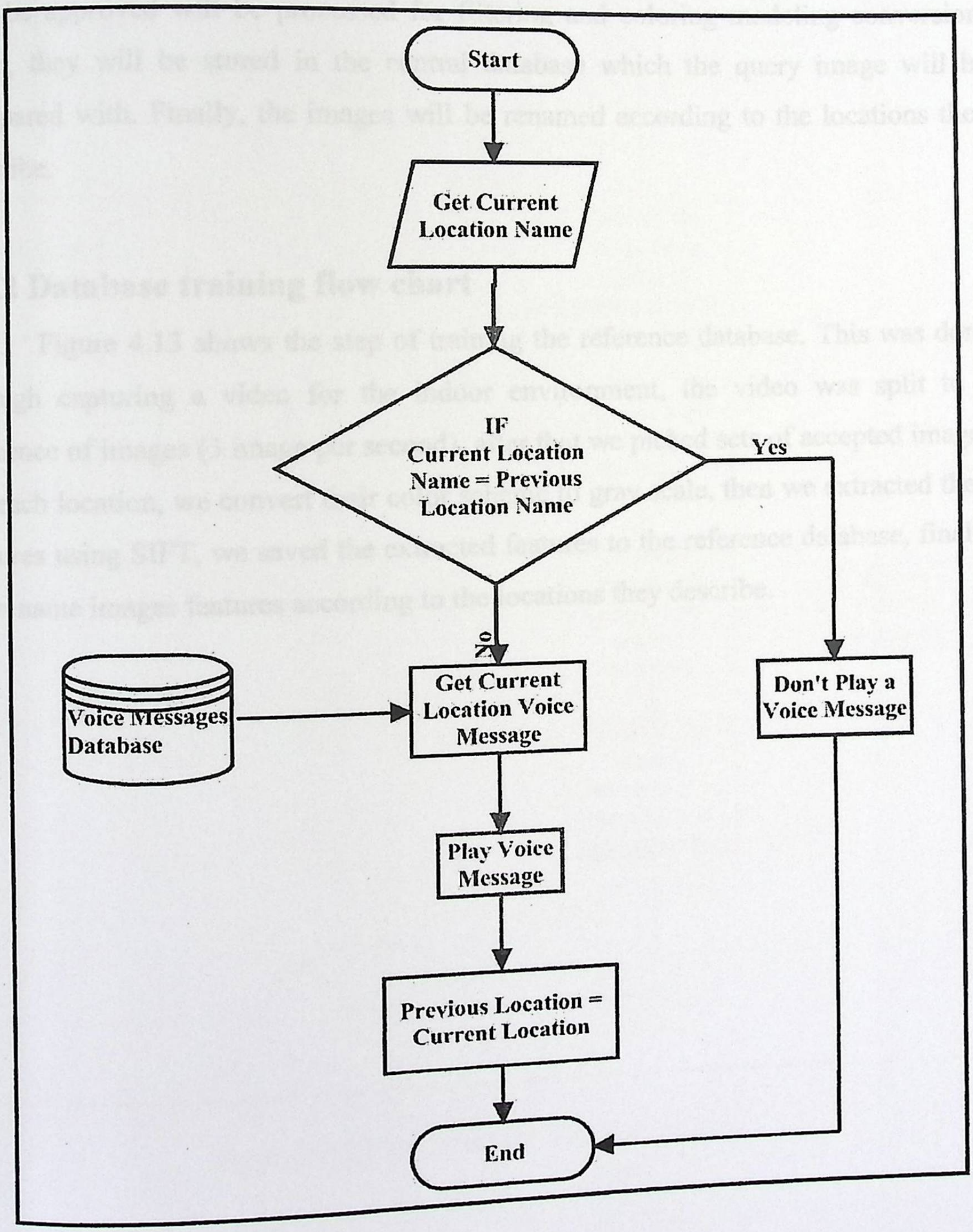


Figure 4.12 Flow Chart for Determining User Location

4.5 Database

4.5.1 Database training

To prepare the database for the indoor environment that the system will operate under; the project team first will photograph the environment, the images that will be approved will be processed for filtering and coloring modeling conversion. Then, they will be stored in the central database which the query image will be compared with. Finally, the images will be renamed according to the locations they describe.

4.5.2 Database training flow chart

Figure 4.13 shows the step of training the reference database. This was done through capturing a video for the indoor environment, the video was split to a sequence of images (3 image per second), after that we picked sets of accepted images for each location, we convert their color scheme to gray scale, then we extracted their features using SIFT, we saved the extracted features to the reference database, finally we rename images features according to the locations they describe.

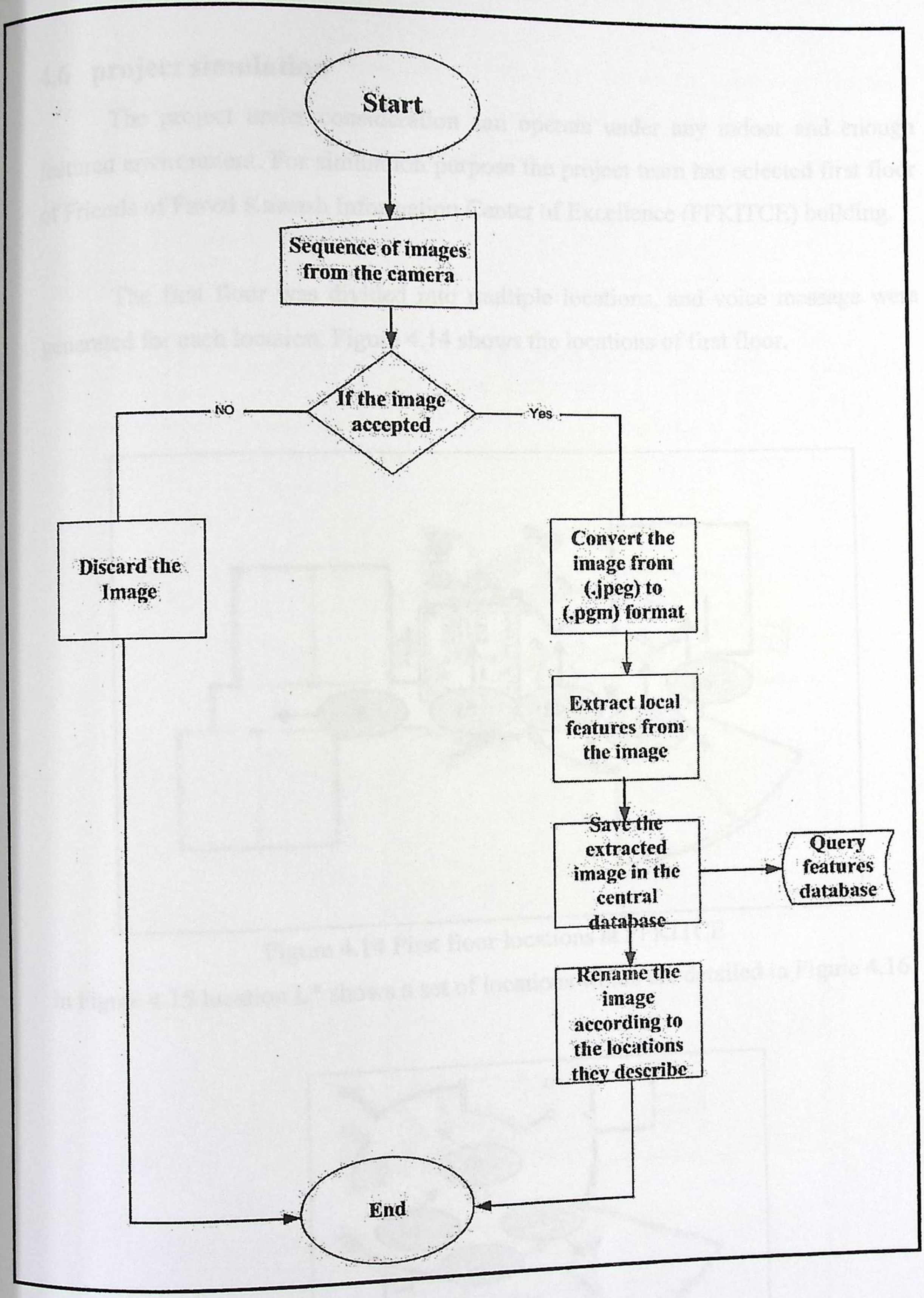


Figure 4.13 Flow chart for database training

4.6 project simulation

The project under consideration can operate under any indoor and enough featured environment. For simulation purpose the project team has selected first floor of Friends of Fawzi Kawash Information Center of Excellence (FFKITCE) building.

The first floor was divided into multiple locations, and voice message were generated for each location. Figure 4.14 shows the locations of first floor.

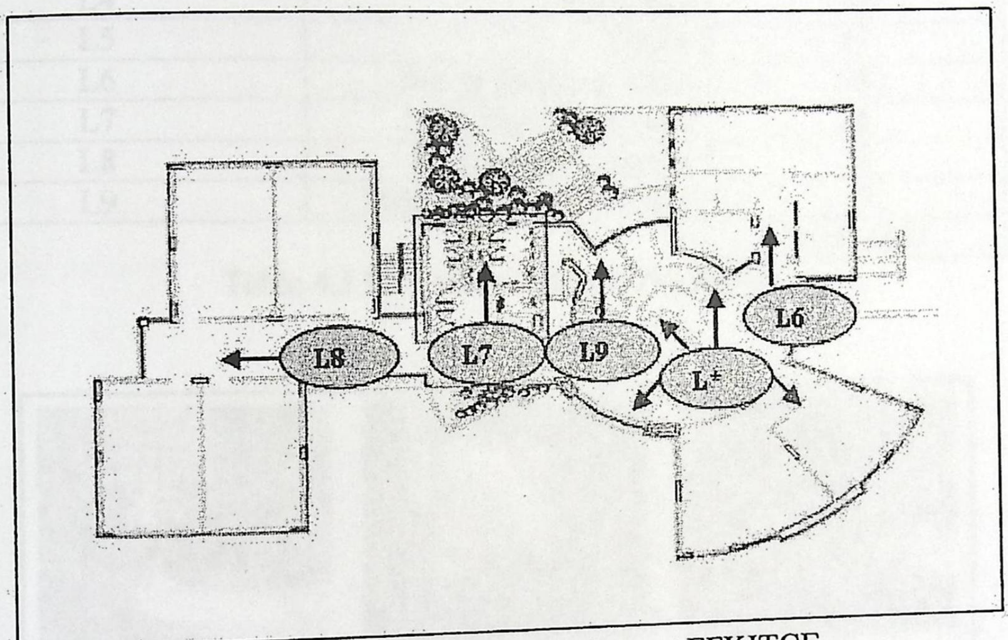


Figure 4.14 First floor locations at FFKITCE

In Figure 4.15 location L* shows a set of locations which are detailed in Figure 4.16.

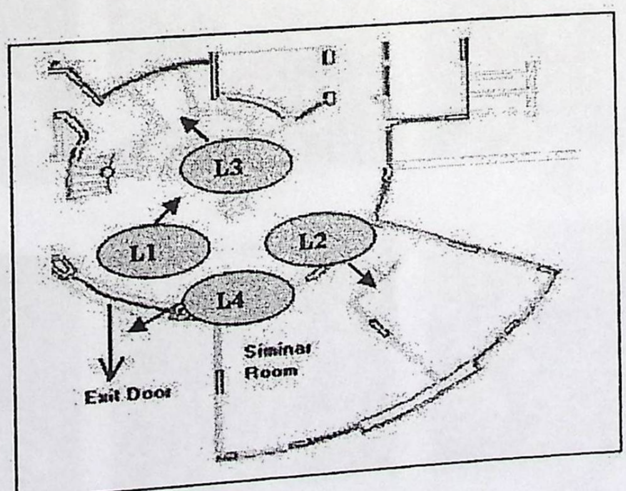


Figure 4.15 Reception hall locations

A set of 262 image were selected which are shown in Table 4.1. The table shows the locations label and number of images that represent each location in the reference database. Figure 4.16 represents samples of the reference database for each location in Table 4.1.

Location # description	Location specification	Number of images in database
L1	Reception hall	48
L2	Seminar and testing room	21
L3	Back of the reception	46
L4	Exit Door	14
L5	Chairs	21
L6	Server and Eng. room	18
L7	First floor bathrooms	20
L8	Laboratories passage	39
L9	Staircase to second floor	35
		Total =262

Table 4.1 First floor locations description

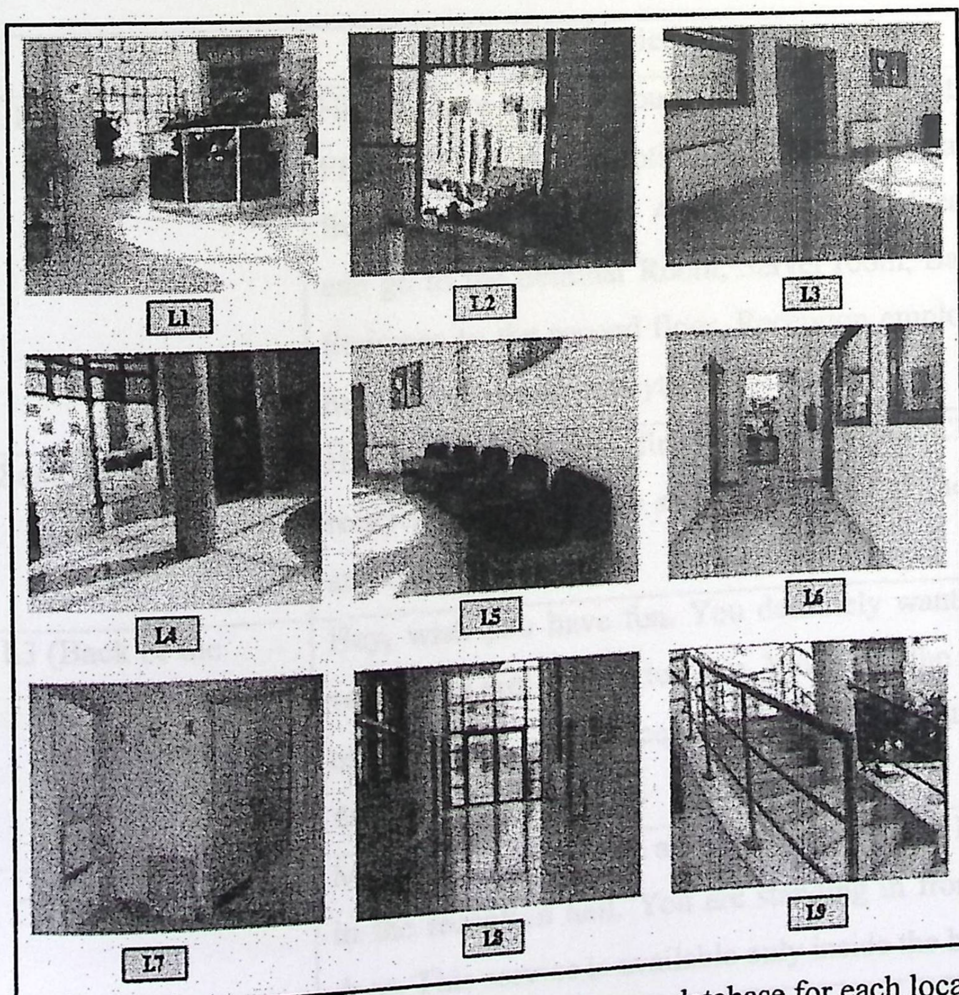


Figure 4.16 Samples of images from the reference database for each location

4.7 Voice message database

Blind persons should listen to suitable, clear, friendly, and in the same time brief information rather than listening to any information. The challenge is that the voice message should be clear enough so the blind user can understand the situation around him/her, and should be brief enough so it could not bother the blind person. We associated a message voice with each location, which describes the location. One voice message was allocated for each location. The message describes the location he/she at, what he is looking at, and the possible locations he/she can go to. The average duration time for each voice message is 40 seconds at maximum. Negative voice messages provide directions that could help the blind user to capture good images.

The voice message starts after a Beeb which prepare the blind person to hear the voice message. Table 4.2 shows voice messages scenario associated with each location.

Location	Voice message
L1 (Reception)	Hi, I'm here to help you. You are at the first floor of FFKITCE in the reception hall. You are standing in front of the Reception. According to your location you can go to the Seminar Room, Server room, Bathrooms, Staircase to the second floor. Reception employees can help you if you need anything.
L2 (Seminar and testing room)	Hi, You are at the first floor of FFKITCE in the reception hall. You are standing in front of the seminar and testing room.
L3 (Back of the reception)	Hey, wish you have fun. You definitely want to know where you are!!! I will tell you. You are at the first floor of FFKITCE in the reception hall. You are standing at the back the Reception.
L4 (Exit Door)	Mr. / Mrs. (XXX) You are at the first floor of FFKITCE in the reception hall. You are standing in front of Exit door. This service is available only inside the building.

Location	Voice message
L5 (Chairs)	Hi again. I hope that you are enjoying our service. You are at the first floor of FFKITCE in the reception hall. You are standing in front of Chairs.
L6 (Server and Eng. room)	Hi, I'm here to help you. You are at the first floor of FFKITCE in the reception hall. You are standing in front of the server and Eng. room.
L7 (First floor bathrooms)	Hi, you are at the first floor of FFKITCE. You are standing in front of First floor Bathrooms. According to your location you can go to the laboratories passage or back to the reception hall.
L8 (Laboratories passage)	Mr. /Miss (XXX) you are at the first floor of FFKITCE. You are standing at Laboratories passage. There are four laboratories here.
L9 (Staircase to second floor)	Hi, you are at the first floor of FFKITCE in the reception hall. You are standing in front of the staircase that is lead to the next floor.

Table 4.2 Voice messages associated with each location

The content of voice massage changed according to the sequence of locations the blind goes through. And it is range between brief messages and friendly, sometimes detailed messages. Blind user can easily control the voice through such as adjusting the voice volume.

4.8 Searching in image database

In order to accomplish an increase of success rate, the project team built locations graph which shows the possibilities of searching depending on the location that the user could look at; thus will prevent the system accepting bad and impossible navigation, and so the success rate will be raised.

This graph will also provide accurate tracking for the blind person, and will discard the locations that the blind person can not move to. For example, assume that the blind person orienting the camera to the first floor bathrooms, searching in the whole database will result in giving the same percentage for first floor bathrooms and the seminar room, for example 81% for each, so where the blind user is? What message voice to play?

The system will answer these questions by comparing the possible locations with the previous location depending on the graph.

4.8.1 Searching graph Figures

The following Figures illustrate the navigation possibilities for each location, these possibilities depend on the locations according to first floor of FFKITCE building, and thus; it will vary too much if the system will operate under another indoor environment.

Figure 4.17 shows the navigation possibilities for location one (reception hall). Navigation possibilities for this location are location two (Seminar and testing room), three (Back of the reception), four (Exit Door), five (Chairs), six (Server and Eng. room), seven (First floor bathrooms), nine (Staircase to second floor), and location one it self.

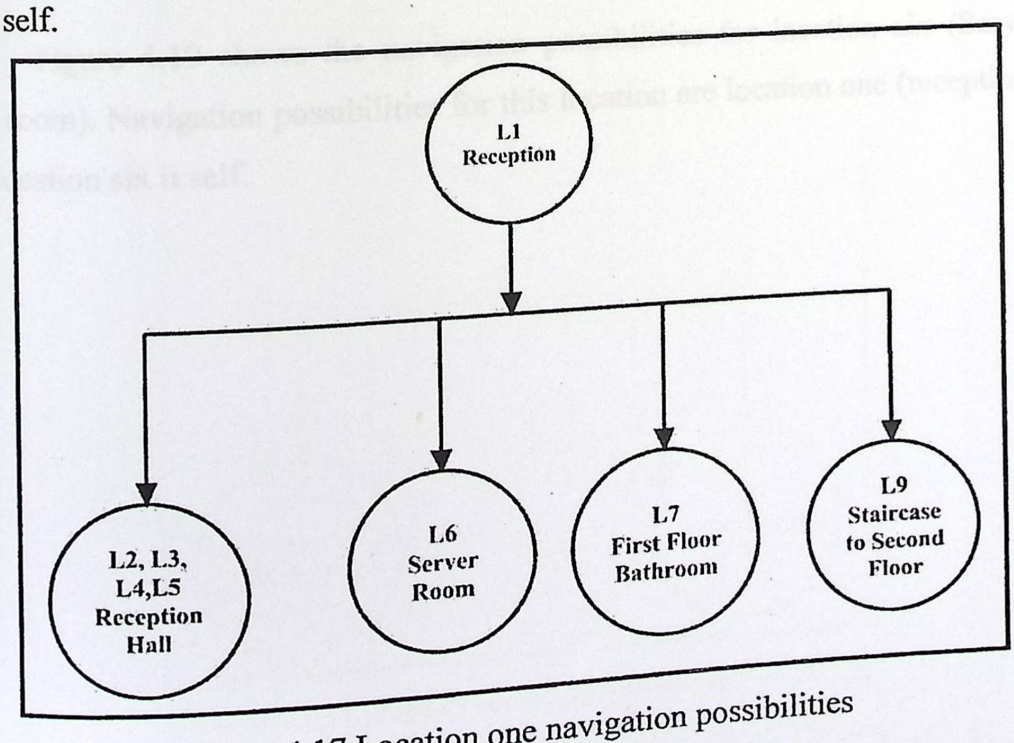


Figure 4.17 Location one navigation possibilities

Figure 4.18 shows the navigation possibilities for location L* (L2, L3, L4, L5). Navigation possibilities for this location are location one (reception hall), and location L* it self.

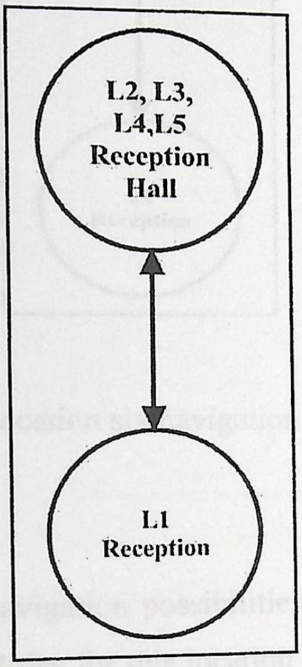


Figure 4.18 Location L* navigation possibilities

Figure 4.19 shows the navigation possibilities for location six (Server and Eng. room). Navigation possibilities for this location are location one (reception hall), and location six it self.

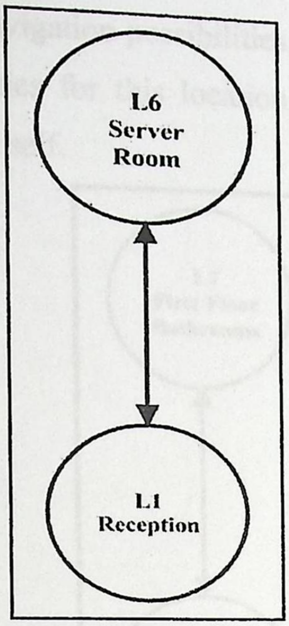


Figure 4.19 Location six navigation possibilities

Figure 4.20 shows the navigation possibilities for location seven (First floor bathrooms). Navigation possibilities for this location are location eight (Laboratories passage), location one (Reception), and location seven it self.

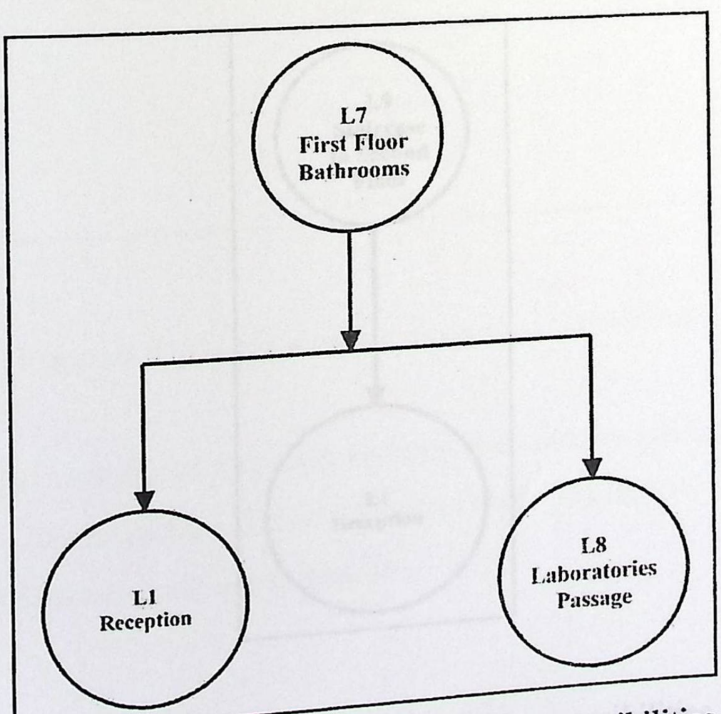


Figure 4.20 Location seven navigation possibilities

Figure 4.21 shows the navigation possibilities for location eight, (Laboratories passage). Navigation possibilities for this location are location seven (First floor bathrooms), and location eight it self.

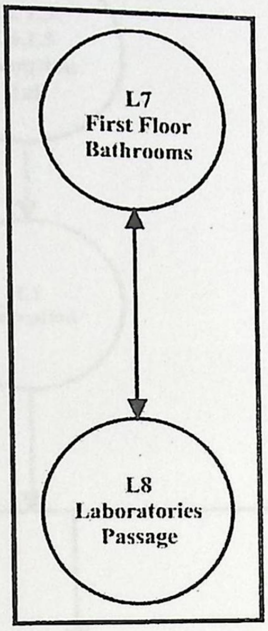


Figure 4.21 Location eight navigation possibilities

Figure 4.22 shows the search possibilities for location nine, (Staircase to first floor). Navigation possibilities for this location are location one (Reception), and location nine it self.

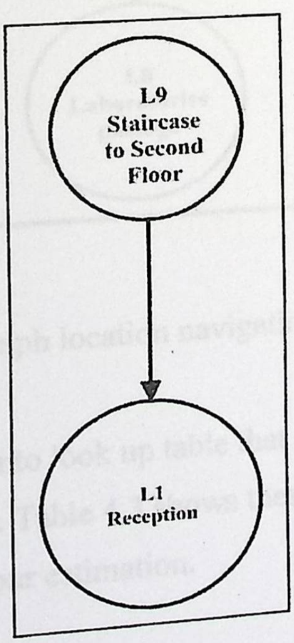


Figure 4.22 Location nine navigation possibilities

Figure 4.23 illustrate the graph locations navigation possibilities for first floor as a whole

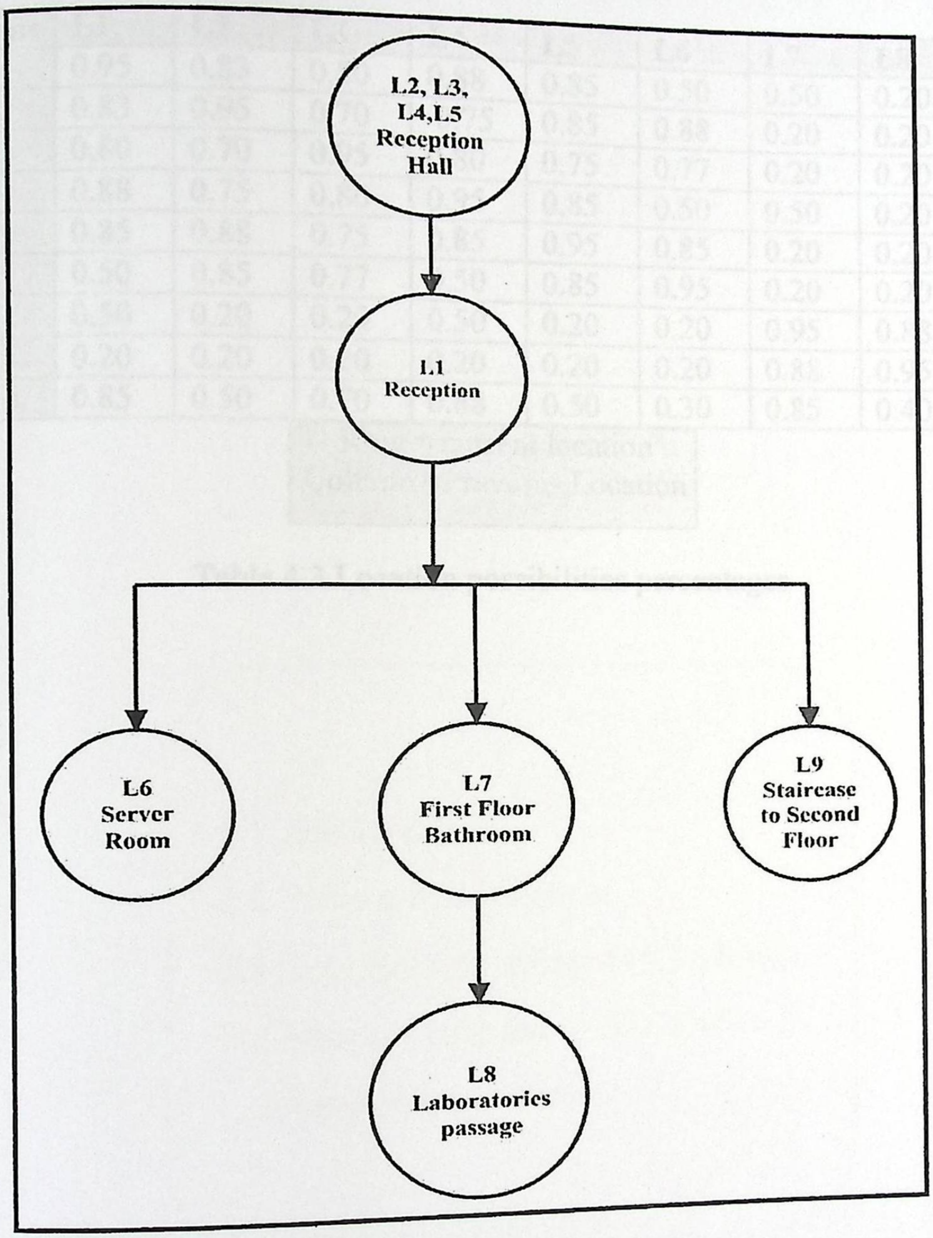


Figure 4.23 Graph location navigation possibilities

This graph is translated into look up table that contains navigation possibilities from a given location to another. Table 4.3 shows these possibilities values which were determined depending on our estimation.

Locations	L1	L2	L3	L4	L5	L6	L7	L8	L9
L1	0.95	0.83	0.80	0.88	0.85	0.50	0.50	0.20	0.85
L2	0.83	0.95	0.70	0.75	0.85	0.88	0.20	0.20	0.50
L3	0.80	0.70	0.95	0.80	0.75	0.77	0.20	0.20	0.70
L4	0.88	0.75	0.80	0.95	0.85	0.50	0.50	0.20	0.88
L5	0.85	0.88	0.75	0.85	0.95	0.85	0.20	0.20	0.50
L6	0.50	0.85	0.77	0.50	0.85	0.95	0.20	0.20	0.30
L7	0.50	0.20	0.20	0.50	0.20	0.20	0.95	0.88	0.85
L8	0.20	0.20	0.20	0.20	0.20	0.20	0.88	0.95	0.40
L9	0.85	0.50	0.70	0.88	0.50	0.30	0.85	0.40	0.95

Row = current location
Column =Previous Location

Table 4.3 Location possibilities percentages

CHAPTER FIVE

RESULTS

5.1 Introduction

5.2 Time consideration

5.3 Success rate using SIFT alone

5.4 Success rate using SIFT with Markov localization

5.1 Introduction

As motioned, the reference database contains 262 images divided into 9 locations, each related with voice message. We tested the reference database with a sequence of images (467 images) taken form the first floor of FFKITCE, these images represent the set of all possible messages that the blind person could capture. We will demonstrate the results in this chapter.

5.2 Time consideration

The average time that is needed for feature extraction for one incoming image is 0.7 seconds this depends on the number of matches that is extracted from each image. Figure 5.1 represents the maximum number of matches for 200 query image when compared with the database. The average time that is needed for comparing one image with the whole data base is 74 seconds because the searching for the keypoints in the database is sequential and not indexed. Indexing the database is quite possible, as proposed by Lowe [17] and will dramatically reduce the searching time and bring the system back into the real-time.

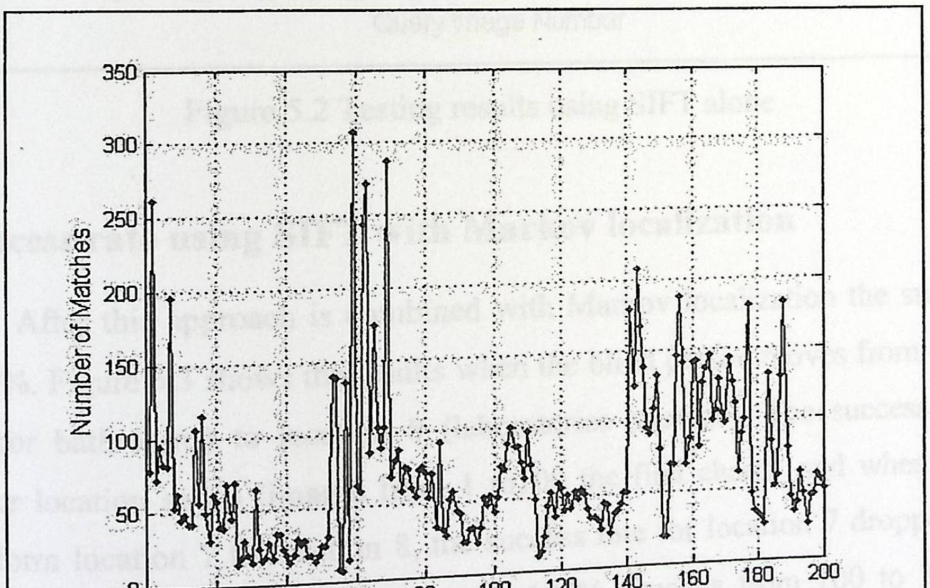


Figure 5.1 Maximum numbers of matches for 200 query images with the whole database

5.3 Success rate using SIFT alone

Figure 5.2 illustrates the results of using SIFT algorithm alone for location determination. The tested locations contain other locations from second floor. The success rate was 85.5% (401 images). As we notice the error occurs when the person move from one location to another.

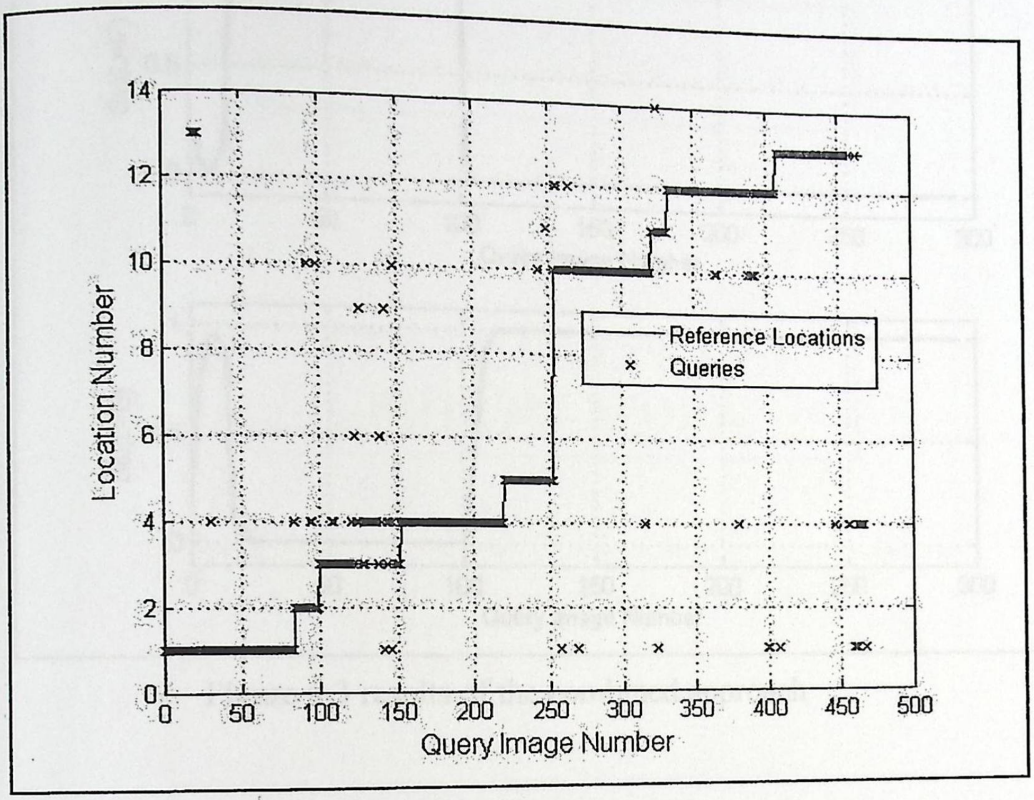


Figure 5.2 Testing results using SIFT alone

5.4 Success rate using SIFT with Markov localization

After this approach is combined with Markov localization the success rate was 100%. Figure 5.3 shows the results when the blind person moves from location 7 (first floor bathrooms) to location 8 (laboratories passage). The success rate was 100% for location seven (images from 1 to100 the first chart) and when the blind moved form location 7 to location 8, the success rate for location 7 dropped to zero while the belief of location 8 raised to be 100% (images from 100 to 300 in the second chart). This is so important because the final decision which is presented by the voice message is taken after processing a sequence of images, and so if the blind user captures meaningless image the rate of the location belief would not affected , so mistaking captured will be handled .

Another issue is that by combining SIFT approach with Markov approach the system could operate under any environment without any problem all what we have to do is just preparing the central database.

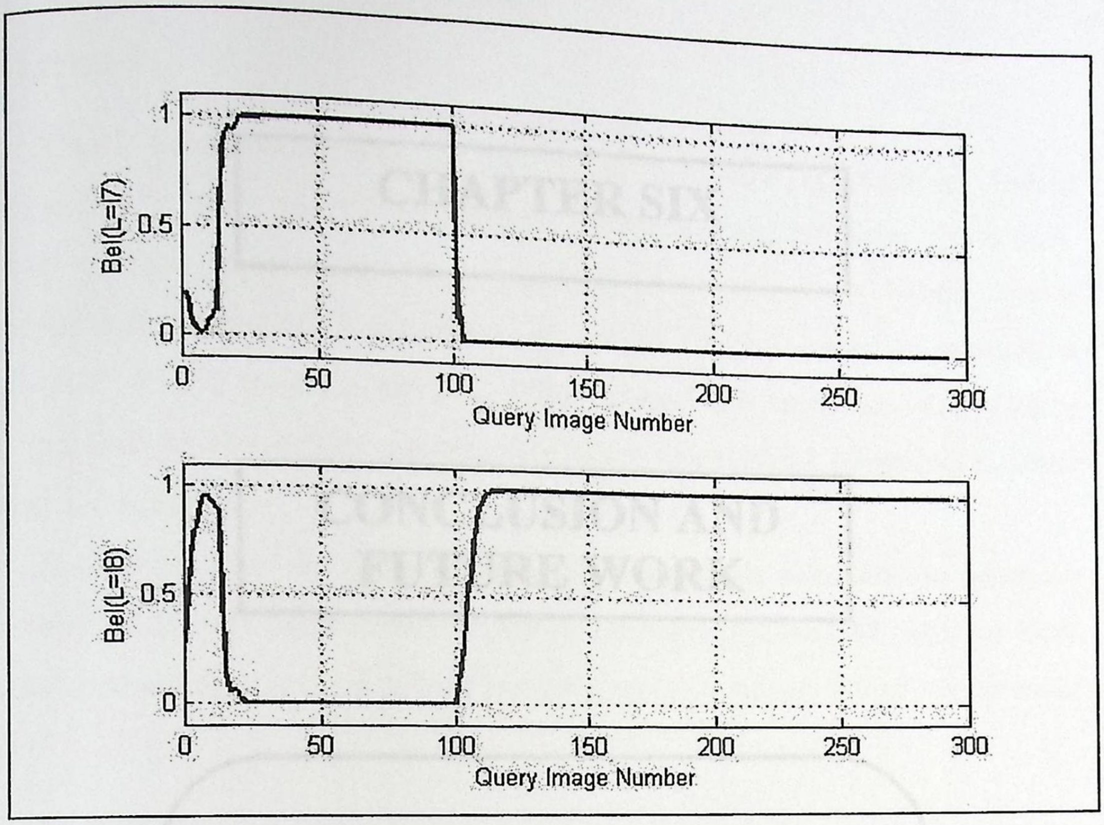


Figure 5.3 results of the combined approach

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Conclusion

6.2 Future work

6.1 Conclusion

Visual Localization Aid for the Blind is a new approach of indoor localization for the blind and visually impaired persons using computer vision only. The most obvious conclusion from experimental results in a public indoor environment is that using SIFT algorithm can lead to a high localization rate with a success rate of 85.5% which is not bad. While using SIFT algorithm with Markov method can lead to above 90% success rate witch is excellent compared to other researches in this domain.

Another issue is that by combining SIFT approach with Markov approach the system could operate in any environment without any problem. All what we have to do is just prepare the central database and look up table for the indoor environment locations.

6.2 Future work

Future work includes the design of path planning to generate intelligent routes between indoor locations so we can reduce the error rate, to give the person appropriate directions. In addition, indexing the database is highly recommended to speed up the searching time.

The end user is actually interested in a fast retrieval of the voice message that is relevant to the location of the query image(s). Today's typical image databases are steadily increasing in size which in turn puts higher demands on the retrieval performance of CBIR system. Thus the speed becomes more and more important [12].

Indexing the database is quite possible, as proposed by Lowe [17] and will dramatically reduce the searching time and bring the system back into the real-time. The project team intends to speed up the system using Best Bin First Search (BBF) algorithm, which expected to return the closest neighbor with high probability [17].

6.1 Conclusion

Visual Localization Aid for the Blind is a new approach of indoor localization for the blind and visually impaired persons using computer vision only. The most obvious conclusion from experimental results in a public indoor environment is that using SIFT algorithm can lead to a high localization rate with a success rate of 85.5% which is not bad. While using SIFT algorithm with Markov method can lead to above 90% success rate witch is excellent compared to other researches in this domain.

Another issue is that by combining SIFT approach with Markov approach the system could operate in any environment without any problem. All what we have to do is just prepare the central database and look up table for the indoor environment locations.

6.2 Future work

Future work includes the design of path planning to generate intelligent routes between indoor locations so we can reduce the error rate, to give the person appropriate directions. In addition, indexing the database is highly recommended to speed up the searching time.

The end user is actually interested in a fast retrieval of the voice message that is relevant to the location of the query image(s). Today's typical image databases are steadily increasing in size which in turn puts higher demands on the retrieval performance of CBIR system. Thus the speed becomes more and more important [12].

Indexing the database is quite possible, as proposed by Lowe [17] and will dramatically reduce the searching time and bring the system back into the real-time. The project team intends to speed up the system using Best Bin First Search (BBF) algorithm, which expected to return the closest neighbor with high probability [17].

APPENDIX A.1

(Paper) Visual Localization Aid for the Blind.
Submitted to International Conference on Computer and Information Technology (PICCIT 07), Palestine, 2007.

Visual Localization Aid for the Blinds

Fatinah Sinnoqrot, Iman Younes and Hashem Tamimi

Department of Information Technology
College of informative science and informatics
Palestine Polytechnic University
P.O.B 198, Hebron, Palestine
htamimi@ppu.edu

Abstract

In this paper we describe a computer-based location determination system for helping blind or visual impaired persons to navigate easily in an indoor environment. Unlike the existing systems, the proposed system requires a light weight personal computer and a small camera only. The camera captures images from the place where the blind exists and sends it to the computer where it will be analyzed. As a response, a voice message is played to the person explaining where she/he is. The method adopted here is based on a Content-Based Image Retrieval (CBIR) approach. Image-based features are extracted using the well known SIFT algorithm. These extracted features are used as the basis for the similarity check between images. This approach is combined with Markov localization to enhance the localization rate. Experimental results in a public indoor environment show that the proposed approach can lead to a high localization rate.

Key words:
Indoor environment, blind users, CBIR, SIFT, Markov localization.

I. Introduction

Blind people are at tremendous disadvantage when they arrive new indoor environment like university, school, supermarket, and others [2]. They should have freedom and ability to locate where they are. Along the path to a certain destination, blind people need exact information about locations, dangers, and appropriate paths [6].

From this point we aim to provide a facility to make the mobility idea in an indoor environment easier using computer vision a lone and no other sensor except for a light weight camera. In addition the proposed system does not require any artificial markers in the environment and relies on the natural decorations of the existing furniture which makes the establishment of such system inexpensive.

Nevertheless, using computer vision for location detection addresses some challenging because of the following reasons. First, it is hard to determine where the person exactly exists in buildings that have the similar design in different places like similar offices or similar corridors. Second, sometimes there are few clues (decorations) in the environment which help the system to find out the right position. Third, there is no Global Positioning System (GPS) signals help to the limitations of the GPS in indoor environments. Forth, for the blind user, capturing a good image is very hard in many cases and the quality of the captured image would be affected by the lighting effects, camera quality, and camera orientation, which accordingly will affect the final result.

1.2 Related Works

This field, navigation and localization assistance for the blind or visual impaired persons has become an active research area. There are a lot of researches that have been dedicated in this domain that can be classified into a few general categories. One category could be the usage of sensors only, like Robotic guide from Kulyukin [1]. This system aims to aid the blind and visually impaired shoppers to find their way around grocery store. For localization, RoboCart relies on (RFID) tags[1] deployed at various locations in the store. For navigation, RoboCart relies on laser range finding. Similar to this approach the system that was developed by Willis and Hilal [2], which found a solution to the problem of way finding for the blind user in a university campus by using RFID information grid and wearable computing system. Each RFID tag is programmed upon installation with spatial coordinates and information describing the surrounding, this allow for self describing, localized information system with no dependency on centralized database or wireless infrastructure for communications. Sonnenblick [3] evaluated and implemented a large-scale indoor navigation system for blind individuals at the Jerusalem Center for Multi-Handicapped Blind Children. It relies on specially installed safely infrared beacons for orienting blind users and on a custom-built end-user device. Although the mentioned researches work very well in new, unknown indoor environments they still rely on certain provided infrastructure for example RFID tags and infrared beacons which must be installed in many locations at the indoor environment because it is one dimensional signal. Another research is the NavBelt from Shoval et al. [4] which produces a 120 degree wide view ahead from the current location of the blind. This information is translated into a stereophonic acoustical sound that allows the user to notice if a certain direction is blocked. This research dedicated in obstacles avoidance information rather than orientation and localization information.

Other category could be the usage of sensors in a line with intelligent image processing to provide a solution for indoor navigation. Hub et al. [5] proposed a new system that assists blind users in orienting themselves in indoor environment using a sensor module that can be handled like a flashlight by a blind user and can be used for

[1] Radio-frequency identification (RFID) is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders. An RFID tag is an object that can be attached to or incorporated into a product, animal, or person for the purpose of identification using radio waves.

searching tasks within the three-dimensional environment. By pressing keys, the user's inquiries concerning their environment are acoustically answered through a text-to-speech engine. The system's limitation is that it is necessary to hit an object precisely using a picking ray within a 3D model in order to allow proper object identification [6]. In a later work Hub et al, developed a new prototype [6] which includes an option to receive augmented navigation hints automatically just by walking in virtual corresponding navigating areas, this orientation assistant consists of two parts – a sensor module and a small portable computer that can be carried in a small backpack. The portable computer has access to 3D environment models. The sensor module itself consists of direction sensors and a stereo camera. The orientation assistant works as follows: The location of the user respectively the orientation assistant within buildings can be determined over a conventional WiFi[1] system via the differences in the signal strength transmitted by the access points [6]. Hub et al. also presented a new method for interactive tracking of various types of moving objects [7]. For example the state of fixed or moving doors can be recognized by comparing the distance sensor data and 3D model. For the identification and model based tracking of free movable objects, like chairs. Further, using a common face detection algorithm, the system could inform the user of the presence of people. And enable the localization of a real person based on interactive tracking of virtual models of humans. Although these systems work very well in new, unknown indoor environments, they still need certain provided infrastructure. And for location detection they still rely on the use of sensors.

1.3 The proposed work

In this paper we present a novel approach for localization aid of the blind person. There are two points that make our work original. First, our project uses computer vision only in analyzing input pictures without using GPS or other sensor, and so, installing the project in a new indoor environment does not require any especial infrastructure like other projects which used RFID tags and other sensors. Second, this project is the first in helping the blind research domain that measures the result quantitatively.

1.4 Content Based Image Retrieval

The project adopts the similar methodology used in CBIR. CBIR is the application of computer vision to the image retrieval problem of searching for digital images in large databases. It involves analyzing the actual contents of the image, which might refer to color, shape, texture, or any other information that can be derived form the image itself [8]. It involves two steps; as shown in Figure 1, the first step is feature extraction in which the image features extracted to a distinguishable extent. The second step is matching which involves matching these features to yield a result that is visually similar [9].

1 (Wi-Fi) Wireless Fidelity is a brand originally licensed by the Wi-Fi Alliance to describe the embedded technology of wireless local area networks (WLAN) based on the IEEE 802.11 specifications. Wi-Fi was developed to be used for mobile computing devices, such as laptops in LANs, but is now increasingly used for more services, including Internet, and basic connectivity of consumer electronics such as televisions, DVD players, and digital cameras.

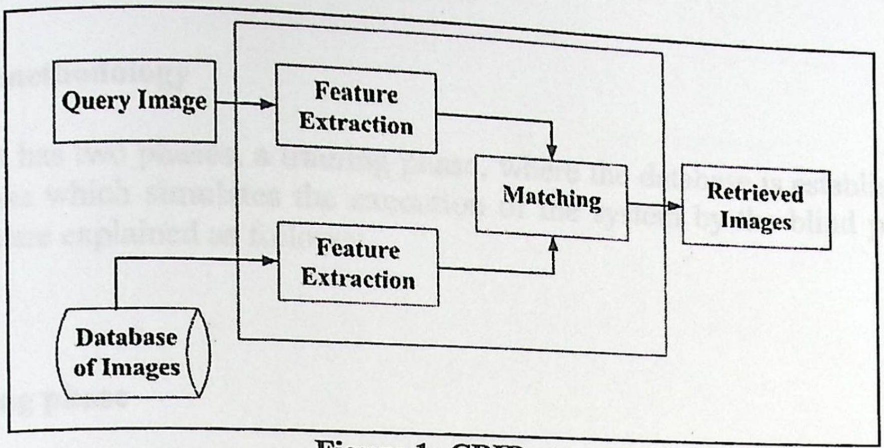


Figure 1: CBIR steps

David Lowe [10] has introduced one of the most successful local feature extraction methods which he called SIFT (Scale Invariant Feature Transforms). The SIFT features are invariant against geometric transformation (e.g. rotation and scaling) and robust to photometric transformation (e.g. illumination, and small changes in viewpoint) [11]. SIFT is an approach for extracting distinctive (local) features from images, that can be used to perform reliable matching between different views of an object or scene. It takes an image and transforms it into a large collection of local feature vectors "keypoints". Figure 2 shows an example of matching produced by the SIFT matching algorithm. Features are extracted from each of the two images, and lines are drawn between features that are similar. In this example, many matches are found and only small fractions are incorrect.

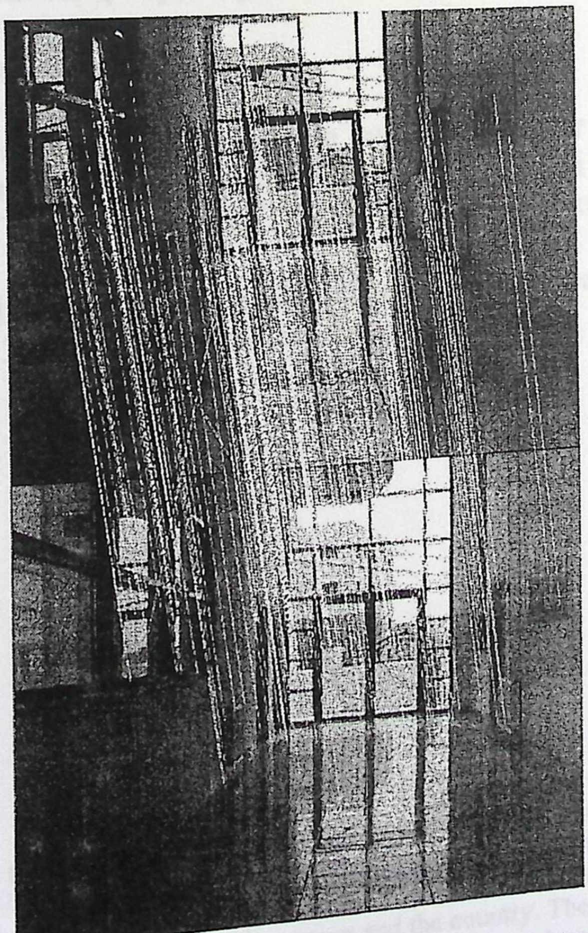


Figure 2: Example of applying SIFT matching to two images

2. Project methodology

The project has two phases, a training phase, where the database is established, and a testing phase which simulates the execution of the system by the blind person. The two phases are explained as follows:

2.1 Training phase

In order to prepare the database for the indoor environment that the system will operate under; the set of images that describe the indoor environment was collected and labeled according to their positions. The features of each image are extracted using SIFT. Finally, the whole features and the corresponding labels restored in a database.

2.2 Testing Phase

In this phase, the system will capture a set of images for processing. Then, the local features will be extracted using SIFT algorithm. The extracted local features will be compared with the whole database of local features that describe the environment. The comparison criterion is to find a location from the database that has the maximum number of matches with the query images based on SIFT algorithm.

2.3 Project simulation

The project under consideration can operate under any indoor and featured enough environment. For simulation purpose the first floor of Fawzi Kawash Information Technology Center of Excellence (FFKITCE)[1] building in Hebron was selected.

The first floor was divided into multiple locations, and generate voice message for each location. Figure 3 shows the locations of first floor.

[1] FFKITCE is one of few purpose-built IT development-training centers in Palestine; it was established at Palestine Polytechnic University-Hebron in 2005. It aims to encourage and support innovative IT research and projects that have significant potential value, and to provide exemplary dissemination, training and support in IT for the University, the region and the country. The IT Business Incubation Service at the Center will enable people with good ideas in IT to find a place to work and develop their ideas.

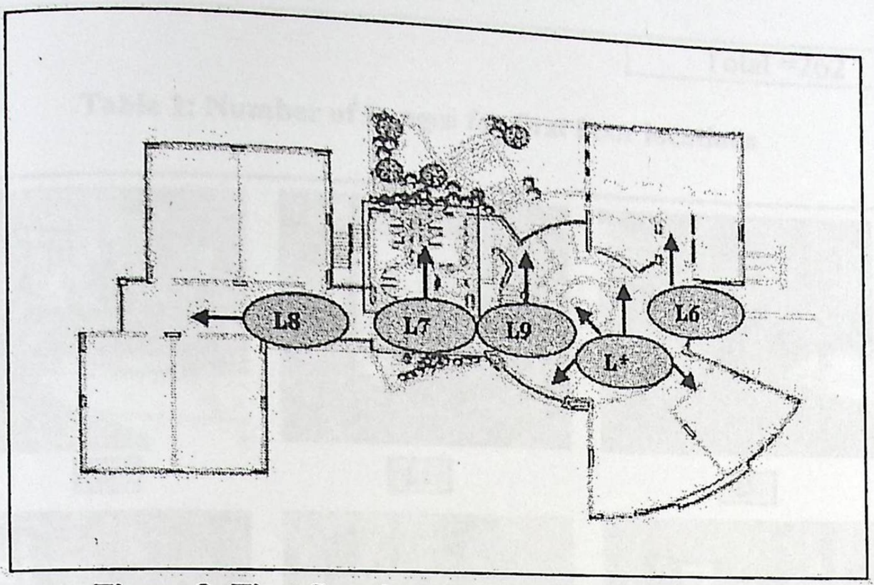


Figure 3: First floor locations at FFKITCE building

In Figure 3 location L* shows a set of locations which are detailed in Figure 4.

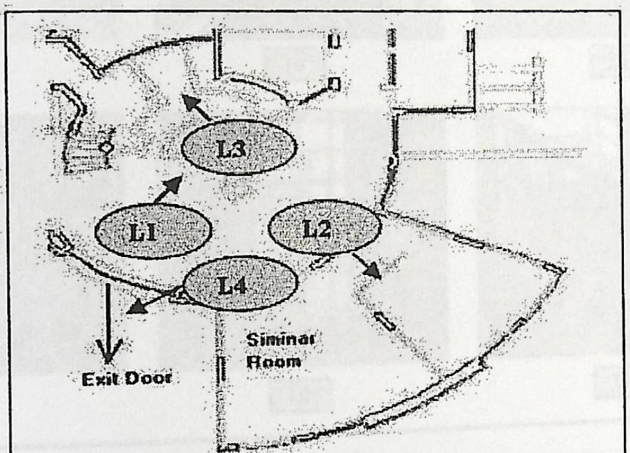


Figure 4: Reception hall locations

A set of 262 image were selected which are shown in Table 1. The table shows the locations label and number of images that represent each location in the reference database. Figure 5 represents samples of the reference database for each location in Table 1.

Location # / description	Location specification	Number of images in database
L1	Reception hall	48
L2	Seminar and testing room	21
L3	Back of the reception	46
L4	Exit Door	14
L5	Chairs	21
L6	Server and Eng. room	18
L7	First floor bathrooms	20
L8	Laboratory passage	39
L9	Staircase to second floor	35

	Total =262

Table 1: Number of images for first floor locations

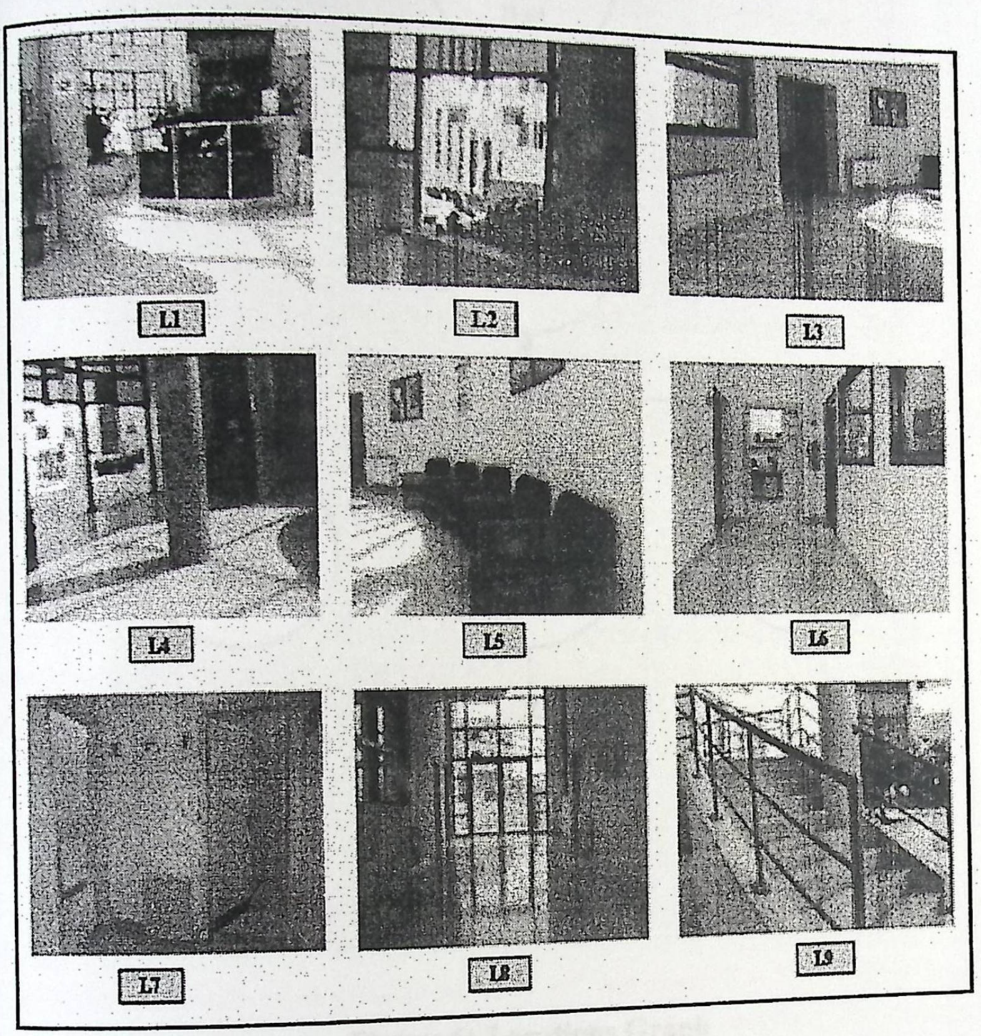


Figure 5: Samples of images from the reference database for each location

2.4 Location Graph

The above approach involves searching in the whole database for each query image which is time consuming. To over come this problem, a location graph was designed as seen in Figure 6, which shows that the possibilities of searching depending on the location that the person at; thus will prevent the system from searching in the whole database and so will decrease the time for searching. This graph also provides accurate tracking for the blind person, and will discard the locations that the blind person can not directly move to.

This graph is translated into look up table that contains navigation possibilities from a given location to another. Table 2 shows these possibilities values which were determined depending on our estimation.

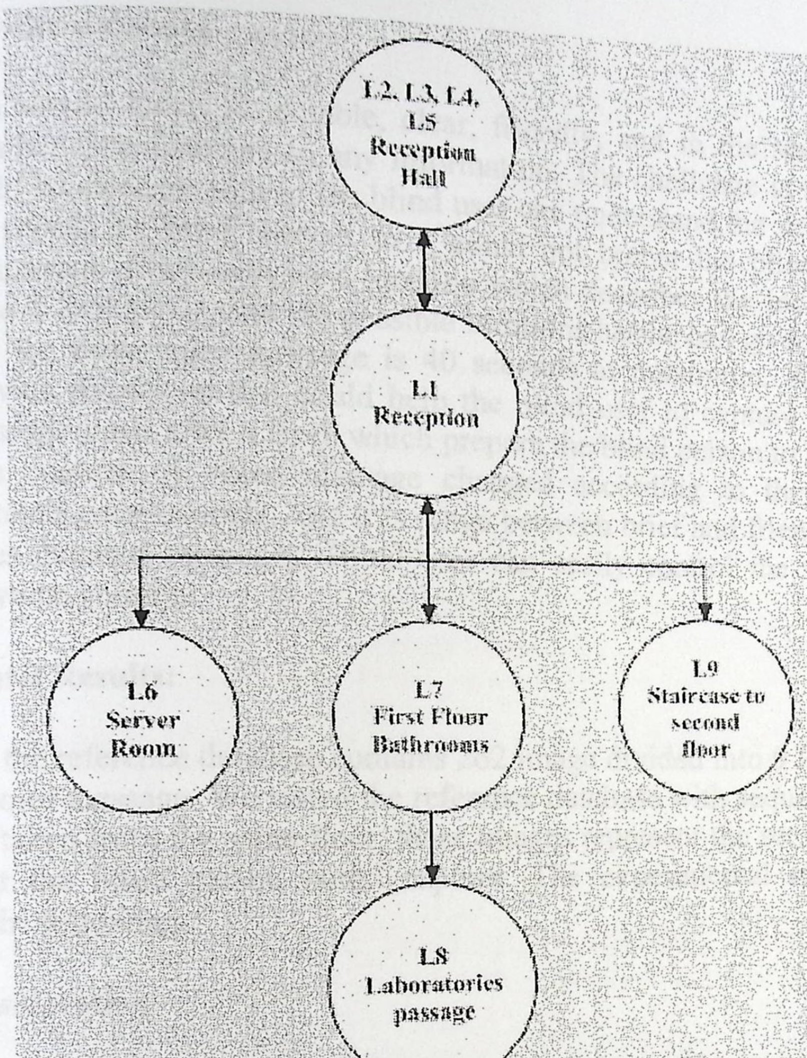


Figure 6: Locations Graph

Locations	L1	L2	L3	L4	L5	L6	L7	L8	L9
L1	0.95	0.83	0.80	0.88	0.85	0.50	0.50	0.20	0.85
L2	0.83	0.95	0.70	0.75	0.85	0.88	0.20	0.20	0.50
L3	0.80	0.70	0.95	0.80	0.75	0.77	0.20	0.20	0.70
L4	0.88	0.75	0.80	0.95	0.85	0.50	0.50	0.20	0.88
L5	0.85	0.88	0.75	0.85	0.95	0.85	0.20	0.20	0.50
L6	0.50	0.85	0.77	0.50	0.85	0.95	0.20	0.20	0.30
L7	0.50	0.20	0.20	0.50	0.20	0.20	0.95	0.88	0.85
L8	0.20	0.20	0.20	0.20	0.20	0.20	0.88	0.95	0.40
L9	0.85	0.50	0.70	0.88	0.50	0.30	0.85	0.40	0.95

Row = current location
Column =Previous Location

Table 2: Location possibilities percentages

.5 Message voice database

lind persons should listen to suitable, clear, friendly, and in the same time brief nformation rather than listening to any information. The challenge is that the voice essage should be clear enough so the blind user can understand the situation around im/her, and should be brief enough so it could not bother the blind person. We ssociated a message voice with each location which describes the location the user , what he/she is looking at, and the possible locations he/she can go to. The average uration time for each voice message is 40 seconds at maximum. Negative voice essages provide directions that could help the blind user to capture good images. he voice message starts after a Beeb which prepare the blind person to hear the voice essage. The content of voice massage changed according to the sequence of ocations the blind goes through. And it is range between brief and friendly messages, nd sometimes detailed messages. Blind user can easily control the voice through djusting the voice volume.

. Experimental Results:

s motioned, the reference database contains 262 image divided into 9 locations, each elated with voice message. We tested the reference database with sequence of images 467 images) taken from the same floor, these images represent the set of all possible essages that the blind person could capture. The experimental results will be xplained as the following.

.2 Time consideration

he average time that is needed for feature extraction for one incoming image is 0.7 econds this depends on the number of matches that is extracted from each image. igure 7 represents the maximum number of matches for 200 query image when ompared with the database. The average time that is needed for comparing one mage with the whole data base is 74 seconds because the searching for the keypoints n the database is sequential and not indexed. Indexing the database is quite possible, s proposed by Lowe [10] and will dramatically reduce the searching time and bring he system back into the real-time.

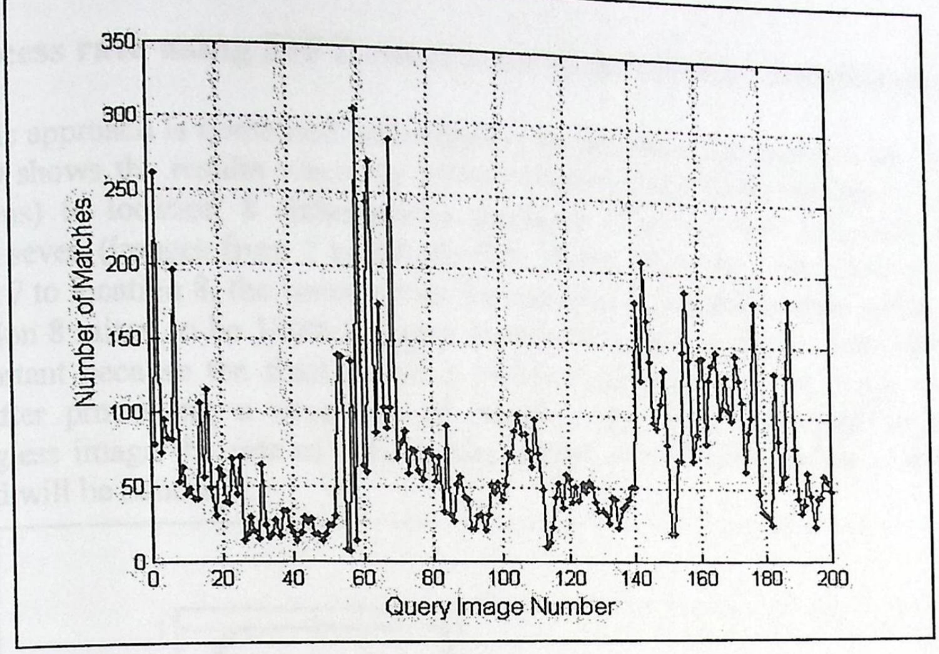


Figure 7: Numbers of matches for 200 query images with the whole database

3.2 Success rate using SIFT alone

Figure 8 illustrates the results of using SIFT algorithm alone for location determination. The tested locations contain other location from second floor. The success rate was 85.5% (401 images).

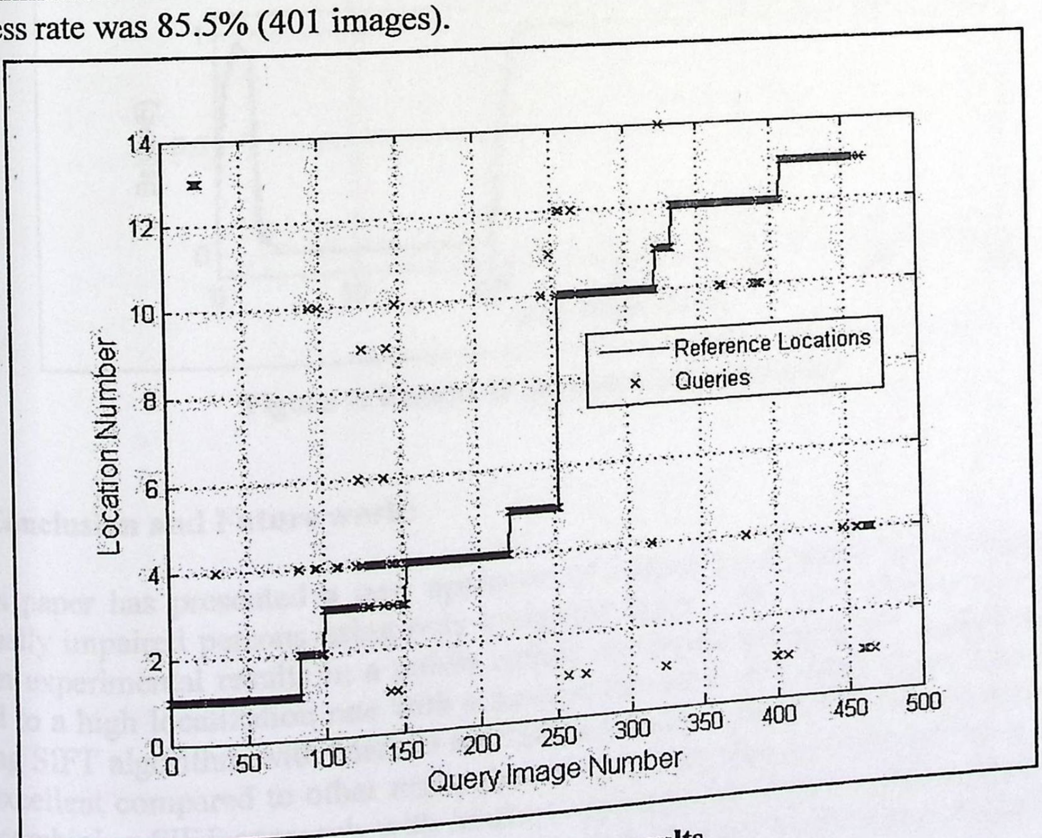


Figure 8: Testing results

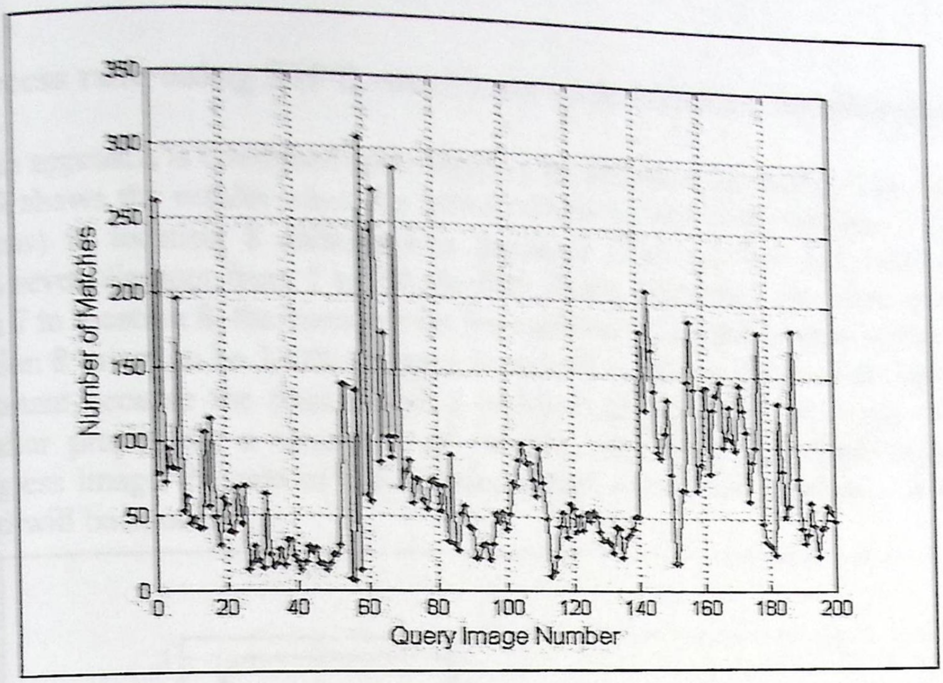


Figure 7: Numbers of matches for 200 query images with the whole database

3.2 Success rate using SIFT alone

Figure 8 illustrates the results of using SIFT algorithm alone for location determination. The tested locations contain other location from second floor. The success rate was 85.5% (401 images).

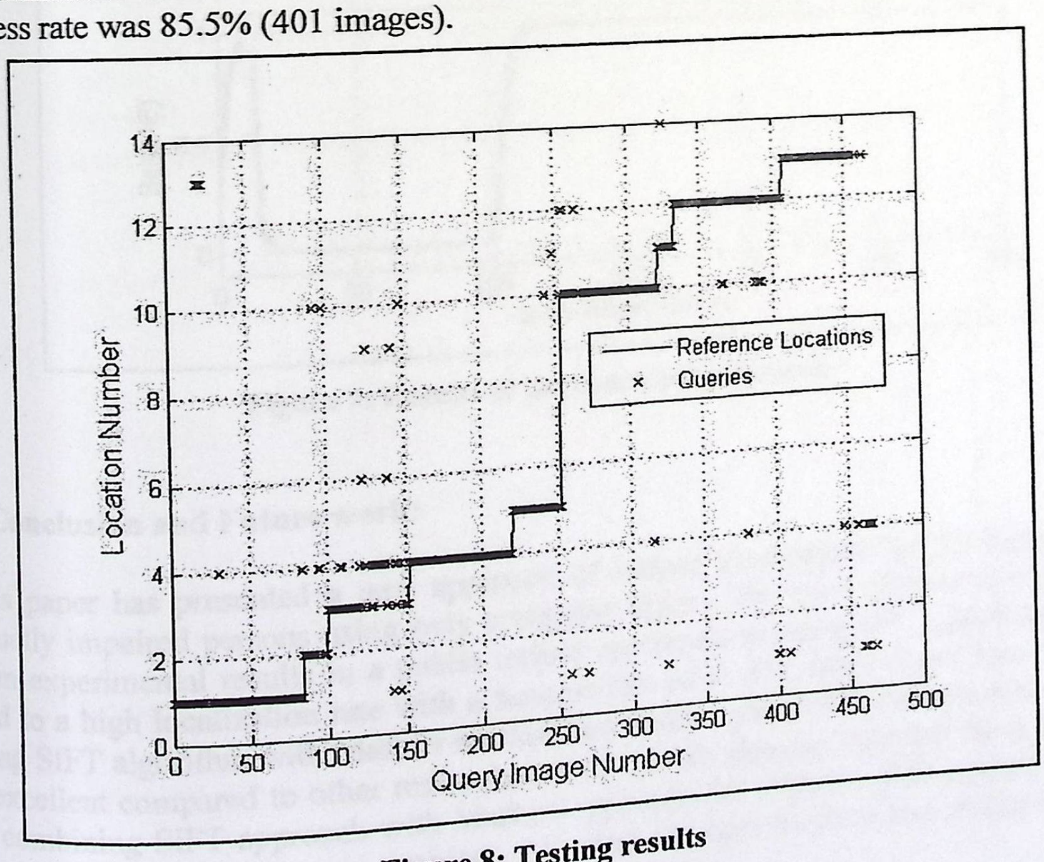


Figure 8: Testing results

3.3 Success rate using SIFT combined with Markov localization

After this approach is combined with Markov localization the success rate was 100%. Figure 9 shows the results when the blind person moved from location 7 (first floor bathrooms) to location 8 (laboratories passage). The success rate was 100% for location seven (images from 1 to100 the first chart) and when the blind moves form location 7 to location 8, the success rate for location 7 dropped to zero while the belief of location 8 raised to be 100% (images from 100 to 300 in the second chart). This is so important because the final decision which is presented by the voice message is taken after processing a sequence of images, and so if the blind user captures meaningless image the rate of the location belief would not affected , so mistaking captured will be handled .

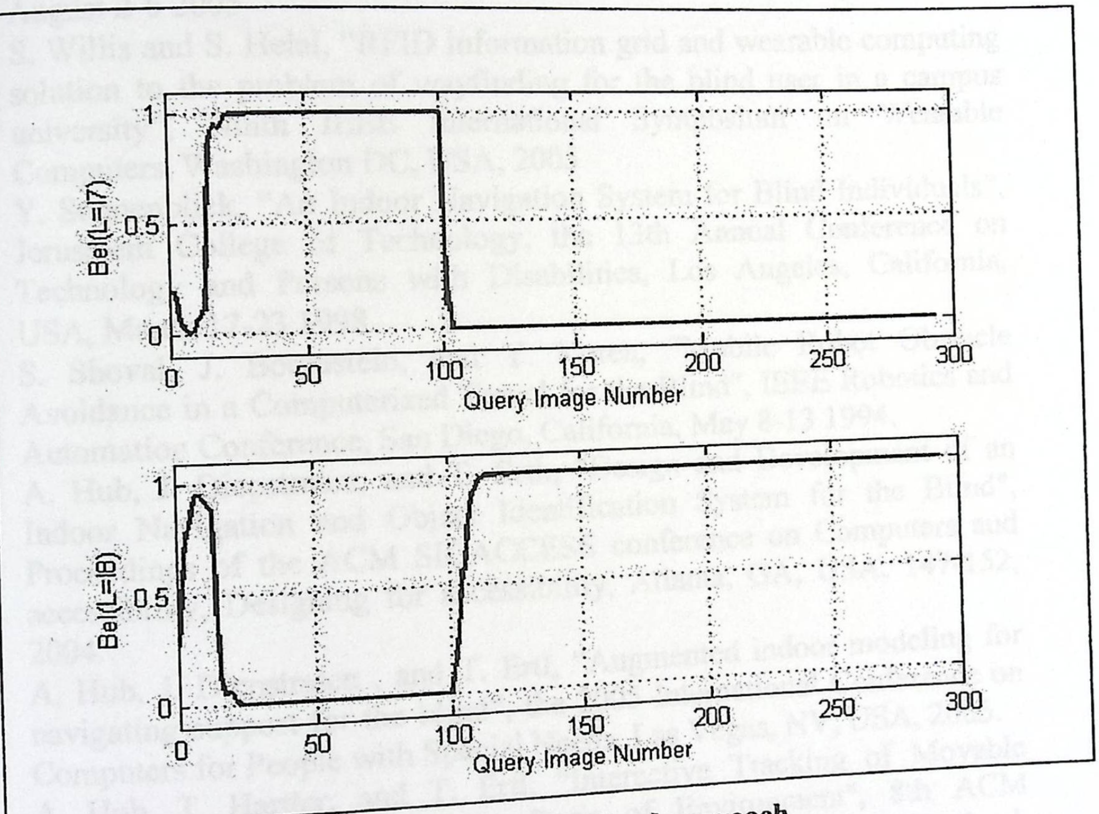


Figure 9: Results of the combined approach

4. Conclusion and Future work:

This paper has presented a new approach of indoor localization for the blind and visually impaired persons using only computer vision. The most obvious conclusion from experimental results in a public indoor environment using SIFT algorithm can lead to a high localization rate with a success rate of 85.5% which is not bad. While using SIFT algorithm with Markov method can lead to above 90% success rate witch is excellent compared to other researches made in this domain. Another thing is that by combining SIFT approach with Markov approach the system could operate under any environment without any problem. All what we have to do is just preparing the central database and look up table.

Future work includes the design of path planning to generate intelligent routes between indoor locations so we can reduce the error rate, to give the person appropriate directions. In addition, indexing the database is quite possible, as proposed by Lowe and will dramatically reduce the searching time and bring the system back into the real-time. The project team intends to speed up the system using Best Bin First Search (BBF) algorithm, which expected to return the closest neighbor with high probability [17].

5. References

[1] V. Kulyukin, C. Gharpure, J. Nicholson, " RoboCart: toward robot-assisted navigation of grocery stores by the visually impaired ", Intelligent Robots and Systems IEEE/RSJ International Conference, August 2-6 2005

[2] S. Willis and S. Helal, "RFID information grid and wearable computing solution to the problem of wayfinding for the blind user in a campus university", Ninth IEEE International Symposium on Wearable Computers, Washington DC, USA, 2005

[3] Y. Sonnenblick, "An Indoor Navigation System for Blind Individuals", Jerusalem College of Technology, the 13th Annual Conference on Technology and Persons with Disabilities, Los Angeles, California, USA, March 17-23 1998.

[4] S. Shoval, J. Borenstein, and Y. Koren, "Mobile Robot Obstacle Avoidance in a Computerized Travel for the Blind", IEEE Robotics and Automation Conference, San Diego, California, May 8-13 1994.

[5] A. Hub, J. Diepstraten, and T. Ertl., "Design and Development of an Indoor Navigation and Object Identification System for the Blind", Proceedings of the ACM SIGACCESS conference on Computers and accessibility, Designing for accessibility, Atlanta, GA, USA, 147-152, 2004.

[6] A. Hub, J. Diepstraten , and T. Ertl, "Augmented indoor modeling for navigating support for the blind", the 2005 International Conference on Computers for People with Special Needs, Las Vegas, NV, USA, 2005.

[7] A. Hub, T. Hartter, and T. Ertl, "Interactive Tracking of Movable Objects for the Blind on the Basis of Environment", 8th ACM SIGACCESS Conference on Computers and Accessibility, Portland, October 23-25 2006.

[8] Wikipedia , Free on-line encyclopedia, "CBIR", available at: http://en.wikipedia.org/wiki/CBIR

[9] H. Muller, N. Michoux, D. Bandon, and A. Geissbuhler, "A Review of Content-Based Image Retrieval Systems in Medical Applications - Clinical Benefits and Future Directions" , International Journal of Medical Informatics, Switzerland, 2004.

[10] D. Lowe, "Distinctive Image Retrieval from Scale-Invariant Keypoints", International Journal of Computer Vision, 2004.

[11] A. Halawani, A. Teynoe, L. Setia, G. Burnner, H. Burkhardt," Fundementals and Applications of Image Retrieval: An Overview", Datenbank-Spektrum, Heft 18, August 2006.

APPENDIX A.2

Project Code

```
/*************** Visual Localization Aid for the Blind ***********/
/* The system reads the query image from database then extract its features using SIFT algorithm then
the extracted features are compare with the all images in reference Data base to determine the blind
user location
*/

/* This code is in the defs.h header file which is share with all files */

/* From the standard C libaray: */
#include <stdlib.h>
#include <math.h>
#include <assert.h>
#include <stdio.h>
#include <string.h>

#define BufferK "Buffer/Keys.txt"             //the path for captured images keys
#define BufferI "Buffer/All_Imgs.txt"         //the path for captured images


#define THRESHOLD 15
#define NOLOCATIONS 10

// Assign Database Path
#define WholeDB "First_Floor/keys.txt"

/*------------------------------ Macros --------------------------*/

#define ABS(x)    (((x) > 0) ? (x) : (-(x)))
#define MAX(x,y)  (((x) > (y)) ? (x) : (y))
#define MIN(x,y)  (((x) < (y)) ? (x) : (y))
#define MAXCHARS 30

//This variable is shared with all files ,it stores the total numb of keypoint in the query image
int total_keypoints;

//Belief
float PrelocBel[NOLOCATIONS]; //Bel for the previous location
float LocBel[NOLOCATIONS]; //Bel for current location
//Predictive Density
float PredeDensity[NOLOCATIONS];

/*--------------------------- Structures --------------------------*/

/* Data structure for a float image*/
typedef struct ImageSt {
  int rows, cols;          /* Dimensions of image. */
  float **pixels;          /* 2D array of image pixels. */
  struct ImageSt *next;    /* Pointer to next image in sequence. */
} *Image;

/* Data structure for a keypoint.  Lists of keypoints are linked
   by the "next" field.*/
typedef struct KeypointSt {
```

```
float row, col;            /* Subpixel location of keypoint. */
float scale, ori;          /* Scale and orientation (range [-PI,PI]) */
unsigned char *descrip;    /* Vector of descriptor values */
struct KeypointSt *next;   /* Pointer to next keypoint in list. */
} *Keypoint;

typedef struct LocationSt{
char ID[2];
int best_no_matches;
float p;//probability
struct LocationSt *Next;
}Location;


Location *firstNode;


/*------------------------- Function prototypes ----------------*/
/* These are prototypes for the external functions that are shared
  between files.
*/

/*From main.c */

void Initiate_Bel(float *locations);

/* From util.c */
void FatalError(char *fmt, ...);
Image CreateImage(int rows, int cols);
Image ReadPGMFile(char *filename);
Image ReadPGM(FILE *fp);
void WritePGM(FILE *fp, Image image);
Keypoint ReadKeyFile(char *filename);
Keypoint ReadKeys(FILE *fp);


/* From match.c */

int FindMatches(Keypoint keys1,Keypoint keys2);
Keypoint CheckForMatch(Keypoint key, Keypoint klist);
int DistSquared(Keypoint k1, Keypoint k2);

/*From Capture_Image.c */
void Read_Qimg();
void Call_Sift(char Keyloc[81],char Imgloc[81]);

/*From Take_decision.c */

void Compare(char *DB,Keypoint qKeys);
void Find_Bel(void);
void Determine_Location(void);
int determine_play_sound(int currentloc);
void Play_Sound(int loc);
void Write_LogF(int loc);
void free_memory(Keypoint k2);
char *strfind(char string[],char c);
char * strfind2(int start,int end,char string[]);
void Creat_Link(char *,int noMatches);
```

```
/* This code is found in main.h file*/

#include "defs.h"

void main(void)
{

	//TurnCameraOn();

	//Initiate locations belief
		Initiate_Bel(PrelocBel);

	//This function Read Query Images then extract there Features then compare them with the
DB
		Read_Qimg();

}
// Th initial locations belief is distributed equally among all locations
void Initiate_Bel(float *locations)
{
int i;
for(i=1;i<10;i++)
	locations[1]=1/(NOLOCATIONS-1);


}


/*This code is found in capture_image.c*/

#include "defs.h"


void Read_Qimg()
{

	FILE *ImgsF,*KeysF;
	char Keyloc[81],Imgloc[81];

	ImgsF=fopen(BufferI,"r");
	KeysF=fopen(BufferK,"a");

	if(!(KeysF || ImgsF))
		{
			FatalError("Can't open Data Bases");
			exit(0);

		}

	while(fscanf(ImgsF,"%s",Imgloc)!=EOF)
	{	firstNode=NULL;
/*To remove pgm extention from the path of the image and replace it with txt extention*/
```

```
        strcpy(Keyloc,strcat(strfind2(0,12,Imgloc),".txt"));


        Call_Sift(Keyloc,Imgloc);

    }
    fclose(ImgsF);
    fclose(KeysF);

}

/*
call_sift function find the keypoints for the 15 images by calling siftWin32.exe
*/

void Call_Sift(char Keyloc[],char Imgloc[])
{
     Keypoint qKeys;
     char str[100]="siftWin32 <";
     strcat(strcat(strcat(str,Imgloc),"> "),Keyloc);

     // Call sift algorithm
     (void)system(str);

     //read query image keypoints
     qKeys=ReadKeyFile(Keyloc);

     /*only Accept the images whose features number more than a threshold */
     if(total_keypoints>=THRESHOLD)
             Compare(WholeDB,qKeys);

}


/*This code is found in Take_Decision.c*/


#include "defs.h"

// Lookup Table
float Lcations_Matrix[10][10]={
0, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0, 0.95, 0.83, 0.80, 0.88, 0.85, 0.50, 0.50, 0.20, 0.85,
0, 0.83, 0.95, 0.70, 0.75, 0.88, 0.85, 0.20, 0.20, 0.50,
0, 0.80, 0.70, 0.95, 0.80, 0.75, 0.77, 0.20, 0.20, 0.70,
0, 0.88, 0.75, 0.80, 0.95, 0.85, 0.50, 0.50, 0.20, 0.88,
0, 0.85, 0.88, 0.75, 0.85, 0.95, 0.85, 0.20, 0.20, 0.50,
0, 0.50, 0.85, 0.77, 0.50, 0.85, 0.95, 0.20, 0.20, 0.30,
0, 0.50, 0.20, 0.20, 0.50, 0.20, 0.20, 0.95, 0.88, 0.85,
0, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.88, 0.95, 0.40,
0, 0.85, 0.50, 0.70, 0.88, 0.50, 0.30, 0.85, 0.40, 0.95
        };

int lastLoc=0;

/*The compare function compares the query image keypoints that have been taken from th webcam
```

with the keypoints from the reference data base
*/

```
void Compare(char *DB,Keypoint qKeys)
{
FILE *orgDBKeys;
char orgKey[81],maxImg[81];
Keypoint k1 = NULL, k2 = NULL;
int noMatches=0,querykeys,i,j;
Location *temp;



        //save the query image total keypoints
        querykeys=total_keypoints;
        orgDBKeys=fopen(DB,"r");
        if (!orgDBKeys)
            {
                        FatalError("Error.Can't open DataBase");
                        exit(0);
            }
        while(fscanf(orgDBKeys,"%s",orgKey)!=EOF )
            {

                        k2 = ReadKeyFile(orgKey);


                        if (k2 == NULL)
                        {
                                FatalError("Command line does not specify all images and keys.");
                                continue;
                        }
                        /*return number of matches between query image keys and refernce image
keys*/
                        noMatches=FindMatches(qKeys, k2);
                        strcpy(orgKey,strfind2(1,3,strrchr(orgKey,'\\')));
                        /* creat a link list for the best number of matches for each location*/
                        Creat_Link(orgKey,noMatches);
                        free_memory(k2);

                }//End of While

        temp=firstNode;

        //Calculate locations probability ;
        while(temp!=NULL)
        {
        temp->p=temp->best_no_matches/(querykeys+0.0);
        temp=temp->Next;
        }
```

```
// Find belief for each location using markove algorithm
Find_Bel();
Determine_Location();

for(i=1;i<NOLOCATIONS;i++)
PrelocBel[i]=LocBel[i];


fclose(orgDBKeys);
free_memory(qKeys);

}


void free_memory(Keypoint k)
{
        Keypoint temp;
        temp=k;
        k=k->next;
while(k!=NULL)
{free(temp);
        temp=k;
        k=k->next;
}
free(temp);
}

void Creat_Link(char loc[],int noMatches)
{
        int found=0; //not found
        int i;

        Location *newNode,*tempptr,*lastNode;

        newNode=(Location *)malloc(sizeof(Location));
        if(newNode==(Location *)NULL)
                {
                printf("\n\nError");
                }

        if(firstNode==NULL)// Create the first node
        {

                //LinkListSize++;
                strcpy(newNode->ID,loc);
                newNode->best_no_matches=noMatches;
                newNode->Next=NULL;
                firstNode=newNode;

        }
        else // increase Rept member in an existing node
        {
                tempptr=firstNode;

        while((tempptr!=NULL) && !found)
        {
```

```
        if(strcmp(tempptr->ID,loc)==0)
        {       if((tempptr->best_no_matches)<noMatches)
                        tempptr->best_no_matches=noMatches;
                found=1;
        }
        lastNode=tempptr;
        tempptr=tempptr->Next;
    }

    if(!found)// Add new node to the end of the link list
    {

            strcpy(newNode->ID,loc);
            newNode->best_no_matches=noMatches;
            newNode->Next=NULL;
            lastNode->Next=newNode;

    }

    }//End of While


}// End of the fuction

void Find_Bel(void)
{
float theta=0.0,sum=0.0;
Location *temp;
int loc,i,j;

temp=firstNode;

//Prediction Phase (motion model)
for(i=1;i<NOLOCATIONS;i++)
{       sum=0.0;
        for(j=1;j<NOLOCATIONS;j++)
                sum+=Lcations_Matrix[i][j]*PrelocBel[j];

        PredeDensity[i]=sum;

}

//Perception Model
while(temp!=NULL)
        {
                loc=atoi(strfind2(1,2,temp->ID));
                LocBel[loc]=temp->p*PredeDensity[loc];

                theta=theta+LocBel[loc];
                temp=temp->Next;
        }
//Normalize the Beliefe
for(i=1;i<NOLOCATIONS;i++)
        LocBel[i]=LocBel[i]/theta;
```

```
// free locations
temp=firstNode;
firstNode=firstNode->Next;
while(firstNode!=NULL)
{free(temp);
        temp=firstNode;
        firstNode=firstNode->Next;

}
free(temp);


}

void Determine_Location(void)
{

int i,loc=0,play=0;

for(i=1;i<NOLOCATIONS;i++)
            if(LocBel[i]>.8)
                    loc=i;

play=determine_play_sound(loc);
            if(play){
                    Play_Sound(loc);
                    Write_LogF(loc);
                    lastLoc=loc;
                            }


}


int determine_play_sound(int currentLoc)
{

if(lastLoc==currentLoc) // currrent location is the same as previous location
        return 0; // don't play the sound
else            // current location is not the same as last loc
                return 1; //Play the sound

}


void Play_Sound(int loc)
{
char soundF[5];
strcpy(soundF,loc);
strcat(soundF,".wav");
system(soundF);

}
```

```
void Write_LogF(int loc)
{
        FILE *logF;
        logF=fopen("LogFile.txt","a");
        fprintf(logF,"%d\n",loc);
        fclose(logF);

}


char *strfind(char string[],char c)
{
        int i=0;
        char *retstr,*orig;
        retstr=(char *)malloc(sizeof(char));

        orig=retstr;

        while(string[i]!=c)
        *retstr++=string[i++];

        *retstr='\0';//End of the string
        return orig;

}


char * strfind2(int start,int end,char string[])
{
        int i=0;
        char *retstr,*orig;
        retstr=(char *)malloc(sizeof(char));

        orig=retstr;
        while((start!=end)&&(*retstr++=string[start++]))
        ;

        *retstr='\0';

        return orig;
}

/*This code is found in match.c */

/************************************************************************/
Demo software: Invariant keypoint matching.
Author: David Lowe

match.c:
This file contains a sample program to read images and keypoints, then
  draw lines connecting matched keypoints.
************************************************************************/

#include "defs.h"


/* Given a pair of images and their keypoints, pick the first keypoint
  from one image and find its closest match in the second set of
```

```
keypoints. Then write the result to a file.
*/
int FindMatches(Keypoint keys1,Keypoint keys2)
{
 Keypoint k, match;
 Image result;
 int count = 0;


 /* Match the keys in list keys1 to their best matches in keys2.
 */
 for (k= keys1; k != NULL; k = k->next) {
  match = CheckForMatch(k, keys2);


   if (match != NULL) {
       count++;

   }
  }

       return count;
}

/* This searches through the keypoints in klist for the two closest
  matches to key. If the closest is less than 0.6 times distance to
  second closest, then return the closest match. Otherwise, return
  NULL.
*/
Keypoint CheckForMatch(Keypoint key, Keypoint klist)
{
  int dsq, distsq1 = 100000000, distsq2 = 100000000;
  Keypoint k, minkey = NULL;

  /* Find the two closest matches, and put their squared distances in
    distsq1 and distsq2.
  */
  for (k = klist; k != NULL; k = k->next) {
   dsq = DistSquared(key, k);

   if (dsq < distsq1) {
        distsq2 = distsq1;
        distsq1 = dsq;
        minkey = k;
   } else if (dsq < distsq2) {
        distsq2 = dsq;
   }
  }

  /* Check whether closest distance is less than 0.6 of second. */
  if (10 * 10 * distsq1 < 6 * 6 * distsq2)
   return minkey;
  else return NULL;
}

// Return squared distance between two keypoint descriptors.
```

```
int DistSquared(Keypoint k1, Keypoint k2)
{
  int i, dif, distsq = 0;
  unsigned char *pk1, *pk2;

  pk1 = k1->descrip;
  pk2 = k2->descrip;

  for (i = 0; i < 128; i++) {
    dif = (int) *pk1++ - (int) *pk2++;
    distsq += dif * dif;
  }
  return distsq;
}
```

/*This code is found in util.c*/

```
/************************************************************************
Demo software: Invariant keypoint matching.
Author: David Lowe

util.c:
This file contains routines for creating floating point images,
reading and writing PGM files, reading keypoint files, and drawing
lines on images:
    Image CreateImage(row,cols) - Create an image data structure.
    ReadPGM(filep) - Returns list of images read from the PGM format file.
    WritePGM(filep, image) - Writes an image to a file in PGM format.
    DrawLine(image, r1,c1,r2,c3) - Draws a white line on the image with the
      given row, column endpoints.
    ReadKeyFile(char *filename) - Read file of keypoints.
*************************************************************************/

#include "defs.h"
#include <stdarg.h>

/* -------------------- Local function prototypes ---------------- */

float **AllocMatrix(int rows, int cols);
void SkipComments(FILE *fp);


/*--------------------- Error reporting ------------------------*/

/* This function prints an error message and exits.  It takes a variable
   number of arguments that function just like those in printf.
*/
void FatalError(char *fmt, ...)
{
   va_list args;
```

```
    va_start(args, fmt);
    fprintf(stderr, "Error: ");
    vfprintf(stderr, fmt, args);
    fprintf(stderr,"\n");
    va_end(args);
    exit(0);
}

/*---------------- Routines for image creation -----------------*/

/* Create a new image with uninitialized pixel values.
*/
Image CreateImage(int rows, int cols)
{
    Image im;

    im = (Image) malloc(sizeof(struct ImageSt));
    im->rows = rows;
    im->cols = cols;
    im->pixels = AllocMatrix(rows, cols);
    im->next = NULL;
    return im;
}

/* Allocate memory for a 2D float matrix of size [row,col]. This returns
   a vector of pointers to the rows of the matrix, so that routines
   can operate on this without knowing the dimensions.
*/
float **AllocMatrix(int rows, int cols)
{
    int i;
    float **m, *v;

    m = (float **) malloc(rows * sizeof(float *));
    v = (float *) malloc(rows * cols * sizeof(float));
    for (i = 0; i < rows; i++) {
        m[i] = v;
        v += cols;
    }
    return (m);
}

/*---------------- Read and write PGM files -----------------*/

/* This reads a PGM file from a given filename and returns the image.*/

Image ReadPGMFile(char *filename)
{
    FILE *file;

    /* The "b" option is for binary input, which is needed if this is
       compiled under Windows. It has no effect in Linux.
    */

    file = fopen (filename, "rb");
```

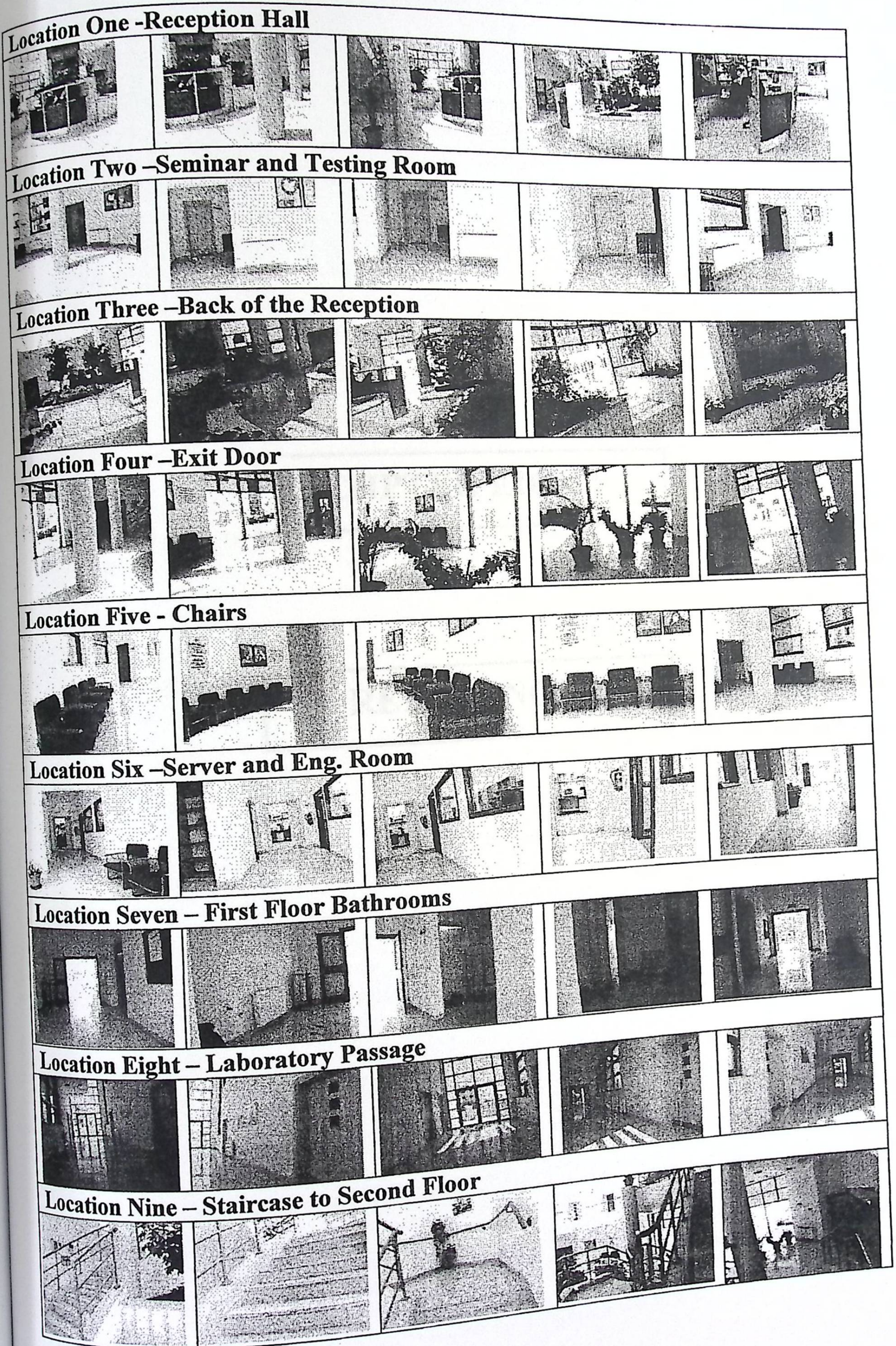
```
if (! file)
    FatalError("Could not open file: %s", filename);

return ReadPGM(file);
}

/* Read a PGM file from the given file pointer and return it as a
float Image structure with pixels in the range [0,1]. If the file
contains more than one image, then the images will be returned
linked by the "next" field of the Image data structure.
  See "man pgm" for details on PGM file format. This handles only
the usual 8-bit "raw" PGM format. Use xv or the PNM tools (such as
pnmdepth) to convert from other formats.
*/

Image ReadPGM(FILE *fp)
{
  int char1, char2, width, height, max, c1, c2, c3, r, c;
  Image image, nextimage;

  char1 = fgetc(fp);
  char2 = fgetc(fp);
  SkipComments(fp);
  c1 = fscanf(fp, "%d", &width);
  SkipComments(fp);
  c2 = fscanf(fp, "%d", &height);
  SkipComments(fp);
  c3 = fscanf(fp, "%d", &max);

  fgetc(fp);  /* Discard exactly one byte after header. */

  /* Create floating point image with pixels in range [0,1]. */

  image = CreateImage(height, width);
  for (r = 0; r < height; r++)
    for (c = 0; c < width; c++)
      image->pixels[r][c] = ((float) fgetc(fp)) / 255.0;

  /* Check if there is another image in this file, as the latest PGM
     standard allows for multiple images. */

  SkipComments(fp);
  if (getc(fp) == 'P') {
    ungetc('P', fp);
    nextimage = ReadPGM(fp);
    image->next = nextimage;
  }
  return image;
}

/* PGM files allow a comment starting with '#' to end-of-line. Skip
   white space including any comments.
*/

void SkipComments(FILE *fp)
{
   int ch;
```

```
fscanf(fp," ");      /* Skip white space. */
while ((ch = fgetc(fp)) == '#') {
  while ((ch = fgetc(fp)) != '\n'  &&  ch != EOF)
      ;
  fscanf(fp," ");
}
ungetc(ch, fp);      /* Replace last character read. */
}

/* Write an image to the file fp in PGM format.
*/

void WritePGM(FILE *fp, Image image)
{
  int r, c, val;

  fprintf(fp, "P5\n%d %d\n255\n", image->cols, image->rows);

  for (r = 0; r < image->rows; r++)
    for (c = 0; c < image->cols; c++) {
      val = (int) (255.0 * image->pixels[r][c]);
      fputc(MAX(0, MIN(255, val)), fp);
    }
}

/*---------------------- Read keypoint file -----------------------*/

/* This reads a keypoint file from a given filename and returns the list
  of keypoints.
*/

Keypoint ReadKeyFile(char *filename)
{
  FILE *file;

  file = fopen (filename, "r");
  if (! file)

                FatalError("Could not open file: %s", filename);


  return ReadKeys(file);
}


/* Read keypoints from the given file pointer and return the list of
  keypoints.  The file format starts with 2 integers giving the   total
  number of keypoints and the size of descriptor vector for each
  keypoint (currently assumed to be 128). Then each keypoint is
  specified by 4 floating point numbers giving subpixel row and
  column location, scale, and orientation (in radians from -PI to
  PI). Then the descriptor vector for each keypoint is given as a
  list of integers in range [0,255].
```

```
*/
Keypoint ReadKeys(FILE *fp)
{
  int i, j, num, len, val;
  Keypoint k, keys = NULL;

//To read number of keypoints (num)  , to read descriptor (len)
  if (fscanf(fp, "%d %d", &num, &len) != 2)
        FatalError("Invalid keypoint file beginning.");

  if (len != 128)
        FatalError("Keypoint descriptor length invalid (should be 128).");

        total_keypoints=num;

                /*This for loop create link list where each node represent one key
                each new node is add to the begining of the link list
        keys is the pointer where it always point to the first node "Iman Younis" */

  for (i = 0; i < num; i++) {
   /* Allocate memory for the keypoint. */
   k = (Keypoint) malloc(sizeof(struct KeypointSt));
   k->next = keys;
   keys = k;
   k->descrip = malloc(len);

   if (fscanf(fp, "%f %f %f %f", &(k->row), &(k->col), &(k->scale),
                  &(k->ori)) != 4)
       FatalError("Invalid keypoint file format.");

   for (j = 0; j < len; j++) {
        if (fscanf(fp, "%d", &val) != 1 || val < 0 || val > 255)
          FatalError("Invalid keypoint file value.");
        k->descrip[j] = (unsigned char) val;
   }

  }

        fclose(fp);

  return keys;
}
```

APPENDIX A.3

Samples from first floor locations


Location One -Reception Hall
Location Two –Seminar and Testing Room
Location Three –Back of the Reception
Location Four –Exit Door
Location Five - Chairs
Location Six –Server and Eng. Room
Location Seven – First Floor Bathrooms
Location Eight – Laboratory Passage
Location Nine – Staircase to Second Floor

CHPTER EIGHT

REFERENCES

8.1 References

[1] S. Willis and S. Helal, “RFID information grid and wearable computing solution to the problem of wayfinding for the blind user in a campus university”, Ninth IEEE International Symposium on Wearable Computers, Washington DC, USA, 2005

[2] A. Hub, J. Diepstraten, and T. Ertl, “Augmented indoor modeling for navigating support for the blind”, the 2005 International Conference on Computers for People with Special Needs, Las Vegas, NV, USA, 2005.

[3] V. Kulyukin, C. Gharpure, J. Nicholson, " RoboCart: toward robot-assisted navigation of grocery stores by the visually impaired ", Intelligent Robots and Systems IEEE/RSJ International Conference, August 2-6 2005

[4] S. Shoval, J. Borenstein, and Y. Koren, “Mobile Robot Obstacle Avoidance in a Computerized Travel for the Blind", IEEE International Conference on Robotics and Automation, San Diego, California, May 8-13 1994.

[5] Y. Sonnenblick, “An Indoor Navigation System for Blind Individuals”, Jerusalem College of Technology, the 13th Annual Conference on Technology and Persons with Disabilities, Los Angeles, California, USA, March 17-23 1998.

[6] A. Hub, J. Diepstraten, and T. Ertl., "Design and Development of an Indoor Navigation and Object Identification System for the Blind", Proceedings of the ACM SIGACCESS conference on Computers and accessibility, Designing for accessibility, Atlanta, GA, USA, 147-152, 2004.

[7] A. Hub, T. Hartter, and T. Ertl, "Interactive Tracking of Movable Objects for the Blind on the Basis of Environment Models and Perception-Oriented Object Recognition Methods", Proceedings of the 8th ACM SIGACCESS Conference on Computers and Accessibility, October 23-25, Portland, OR, USA, 111-118, 2006.

[8] A. Hub, S. Kombrink, and T. Ertl, "Tactile-Acoustical Navigation Assistant for Real and Virtual Explorations of the Environment", Conference Proceedings of the 1st Multi-disciplinary Vision Rehabilitation & Research Conference ENVISION 06, September 21-24, Kansas City, MO, USA, 48, 2006.

[9] M. May, "outdoor\indoor navigation for blind and visually impaired individuals ", 20th Conference on Technology and Persons with Disabilities, Los Angeles, California, USA, 2006.

[10] School of Library, online Archival and Information Studies, "image indexing", [Online Document], Available at: http://www.slais.ubc.ca/people/students/studentprojects/C_Wanczyck i/libr517/future.html

[11] Y. Rui, T. Hugan, M Ortega, S. Mehrotra, "Image Retrieval Current Techniques, Promising Direction and Open Issues ", Published in the Journal of Visual Communication and Image Representation, 1999.

[12] A. Halawani, A. Teynoe, L. Setia, G. Burnner, H. Burkhardt," Fundementals and Applications of Image Retrieval: An Overview", Datenbank-Spektrum, Heft 18, August 2006.

[13] Wikipedia , Free on-line encyclopedia, "CBIR", available at: http://en.wikipedia.org/wiki/CBIR

[14] H. Muller, N. Michoux, D. Bandon, and A. Geissbuhler, "A Review of Content-Based Image Retrieval Systems in Medical Applications - Clinical Benefits and Future Directions" , International Journal of Medical Informatics, Switzerland, 2004.

[15] Sharmin Siddique, "A Wavelet Based Technique for Analysis and Classification of Texture Images", Indian Conference on Computer Vision, Graphics & Image Processing, 2002.

[16] Lexico Publishing Group, LLC, "shape", [Online Document], Available at http://dictionary.reference.com/search?q=shape

[17] D. Lowe, "Distinctive Image Retrieval from Scale-Invariant Keypoints", International Journal of Computer Vision, 2004.

[18] David G. Lowe "Object recognition from local scale-invariant features", Computer Science Department, University of British Columbia, Canada, unknown year.

[19] D. Fox, W. Burgard, F. Dellaert and S. Thrun, "Markov Localization for Mobile Robots in Dynamic Environments", Journal of Artificial Intelligence Research, 1999, and International Conference on Robotics and Automation (ICRA99), May, 1999.