Palestine Polytechnic University
College of Administrative Science and Informatics
Department of Information technology

Mobile Localization Using Particle Swarm Optimization

Project students:
Anas Al-sharabati
Rafat Ihrezat
Hamouda Hammad

Supervisor:
Hashem Tamimi

2009

Contents

List of tables

List of figures

Abstract

This project aims to guide a blind person in order to find her/his location in indoor environment, where no Geographical Positing System can help.

The environment that surrounds the person is represented by a set of images taken from by a mobile phone. Initially the system captures images of many locations, extracts their features and stores them. After that, a new feature is extracted from the image of the unknown location and is compared with the stored features. Finally, the location is sent to the blind person using a voice message. To overcome the problem of having different locations with similar images, the particle swarm optimization approach is used.

A simulation program has been developed in this project; it has a set of stored images that represents the environment and it accepts input images to demonstrate the localization progress. This program allows the researcher to change some variables and observe the effects on the searching process. The user can also select a search path and the search process is shown graphically until best location is reached.

The project will be available at the PPU website so that the researchers, who are interested in the localization process using PSO, can benefit from it. This will also reflect the research done at the PPU and increase the popularity of its website.

CHAPTER ONE

1

Introduction

Chapter one

- *Overview.*
- *Project objectives.*
- *Project hypothesis and assumptions.*
- *General description of the system.*
- *The organization of the document.*

1.1 Overview

People depend on sight to achieve most of their work, including determining their location, Blind people have trouble with action that requires sight, this is why they use tools to aid them in such situation, such as using the walking stick to aid them on walking so they can avoid hitting objects while moving, and when they want to go to far places they will need people to give them directions.

There is a lot of research on tools that help the blind achieve their work easily. One of the oldest and most known is Braille system which is a method used by the blind to read. Some of the resent researches focuses on today's technology like computers, the Internet and the mobiles. For example blind people can use computers by using hardware and software that accepts voice commands or hardware that produce output in Braille system.

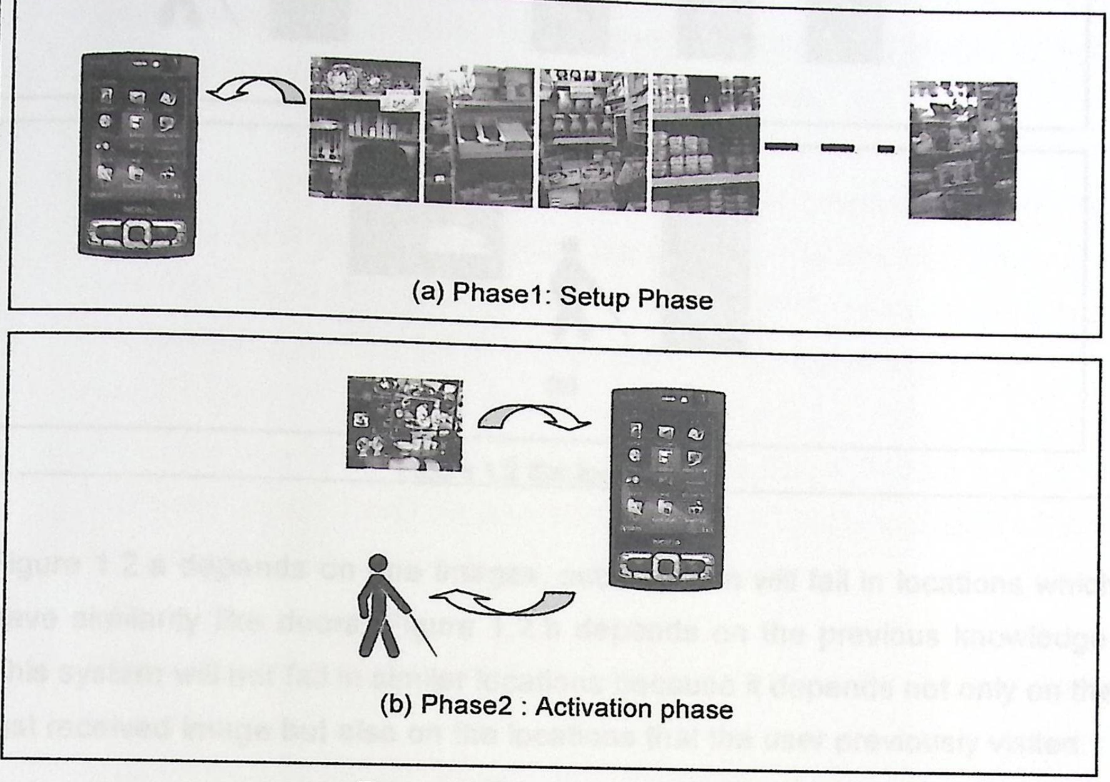
In this research, we propose a computer based approach that help the blind person find her/his location in indoor environment using a mobile phone, which improve the previous research "Mobile-based Visual Localization Aid for the Blind" [1].

The scenario of the localization process is done in two phases as follows:

1. Setup Phase: First, the system should have previous knowledge about the environment, by storing the important features from the images of the environment into the mobile phone. The description of each location is also stored in the mobile. This phase can be accomplished by a helping person rather than the blind person himself.

2. Activation Phase: While the system is running, it keeps receiving new images from the surrounding and comparing them with the previous knowledge. The result of this comparison is the stored location with the most similarity to the new location. Finally, a voice message is send back to the blind person with the description of that location.

The scenario is shown in the following figure:



(a) Phase1: Setup Phase

(b) Phase2 : Activation phase

Figure 1.1 The system's scenario

Figure 1.1.a shows the process of storing the environment; the process needs a large amount of images that covers the entire environment and the sequence is important. Figure 1.1.b shows how activation phase works, it depends on an image to compare it with stored information in order to find the result, and then send the location message to the user.

The scenario above works if each location has some distinctive images that makes it different from other locations. This is not true in real situations. For example, there are doors in many locations that are similar.

To solve this problem, the system should depend on the history of the locations that the blind person visited before the current location.

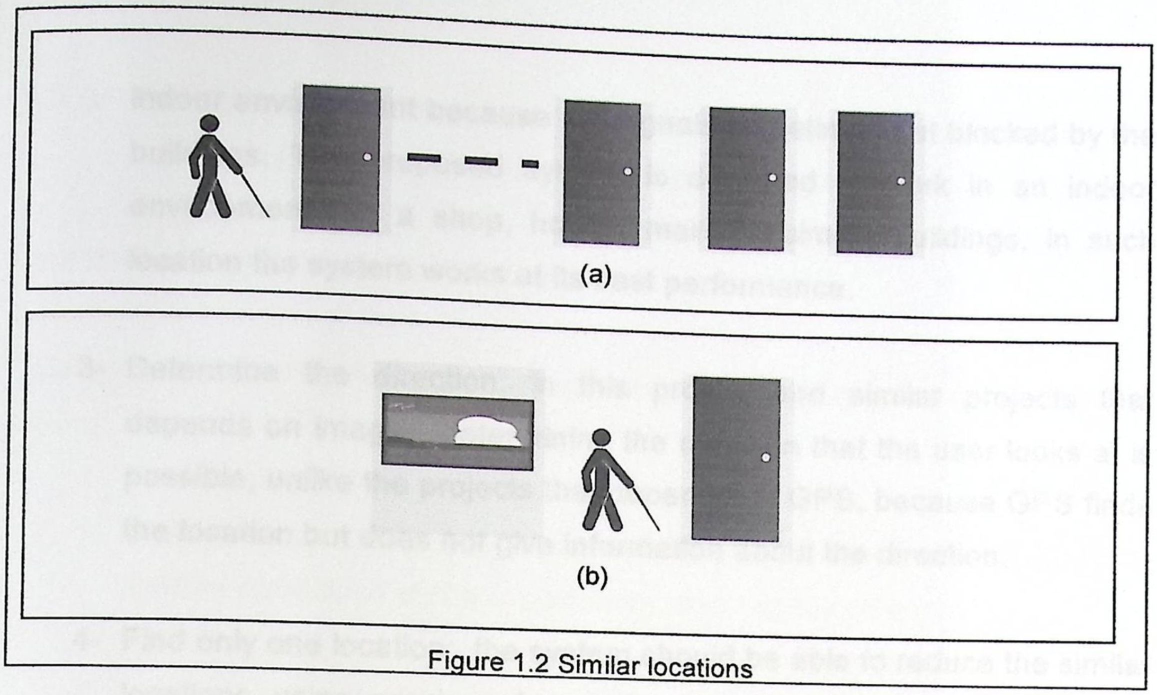


Figure 1.2 Similar locations

Figure 1.2.a depends on one images, such system will fail in locations which have similarity like doors. Figure 1.2.b depends on the previous knowledge. This system will not fail in similar locations because it depends not only on the last received image but also on the locations that the user previously visited.

To take the previous information into consideration and combine it with the current knowledge, we need an artificial intelligence approach. In this research we use Particle Swarm Optimization (PSO) to the track the location.

1.2 Project Objectives:

This project is expected to achieve the following:

1- Determine the location with accuracy: the location which the system finds must be very close to the place that needs to be located, in other words, the distance between the person and the result of the system must be small.

2- Finding the location in indoor environment: Finding the location in an indoor environment is harder than it in outdoor environment because technologies like Global Positioning System (GPS) do not work in an indoor environment because the signals sometimes get blocked by the buildings. The proposed system is designed to work in an indoor environment like a shop, house, malls or similar buildings, in such location the system works at its best performance.

3- Determine the direction: in this project and similar projects that depends on images, determining the direction that the user looks at is possible, unlike the projects that depends on GPS, because GPS finds the location but does not give information about the direction.

4- Find only one location: the system should be able to reduce the similar locations using previous knowledge images. This means the blind person will hear one description for his location.

5- Reduce storage space for storing the environment: the system stores the environment as a set of features which reduces the storage needed greatly compared to storing the environment as images, this is important for a system that is expected to work on mobile phones which have limited memory.

6- Real-time response: the processing time must be reduced as much as possible to achieve real time response; the blind person must not wait for a long time for the result.

7- Accurate testing: the strategies and methods used should be tested to ensure that the hypotheses are correct.

8- System cost: the system should be of a low cost. In the process of capturing the environment only a good quality camera is needed to give the images and the output environment file is a simple file that does not need special software to deal with. The user of the system only needs only a modern mobile phone embedded with a camera to use the system.

9- Detection and recovery: there should be an approach to detect if the system is not able to find the right location and recover.

1.3 Project hypothesis and assumptions

The project uses the following hypotheses and it is expected to work under them:

1- The environment is an indoor environment: - this means that the lighting conditions of the environment can slightly change.

2- A representation of the environment is previously stored with enough samples of images: The system only searches within previously entered samples and the number of these samples must be large enough to reduce the probability of error.

3- The system reads the images from the locations sequentially: in the setup phase, the images should be captured as a sequence while the user walks in the environment; if the captured Images are not sequential the error will increase.

4- During the setup phase the images should be of a high quality and contains good information in order to extract valuable and representative information.

5- The images should contain information about the place: if an object blocked the sight of the location, like a person standing in front of the location, the accuracy will be reduced.

1.4 General description of the system

The environment is captured as a sequence of images with a slightly difference between neighbor images, then the system extracts the environment's features and stores them into a file.

The system work in the following steps:

1. Load the data that represent the environment.
2. Wait for a new image from the blind person.
3. Find the features of the new image.
4. Compare the features with stored data that represent the environment.
5. Use PSO to find the corresponding location.
6. Output the location information as voice.
7. Wait for another image and repeat 3, 4, 5 and 6.

1.5 The organization of the document

1- Chapter 2 presents the technologies and methods that are used in this project like PSO and Bhattacharyya distance also some related studies.

2- Chapter 3 explains the project in detail, describing the theory, and shows all the system components, how each component works and how they work with each other.

3- Chapter 4 shows how the project was implemented in a simulation program.

4- Chapter 5 discuses the testing environment and the testing process, then show the results that the testing found.

5- Chapter 6 concludes this report. Some recommendations for future works are discussed.

CHAPTER TWO

2

Background

Chapter Two

- ***Overview.***
- ***Related projects.***
- ***Image features.***
- ***Content-based image retrieval.***
- ***RGB color model.***
- ***Histogram.***
- ***Bhattacharyya distance.***
- ***Particle swarm optimization.***

2.1 Overview

This chapter reviews some pervious projects that aim to support the blind persons. Also it explains the technologies and methods that are used in this project so that the reader can understand the project details.

2.2 Related projects

This project and the coming related projects does not depend on the Global Positioning System (GPS), because it fails in the in-door environments as a result of losing the satellite signal when blocked by a building, also the GPS is very expensive for high accuracy and inefficient with low accuracy. GPS does only give information about the position of the user, in a system that helps the blind the direction also matters.

The first related project which is called "Body Mounted Vision System for Visually Impaired Outdoor and Indoor Wayfinding Assistance" [10]. It works in two steps, the first one is to capture the environment as a video then extract the features to build a 3D-map of the environment then provide this map to the system. The second step is to use the system, the user should have a wearable computer that has a camera then as the user walks into the environment the system will check the 3D-map and give the user directions. the advantage of this system that it does not fail when a new object pass through the environment like a passing person, but this project needs to make the blind person carry big equipment as shown in Figure 3.1.

Another related project depends on a single image to find the location it is called "Mobile-based Visual Localization Aid for the Blind" [1]. In this project the environment is entered to the system as a set of histograms, and the system with the stored environment are installed into a mobile phone, the user enters an image of his surrounding area to get information about the location by a voice message. This project does not need any special equipment and the process of capturing the environment is easy, but this project can only be applied in place that are different from each other, if this project is used in a place that have a number of similar places it will fail to find the location.

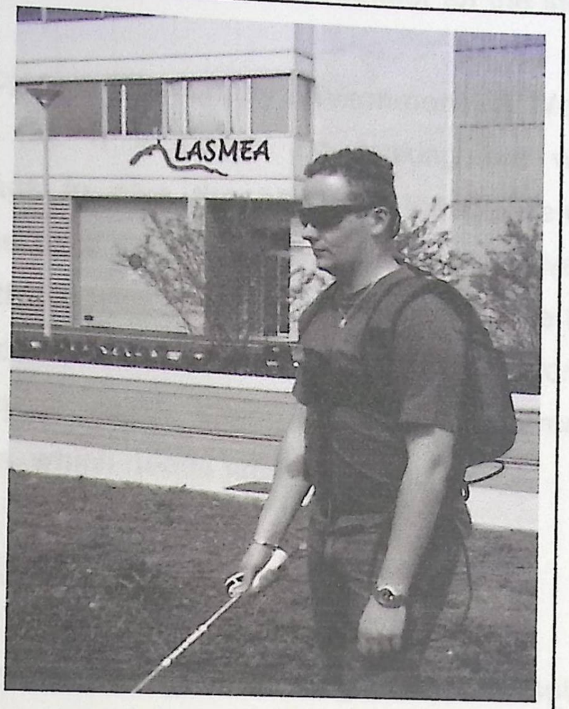


Figure 2.1 Project equipment

Picture taken from [10]

The next project uses augmented objects it is called “Augmented Indoor Modeling for Navigation Support for the Blind” [3]. The user carries a small device with him, this device has several sensors, when the device is directed toward an object, it gives information about it. All objects in the environment must have radio-frequency identification (RFID) or simply a tag that responds to the reader and sends its information to the user, also all parts of the environment that contains stairs, doors or other critical objects must have a special tag on the ground to inform the user about what is around him. This project is accurate but the place that uses this project must be pre-engineered with RFIDs which is inefficient and cost a lot of time and effort.

Another project uses stereo images called "Interactive Localization and Recognition of Objects for the Blind" [4], the environment must be have its features extracted first and the information about the locations is associated to the features. Then when the user activates the system, it compares the captured images with the stored features to find the location and give the user a feedback. This uses stereo images which enable depth detection, but it still needs a wearable computer and a special sensor for the stereo images.

The final related project is "The Chatty Environment - A World Explorer for the Visually Impaired" [2]. It is based on making the whole environment augmented with tags to give the blind an independent environment for themselves, and the user should have a device that gives him information about the objects, this device can communicate with the tags and also with the user. This project focuses on giving the user information about the objects and about the way to reach them, but the environment must be fully augmented with tags, which needs time and effort.

2.3 Theoretical background

In the coming section, we will give a background on a set of topics that will help the reader understand the report.

2.3.1 Image features

An image feature is a small piece of information that represents the image itself by applying some operations to the image which is called feature extraction [8]. Image features are important in image processing; most of the image processing application needs to extract features from the images in order to understand the image content.

Image feature may be local or global depending on the way that the features are extracted. If the feature is taken by applying some operations on some certain pixels and on neighbor pixels, then the feature is local, the result of such operation is mostly a point of interest which is a place in the image that has special characteristics. When extracting the feature by using an operation that is applied to the whole image it is called a global feature, the most known global feature is the image histogram.

2.3.2 Content-based image retrieval

Content-based image retrieval (CBIR) is the way of searching for and retrieving an image from database that contains a large number of images, unlike image search which needs a text description for each image, Content-based image retrieval only needs an input image and the system will extract and match the features [7].

The Content-based image retrieval system has a large set of images with their features extracted and stored. When an image is fed to the system, it will extract the image features then compare the extracted features with the stored ones in order to find the most similar image or images to the input image. Figure 2.2 shows the process:

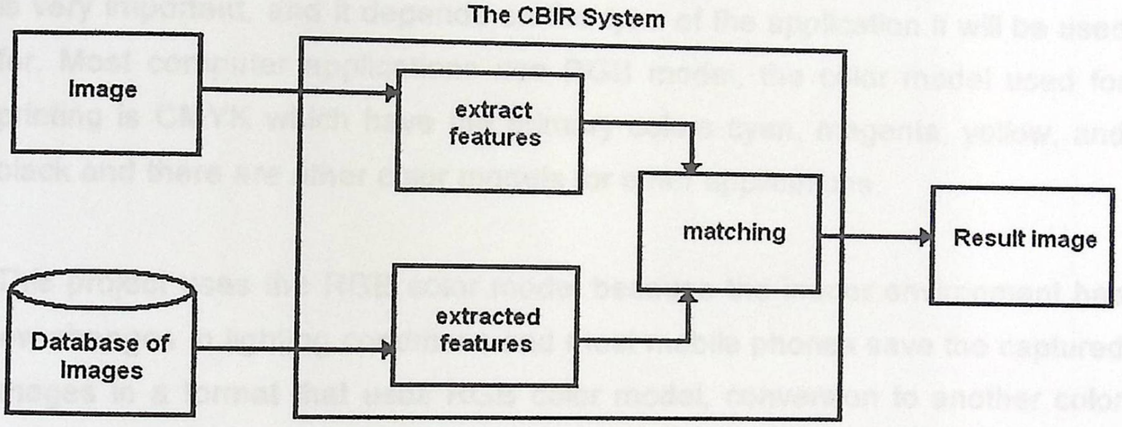


Figure 2.2 Block diagram of The Content-based image retrieval.

The idea of CBIR is used in this project to find the most similar locations by comparing the user location with the stored locations; the system captures an image from the user location then use CBIR to look for the similar locations.

2.3.3 RGB color model

RGB color model is one of the most used color models, RGB stands for Red Green and Blue. The RGB color model is a color model that uses these three intensity components to represent color [8].

The RGB color model represent color by giving a value to each of its intensity components, a higher value means higher intensity. The other colors are displayed as the result of combining the primary color intensities, because RGB is an additive color model.

For example the result of add the three components at their highest intensity is white, the absence of the three color results black, and when the three color are equal but not at their highest or lowest value the result is gray. Different intensities between components result other colors. The value for each component is from 0 to 255 which gives 16777216 colors that the RGB color model can represent.

In image processing, the color model is important because most operations in image processing depend on the color value. The selection of the color model is very important, and it depends on the type of the application it will be used for. Most computer applications use RGB model, the color model used for printing is CMYK which have the primary colors cyan, magenta, yellow, and black and there are other color models for other applications.

This project uses the RGB color model because the indoor environment has low changes in lighting conditions and most mobile phones save the captured images in a format that uses RGB color model, conversion to another color model will consume the processer speed.

2.3.4 Histogram

The histogram is one of the images features, it shows how individual brightness levels are in the image [8], so when you look at a histogram you can tell if the color is mostly have high or low intensity.

The histogram is made by counting each color in the image, by going through the image's pixels to find its color and increase a counter for each. A gray-scale image will have one histogram while a color image that uses RGB color model will have three histograms one for the red component and other two for green and blue components. The data in the histogram are divided on a number of bins or intervals, for example a 64 bins histogram for a gray-scale image will store every four values together. For example the color intensities 0, 1, 2 and 3 will be stored in the one bin and the color intensities 4, 5, 6 and 7 will be stored in another bin.

The histogram is effective in image retrieving process if the image is slightly different in color, but if the color change much, like if the image is taken once in a sunny day then the same image is taken in a rainy day, it will not be effective. Figure 2.3 shows two gray-scale images on the left a slightly dark one, on the right a brighter image see how the histogram is affected.

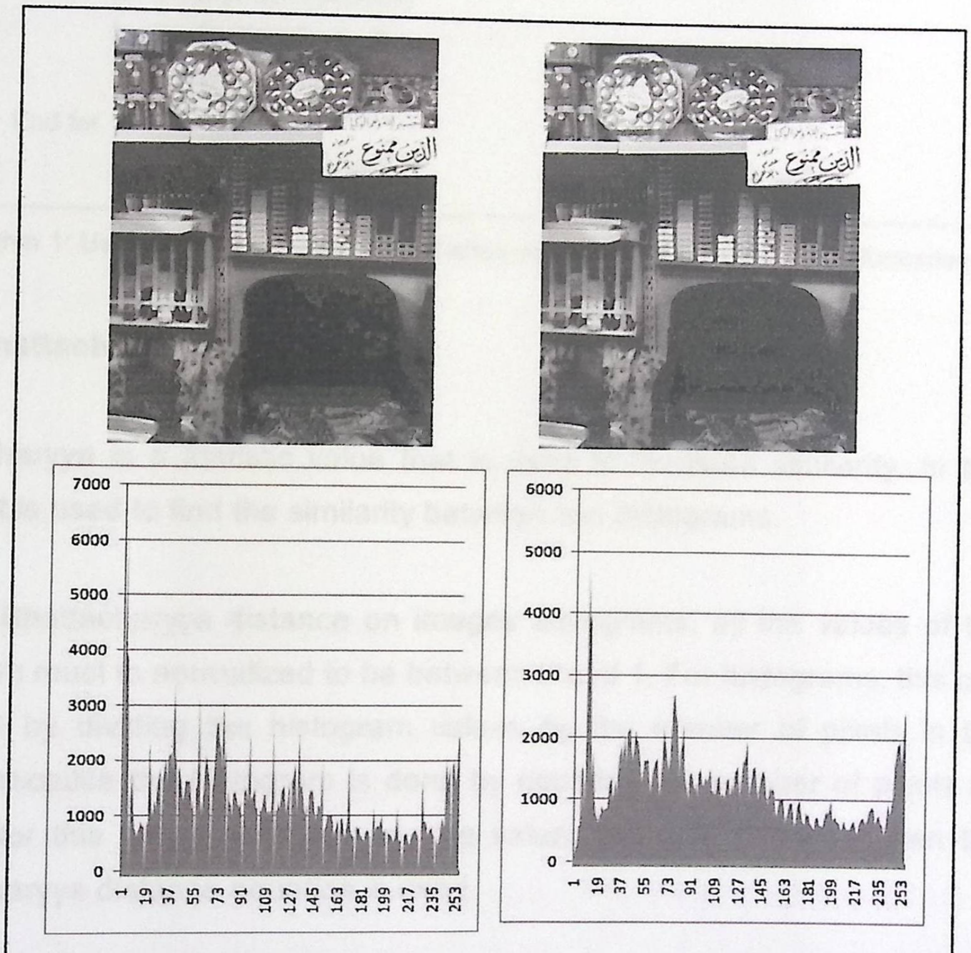


Figure 2.3 Image histogram

3D histogram can be used when dealing with RGB colors, it stores the values in a 3 dimension array of bins. The array has three indexes for red, green and blue. Using 3D histogram is better than using three histograms one for each color intensity, because 3D histogram is more representative.

The following algorithm shows how to build 3D histogram

```
D = Max_COLOR/BINS
Initialize hist3 elements to zero
For index1 = 0 to IMG_WIDTH
        For index2 = 0 to IMG_HEIGHT
                r=Red[index1][index2]
                g=Green[index1][index2]
                b=Blue[index1][index2]
                hist3[r][g][b]++
        End for
End for
```

Algorithm 1: Using this algorithm the result array will have the 3D histogram information.

2.3.5 Bhattacharyya distance

Bhattacharyya is a statistic value that is used to measure similarity. In this project it is used to find the similarity between two histograms.

To use Bhattacharyya distance on images histograms, all the values of the histogram must to normalized to be between 0 and 1. For histograms, this can be done by dividing the histogram values by the number of pixels in the image, because the histogram is done by counting the number of pixels for each color this way guarantees to give values between 0 and 1, then the Bhattacharyya distance equation is used.

$$\text{Bhattachar yya} = \sqrt{1 - \sum_{i}^{n} \sqrt{\sum a_i . \sum b_i}} \quad \ldots(2.1)\ [6]$$

Where a, b are the first histogram and second histograms. N is the number of bins.

If the value of Bhattacharyya distance is close to 0 the images are similar, the farther the value is from 0 the more are the images different.

Similar images are most likely to represent the same location, while images with a value close to 1 are from different locations. The value of Bhattacharyya distance in this project reflects the possible match between locations.

Another way to measure the similarity between two histograms is mean square error (MSE), by applying the following equation.

$$MSE = \frac{1}{N}\sum_{i}^{n}\left(a_i - b_i\right)^2 \quad \ldots(2.2)\ [9]$$

Where a, b are the first histogram and second histograms. N is the number of bins.

Bhattacharyya distance is used in this project because it can compare images with different sizes.

2.3.6 Particle swarm optimization

Particle swarm optimization (PSO) is an evolutionary algorithm and it is a part of the swarm intelligence algorithms that tries to simulate social models, like the birds flocking, ant colonies and fish schooling. PSO was described the first time in 1995 by James Kennedy and Russell C. Eberhart [5]. This algorithm tries to reach the best solution in a search space; such algorithm is good for crowd simulation.

In this algorithm a number of particles search the search space for a target. Each particle search its surrounding area for a local best solution and all the particles are affected by the best location found which is called the global best. The movement of the particles toward the goal is iterative and is controlled by the following two equations:

$$v_i(t) = \omega \cdot v_i(t-1) + c_1 \cdot \alpha_1 \cdot (pl_i - x_i(t-1)) + c_2 \cdot \alpha_2 \cdot (pg - x_i(t-1)) \ldots(2.3)$$

$$x_i(t) = x_i(t-1) + v_i(t) \ldots\ldots(2.4)$$

i the particle number.

t the iteration number.

x_i is the position of the particle.

v_i is the particle's velocity.

ω, c_1, c_2 are scalar values.

pg is the global best.

pl_i is a local best.

α_1, α_2 are random values.

Equation 2.3 calculates the velocity of the particle, which is how many unites the particle will move, this decision is affected by surrounding area of the particle and the global best value and random values to make the particles search different locations.

Equation 2.4 is for changing the pervious position with a new one using the velocity, the velocity is added to the current position of the particle which gives the new position.

The choice of the formula's parameters is critical for the system; choosing bad values will make the particles move in an inefficient way. Some formula's parameters have some recommended values by the previous studies, still the values needs little change depending on the project implementation and the system needs, the following Figure 2.3 shows the general process of PSO.

The loop in the algorithm is stopped by some criteria like finding the target, or after a number of iterations.

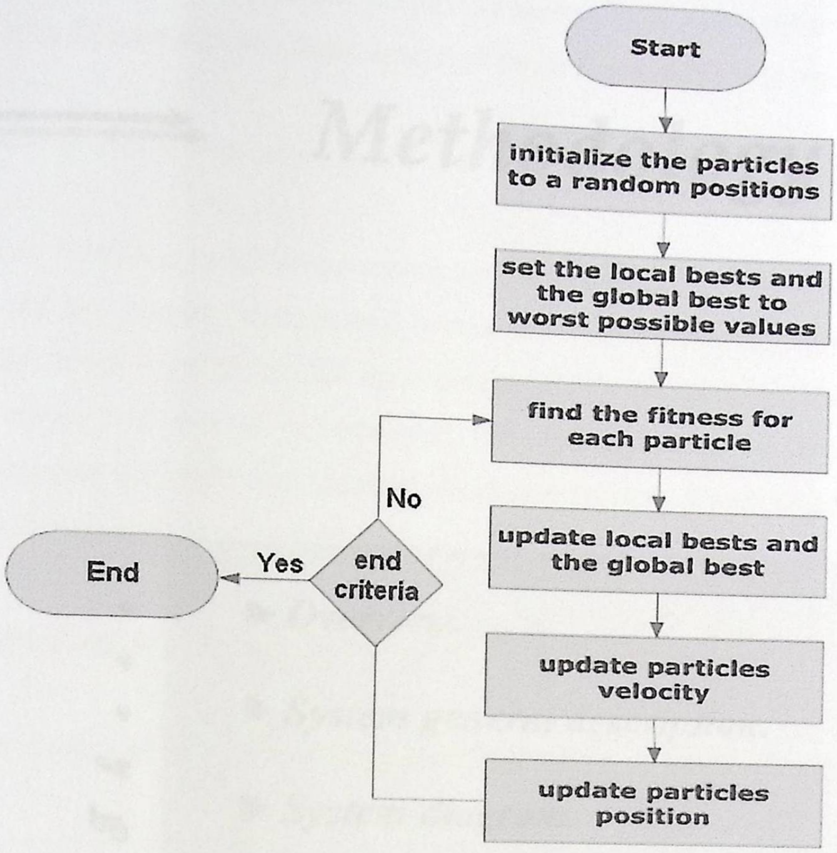

Start
initialize the particles to a random positions
set the local bests and the global best to worst possible values
find the fitness for each particle
update local bests and the global best
update particles velocity
update particles position
No
end criteria
Yes
End

Figure 2.4 The PSO algorithm flowchart

We will use PSO in this project in order to search for the location of the blind person and track his position as he moves.

Summary

We reviewed some projects that are related to this project. And we gave background of topics that are important for understanding the system.

CHAPTER THREE

3

Methodology

Chapter Three

- *Overview.*
- *System general description.*
- *System diagram.*
- *The system detailed description.*
- *The Testing methodology.*

3.1 Overview

This chapter will describe in details how the system works, by showing how all the system parts work with each other, the technologies and methods that are described in Chapter 2 and their contribution in the system will be discussed.

3.2 System general description

The system needs a predefined environment to search in, this environment is represented as images histograms. When the system starts, it first loads the histograms, then it waits for an input image which will come from the mobile camera, when the image is received, the system finds its histogram, the system process it to find the location. Finally it returns a message describing the location and wait for another image.

3.3 System diagram

The general view of the system is shown in the diagram in Figure 3.1:

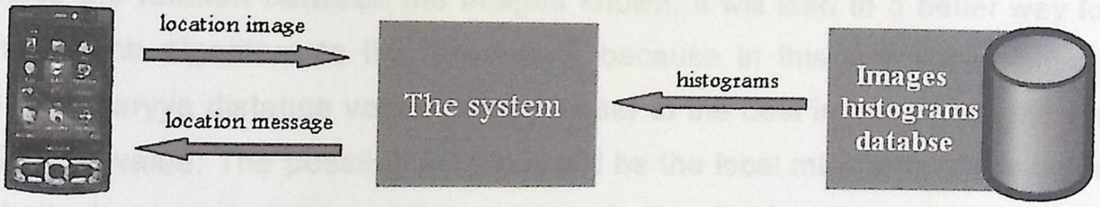


Figure 3.1 System diagram

The system starts by loading the histograms, then keeps taking images and uses them to find the location and output the location information

3.4 The system detailed description

The system needs a predefined environment which consists of large set histograms extracted from a large number of images of environment. These images should cover all the areas in the environment and every area must have several images in order to increase the accuracy of the system.

Calculating the histogram is done before the search system starts, this will reduce the time that the system needs to start and will reduce the storage space than storing them as images. The sequence of creating the environment should be in a way that enables the system to know the neighborhood relations between the locations.

When the system receives an input image it calculates the image histogram, as described it Chapter 2, to use it in the comparing process. Then the system compares the captured image histogram with all the environment histograms by finding Bhattacharyya distances (see Equation 2.1) the result will be a set of Bhattacharyya distances that is equal to the number of images in the environment. The Bhattacharyya distance is considered as the fitness function in the PSO algorithm. The best fitness is the value of 0 and the worst is 1.

Local and global minimum

Since the relation between the images known, it will lead to a better way for the search algorithm to find the result, because in this way the neighbors Bhattacharyya distance values will be closer to the best image Bhattacharyya distance value. The possible locations will be the local minimum values in the Bhattacharyya distance value set, and the best value of the whole environment is the global minimum value which represents the best location that must be found.

In order to make this clearer, see Figure 3.2, where the captured image is similar to the one at index 34, the neighbors values are close to its value and the index 34 is a local minimum in its area and a global minimum for the environment.

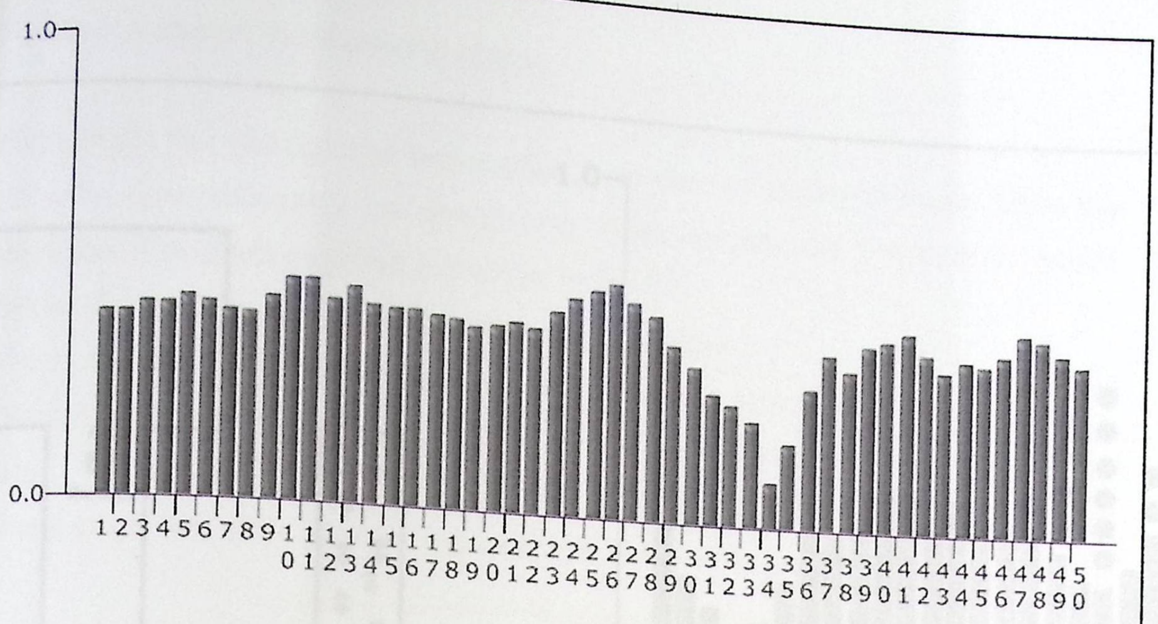


Figure 3.2 Bhattacharyya distance chart, where X axis is the image number and Y axis is the Bhattacharyya distance value

If several locations are similar they will be represented as local minima, the particles will fall into them. The PSO approach will drag the particles into the global minima since the global best affects the velocity value (Equation 2.3).

The particles will stay in the global minima until the Bhattacharyya distance value starts to get lower after a number of images, this will make the particles go to the next global best location.

Getting lost

The system has the ability to correct itself in case it get lost, this is done by observing the Bhattacharyya distances for some time, when the system finds that all distances values are larger than 0.25 (the value is selected after some experiments) it will realize that the system is following wrong path or that the environment is an unknown one, in this case the system will restart the search process in order to find the correct location by going into the initialization step (see Figure 3.4 description). The figure 3.3 describes how the system gets lost.

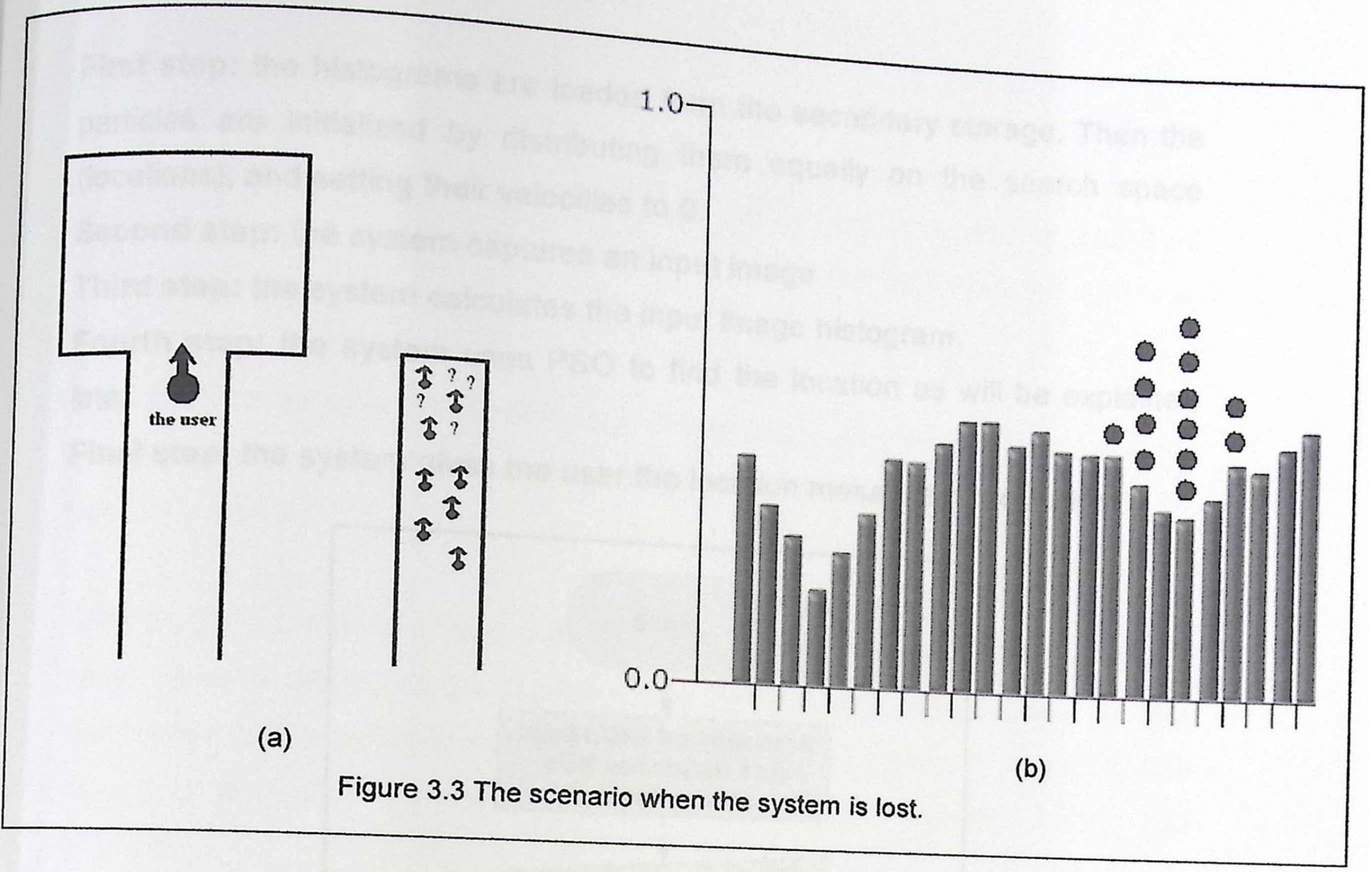


Figure 3.3 The scenario when the system is lost.

In Figure 3.3.a the lost case occurred because all the particles follow a similar location, which is another corridor. In Figure 3.3.b shows that the Bhattacharyya distance value where the particles are grouped (the bold dots) became high and the real location is not followed, so the system will find a need to restart the search.

Each location must be associated with a message that describes the location, this is essential for the system to do the output process correctly. When the location is found the system uses the message that associated with the location (voice message in case of a blind user) and outputs it to the user. The flowchart in Figure 3.4 shows the complete process of the system.

Figure 3.4 shows the following steps:

First step: the histograms are loaded form the secondary storage. Then the particles are initialized by distributing them equally on the search space (locations), and setting their velocities to 0.

Second step: the system captures an input image

Third step: the system calculates the input image histogram.

Fourth step: the system uses PSO to find the location as will be explained later.

Final step: the system gives the user the location message if requested.

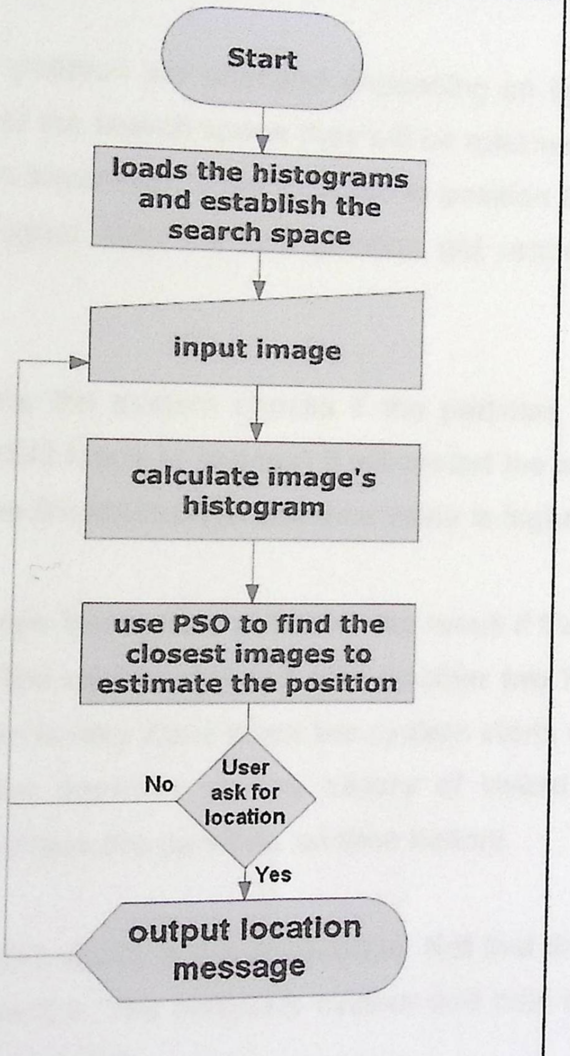

Start
loads the histograms and establish the search space
input image
calculate image's histogram
use PSO to find the closest images to estimate the position
User ask for location
No
Yes
output location message

Figure 3.4 The system's flowchart

The system and PSO

The system uses two iterations in PSO for each image, a higher number will make the particles quickly gather which reduce the probability of finding similar locations. Then for each PSO loop the system updates the local best depending on the particles location, then it takes the best local minimum as the global best.

Then it calculate how the particle must move depending on the local best and the global best using Equation 2.3, the result of this equation will change the particles velocity.

After that the particles position are changed depending on their velocities, if some particles get out of the search space they will be returned to it, a particle that moves to a position lower than 0 get returned to position 0, a particle that moves to a position higher than the last position get returned to the last position.

At the end of each loop the system checks if the particles are lost, if the system stay lost for 8 PSO loops (4 images) it will restart the search process. We mean by lost that the Bhattacharyya distance value is higher than 0.25.

At the end of the two loops the system will show the result if the user request, and then it will capture the next image and start another two PSO iterations. Note that the initialization is only done when the system starts or when it gets lost because the system depends on the history of visited locations the initialization process will erase the pervious location history.

Figure 3.5 shows how PSO works in this application. Not that the figure is part of the PSO step in Figure 3.4. The complete system and how all its parts are connected is shown in Figure 3.6.

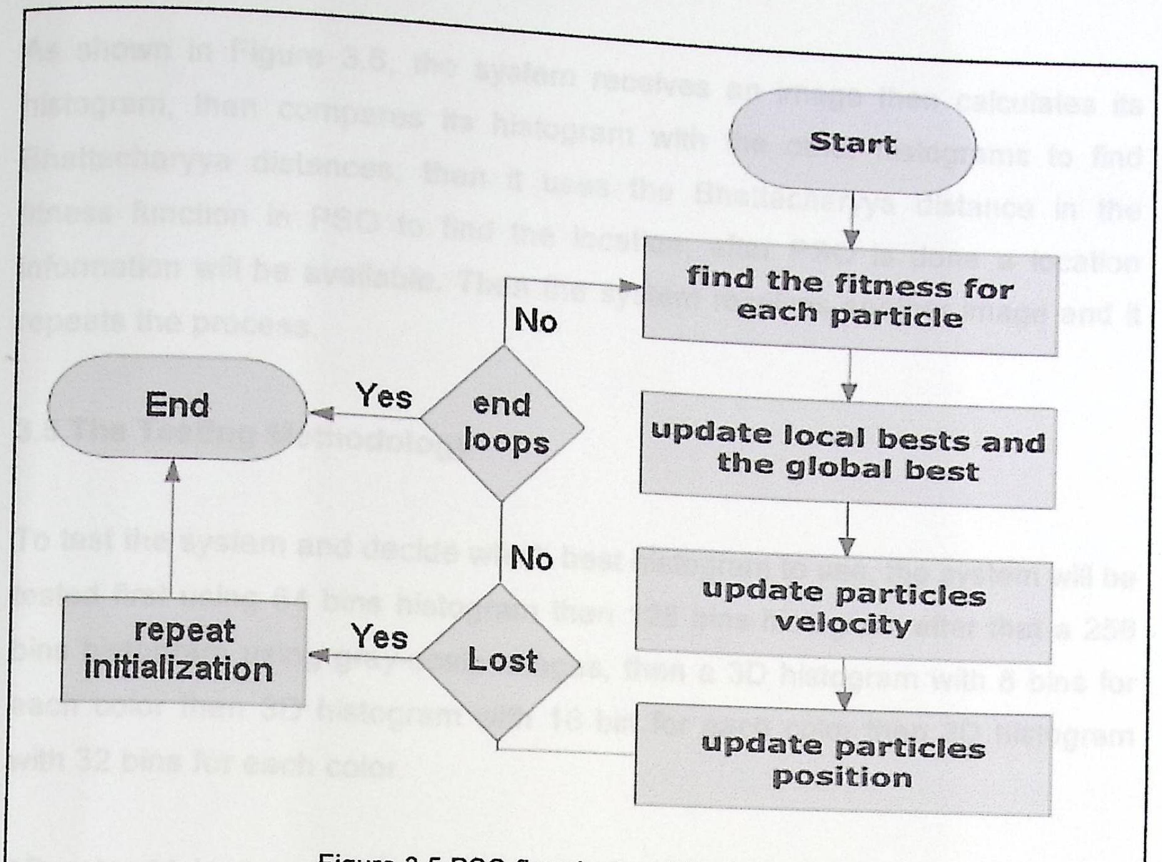

Start
find the fitness for each particle
update local bests and the global best
update particles velocity
update particles position
No
end loops
Yes
End
No
Lost
Yes
repeat initialization

Figure 3.5 PSO flowchart and the lost case

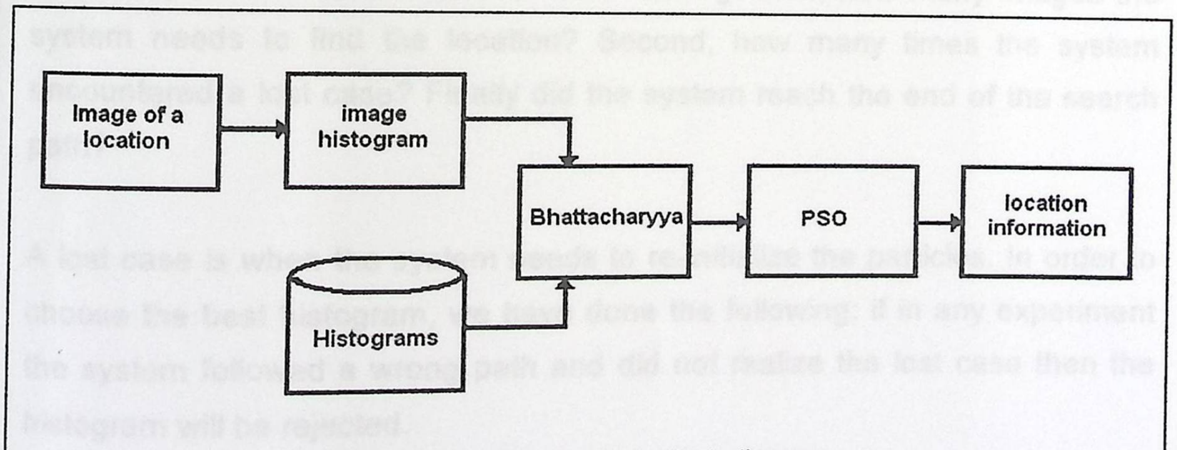


Figure 3.6 The system's block diagram

As shown in Figure 3.6, the system receives an image then calculates its histogram, then compares its histogram with the other histograms to find Bhattacharyya distances, then it uses the Bhattacharyya distance in the fitness function in PSO to find the location, after PSO is done a location information will be available. Then the system receives another image and it repeats the process.

3.5 The Testing Methodology

To test the system and decide which best histogram to use, the system will be tested first using 64 bins histogram then 128 bins histogram after that a 256 bins histogram using gray-scale images, then a 3D histogram with 8 bins for each color then 3D histogram with 16 bin for each color then 3D histogram with 32 bins for each color.

1D color histogram was ignored in this project because it does not take in consideration the color combinations, which might lead sometimes to same Bhattacharyya distance even that when the image is different, such thing has a lower probability of occurrence in 3D histograms.

Three aspects will be considered when testing: First, how many images the system needs to find the location? Second, how many times the system encountered a lost case? Finally did the system reach the end of the search path?

A lost case is when the system needs to re-initialize the particles. In order to choose the best histogram, we have done the following: if in any experiment the system followed a wrong path and did not realize the lost case then the histogram will be rejected.

PSO values will be constant in this testing, the values were decided by observing their movement after changing them several times values. Two PSO iterations will be used, C1, C2 and W value will assigned to 0.5.

Summary

This chapter showed the project in details, and also how each part of the system works. Flowcharts and diagrams were provided to make the description of the system more clear.

CHAPTER FOUR

4

Implementation

Chapter Four

- ► *Overview.*
- ► *The programs objectives.*
- ► *The program description.*
- ► *The system internal structure.*

4.1 Overview

In this project a simulation program was made, this chapter will give details about the program and its objectives and implementation. The class diagram of the system is also explained in this chapter.

4.2 The program's objectives

The simulation program was made to achieve the following:

1. Test the system: using the simulation program, the theory of the system is tested.

2. Study the PSO algorithm: this simulation program enables the user to change the PSO variables and study the effects of the variables on the particles movements.

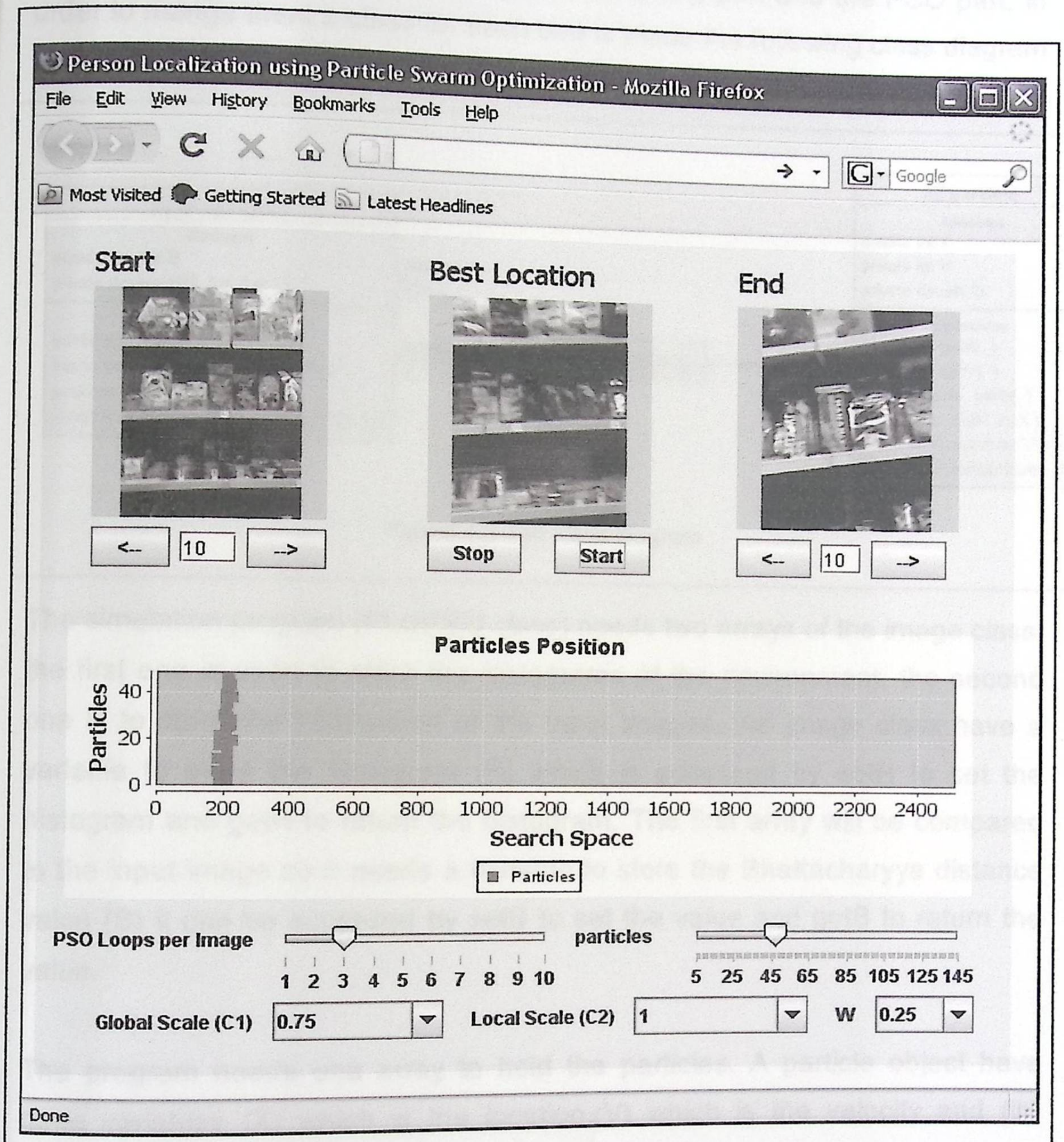
3. Benefit other researchers: this simulation program is a java applet which is an internet application. Researchers on PSO and localization can try the system on the internet.

4. Benefit the university: this simulation program is planned to be placed on the university web site, this will make the researchers enter the web site and see one of the university's projects.

4.3 The program description

This program is a java applet which is made to be used in the university web site. The program has its own sample to be used by the researchers worldwide, these samples are taken from a mid size store. The researchers can use by choosing the start and end points of the search path, the user can change the PSO search variables before the search starts. When the user starts the search the system will automatically input a sequence of images the which are between the start and end points, during the search process the program will show an image at the end of the PSO loops for each captured image, and also the particles positions will always be shown.

The following figure shows the applet:



PSO Loops per Image

particles

Global Scale (C1) 0.75

Local Scale (C2) 1

W 0.25

Figure 4.1 The simulation program

4.4 The system internal structure

The program runs as the system was described in Chapter three. The main two parts of the system are the image processing part and the PSO part, in order to mange them a class for each one is made the following class diagram shows the program classes implementation:

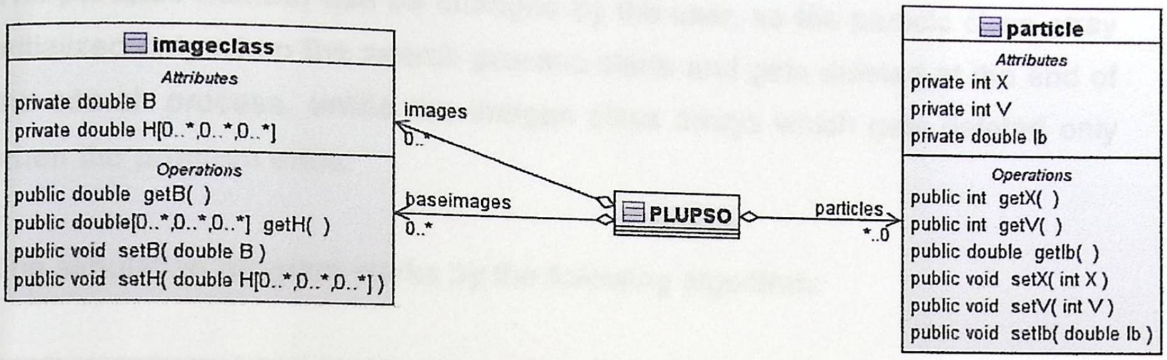


Figure 4.2 The class diagram

The simulation program (PLUPSO class) needs two arrays of the image class, the first one is used to store the histograms of the environment, the second one is to store the histograms of the input images, the image class have a variable to store the histogram (H) which is accessed by setH to set the histogram and getH to return the histogram. The first array will be compared to the input image so it needs a variable to store the Bhattacharyya distance value (B) it can be accessed by setB to set the value and getB to return the value.

The program needs one array to hold the particles. A particle object have three variables (X) which is the location,(V) which is the velocity and (Ib) which is the Bhattacharyya distance for the position from the captured image, the variables can be accessed by get and set methods.

When the program starts, it define the search space by allocating the first array of the image class and loading the histograms into it, then define the input images by allocating the second array and load the input images histograms into it.

These histograms do not change in the program, when the input image is determined, the program compares it with the search space histograms and calculates the Bhattacharyya distance value for each of them.

The particles number can be changed by the user, so the particle class array initialized only when the search process starts and gets deleted at the end of the search process, unlike the images class arrays which gets deleted only when the program ends.

The simulation program works by the following algorithm:

```
Load the histograms for both input and search path
Wait for the user input
Initialize the particles
While the search path not finished
{
  Get the input image histogram
  Loop for all search space images
    Calculate Bhattacharyya distance and store it
  Set global best to a bad value
  Do PSO loops
  Update user output
}
Wait for user input
```

Algorithm 2: the general way of how the simulation system works.

Summary

This chapter provided information about the simulation program, giving the program objectives and description. The internal structure of the program was also provided.

CHAPTER FIVE

5

Experiments and results

Chapter Five

- ► ***Overview.***
- ► ***The sample.***
- ► ***Experiments and results.***

5.1 Overview

This chapter contains the information about the experiments that has been made on the system, and the results of these experiments.

The first section will describe the samples that are used in the experiments. In the second section using the simulation program experiments will be done to test the theory, results of the test process will show the system performance level with different histograms.

5.2 The sample

The sample that is used in the system is from a mid-size store (around 10m^2), this store contains 16 locations, and the number of images taken is 2748 image. The following table shows how the images are distributed on the locations:

Table 5.1 Sample images

Location	Number of images
Location 1	115
Location 2	366
Location 3	78
Location 4	336
Location 5	269
Location 6	176
Location 7	215
Location 8	215
Location 9	48
Location 10	148
Location 11	152
Location 12	137
Location 13	225
Location 14	125
Location 15	85
Location 16	58
	2748

The store contains locations with different properties such as the snacks section, the drinks section and the detergents, and some locations looks similar which make these samples good for testing the system, also the size of the sample allows using several test paths.

The images that were taken are of a resolution of 480*640 with RGB colors, the captured images are not necessary with the same resolution because Bhattacharyya distance will not be affected by the difference since it uses a normalized histogram.

In Figure 5.1 some sample images of the store locations.

5.3 Experiments and results

In order to test the system, several test paths were used and observed to measure the system performance. For each path the number of images needed to find the location (N) and the number of times an error occurred (E) were recorded. Note that an error is when the system needs to re-initialize the particles, also if the system reached the final destination or not. The system was first tested using gray-scale images, and a 64 bin histogram. The results are in the following table:

Table 5.2 Testing results with 64 bins histogram

Test path	N	E	Reached destination?
Location 1 to 2	1	2	No
Location 2 to 4	3	0	Yes
Location 6 to 4	4	0	Yes
Location 8 to 10	2	0	Yes
Location 12 to 10	-	-	No
Location 15 to 12	-	-	No
Location 14 to 12	-	-	No
Location 14 to 16	-	-	No
	-	-	37.5%

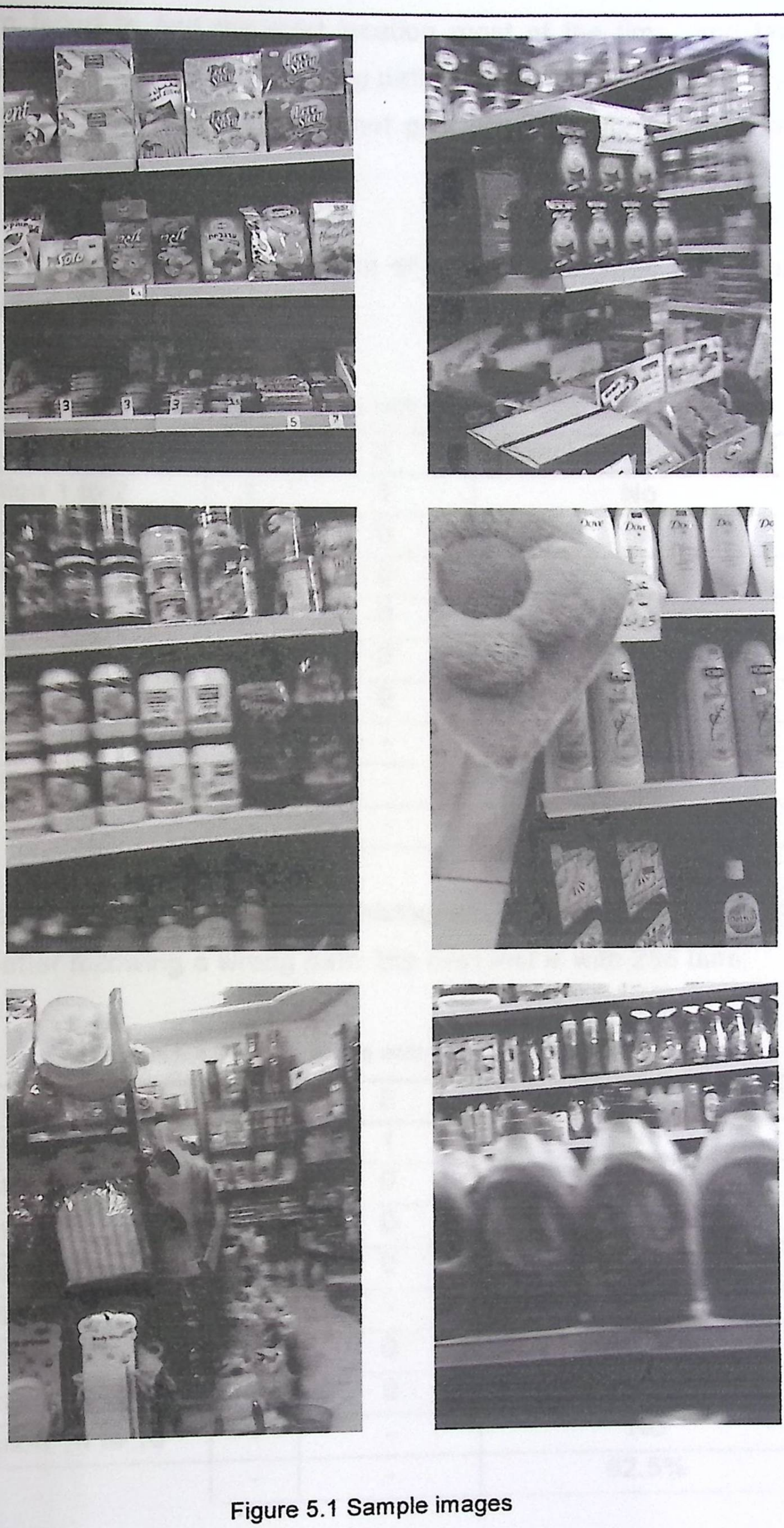


Figure 5.1 Sample images

The system failed to find the start location most of the time, and failed to correct itself when it followed the wrong path, which means that the gray-scale images with 64 bin histogram does not give enough information about the image.

The next test uses a 128 bin histogram of gray-scale images, the results are in the following table:

Table 5.3 Testing results with 128 bins histogram

Test path	N	E	Reached destination?
Location 1 to 2	1	1	No
Location 2 to 4	2	0	Yes
Location 6 to 4	3	0	Yes
Location 8 to 10	3	0	Yes
Location 12 to 10	4	0	Yes
Location 15 to 12	3	0	Yes
Location 14 to 12	-	-	No
Location 14 to 16	-	-	No
	-	-	62.5%

The system improved using 128 bins histogram, but the system is still unable to recover after following a wrong path, the next test is with 256 bins:

Table 5.4 Testing results with 256 bins histogram

Test path	N	E	Reached destination?
Location 1 to 2	1	1	No
Location 2 to 4	2	0	Yes
Location 6 to 4	4	0	Yes
Location 8 to 10	4	0	Yes
Location 12 to 10	-	-	No
Location 15 to 12	4	0	Yes
Location 14 to 12	3	0	Yes
Location 14 to 16	-	-	No
	-	-	62.5%

There was a low difference between using 128 bins and 256 bins, the system sometimes fails to find the beginning of the path, and fails to recover after errors, this shows that the gray-scale images are not effective for this system.

RGB images should provide more information about the location, so a 3D histogram is used for the next test, this will make places with different colors have different histogram, which may has been the drawback of the gray-scale images in the previous tests. 8 bins for each intensity is used, this makes a low change in the lighting have a low effect. The results of using 3D histogram are in the following table:

Table 5.5 Testing results with 3D histogram

Test path	N	E	Reached destination?
Location 1 to 2	1	0	Yes
Location 2 to 4	2	0	Yes
Location 6 to 4	3	0	Yes
Location 8 to 10	3	0	Yes
Location 12 to 10	3	0	Yes
Location 15 to 12	3	0	Yes
Location 14 to 12	3	0	Yes
Location 14 to 16	3	0	Yes
	2.6	0	100%

As Table 5.5 shows, very high performance of the system, it only needed three images to find the location and in all tests the system reached the final destination.

We can conclude that using color information can overcome the problem of low differences between the images, the system is able to find the path only after three images, and it did not encounter any error during the tests which leads us to say that a 3D histogram with 8 bins for each color is accurate enough for the system.

But still can the system recover from error? In order to find that out in the tests, we made some deliberate errors, the system was able to recover from the errors after 4-7 images get captured.

After the test the system shows that it is successful in finding the path, following it and recovering from errors.

Summary

This chapter showed information about the test sample, also it gives the information and results of the experiments that were done on the system. The final result was that 3D histogram is the most suitable for this system.

CHAPTER SIX

6

Conclusion

[illegible] way of helping a blind person navigate indoor environment using a mobile phone, a simulation program has been developed to test the theory and to be usable for other researchers.

As the system and the theory worked successfully, we can conclude that it is possible to build a navigation system to help the blind that does not need a GPS, also this system can overcome the [illegible] GPS encounter in indoor environments.

This system can be developed on a mobile phone that have an embedded camera which means that the user does not need to carry special equipments as in similar projects.

Depending on historical information to find the location is better than depending on one image, because depending on history solves the problem of similar places in the environment.

An inexpensive navigation system without any setup cost has been made possible in this project. During the setup, this system only needs a video of the environment to take its features, and only a modern mobile with camera to install the system on it which is possible to provide to most people.

In the experimental work, we have concluded that the 3D color histogram is more successful than the gray scale histogram. Using the 3D color histogram the system can find a location using three images from the surrounding and it did not fail in any lost case.

The proposed system is robust and can detect if it fails in a lost case where it re-initiates the search process in order to recover its correct place.

In this project we have tested a way for helping a blind person navigate in an environment using a mobile phone, a simulation program has been designed to test the theory and to be usable for other researchers.

As the system and the theory worked successfully, we can conclude that it is possible to build a navigation system to help the blind that does not need a server, also this system can overcome the problem that the GPS encounter in the in-door environments.

This system can be developed on a mobile phone that have an embedded camera which means that the user does not need to carry special equipments as in similar projects.

Depending on histological information to find the location is better than depending on one image, because depending on history solves the problem of similar places in the environment.

An inexpensive navigation system without any setup cost has been made possible in this project. During the setup, this system only needs a video of the environment to take its features, and only a modern mobile with camera to install the system on it which is possible to provide to most people.

In the experimental work, we have concluded that the 3D color histogram is more successful than the gray scale histogram. Using the 3D color histogram the system can find a location using three images from the surrounding and it did not fall in any lost case.

The proposed system is robust and can detect if it falls in a lost case where it re-initialize the search process in order to recover its correct place.

Future work

This project can be improved to accept multiple environments by integrating it with a wireless network that estimate the user's area then provide him with the environment features and information, this will make the user need to connect to a server but it will allow the system to use large environment with lower processing.

Also it is possible to improve the system by decrease the lightness effect, this can be done either by adding additional feature to the histogram or changing the color model to HSV (Hue, Saturation, Value) which can with some criteria overcome some lightness problems.

This program can be improved by using a feature that uses points of interest to be less affected by the presence of obstacles, small changes in environment, or changes in the user distance from the object.

Also implementing the system on the mobile can now be with less effort, the theory is tested and succeeded and the programming logic is available in the simulation program code.

References

[1]Al-Tamimi, M. and Tamimi, H., "Mobile-based Visual Localization Aid for the Blind", Graduation Project (B.Sc.), Palestine Polytechnic University, Hebron, Palestine, June 2008.

[2] Coroama, V. "The Chatty Environment - A World Explorer for the Visually Impaired". Adjunct Proceedings of Ubicomp, 2003.

[3] Hub, A., Diepstraten, J., Ertl, T. "Augmented Indoor Modeling for Navigation Support for the Blind". Proceedings of the 2005 International Conference on Computers for People with Special Needs, Las Vegas, Nevada, USA, 54-59, 2005.

[4] Hub, A., Hartter, T., Ertl, T. "Interactive Localization and Recognition of Objects for the Blind". California State University, Northridge Center on Disabilities' 21st Annual International Technology and Persons with Disabilities Conference, Los Angeles, CA, USA, 2006.

[5] Kennedy, J. and Eberhart, R. C. "Particle swarm optimization" (1995).. Proc. IEEE Int'l. Conf. on Neural Networks, IV, 1942--1948.

[6] L. Mihaylova, P. Brasnett, N. Canagarajah and D. Bull, "Object Tracking by Particle Filtering Techniques in Video Sequences", Advances and Challenges in Multisensor Data and Information Processing, Vol. 8, NATO Security Through Science Series: Information and Communication Security, E. Lefebvre (Ed.), IOS Press, the Netherlands, 2007, pp. 260-268.

[7] M. Mirmehdi, R. Perissamy, "CBIR with Perceptual Region Features". Proceedings of the 12th British Machine Vision Conference. T Cootes, C Taylor, (eds.). ISBN 1 901725 16 2, pp. 511–520. September 2001.

[8] M. Nixon, A. Aguado, "Feature Extraction and Image Processing", (Butterworth Heineman, GB: Newnes-Oxford, 2002).

[9] S. Jeong, C. S. Won, and R. M. Gray: "Histogram-Based Image Retrieval Using Gauss Mixture Vector Quantization", IEEE ICASSP, Hong Kong, April 2003.

[10] Sylvie Treuillet, Eric Royer, Thierry Chateau, Michel Dhome, Jean-Marc Lavest: "Body Mounted Vision System for Visually Impaired Outdoor and Indoor Wayfinding Assistance".