DIP BASED INDOOR GUIDANCE SYSTEM FOR BLIND PERSON

By

NC Muhammad Usman Aftab

NC Muneeb ur Rehman

NC Abdul Rauf

NC Ahsan Khurshid

Proffered to the Faculty of Electrical (Telecom) Engineering, Military College of Signals, National University of Sciences and Technology, Rawalpindi in fulfilment for the requirement of a Bachelors Degree in Electrical Telecommunication Engineering

JUNE , 2017

In the name of ALLAH, the Most Benevolent , the Most Courteous

ABSTRACT

DIP based Indoor guidance system for Blind Person

This Projects aims at the Development of a portable device that will help the blind users to navigate in a building independently by the provision of appropriate guidance and instructions. Pictures taken by the user through the device, will be matched with an already stored database of images, the best match found will dictate the corresponding audio instructions about the user's current location in the building.

The device is now capable of finding the best matched image from the database and guide the blind person by generating the appropriate sound according to the location of blind person. Picture taken from the camera which is interfaced with raspberry pi 3 is matched with the images of database and guide the blind person. This is possible by using SURF algorithm and FLANN index. The key points are extracted by the SURF algorithm of the test image and FLANN index make the index of descriptor vectors and match the key points of test image with the images already stored in the database one by one and generating the audio commands for every possible match that could occur during the process. These Audio Commands will contain the instructions that will update the user about his/her current precise location in the building.

JUNE 2017

CERTIFICATE

It is hereby certified that the contents and the form of the project entitled “DIP based indoor guidance system for blind user” submitted by the syndicate of

1) Muhammad Usman Aftab
2) Muneeb-ur-Rehman
3) Ahsan Khurshid
4) Abdul Rauf

has been observed satisfactory as per the demand of the B.E. Degree in Electrical (Telecom) Engineering.

Name (Supervisor): **Asst. Prof. Dr. Mir Yasir Umair**

Signature: ____________________________

Date: ____________________________

DECLARATION

No percentage of the work presented in this proposal has been proffered in assist of another award or qualification either at this university or elsewhere.

DEDICATED TO

Allah The Omnipotent ,

Faculty/Supervisor for their help,

And our parents for their blessings and support

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH Almighty. We thank our FYP Syndicate Supervisor Asst. Prof Dr Mir Yasir Umair from the depth of our hearts. He has been extremely supportive to us from the very beginning. He has provided his technical expertise & experience on even the most minor problems regarding our project. He has been a great guide to take us through the whole process of FYP comprehensively .None of this would‘ve been possible without the support & encouragement of our parents. They’ve supported us in every possible way & we owe our success to their untiring efforts and countless prayers. May Allah bless all of them.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Organization of Document……………………………………………….…...1

1.2 Overview…………………………………………………………………...1

1.3 Problem Statement…………………………………………………………1

1.4 Approach……………………………………………………………….. 2

1.5 Salient Features……………………………………………………………2

1.6 Objectives………………………………………………………………… 3

2. BACKGROUND

2.1 Indoor Positioning…………...……………………………………………… 4

2.2 WiFi based indoor positioning……………………………………………4

2.3 Bluetooth based indoor positioning……………………………………… 4

2.4 Digital image processing based indoor positioning……………………… 5

3. LITERATURE REVIEW

3.1 SIFT Algorithm…………………………………………………………… 6

3.1.1 Extrema detection of scale space……………………………….… 6

3.1.2 Localization of key points…………………………………….….. 7

3.1.3 Assignment of orientation…………………………………….….. 7

3.1.4 Formation of descriptor of key points…………………………..…... 7

3.1.5 Disadvantages of SIFT…………………………………………….…... 8

3.2 Speeded up Robust Features (SURF)…………………………………….….. 8

3.2.1 Scale-space representation and interest's points localization….…… 9

3.2.2 Descriptor…………………………………………………….……... 9

3.2.3 Orientation Assignment……………………………………….……….10

3.2.4 Descriptor depending on the combination of Haar wavelet response..10

3.3 Brute Force Matcher……………………………………………………….… 10

3.4 FLANN Algorithm……………………………………………………... 11

3.5 Implementation Concept…………………………………………………...11

4. DESIGN AND DEVELOPMENT

4.1 Technical Specifications…………………………………………………….. 12

4.1.1 Hardware Specifications………………………………………………12

4.1.1.1 Raspberry Pi 3 Model B……………………………………...12

4.1.1.2 Camera……………………………………………………… 16

4.1.1.3 Push Button……………………………………………………..16
4.1.1.4 Headphones……………………………………………………..17
4.1.1.5 Power Bank……………………………………………………..17

4.1.2 Softwares Used……………………………………………………….. 17

4.2 Design Summary……………………………………….………………..18

4.3 Development Phase…………………………………….…………………..19
4.3.1 Development of Database……………………..………………….. 19
4.3.2 Implementation of SURF……………………………………………19
4.3.3 Testing…………………………………………………………………..20
4.3.4 Final Product…………………………………………………………20
4.3.5 Result analysis of algorithms used……………………………......20

5. ANALYSIS AND TESTING

5.1 Introduction…………………………………………………………... 24

5.2 Testing………………………………………………………….……..24

5.3 Real Time Implementation……………………………………….……….. 24

5.4 Conclusion and Future Enhancements……………………….…………25

6. REFERENCES

Appendix A…………………………………………………..…………..,,,27

Appendix B…………………………………………………...…………… 29

Appendix C…………………………………………………...……………30

LIST OF FIGURES

Fig 1.1: Approach ………………………………………………………………… 2

Fig 4.1: Raspberry pi 3 Model B……………………………………………………..13

Fig 4.2: Pin Configuration of Raspberry Pi ……………………………………...14

Fig 4.3: **Diagram of GPIO**…………………………………………………………… 15

Fig 4.4: Functions of different pins………………………………………………. 15

Fig 4.5: Camera………………………………………………………………………16

Fig 4.6: **Design Summary**…………………………………………………………. 19

Fig 4.7: **Key points Extraction**……………………………………………………. 20

Fig 4.8: **Key points Matching**…………………………………………………... 22

Fig 5.1: Test image…………………………………………………………………24

Fig 5.2: **Matched image from database**.. 25

CHAPTER 1- INTRODUCTION

1.1 Organization of Document:

This file consists of six main parts:

- First part gives the introduction of the project.
- Second part tells about background related to this field.
- Third part is about the literature review.
- Fourth part is about design, implementation and analysis of results.
- Fifth part enlightens future improvements that can be made.
- Final part is about references and bibliography

1.2 Overview:

The project aims to develop a guidance system for a blind user that will help him navigate in a building independently by the provision of appropriate guidance and instructions. Pictures taken by the user through the device, will be matched with an already stored database of images, the best match found will dictate the corresponding audio instructions about his present location in the indoor environment.

This process guides the user to localize himself inside the building, which is the first step towards navigation. Based on the location of the user, appropriate instructions will be generated.

1.3 Problem Statement

People with disability to see, often suffer while moving in an indoor environment. Conventional use of sticks is not much reliable and helpful, especially when exposed to large buildings and new environments. An extra human guide is required to help these blind people which makes them dependent on others. Thus, there is a problem for the blinds in effectively and independently navigating through an indoor environment.

GPS is not helpful at all in the indoor environment because of no line of sight inside the building. This makes GPS useless inside the building.

One measure for indoor positioning is WiFi Client Based Guidance System but it results in signal fluctuations that can lead to significant errors and inaccuracies. Relatively inaccurate (5-15m). There are multiple factors on which the accuracy depends like reflection, scattering, ceilings, own body and shielding through the walls.

At times there are large signal fluctuations, or problems of power, as it needs constant WiFi connectivity, which result in errors and inaccuracies.

Thus, indoor positioning is always a challenging task with various problems to overcome. The problems include inaccuracies, environmental limitations and systematic errors.

1.4 Approach :

The approach to achieve the project objective is as follows:

Extensive Research on Image Matching, Algorithms, Selection of Controller and its interfacing with other hardware modules. Installation of OS on the controller selected (Raspberry Pi) and Updating the required libraries to run the selected algorithms. Once, the controller is ready and up to date with its libraries, it's time to interface and integrate the other modules that includes the camera and ultra-sonic module, with the raspberry pi.

As the camera gets integrated with Raspberry Pi's Operating System, next task is to form the database of images using this camera. The database includes images of whole interior of the test-building, which will later be matched with the picture that blind user will take through this same camera as part of the device. The next task is to implement the algorithms, which precedes the matching process. The algorithms are responsible for finding out the key points in images. Next, is the Programming the Pi for image matching using the information generated by the algorithms, and generating the audio commands for every possible match that could occur during the process, these Audio Commands will contain the instructions that will update the user about his/her current precise location in the building.

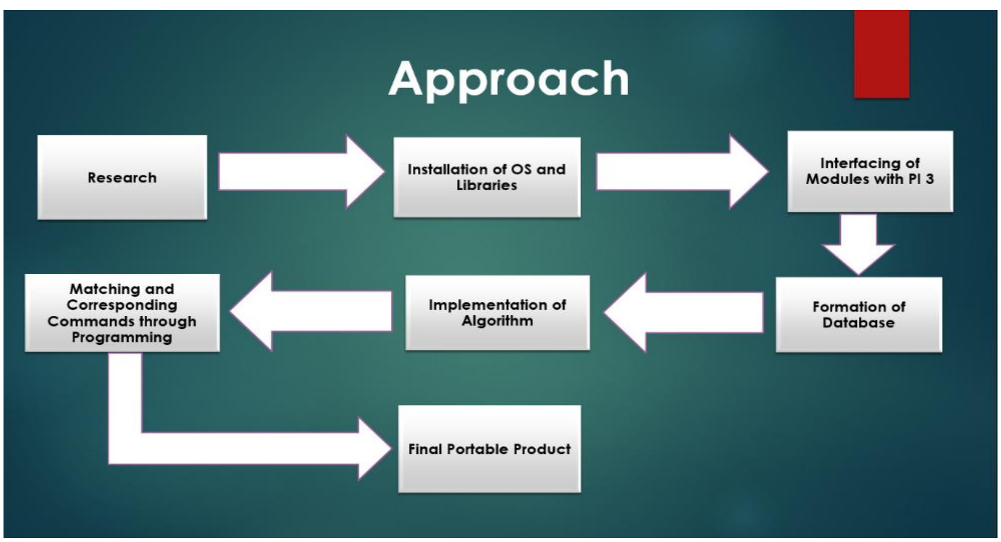


Fig 1.1

1.5 Salient Features

- Accurate
- Portable Product
- Less Margin of Error

- Use of Databases
- Use of Image Processing Algorithms
- Use of Raspberry pi for processing

1.6 Objectives

The project aims to develop a guidance system for a blind user that will help him navigate in a building independently by the provision of appropriate guidance and instructions.

CHAPTER 2- BACKGROUND

2.1 Indoor Positioning:

Indoor positioning system is used to locate objects or people inside an indoor environment with the help of acoustic signals, magnetic field, radio waves or other sensory information gathered by mobile devices. No standard exists for an IPS system since there are several commercial systems on the market.

2.2 Wi-Fi based indoor positioning:

There are different methods for indoor positioning. When considering indoor environments Wi-Fi is a good alternative to GPS. Most of the cases Wi-Fi positioning system (WPS) is easy to install, because the access points of Wi- Fi already exist in most indoor environments. The advantages are cash register systems that already exists, hotspots of the public and shops access points can be used. It is not necessary for user to connect with the Wi-Fi, Wi-Fi enabling is sufficient.

The localization technique which is most commonly used for locating with access points is based on calculating the received signal strength concentration (received signal strength indication) and the technique of fingerprinting. The constraints which are used to locate the wireless access point service set identification and the access point's MAC address. The Wi-Fi hotspot database is made by finding the correlation between the Wi-Fi hotspot MAC addresses and mobile device GPS location data.It's pre-requisite is to install an app on the smartphone that is used to calculate the current location. The more the number of points entered into the database the more accurate is the result.

There are multiple factors on which the accuracy depends like reflection, scattering, ceilings, own body and shielding through the walls. The accuracy of depends on the preconditions and varies from 5 to 15 meters.

2.3 Bluetooth based indoor positioning:

Bluetooth technology is a standard enabling wireless connectivity of devices that operates in the ISM band. This low energy consumption is achieved as no paired connection is required between two BLE devices when one is transmitting frames and the other receiving them. A BLE beacon typically broadcasts at a certain interval frames that contain a unique identifier. User carries a device, which listens to optical or radio frequency (RF) signals from beacons in the environment. The strength, phase, or time-of-flight of the blue tooth signals is measured and utilized to determine the position of the person.

For indoor positioning low energy solution is BLE. It depends on many factors likes signal power and frequency. It runs for years on a coin cell battery. Unlike other wireless technologies. Stable signal strength is maintained by BLE .It is important for distance determination because it has a high sample rate. Short duration messages that can be an advertising message or a data message reduces the power consumption. For positioning advertising messages are used. BLE has unremarkable hardware, is cost effective, has consumption of low energy and integrates into the existing infrastructure.

2.4 Digital Image Processing based indoor positioning:

DIP based indoor positioning depends on extracting the key points from test image and matching it with the key points of database. Database is to be made by taking the pictures of the interior of the building from different angles. It gives more accurate results as compare to the above methods because of no reflection and scattering of signals.

CHAPTER 3- LITERATURE REVIEW

The project is based on the digital image processing so different algorithms of image processing for key points extraction are studied like SIFT, SURF and different algorithms for key points matching like brute force and FLANN are to be viewed in depth.

3.1 SIFT algorithm:

Scale invariant feature transform is a computer vision procedure to describe and detect structures in images. For any key object in an image, fascinating points on the key object can be withdrawn to provide a "feature description" of the entity. This description which is mined from a trained image, can then be used to recognize the object when trying to locate the object in a test image containing many other prominent objects. To accomplish recognition, it is vital that the features mined from the trained image are noticeable even under variations in image scale, illumination and noise . Such points usually lie on high contrast regions of the image, such as entity edges.

SIFT comprises of 4 steps:

1. Extrema detection of scale space
2. Localization of key points
3. Orientation assignment
4. Key point description

3.1.1 Extrema detection of scale space:

So First step is detecting points of interest, which are termed key points in the SIFT structure. In this complete process the image at different scales is convolved with Gaussian sifters and then on the basis of difference Gaussian images are taken. Maxima/Minima of the Difference of Gaussians which occur at several scales tell us about the key points . Specifically, a Difference of Gaussian image is given by

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma),$$

where L(x,y,k) is the convolution of the actual image I(x,y) with the Gaussian blur G(x,y,k) at measure i.e.

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$$

Hence a DoG image between scales ki and kj is just the difference of the Gaussian-distorted images at scales ki and kj. For scale space extrema detection in the SIFT algorithm, first the image is convolved with Gaussian-blurs at diverse scales. The convolved images are gathered by octave, and the value of ki is selected so that we obtain a static number of convolved images per octave. Then the Difference-of-Gaussian images are taken from neighboring Gaussian-blurred images per octave.

Once DoG images have been acquired, key points are recognized as local minima/maxima of the DoG images through scales. This is done by linking each pixel in the DoG images to its eight neighbors at the same scale and nine conforming neighboring pixels in each of the neighboring scales. On the basis of pixel value whether maximum or minimum when related with all the pixels ,then it is selected as a contender for key point .

3.1.2 Localization of key points:

Too many candidates for key point are produced by extrema detection of scale-space among which some are unstable .Detailed fit is performed as next step in the algorithm to all the adjacent data for scale ,ratio of principal curvatures and accurate location. This evidence received helps us to reject those points having low contrast (sensitive to noise) or those which have bad localization along an edge

3.1.3 Assignment of orientation:

Orientation assignment of key points is done in this stage on the basis of gradient directions of local image. Invariance to rotation is achieved in this key step , which helps to represent the key point descriptor comparative to orientation and hence invariance to image spin is attained .

For every pixel present in the neighboring region of the key point in the blurred image Gaussian , a calculation of magnitude and direction is executed .A 36 bin orientation histogram is then formed in which each bin is covering 10 degrees .

Every sample of the adjacent window is added to a histogram bin and is weighted by a Gaussian weighted circular window and gradient magnitude with that is 1.5 times of the key point’s scale .Dominant orientations are shown by the peaks in histogram .After filling of histogram, the orientations which correspond to the highest peaks and local peaks that are within 80 % of highest peaks are allocated to the key points .In the case of many orientations assignment , an additional key point is produced having the same scale and location as the actual key point for each extra orientation.

3.1.4 Formation of descriptor of keypoints:

Previously location of key points of particular scales were found and then orientation was assigned to them which ensures rotation ,scale and invariance to image location Now at this stage we want to figure a descriptor vector for each key point such that the descriptor is extremely distinguishing and moderately constant to the residual changes such as brightness, 3D perspective, etc. This step is achieved on the image neighboring in scale to the key point's scale.

First a set of alignment histograms is formed on 4x4 pixel areas with 8 bins each. These histograms are calculated from greatness and alignment values of samples in a 16 x 16 region around the key point such that each histogram contains samples from a 4 x 4 sub region of the unique neighborhood area. The magnitudes are additionally weighted by a Gaussian function with equal to one semi the width of the descriptor opening. The descriptor then develops a vector of all the values of these histograms. Since there are 4 x 4 = 16 histograms each with 8 bins the vector has 128 essentials. This vector is then normalized to unit length in order to improve invariance to affine variations in lighting. To decrease the effects of non-linear illumination a onset of 0.2 is used and the vector is another time normalized.

Although the measurement of the descriptor, i.e. 128, seems high, descriptors with lesser dimension than this don't accomplish as well across the range of matching tasks and the computational charge remains low due to the estimated BBF (see below) method used for discovering the nearest-neighbor. Lengthier descriptors continue to do better but not by much and there is an extra danger of increased sensitivity to misrepresentation and occlusion. It is also shown that feature matching correctness is above 50% for perspective changes of up to 50 degrees. Therefore, SIFT descriptors are invariant to slight affine variations.

To test the uniqueness of the SIFT descriptors, matching correctness is also measured compared to changing number of key points in the testing database, and it is exposed that matching exactness reduces only very slightly for very huge database sizes, thus representing that SIFT structures are extremely unique.

3.1.5 Disadvantages of SIFT:

1. The drawback is that it is mathematically complicated and computationally heavy.
2. SIFT utilizes histogram of gradients. That is, computation for gradient of each pixel in patch should be done and these computations cost time.
3. It is not effective for low powered devices.

Due to these disadvantages, it is preferred to implement SURF algorithm instead of SIFT algorithm.

3.2 Speeded up Robust Features (SURF):

In case of computer vision, speeded up robust features (**SURF**) is a original local feature detector and descriptor. It can be used for tasks such as 3D reconstruction , image registration, object recognition and classification

To notice concerned points, SURF uses an integer estimate of the determinant of Hessian blob detector, which can be calculated with 3 integer operations using a pre calculated integral image. Its feature descriptor is based on the totality of the Haar wavelet response around the argument of interest. These can also be calculated with the help of the integral image. SURF descriptors have been used to recognize and locate people, faces or objects, to track objects, to reconstruct 3D scenes and to mine points of interest. The image is converted into coordinates, using the multi-resolution pyramid method, to duplicate the original image with Pyramidal Gaussian or Laplacian Pyramid shape to get an image with the same size but with decreased bandwidth. This attains a special blurring effect on the original image, called Scale-Space and certifies that the points of attention are scale invariant.

The algorithm has two key parts: local locality description and curiosity point discovery.

$$S(x,y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i,j)$$

The summation of the actual picture within a rectangle can be assessed rapidly using the integral image, needing assessments at the rectangle's four turns.

SURF also utilizes the determinant of the Hessian for choosing the scale, as is also finished by Lindeberg. Given a point p=(x, y) in an image I, the Hessian matrix H(p, σ) at point p and scale σ, is:

$$H(p,\sigma) = \begin{pmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{pmatrix}$$

3.2.1 <u>Scale-space representation and interest's points localization:</u>

Curiosity points can be found at altered scales, partly because the hunt for correspondences often needs assessment images where they are understood at diverse scales. In additional feature detection algorithms, the scale space is usually understood as an image pyramid. Images are frequently levelled with a Gaussian filter, then they are subsampled to develop the next higher level of the pyramid. Therefore, several floors or stairs with numerous calculations of the masks are computed:

$$\sigma_{\text{approx}} = \text{Current filter size} * \left(\frac{\text{Base Filter Scale}}{\text{Base Filter Size}} \right)$$

The scale space is separated into a amount of octaves, where an octave refers to a sequence of replies maps of covering a replication of scale. In SURF, the bottom level of the scale space is attained from the output of the 9×9 filters.

Hence, contrasting earlier approaches, scale spaces in SURF are executed by applying box filters of diverse sizes. Accordingly, the scale space is examined by up scaling the filter size rather than iteratively dropping the image size. The output of the above 9×9 filter is measured as the first scale layer at scale s=1.2 (agreeing to Gaussian derivatives with σ=1.2). The following layers are attained by filtering the image with slowly bigger masks, taking into explanation the discrete nature of integral images and the precise filter structure.

3.2.2 <u>Descriptor:</u>

The objective of a descriptor is to deliver a exclusive and robust description of an image feature, e.g., by relating the intensity spreading of the pixels within the neighborhood of the point of concentration. Most descriptors are thus calculated in a local way, hence a description is gained for every point of notice recognized before.

The dimensionality of the descriptor has straight influence on both its computational difficulty and point-matching robustness/precision. A short descriptor may be more vigorous against appearance changes, but may not propose adequate discrimination and thus give too numerous wrong positives.

The first phase contains setting a reproducible alignment based on evidence from a circular region about the concerned point. Then we build a square region associated to the nominated orientation, and mine the SURF descriptor from it.

3.2.3 **Orientation assignment:**

If we want to accomplish the turning invariance, we have to find the alignment of the point of attention. Now we compute the Haar wavelet responses in both directions i-e x & y, within a circular region of the point of curiosity. A Gaussian function at the point of interest weighs the responses obtained By calculation of the totality of all responses we can estimate the dominant orientation within a window of the size π/3. Within this window, we sum the horizontal and vertical responses. A local alignment vector can then be obtained with the help of the summed responses. Thus, the point of interest is defined by the longest such vector. In order to attain the wanted balance among angular resolution and robustness, The size of the sliding window needs be chosen carefully.

3.2.4 **Descriptor depending on the combination of Haar wavelet response:**

A square region placed on interest point is to be mined, in order to describe the region around the point, and is then oriented along the alignment as selected above. The size of this window is 20s.

The concerned region is split into smaller 4x4 square sub-regions, and for each one, at 5x5 frequently spaced sample points we extract Haar wavelet responses. These responses are weighted with a Gaussian to achiever more robustness for noise,deformation and translation.

Major advantage of SURF over SIFT is that SURF is very fast as compare to SIFT. The project requires minimum time to be taken for the detection of key points, so we chose SURF algorithm to achiever desired results for the project.

After having extracted the key points, our next task is to correlate the key points of test image with the key points of the images in the database.

3.3 **Brute Force Matcher:**

Brute Force matcher rather is straight forward and simple. It takes the descriptor of one feature in first set and matches it with all the other features in next set using distance calculations. The closest one is returned as result.

The first step in brute force matching is to create the BF Matcher object using cv2.BFMatcher(). It takes two optional constraints. First one is norm type, distance measurement is specified by this parameter. By default, it is cv2.Norm_L2. Once it is created, two significant approaches are brute force Matcher.match() and Brute force Matcher.knnMatch(). First method returns the finest match overall while the Second method returns k finest matches where k is defined by the user.

By using cv2.draw Key points () to draw key points, we can draw matches using cv2.drawMatches(). Two images are stacked horizontally and lines are drawn from first image to second image presenting finest matches. There is also cv2.drawMatches Knn which draws all the k finest matches. If k=2, it will draw two match-lines for each key point. So a mask has to be passed if it is required to selectively draw it. The disadvantage of brute force matcher is that it takes too much time for image matching.

3.4 Fast Library for Approximate Nearest Neighbors (FLANN):

FLANN stands for Fast Library for Approximate Nearest Neighbors. It contains a gathering of algorithms which are enhanced for a fast nearest neighbor search in large datasets. It works faster than brute force Matcher for huge databases. It performs FLANN calculations among two sets of feature vectors. As a consequence we get two numpy arrays with the first one being a list of indexes of the matches and the next one contains the match distance value. Lower distances mean better matches.

3.5 Implementation Concept:

1. For each image in the database, First off, descriptors are required to be computed with the algorithm best meeting the requirements. We are using SURF, because our main concern is speed. however depending on the requirements e.g required license, performance a different algorithm might suite the requirements better.

2. Combine all the descriptors into one big matrix while keeping track of the descriptors' identity, that is we must know which descriptor range within the matrix came from what image. Actually, to achieve this goal, rows from all matrices are concatenated and the row number is saved, at the start and end of the newly appended descriptors.

3. Build a FLANN index from the MATRIX where all the descriptors lie.

4. Take the query image and Compute its descriptors

5. Run KNN search over the FLANN index. A K-Nearest-Neighbors search calculates the space between a query descriptor and the descriptors of the all of the images in database, and then returns the K pairs with least distance. Two matrices are produced as a result – Indeces matrix and Distances matrix which contain the indices of descriptors within the concatenated matrix which had the lowest distance and the distances themselves, respectively.

6. Filter out/screen out all the inadequate matches computed in the previous step.

7. Since we know what descriptors within the concatenated matrix belong to what image, We can find images matching the query image. Image giving the highest number of descriptor matches or more precisely descriptor indexes in the Indices matrix, will be our result image of this whole process.

CHAPTER 4- DESIGN AND DEVELOPMENT

4.1 Technical Specification:

4.1.1 Hardware Components:

- Raspberry Pi 3 Model B Complete Kit
- Camera
- Push Button
- Headphones
- Power Bank

4.1.1.1 Raspberry Pi 3 Model B:

Raspberry Pi 3 Model B is the latest Raspberry Pi as of January 2017. Scratch and Python are the main languages that are supported by Raspberry Pi 3. It supports many other languages. Raspberry Pi models cannot keep track of the time of day because they do not have a built-in real-time clock. The Raspberry Pi 3, has a processor of quad-core Cortex-A53 which has 10 times the act than that of a Raspberry Pi 1. Raspberry Pi 3 is about 80% efficient and faster than the Raspberry Pi 2 in the tasks which are parallel. A program which runs on the Pi can recover the time from input of the user at boot time, thus it recognizes the time when powered on. Pi automatically retains the time on shutdown, and re-installs that time at boot so that it provide reliability of time for the file system,

DS1307 which has a real-time hardware clock and has a battery backup and is fully binary coded, may be added

A quantity of applications and developers have been there that influence the Raspberry Pi, efforts are made by programmers to adjust the Raspberry Pi into a solution that is cost effective in power consumption and energy monitoring. Raspberry Pi is an economical solution to expensive commercial alternatives because of the relatively low cost.

Technical Specifications are given as follows:

- CPU (1.2 GHz with 64 bit Quad core ARM v 8)
- 4 HDMI, Ethernet and USB ports
- Wireless LAN 802.11n
- 40 general input output pins
- Combined composite video and 3.5mm audio jack
- RAM of 1GB
- Interface for camera
- Push-pull micro SD card slot
- Interface for display

Fig 4.1:Raspberry pi 3 Model B

<u>Pin Configuration of Raspberry pi :</u>

The physical interface between the outside world and Pi is GPIO pins. They are considered as switches at the simplest level that can be turn on or off .26 are GPIO pins out of the 40 pins while others are ground or power pins with additional two ID EEPROM pins.

To interact with the real world these pins can be programmed. It is not necessary that input comes from a switch; input can be signal from another device or computer or can be from a sensor. The output can be turning on an LED, sending a signal to another device or anything. Devices attached to Raspberry pi are controlled from anywhere by Raspberry pi and the data can be send back by those devices if the devices are in a network. Raspberry Pi is considered as ideal for control of physical devices over the internet.

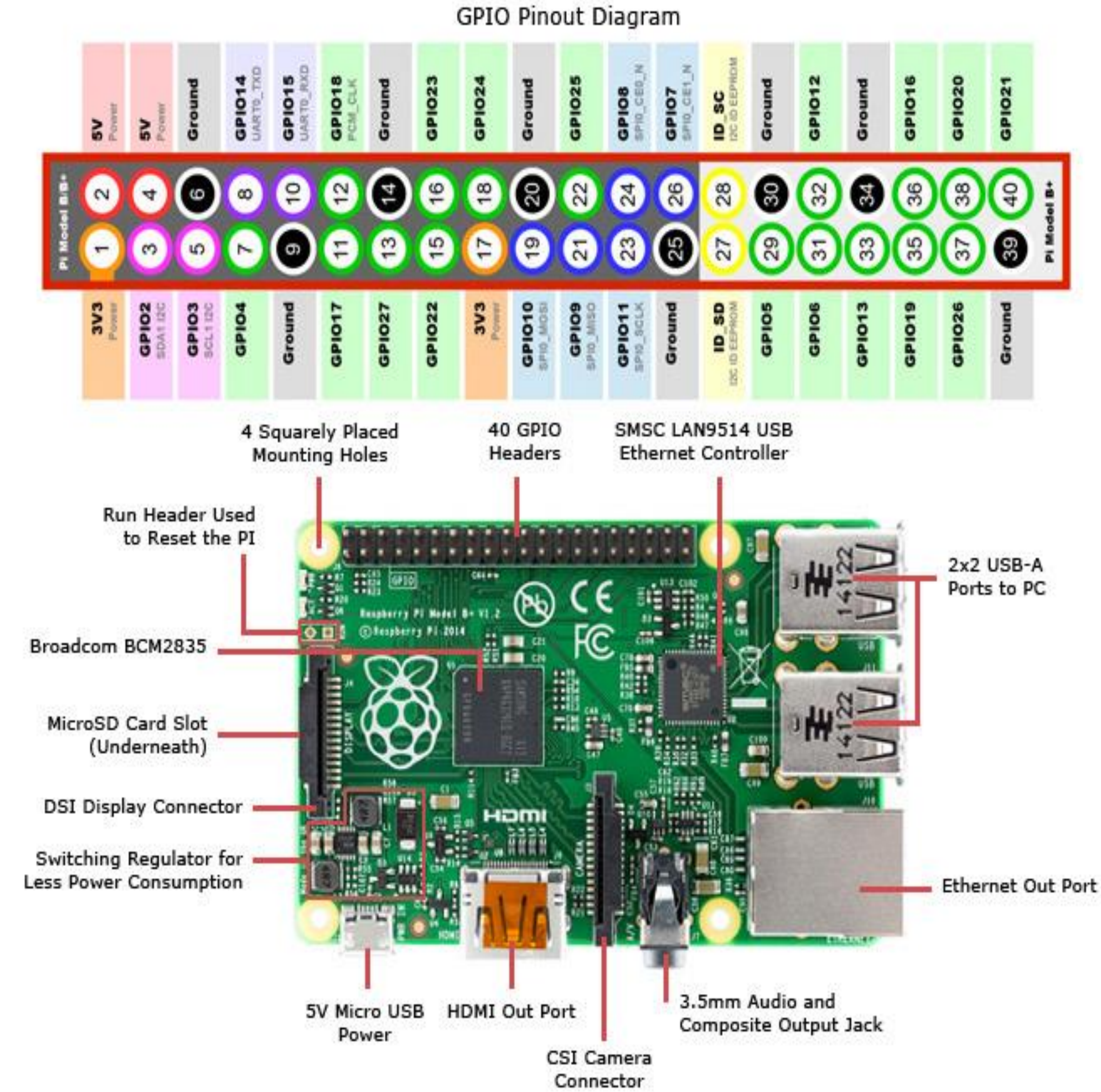


Fig 4.2: Pin Configuration of Raspberry Pi

Working of GPIO:

It is safe to follow the instructions when messing with the GPIO. When wires and power sources are plugged randomly into Pi, then it is the possibility that Pi may kill it. If things that use a lot of power connect to the Pi bad things can happen; LEDs use less power so they are fine while motors are not. If Raspberry Pi is not taken and considering the electrical circuit in which there is a battery connected to a switch and a light source .

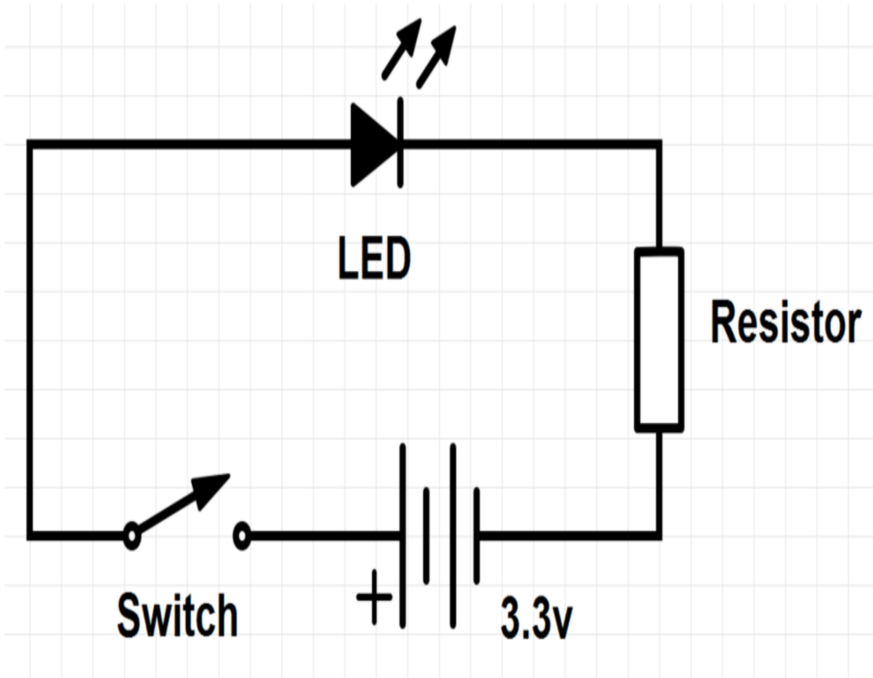


Fig 4.3: Diagram of GPIO

When GPIO pin is used to take output, then both the switch and the battery is replaced by Raspberry Pi. In computing terms each pin can be turn on or off. Pin generates the 3.3V output voltage when it is HIGH and it is off when it is low.

This is exactly the same circuit when considering Raspberry Pi. At the GPIO pin which can generate output +3v and a ground pin (which is at zero potential) LED is connected.

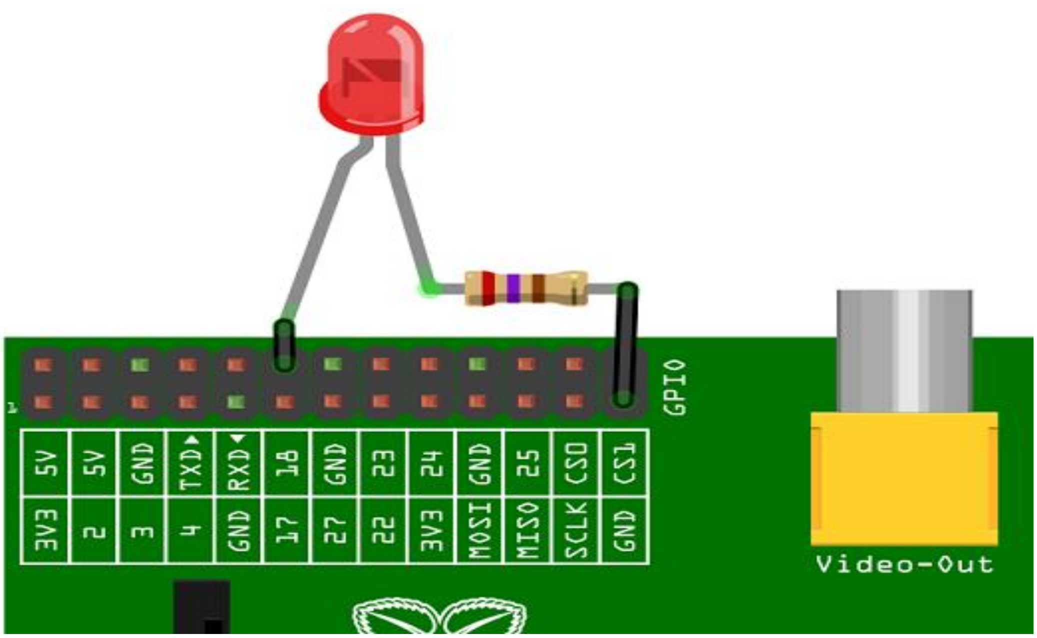


Fig 4.4: Functions of different pins

GPIO outputs can be HIGH or LOW, 3v3 or 0v, on or off .So GPIO outputs are easy. On its inputs digital devices work so GPIO inputs are trickier. When a button is connected across a ground pin and a input pin, the Pi can get confused that whether switch is on or off. It might work correctly, it might not. Without any reference it is difficult to know whether the pin is up or down, or even.. It is necessary to know "pull up" and "pull down" in Raspberry Pi. It's a method of giving the input pin a reference due to which it distinguishes when an input is obtained .

4.1.1.2 <u>Camera:</u>

A 5megapixel camera is used for the purpose of taking pictures by the user. It can be easily interfaced to the module that is Raspberry pi. The picture taken is than matched by a database already installed in raspberry pi. The performance of the algorithm is not affected by the quality of camera being used by the blind person.

For example, if database images are taken by a camera of 8MP and the user has a camera interface of 5MP, it does not have any effect on the results as the key points extracted by the algorithm can still be matched despite having mere differences. Additionally, Threshold for the matching algorithm can also be varied according to the requirement.

There is a flexible cable in the camera that is inserted into the CSI connector which is present between the HDMI ports and Ethernet. It is easy to use the camera in Raspbian by selecting the camera option and running the Raspi-config

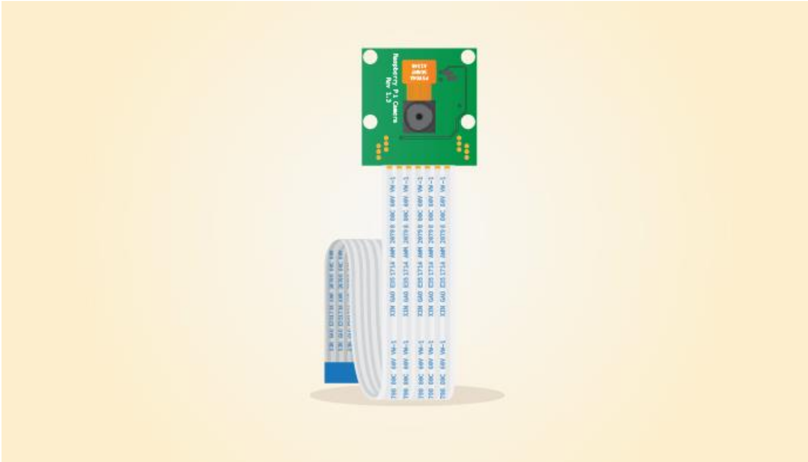


Fig 4.5:Camera

4.1.1.3 <u>Push Button:</u>

A push button is interfaced at ‘general pin input/output’(GPIO) slots of raspberry pi.

The main purpose of the push button is to help blind person take the image. The user just have to push the button, by doing so, some commands will run and an image has been taken by the device capturing his front view. This image is then automatically be uploaded on the code and then, matched with the already stored database. Finally, as a result of matching, appropriate instructions and commands are converted into voice and are fed into the user's ears.

4.1.1.4 Headphones:

Any types of headphones can be used by the user which is interfaced by the audio jack present in the device.

4.1.1.5 Power bank:

Power bank is used for the continuous supply of power to the device. It can be recharged according to requirement. Fully charged, It can supply power to the device for about 2-3 hours.

4.1.2 Softwares Used:

- Raspbian (OS) (Version Jessie)
- Python (Version 2.7)

Raspberry pi can support various operating systems which can be installed through SD cards which is then placed in the SD card slot present on the bottom .

Initially Raspberry Pi practices a Debian based Linux working system which is Raspbian. Raspbian uses main desktop environment as of the latest update that is PIXEL, Pi Improved X windows Environment, Lightweight .It is comprised of Open box stacking window manager a modified LXDE desktop environment with a new theme and few other changes.

There are operating systems like Ubuntu MATE, Windows 10 internet of things core, classroom management, RISC OS and that are available via the official website

Python is a high level general purpose programming which is widely used. It has a design viewpoint which stresses code readability (instead of using curly braces or keywords whitespace indentation is used to define code blocks), and its syntax ease the programmers and allow them to express ideas in fewer lines of code which is not possible in C++ or Java. Python provides constructs so enable writing clear programs on both a large and small scale. It features automatic memory management, dynamic type system including imperative, functional programming, object oriented and procedural styles.

Python's greatest strength is it's large standard library. It also provide tools that are suited to many tasks.

Many standard protocols for internet applications (such as HTTP and MIME) are supported. The different modules are also included in python like modules for creating graphical user interfaces, generating pseudorandom numbers, connecting to relational databases, arithmetic with arbitrary precision decimals, perform unit testing and manipulating regular expressions

Python is being used as primary language responsible for translating and implementing the algorithms which is imported from Open CV Libraries. The version used is 2.7.

Open CV (Open Source Computer Vision) is developed by Intel. The primary interface of Open CV is C++ and is also written in C++, but it still holds a widespread, older C interface which is less compressive . It is a library of functions that largely aimed at real-time computer vision. This library is free for use. Python has bindings of Java and MATLAB/OCTAVE. In the online documentation the API for these interfaces can be found. Wrappers have also established in other languages such as C#, Perl, Haskell and Ruby

The application areas of Open CV include:

- Identification of objects
- Recognition and Segmentation
- Mobile robotics
- Recognition of gestures
- Structure from motion (SFM)
- Human–computer interaction (HCI)
- 2D and 3D feature toolkits
- Understanding the motion
- Tracking of motion
- Estimation egomotion
- Augmented reality
- Facial recognition system

In order to have a support in the above areas, Open CV has a static library that contains:

- Random forest
- Artificial neural networks
- Boosting
- Support vector machine (SVM)
- Decision tree learning
- k-nearest neighbor algorithm
- Expectation-maximization algorithm
- Gradient boosting trees
- Naive Bayes classifier

Open CV has a support on a different platforms .

Desktop : Windows, FreeBSD, Linux,Open BSD;

Mobile: Android , BlackBerry 10 ,Maemo ,iOS

4.2 <u>Design Summary:</u>

The raspberry pi is acting as the brain of the device which takes input from all modules and processes the information. Camera and Ultra-Sonic Module are to be interfaced with Raspberry Pi which is acting as Controller or CPU. Database is to be made by snapping pictures of interior of the building from most of possible different angles, and uploading to the memory of Raspberry Pi. Pictures taken through Camera by Blind Person are matched with the Database, followed by generation of appropriate command according to results of best matched image/picture.

This command is, then converted to audible format via espeak module.

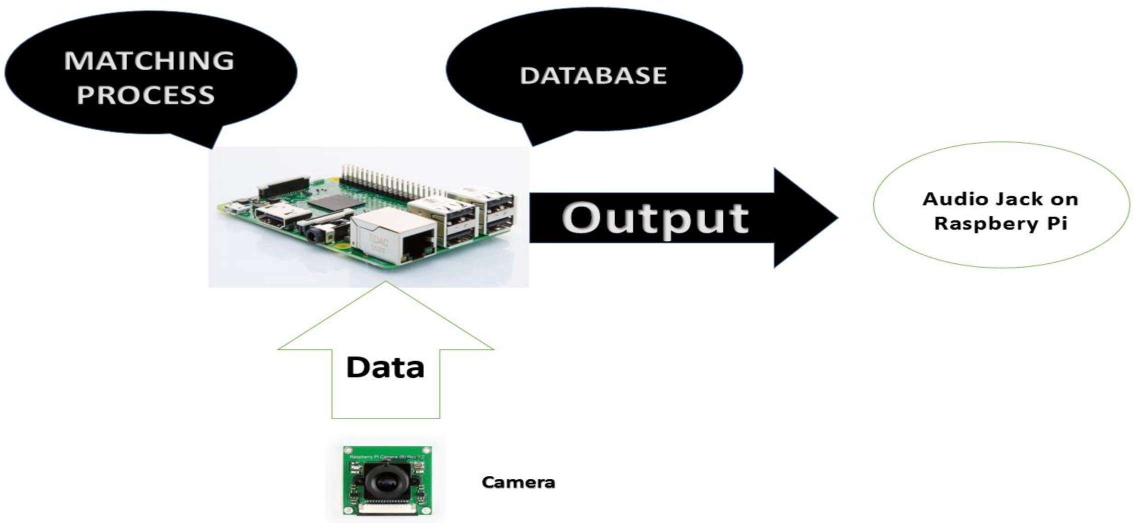


Fig 4.6: Design Summary

4.3 Development Phase:

4.3.1 Development of Database:

Once the camera and LCD have been interfaced with Raspberry Pi, we have formed the Database, Our test-building is EE Department, The database includes hundreds of images of the test-building .Pictures are taken by the Camera of Raspberry Pi, from distinct angles, distances, and under different circumstances, i.e Normal, Lights, Brightness etc. And stored in the memory card of Raspberry Pi.

4.3.2 Implementation of SURF algorithm:

As the database has been developed and stored onto the memory of Raspberry Pi. The next work we have done is the implementation of SURF which we have used for extracting out the key-points and features from the Images, Open CV includes the functions that our algorithms is using during their execution.

The Execution of the Algorithm is done using Python as the primary language. Python has:

- simpler syntax compared to C/C++
- Python has more robust functionality in shorter codes compared to C/C++
- Once integrated with Open CV, it acts as wrapper and imports the libraries and functions from Open CV

4.3.3 Matching and Generation of Appropriate Commands:

The image matching is performed by FLANN algorithm which is responsible to find the best match, based on information extracted by SURF. Appropriate commands containing instructions to guide the user are generated according to every possible match that could occur during the process.

4.3.3 <u>Testing:</u>

Our practical test building is EE department. The initial prototype is tested indoor and be checked for any possible errors or inaccuracies in the system followed by the refinement process.

4.3.4 <u>Final Product:</u>

Upon Successful Completion of Testing, the device will be made as compact as possible and be given its final shape, that will make it easy to carry and ready to use.

4.3.5 <u>Result Analysis of Algorithm used :</u>

After analyzing the key points extraction and key points matching results we found that with the increase in accuracy the processing time increased and with less features we got reduction in time .So we have to choose a optimal point or trade off point and in our case value of parameter is 400 at trade off point.

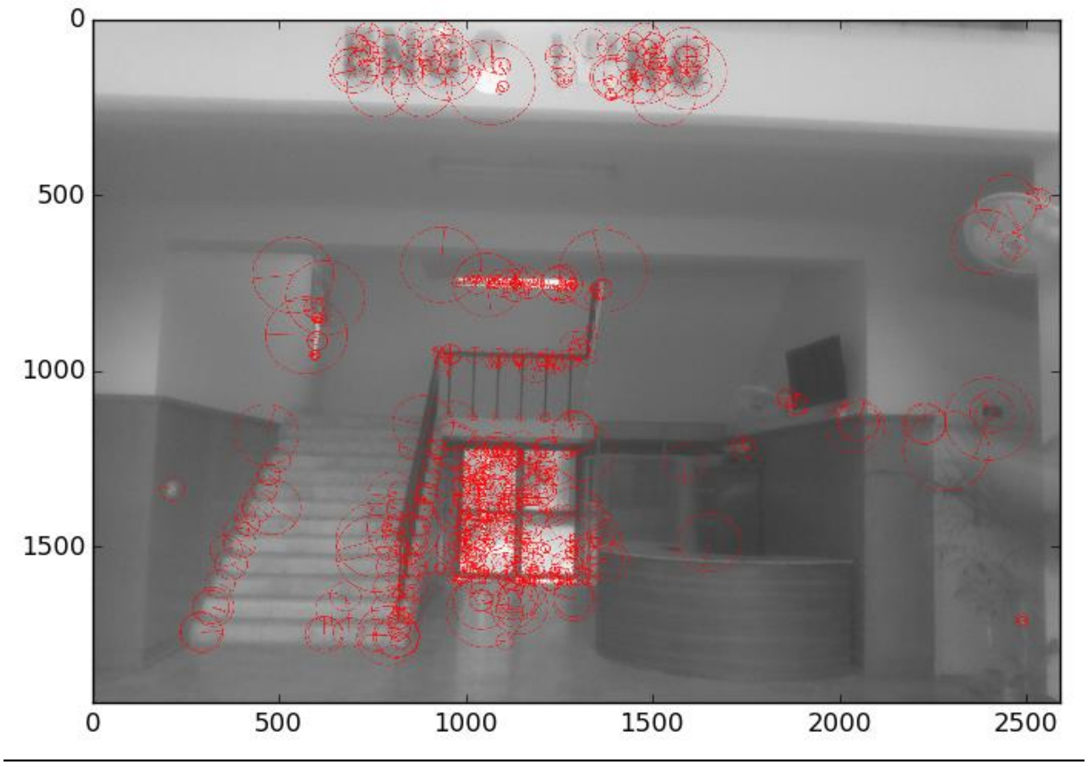


Fig 4.7 (a)

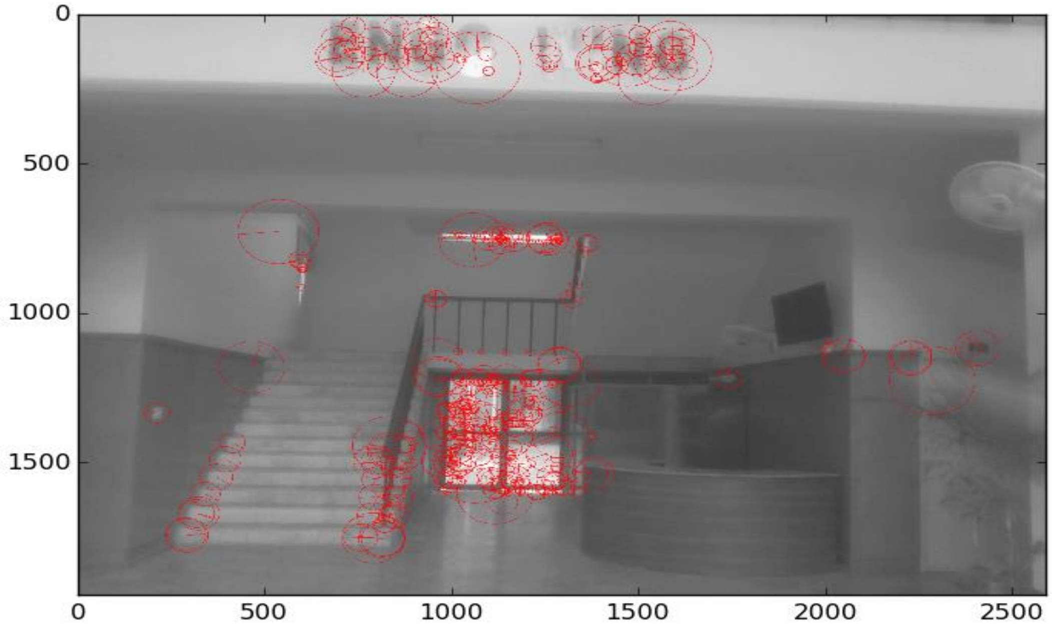


Fig 4.7 (b)

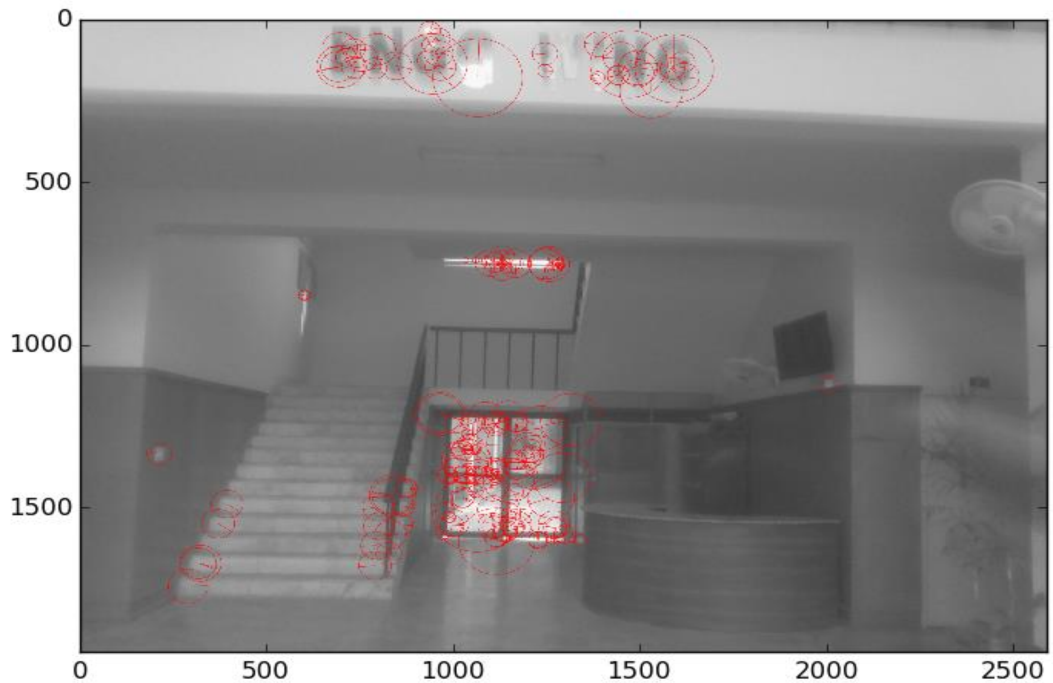
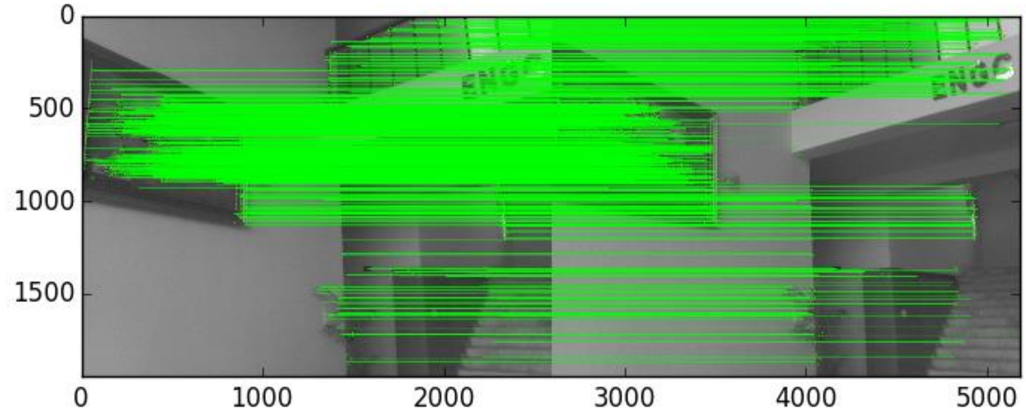


Fig 4.7 (c)

<u>**Matching Two Images One To One :**</u>

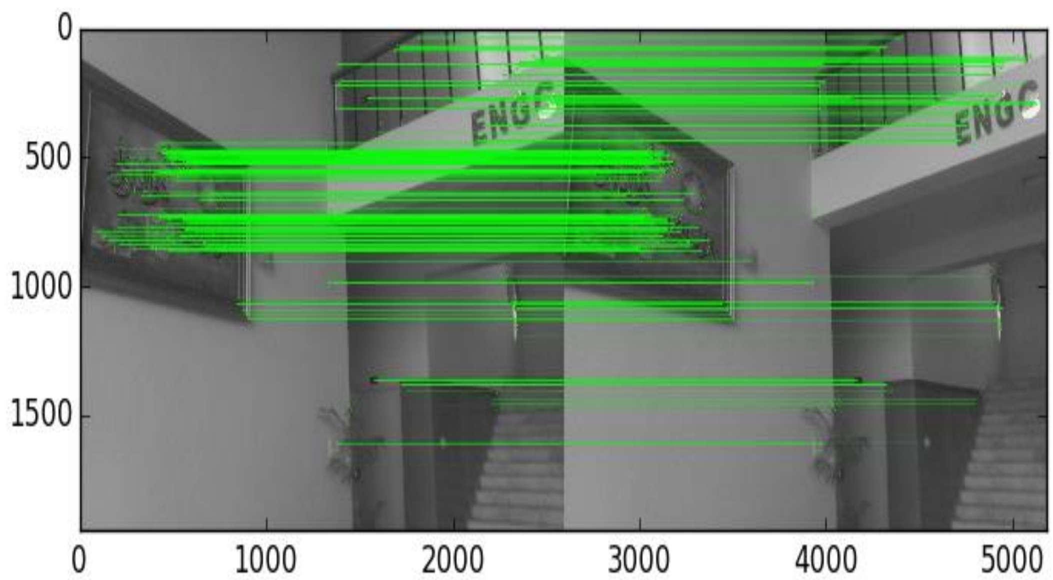
Parameter Value Set To 100:



No Of Matches=2292

Fig 4.8 (a)

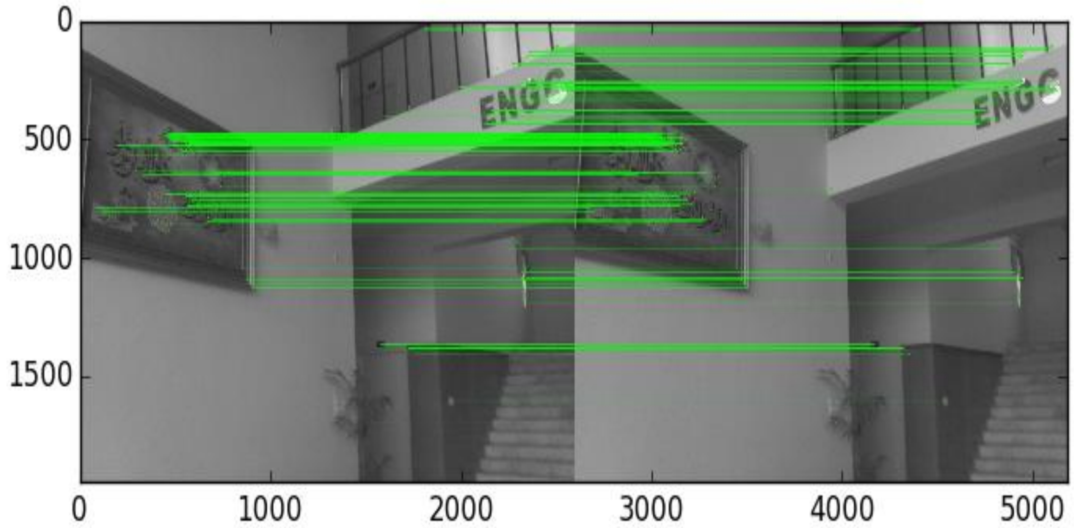
Parameter Value Set To 400:



No Of Matches=547

Fig 4.8 (b)

Parameter Value Set To 700:



No Of Matches=278

Fig 4.8 (c)

CHAPTER 5-ANALYSIS AND TESTING

5.1 Introduction:

This chapter describes the setup for testing the complete DIP based guidance system implemented in hardware and software. The results are discussed and possible future enhancements for the project are described.

5.2 Testing:

For testing the hardware and software assembly we setup the raspberry pi .We stored the data base of images of EE department, code of SURF and FLANN in the raspberry pi and interface it with the camera and button.

5.3 Real Time Implementation:

Blind user press the button of portable device and capture the image. Picture taken by the blind person is correlated with the already stored images and the best matched image guide the blind user about his position by the appropriate command. So the blind person is now capable of taking the next step. This process takes 4 to 5 seconds .In this way blind person easily knows his position inside the building.

Fig 5.1 Test image

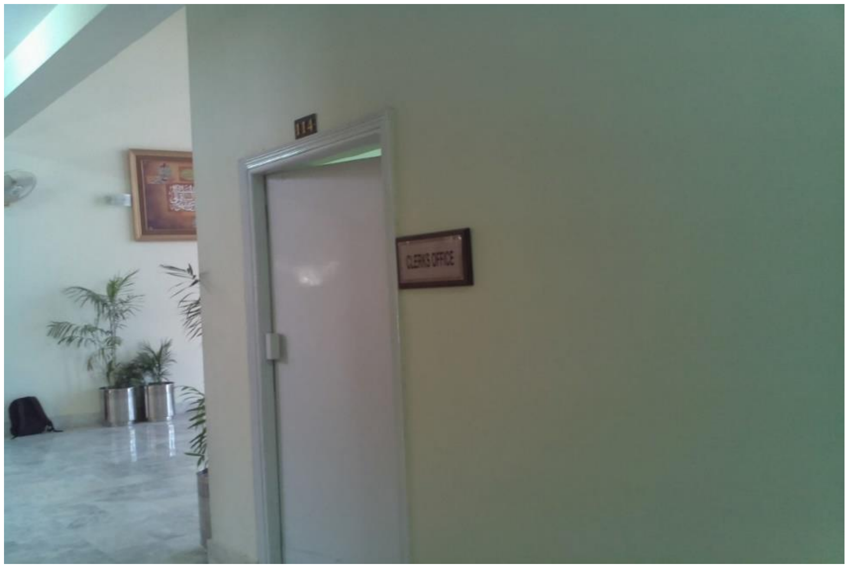


Fig 5.2 Matched image from database

5.4 <u>Conclusion and Future Enhancements:</u>

From all the above discussion we conclude that DIP based indoor guidance system is a device which guides the blind user in some seconds about his position inside the building. It implement the algorithms of image processing. Matching in just few seconds is the most useful one of this portable device.

However for future advancements, further algorithms and routines can be defined so that the device can perform other functionalities like how to freely move inside a building in a new environment by the dynamic database that is when the user enters in a new environment some other person downloads the database of that building from internet and the user will now easily move inside a building.

CHAPTER 6-REFERENCES

Related Work Done So Far with References:

Work on Indoor Positioning through wifi client based method has been done under the supervision of Dr. Mir Yasir, 2015-16 Also, work on the use of Digital Image Processing to recognize Currency Notes has also been completed under the supervision of Lt Col. Abdul Ghafoor , 2015-2016. Our Project is derived from these projects.

[1] Dr. Mir Yasir's Syndicate, "Indoor Positioning using Wifi-Client based Technique", Military College of Signals, 2015-16

[2] Thesis, "Mobile based indoor positioning utilizing wi-fi localization and image processing", College of Computer Studies De La Salle University Manila, 2012

[3] PPM Seminar, "Image Processing with Open CV" Fabrizio Dini, Giuseppe Lisanti, 2010

APPENDIX A: SYNOPSIS

DIP Based Indoor Guiding System For Blind Person/Handicapped

Extended Title: Indoor Guiding System For Blind Person/Handicapped without GPS and access point utilizing SIFT algorithm(DIP BASED) .
Brief Description of The Project / Thesis with Salient Specifications: The project designs and implements a guidance system using digital imaging processing technique that will help blind users to move around in a given building. The project provides the user with a compact device comprising of a camera, controller and a storage device ,that device will utilize the SIFT algorithm that will provide the user the information he needs to move around in the building
Scope of Work : GPS cannot be used for indoor positioning and WiFi client based positioning is not possible due to possible signal fluctuations therefore using dip based algorithm For positioning purpose.
Academic Objectives : *This project would give hands on- experience of:* • digital image processing in image matching and positioning • C,C++ and Matlab languages for programming • Embedded Systems • modules/ Interfacing • Ultrasonic Module
Application / End Goal Objectives : Development of a guidance system for blind users as well as for strangers for steering into a new environment. The End Goal Objective would be to provide a

working device for the market that could be enhanced and implemented on a large scale in the market

Previous Work Done on The Subject :

Work on dip based guidance system has not been done in MCS yet

Material Resources Required:

- Algorithm Processor
- Microcontroller
- Storage Device.
- Camera

No of Students Required : 4

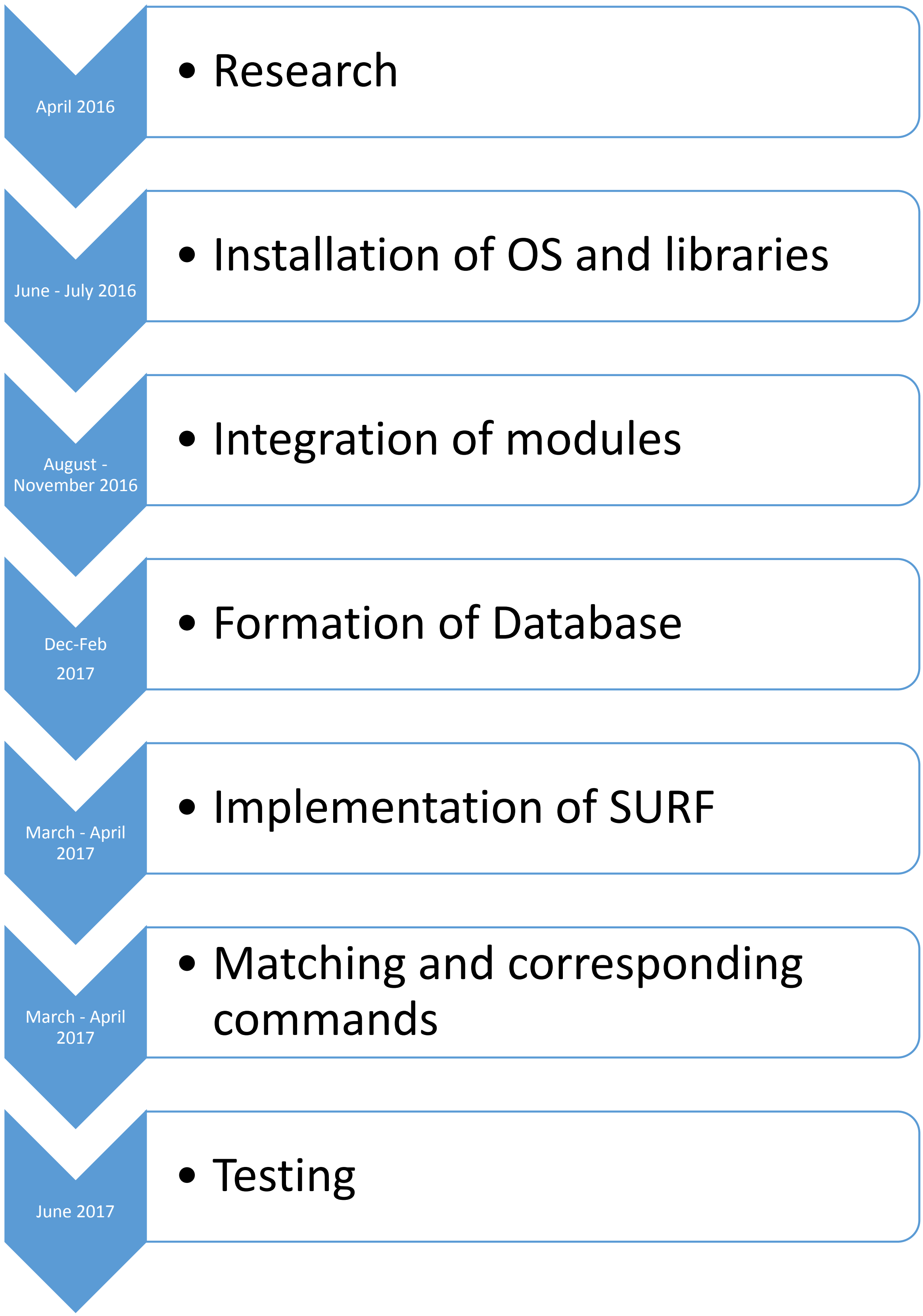
Group Members:

- Muhammad Usman Aftab
- Abdul Rauf
- Ahsan Khurshid
- Muneeb Ur Rehman

Special Skills Required:

Learning of programming languages, Utilization of image processing algorithm development of data base and its communication with Blind person

APPENDIX B: PROJECT PROGRESS



APPENDIX C: CODE

Part 1:

```
import RPi.GPIO as GPIO
import time
import cv2
import numpy
import os
import collections
import operator
import pyttsx
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(21, GPIO.IN, pull_up_down=GPIO.PUD_UP)
while True:
    input_state = GPIO.input(21)
    if input_state == False:
        print('Button Pressed')
        time.sleep(1)
        execfile('modified.py')
        time.sleep(1)
```

Part 2:

```
image = get_image("./img/%s" % (f,))
keypoints, descriptors = get_image_features(image)
numpy.save(f+'.npy', descriptors)
descriptors = numpy.load(f+'.npy')
```

Part 3:

```
import cv2
import numpy
import os
import collections
import operator
import pyttsx
import time
files = []
matcher = None
dos = None
def get_image(image_path):
	return cv2.imread(image_path)
def get_image_features(image):
	# Workadound for missing interfaces
	surf = cv2.xfeatures2d.SURF_create(400)
	# Get keypoints from image
	keypoints = surf.detect(image, None)
	# Get keypoint descriptors for found keypoints
	keypoints,descriptors= surf.detectAndCompute(image,None)
	return keypoints, numpy.array(descriptors)
def train_index():
	# Prepare FLANN matcher
	flann_params = dict(algorithm = 1, trees = 4)
	matcher = cv2.FlannBasedMatcher(flann_params, {})
	# Train FLANN matcher with descriptors of all images
```

```
    for f in os.listdir("img/"):
        print "Processing " + f
        descriptors = numpy.load(f+'.npy')
        #descriptors=numpy.fromfile(f+'.dat')
        print descriptors.shape
        #print type(keypoints)
        matcher.add([descriptors])
        files.append(f)
    print "Training FLANN."
    matcher.train()
    print "Done."
    return matcher
def match_image(index, image1):
    # Get image descriptors
    global ID
    image1 = cv2.imread('Test_Image.JPG',0)
    image1 = cv2.resize(image1, (600, 400))
    keypoints, descriptors = get_image_features(image1)
    # Find 2 closest matches for each descriptor in image
    matches = index.knnMatch(descriptors, 2)
    # Cound matcher for each image in training set
    print "Counting matches..."
    count_dict = collections.defaultdict(int)
    for match in matches:
# Only count as "match" if the two closest matches have big enough distance
        if match[0].distance / match[1].distance < 0.3:
```

```
                continue
            image_idx = match[0].imgIdx
            count_dict[files[image_idx]] += 1
        # Get image with largest count
        matched_image = max(count_dict.iteritems(), key=operator.itemgetter(1))[0]
        ID = matched_image
        # Show results
        print "Images", files
        print "Counts: ", count_dict
        print "==========="
        print "Hit: ", matched_image
        print "==========="
        ID = matched_image
        return matched_image
if __name__ == "__main__":
        print "OpenCV Demo, OpenCV version " + cv2.__version__
        start_time = time.time()
        flann_matcher = train_index()
        print "\nIndex generation took ", (time.time() - start_time), "s.\n"
# ============= Training done, image matching here ==============

        start_time = time.time()
        match_image(flann_matcher, "Test_Image.JPG")
        print "Matching took", (time.time() - start_time), "s."
        print ID
```

```
if ID == 'IMG_0027.JPG' or ID =='IMG_0029.JPG'or ID =='IMG_0031.JPG'or ID ==
'IMG_0033.JPG'or ID =='IMG_0034.JPG'or ID =='IMG_0035.JPG'or ID =='IMG_0041.JPG'or
ID=='IMG_0045.JPG'or ID=='IMG_0049.JPG':

    engine = pyttsx.init()

    rate = engine.getProperty('rate')

    engine.setProperty('rate', rate-30)

    engine.say('you are standing at the entrance of double e department .... on your left are the
clerk office and hod office ..  on your right are faculty offices...... on your front right is
information desk ...... on your front left is stairs to the first floor ..... right on front is exit')

    engine.runAndWait()


elif ID == 'IMG_0059.JPG' or ID == 'IMG_0055.JPG'or ID =='IMG_0053.JPG'or
ID=='IMG_0051.JPG'or ID =='IMG_0058.JPG':

    engine = pyttsx.init()

    rate = engine.getProperty('rate')

    engine.setProperty('rate', rate-30)

    engine.say('you are standing in front of office of doctor mir yasir office')

    engine.runAndWait()


elif ID == 'IMG_0080.JPG' or ID =='IMG_0063.JPG'or ID =='IMG_0054.JPG':

    engine = pyttsx.init()

    rate = engine.getProperty('rate')

    engine.setProperty('rate', rate-30)

    engine.say('you are in ground corridor having faculty offices ')

    engine.runAndWait()


elif ID == 'IMG_0074.JPG' or ID =='IMG_0067.JPG'or ID== 'IMG_0069.JPG':

    engine = pyttsx.init()
```

```
        rate = engine.getProperty('rate')

        engine.setProperty('rate', rate-30)

        engine.say('you are standing in front of office of assistant professor atiya obaid ,Lecturer Aimen aakif ,Lecturer maryam rasool and lecturer amal haider')

        engine.runAndWait()


elif ID == 'IMG_0085.JPG' or ID =='IMG_0081.JPG':

        engine = pyttsx.init()

        rate = engine.getProperty('rate')

        engine.setProperty('rate', rate-30)

        engine.say('you are standing in front of Traing office of I S department')

        engine.runAndWait()


elif ID == 'IMG_0096.JPG':

        engine = pyttsx.init()

        rate = engine.getProperty('rate')

        engine.setProperty('rate', rate-30)

        engine.say('you are standing in front of Dean office')

        engine.runAndWait()


elif ID == 'IMG_0104.JPG':

        engine = pyttsx.init()

        rate = engine.getProperty('rate')

        engine.setProperty('rate', rate-30)

        engine.say('you are standing in front of S o Cordination  engineering wing')

        engine.runAndWait()
```

```
elif ID == 'IMG_0120.JPG':

        engine = pyttsx.init()

        rate = engine.getProperty('rate')

        engine.setProperty('rate', rate-30)

        engine.say('you are standing in front of office P A to C i e ')

        engine.runAndWait()



elif ID == 'IMG_0126.JPG' or ID =='IMG_0128.JPG' or ID =='IMG_0125.JPG' or ID
=='IMG_0131.JPG':

        engine = pyttsx.init()

        rate = engine.getProperty('rate')

        engine.setProperty('rate', rate-30)

        engine.say('on your front is double E exit ... on your right are stairs and Leutinent Colonel
Abdul Gafoor office ')

        engine.runAndWait()



elif ID == 'IMG_0136.JPG':

        engine = pyttsx.init()

        rate = engine.getProperty('rate')

        engine.setProperty('rate', rate-30)

        engine.say('entrance of double e')

        engine.runAndWait()


elif ID == 'IMG_0153.JPG':
```

```
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('c r 28')
        engine.runAndWait()

elif ID == 'IMG_0156.JPG':
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('c r 30')
        engine.runAndWait()

elif ID == 'IMG_0172.JPG':
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('H o d office')
        engine.runAndWait()

elif ID == 'IMG_0216.JPG'or ID == 'IMG_0217.JPG' or ID =='IMG_0220.JPG' or ID
=='IMG_0226.JPG' or ID =='IMG_0222.JPG' or ID =='IMG_0223.JPG'or ID =='IMG_0231.JPG'
:
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('you are in E c r ')
```

```
    engine.runAndWait()


elif ID == 'IMG_0232.JPG':
    engine = pyttsx.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate-30)
    engine.say('Second entrance of e c r ')
    engine.runAndWait()


elif ID == 'IMG_0249.JPG' or ID =='IMG_0251.JPG' or ID =='IMG_0252.JPG':
    engine = pyttsx.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate-30)
    engine.say('On your right is Brigidaier retired Khans office... on your left is the lecturer narmeens office')
    engine.runAndWait()


elif ID == 'IMG_0247.JPG' or ID =='IMG_0147.JPG' or ID =='IMG_0143.JPG':
    engine = pyttsx.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate-30)
    engine.say('On your left are the washrooms..on your immediate right are the stairs to first floor..and on left front and right front are the Class rooms')
    engine.runAndWait()


elif ID == 'IMG_0159.JPG' or ID =='IMG_0148.JPG':
    engine = pyttsx.init()
```

```
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('You are infront of C R TWENTY NINE')
        engine.runAndWait()

    elif ID == 'IMG_0152.JPG' or ID =='IMG_0144.JPG':
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('You are infront of C R TWENTY SEVEN')
        engine.runAndWait()

    elif ID == 'IMG_0165.JPG' or ID =='IMG_0167.JPG':
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('On your left is O I C M I S  cell')
        engine.runAndWait()

    elif ID == 'IMG_0174.JPG' or ID =='IMG_0170.JPG':
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('you are infront of clerk office')
        engine.runAndWait()
```

```
elif ID == 'IMG_0188.JPG' or ID =='IMG_0185.JPG' or ID =='IMG_0178.JPG':
    engine = pyttsx.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate-30)
    engine.say('double e entrance')
    engine.runAndWait()

elif ID == 'IMG_0200.JPG' or ID =='IMG_0198.JPG':
    engine = pyttsx.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate-30)
    engine.say('Turn right and climb up stairs to first floor')
    engine.runAndWait()

elif ID == 'IMG_0273.JPG' or ID =='IMG_0274.JPG':
    engine = pyttsx.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate-30)
    engine.say('You are infront of office of assistant professor engineer fazal ahmed')
    engine.runAndWait()

elif ID == 'IMG_0203.JPG' or ID =='IMG_0202.JPG':
    engine = pyttsx.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate-30)
    engine.say('Infront of E C R at ten feet away')
```

```
        engine.runAndWait()

elif ID == 'IMG_0206.JPG' or ID =='IMG_0215.JPG':
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('infront of doctor safia akrams office')
        engine.runAndWait()

elif ID == 'IMG_0238.JPG' or ID =='IMG_0236.JPG' or ID =='IMG_0234.JPG':
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('infront of computer lab 1')
        engine.runAndWait()

elif ID == 'IMG_0240.JPG' or ID =='IMG_0237.JPG':
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
        engine.setProperty('rate', rate-30)
        engine.say('exit door of first floor')
        engine.runAndWait()

elif ID == 'IMG_0241.JPG' or ID =='IMG_0243.JPG':
        engine = pyttsx.init()
        rate = engine.getProperty('rate')
```

```
    engine.setProperty('rate', rate-30)
    engine.say('On your left is computer lab 1 and left front E C R')
    engine.runAndWait()

elif ID == 'IMG_0046.JPG':
    engine = pyttsx.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate-30)
    engine.say('Standing right infront of main stairs to floor 1')
    engine.runAndWait()
```