Department of Electrical
and
Computer Systems Engineering

Technical Report
MECSE-25-2005

Vision-based indoor localization of a motorized wheelchair

P. Chakravarty

MONASH
UNIVERSITY

Vision-based Indoor Localization of a Motorized Wheelchair

Punarjay Chakravarty
Department of Electrical and Computer Systems Engineering,
Monash University, Clayton, Vic 3800, Australia
Punarjay.Chakravarty@eng.monash.edu.au

Abstract

This project has achieved the localization of a motorized wheelchair in an indoor environment. A camera, mounted on top of the wheelchair is the only sensor. The SIFT algorithm, which identifies images and objects irrespective of changes in scale and viewpoint perspective has been implemented. This algorithm has been used to obtain the wheelchair's current position on a (previously built) topological map of a corridor in the environment. Each topological node on the map is associated with a set of (previously taken) pictures, stored in a database. The SIFT algorithm is used to compare the current picture from the camera to the pictures in the database. The best matched picture gives the current topological position of the wheelchair on the map.
A Principal Components Analysis (PCA) based modification of the basic SIFT algorithm that reduces its dimensionality is discussed. A Hidden Markov Model modeling the transition probabilities from one topological node to another improves the robustness of the localization.

Acknowledgements

This project was carried out during 2 months spent at the IRIS (Institute for Robotics and Intelligent Systems) Lab in Bangalore, India. I thank Dr. Sitaram, director and Dr. Naidu, project director for giving me the opportunity to conduct research at this institution. I am grateful to Mr. Sartaj Singh, my supervisor, and lead scientist of the Robotics group, for giving me full rein of the motorized wheelchair, which was used as the mobile platform to test the algorithm. I thank Mr. Manohar (lead scientist of the Virtual Reality group) for his help in converting some of the Matlab code to C. I am also grateful to Dr. Subrata Rakshit (head of Computer Vision group) for inputs regarding dimensionality reduction.

Table of Contents

1. Introduction

This project investigates the use of Computer Vision for Localization. The objective is to localize a motorized wheelchair in an indoor environment, by matching the current image from the wheelchair camera to prior pictures taken in that environment. The environment should not have been modified in any way to make the image matching easier. The mounting of a webcam on the wheelchair is the only modification made to it. The wheelchair is used as the mobile agent, because such a wheelchair is readily available at the IRIS Lab (a motorized wheelchair was one of the past projects of the organization) and because the testing of the system is much easier on a wheelchair. One can drive the wheelchair to different locations in the environment and take pictures. A laptop carried along by the person driving the wheelchair grabs the pictures from the webcam, runs the localization algorithm and displays the position of the wheelchair on an on-screen map.

Figure 1 Motorized wheelchair equipped with camera and laptop acts as the mobile platform for testing the localization algorithm

Localization using vision involves identifying landmarks in an environment, and ascertaining the wheelchair's position with respect to the landmark. The visibility region for each landmark (the positions on the ground where the landmark is significantly visible) constitute topological nodes on a map that the position of the robot is localized to. The recognition of natural landmarks in images is essentially a problem of image matching (to a database of pre-stored images). A common approach to image matching involves the extraction of interest points from the image, calculating an image descriptor associated with each interest point (based on the local image characteristics) and matching these descriptors. The search for an image descriptor that is invariant to changes in scale, orientation, illumination and 3D camera viewpoint has been on for the last 25 years. The Harris Corner detector [1] is one such interest point operator which works on the principle that at a corner, the derivative of the intensity gradient is large in perpendicular directions to each other.

Zhang et.al [2] matched image regions around the Harris Corner detector by using a normalized cross-correlation function. A fundamental matrix describing the geometric relationship between 2 views of the scene was computed using the matched image points and outlying points (which did not match the majority solution) were removed. This method was effective in short-baseline stereo matching where there was not much disparity in the images. Schmidt and Mohr [3], in 1997 devised a rotationally invariant image descriptor that was calculated from an image patch around the Harris Corner points. But because the Harris corner points are not invariant to changes in scale, therefore, this method was also not invariant to scale.

In 1999, David Lowe came up with the SIFT (Scale Invariant Feature Transform) algorithm [4] which has, in the last few years gained increasing popularity for object and scene recognition [5-7]. It has been used, in combination with probabilistic models (Hidden Markov Models, Markov Chain Monte Carlo methods etc.) to localize a mobile robot in an indoor environment through pictures taken (by the same agent) in that environment at periodic intervals.

2. Background behind SIFT

The SIFT algorithm identifies interest points at the maxima of a Difference of Gaussian function. The Gaussian function is used to locate features at a variety of different scales. This is illustrated in Figure 2. A Gaussian pyramid is constructed by successively Gaussian blurring and downsampling an image. (Blurring involves convolution of the image with a Gaussian kernel). The Gaussian pyramid in Figure 2 illustrates that lower layers of the pyramid (toward the bottom of the figure) reveal finer details, like the writing on the TV Guide and the design on the tablecloth but higher layers of the pyramid reveal only coarser details like the outline of the TV Guide and the table.

Lowe [8] quotes work by Koenderink and Lindeberg that under a variety of reasonable assumptions the only possible scale-space kernel is the Gaussian function. He also quotes work by Mikolajczyk [9] that the maxima of a Difference of Gaussian Function is the most stable image feature. He indicates that interest points that are most robust against changes in scale, are most efficiently extracted from the Difference of Gaussian function.

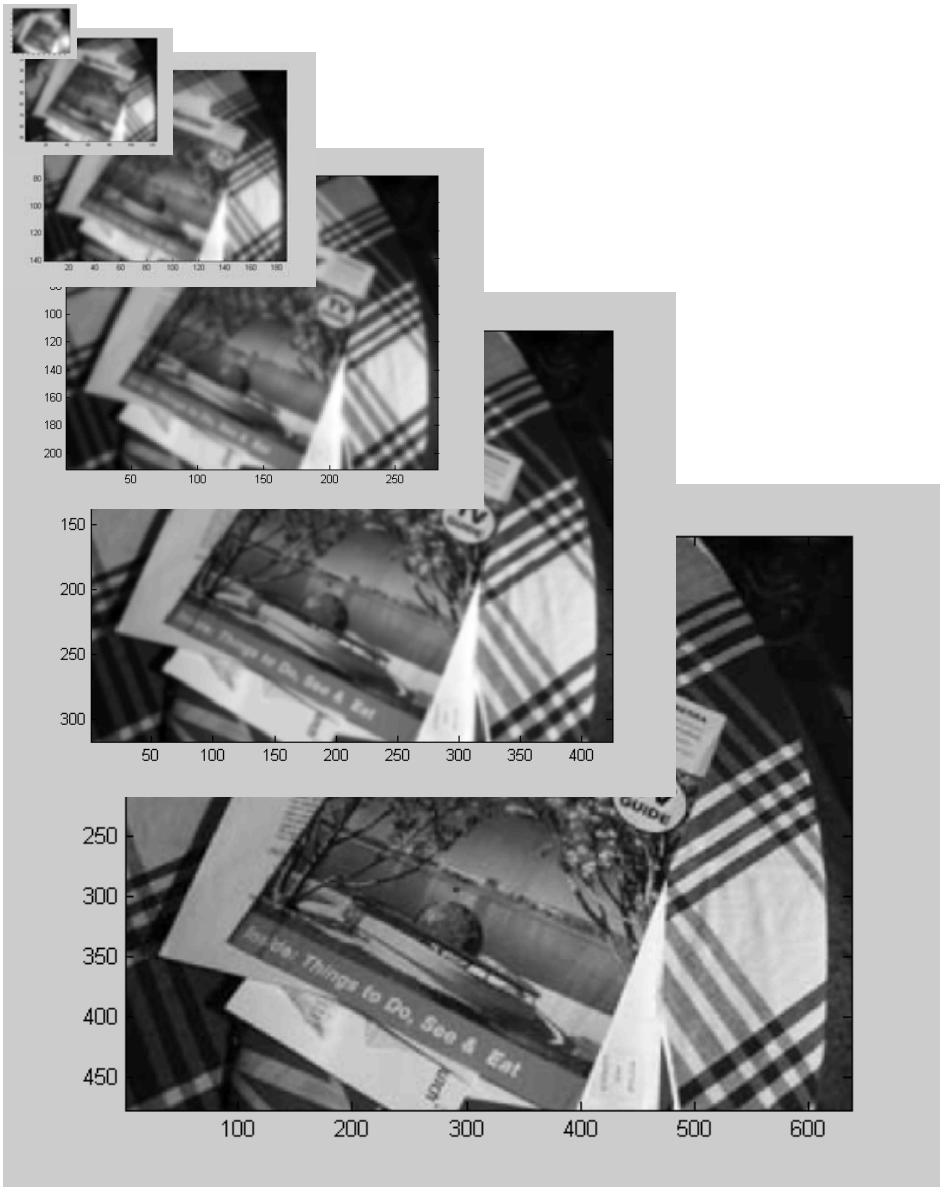

Figure 2 A Gaussian pyramid constructed by successively Gaussian blurring and downsampling an image. Lower layers of the pyramid (toward the bottom of the figure) reveal finer details, like the writing on the TV Guide, but higher layers of the pyramid reveal only coarser details like the outline of the TV Guide.

3. SIFT

The SIFT algorithm consists of the following major steps:

1. Scale-space peak detection
2. Accurate key-point localization
3. Majority orientation assignment
4. Computation of the local image descriptor

3.1 Step 1: Scale-space peak detection

In the 1st step, the image is searched for peaks over both location and scale. This is efficiently implemented by constructing a Gaussian pyramid and searching for local peaks in successive Difference of Gaussian (DoG) images. The Gaussian pyramid is constructed by taking the image and successively blurring it with a Gaussian kernel. Successive layers of this pyramid are subtracted to get another pyramid of DoG images (Figure 3).

(Note: In the implementation, the original image is initially upsampled by a factor of 2 before the pyramid is constructed. Gaussian blurring the original image is equivalent to low-pass filtering it, discarding the higher frequencies. Upsampling the initial image has the effect of introducing a guard band preserving the information content in the highest frequencies) Each pixel in the DoG images are checked for being a local maxima by comparing with the 8 neighbouring pixels on the same level of the pyramid and the 18 pixels from the image patches in the immediately adjacent layers as shown in Figure 4.

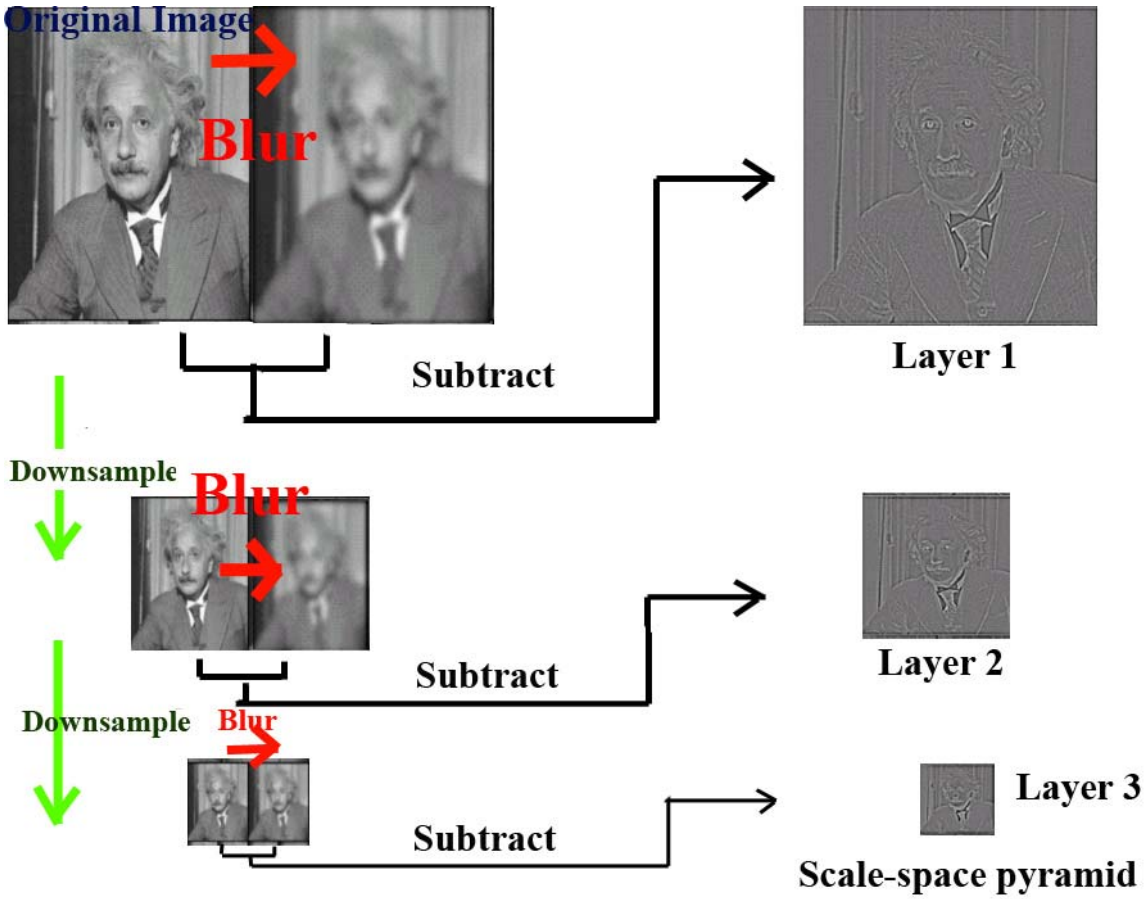

Figure 3 Construction of the 1st 3 layers of the Difference of Gaussian (Scale-space) pyramid

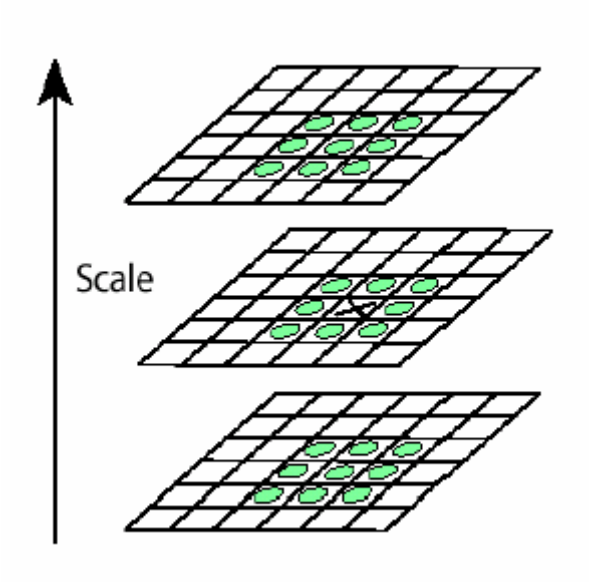

Figure 4 Each pixel in the DoG images (marked with an x) are checked against its neighbours (marked with green circles) on the same and adjacent layers for peak detection. (Figure from [8])

3.2 Step 2: Accurate Keypoint Localization

An accurate position fix on the keypoints located in the previous step has been implemented (as suggested as an extension to the original SIFT algorithm by [10]), by fitting a 3D quadratic function to the local sample points. Subsequently, all poorly defined peaks (termed edge response by [8]) that have larger derivative of the intensity gradient in 1 direction than the other perpendicular direction are eliminated. This can be ascertained from the matrix H:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

If the one eigen value of H is significantly larger than the other, then the point is an edge. (Note: The derivatives in the matrix H are determined by taking differences of neighbouring sample points.)
[8] shows that an efficient method of ascertaining if the ratio of the eigen values is large, is by checking Equation 1. (More details in this step can be obtained from [8]).

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \qquad \textbf{Equation 1}$$

3.3 Step 3: Majority Orientation Assignment

This step makes the descriptor rotation invariant. It involves calculating the gradient vectors in a window around the SIFT feature on the scale at which the feature was detected. So, for example, if a SIFT feature was detected with scale 3 (third level of the DoG pyramid), the 3rd layer of the Gaussian pyramid is accessed and the gradients (magnitude and direction) of the intensity values around the SIFT feature are calculated:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad \textbf{Equation 2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \quad \textbf{Equation 3}$$

where $m(x,y)$ and $\theta(x,y)$ indicate the magnitude and direction of the gradient at location (x,y). $L(x,y)$ indicates the intensity of the pixel at location (x,y).

The gradients are pre-computed on all the levels of the Gaussian pyramid. This is done by convolving the layers of the Gaussian pyramid with the 2 Prewitt masks (shown in Figure 5) to get the x and y directional derivatives (xgrad(x,y) and ygrad(x,y)) at each pixel position.

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Figure 5 x and y Prewitt masks

Then, the magnitude and direction of the gradient vector at each pixel position can be found out as follows:

$$m(x,y) = \sqrt{xgrad^2 + ygrad^2} \quad \textbf{Equation 4}$$

$$\theta(x,y) = \tan^{-1}(ygrad / xgrad) \quad \textbf{Equation 5}$$

The gradient orientations are computed in a 16x16 window around the SIFT feature and quantized in 8 steps of 45 degree intervals. (Note: [8] cites a 45-deg quantization interval, but I have used a 5-deg quantization interval for increased accuracy.) The histogram (Figure 7) is incremented by a sample that is weighted by its gradient magnitude and a Gaussian-weighted circular kernel that is placed on top of the 16x16 window. This has the effect of giving a higher weight to the samples near the centre of the window.

The majority gradient direction is subtracted from each of the gradient directions in the window and the window is rotated so that the majority gradient is perpendicular to the top margin of the window (Figure 8).

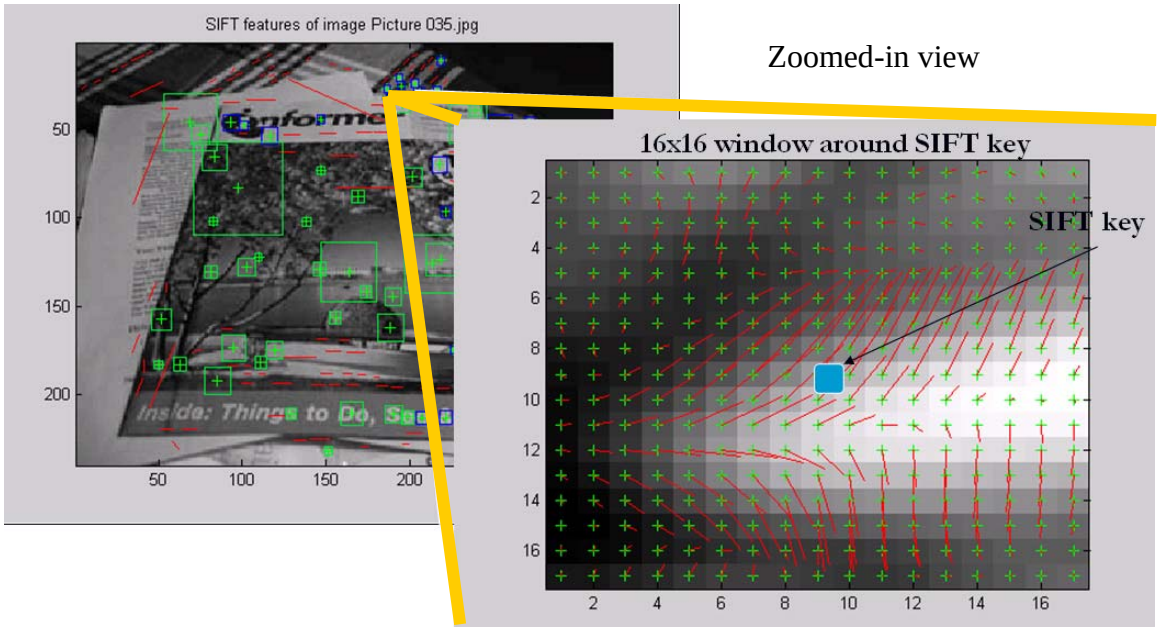

Figure 6 A Close-up look at a 16x16 window around a SIFT key (blue). Red vectors represent the intensity gradients in this window. Green crosses indicate the pixel centres.

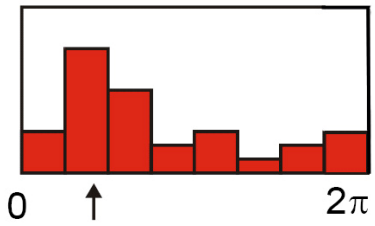

Figure 7 Orientation histogram with 8 bins at intervals of 45 deg

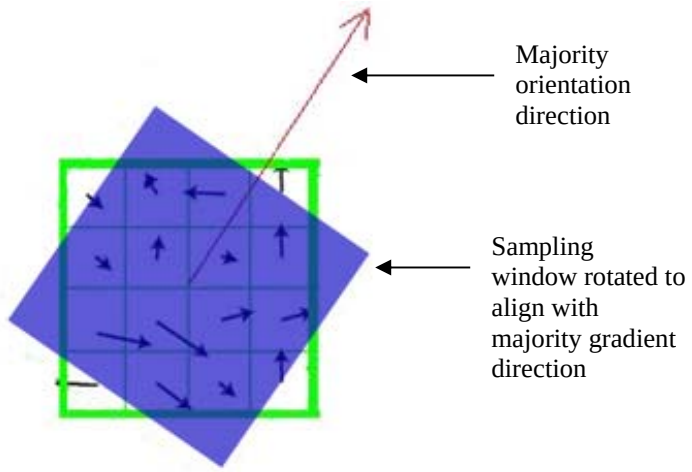

Figure 8 Sampling window (green) around SIFT feature and intensity gradient directions (black vectors). Majority orientation direction shown in red. Sampling window is rotated (rotated window indicated in blue) to align with majority gradient orientation.

3.4 Step 4: Computation of the local image descriptor

This step associates each SIFT feature point with a 128-element feature vector that uniquely identifies that point.

The gradient orientations obtained at the end of Step 3 (weighted by their magnitudes and by a circular Gaussian kernel) are arranged into 16 histograms (as shown in Figure 9, except that the figure shows an 8x8 window and 4 orientation histograms to reduce clutter). Values of the orientation histogram constitute the 128-dimensional vector (8 orientations x 16 histograms). This 128-dimensional vector is the SIFT feature vector.

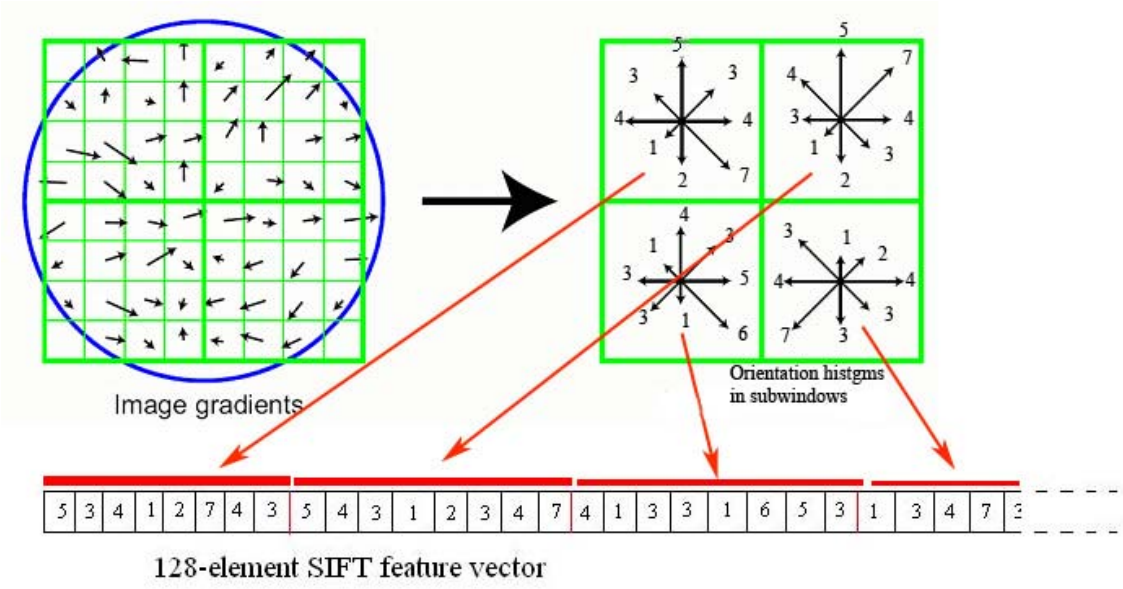

Figure 9 Values of orientation histograms calculated in sub-windows arranged as elements of the SIFT feature vector

4 PCA-SIFT

In 2004, Ke and Sukthankar [11] introduced a PCA-based local descriptor that is more invariant to image deformations and more compact than the SIFT feature vector.
The PCA-SIFT algorithm accepts the same input as the SIFT operator, i.e., the sub-pixel image locations, scale and dominant orientation of the keypoint. However, after this, the method of construction of the feature vector differs from Lowe's SIFT algorithm.
A 40-by-40 image patch is extracted around the interest point and rotated so that its dominant orientation is aligned with a majority orientation direction (as in Step 3 of the normal SIFT). The horizontal and vertical components of gradient vectors in this window are concatenated to form a 3200-element vector (40x40x2).

Then, the following steps are performed:

1. An Eigen-space representation of the gradient information of local image patches is pre-computed off-line by taking a large number of gradient patches

around SIFT interest points from many images. A large number of 3200-dimensional vectors extracted from the gradients around these patches are used to construct a covariance matrix.

2. The matrix consisting of the top 20 eigenvectors (of the covariance matrix) is used as the projection matrix.
3. Online, given an image patch around a SIFT feature located in the current (query) image, its 3200-element feature-vector is extracted and is projected using the projection matrix to get a 20-dimensional feature vector.

Thus, the PCA-SIFT algorithm has achieved a reduction in dimensionality from 128 (in the original SIFT vector) to 20.
It is also found to give better results in image matching than the basic SIFT (as claimed by [11]).

To summarize, the SIFT algorithm identifies stable key locations in scale space. This means that scale changes of objects in an image will have no effect on the key locations selected. An explicit scale is determined at each point, allowing the image description vector to be sampled at an equivalent scale in each image. Also, the dominant gradient direction is identified at each SIFT point, which means that the transform is invariant to object rotation. At each SIFT point, the gradients around the point (at the identified scale) is arranged as a 128-element feature vector (and reduced to a 20-dimensional vector using the PCA-SIFT extension) which characterizes that SIFT feature. It is this vector that is matched to a database of feature vectors.

5 Wheelchair Localization

The distinctive invariant features of SIFT are used to match digital image content between 2 views of a scene. Images corresponding to views taken from the wheelchair camera as it moves through a building can be matched to a database of images using the SIFT algorithm. Each image in the database is associated with a topological node on a map of the environment. Then later, when the wheelchair webcam views a scene, the algorithm can find out where it is on the topological map based on the best-matched image. This is necessary for an algorithm that needs to navigate the wheelchair (or a robot) from one room to another.

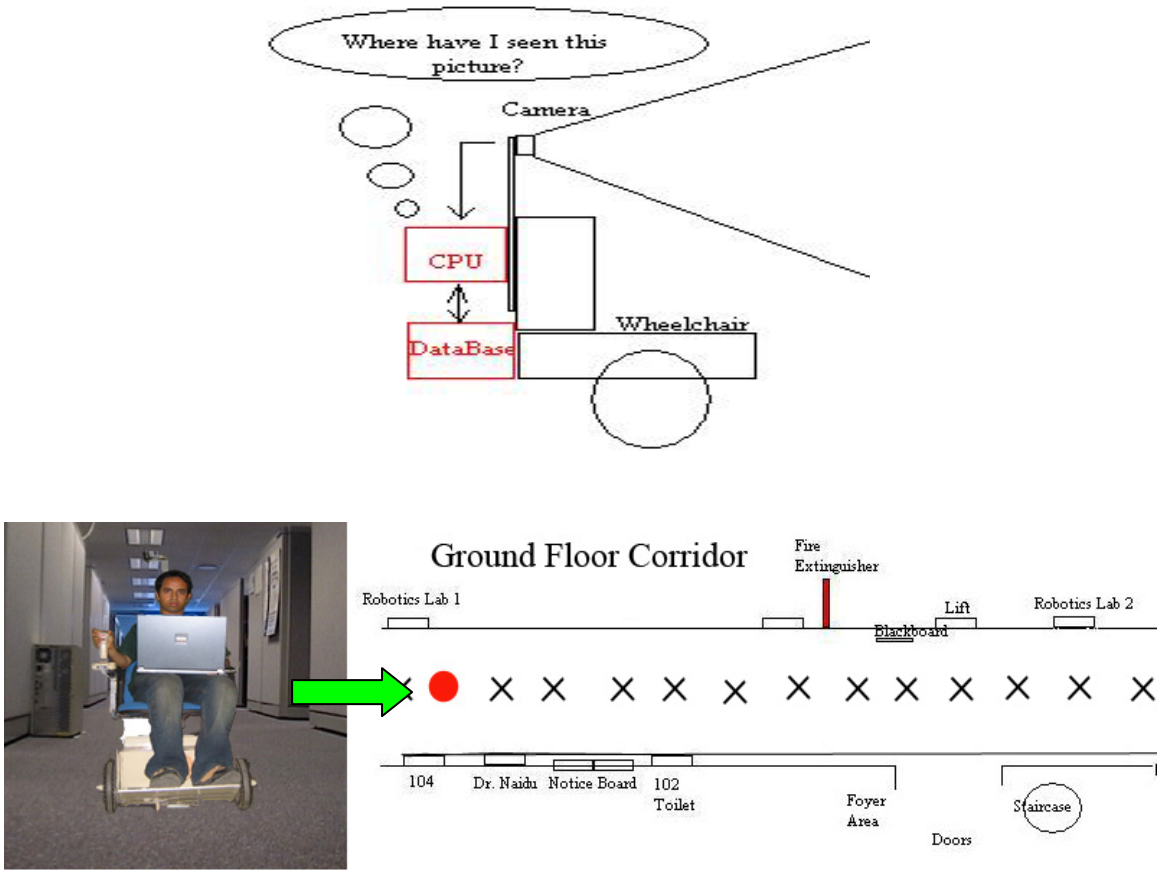

Figure 10 shows the idea behind the vision-based localization

A topological map of ground-floor was made, with topological nodes representing the places at which images were taken. The wheelchair was taken to each topological node and 8 pictures taken representing each node (360 degrees covered in eight 45-deg intervals). SIFT features for each picture were calculated offline and stored in the database. During localization, the current picture from the camera was compared with images in the database. The closest match gave the location of the wheelchair on the topological map.

Figure 11 Left column shows images from webcam during localization and right column shows best matched image in database

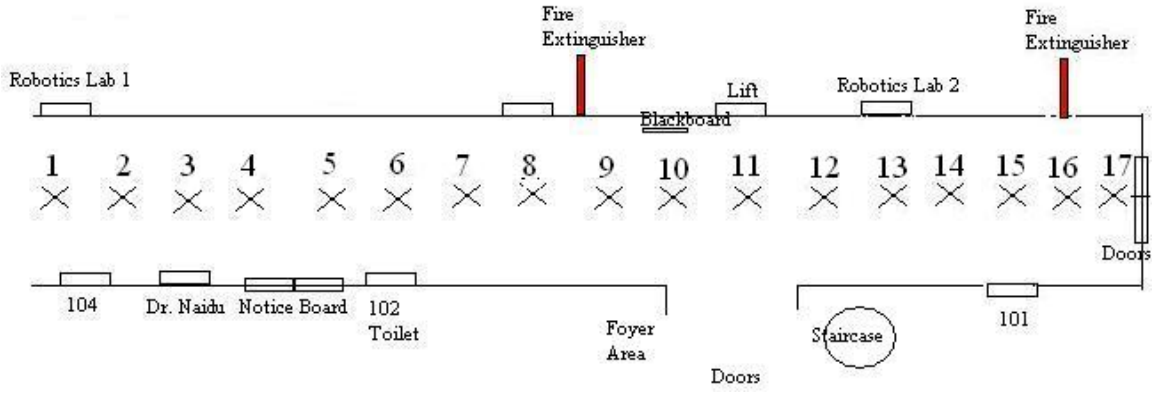

Figure 12 Figure shows topological nodes (marked with X) in IRIS corridor

5.1 Hidden Markov Model

A Bayesian filter incorporating a Hidden Markov Model [6, 12, 13] that models the transition probability between topological locations is used to increase the robustness of the localization. For example, say that at time k, the wheelchair is localized to be at node 1 in front of Robotics Lab 1 (Figure 12) and at time k+1, because of faulty image matching, its location is given to be at node 11, in front of the lift. This transition is impossible in 1 time step, and is given a very low probability by the Hidden Markov Model.

A recursive Bayesian filter that characterizes the probability of the current state x (indicating the current topological location), given the sequence of past observations z^k upto time k (past observations are the past image matches) is given by:

$$P(x \mid Z^k) = \frac{(P(z_k \mid x)P(x \mid Z^{k-1})}{P(z_k \mid Z^{k-1})} \qquad \textbf{Equation 6}$$

where the sensor model is calculated by

$$p(z_k \mid x_k = x_i) = \frac{C(i)}{\sum_j C(j)} \qquad \textbf{Equation 7}$$

The LHS of Equation 7 gives the probability of making the observation z at time k, given that the wheelchair is at state x_i (topological location) at the same time. C(i) indicates the number of matching SIFT feature vectors between the current image and the closest image in the database (that are below a threshold Euclidean distance). This value is normalized by the total number of matches (below a threshold Euclidean distance) to the entire database of images.

The transition between states x_i and x_j (locations x_i and x_j on the topological map) are modeled by a Hidden Markov Model. The probability of being at state x_i , given that the state at the immediately previous time step was x_j ($p(x_k = x_i \mid x_{k-1} = x_j)$)is given by A(j,i). A is a NxN matrix, where N is the number of topological locations. Entries in the matrix corresponding to adjacent locations are given a value one, and the final matrix is normalized across each row.

The conditional prior $P(x|z_{k-1})$ (probability of being at state x, given past observations until time k-1) in Equation 6 is found using the transition matrix A(j,i) as follows:

$$p(x_k = x_i \mid Z^{k-1}) = \sum_j^n A(j,i)\, p(x_{k-1} = x_j \mid Z^{k-1}) \qquad \textbf{Equation 8}$$

A localization experiment involving driving the wheelchair down the corridor from node 1 to 17 was performed 2 days after the database was constructed, and at a

different time of day (database images were taken in the morning, and localization was performed in the afternoon).The results from the experiment are shown in Table 1. The actual ground truth position is indicated by G, the results from the image matching by I, and the Bayesian-HMM filtered position indicated by H.

The whole process of SIFT feature point localization, key extraction and matching the key to keys of images stored in a database took on an average of 7 seconds per location for a Matlab-based implementation. (Note: The calculation of SIFT keys for the pictures in the pre-constructed database was done offline.) Computationally intensive parts of the SIFT key extraction, like image re-sampling, gradient quantization and weighted histogram calculation were re-coded in CMex, for faster execution.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	G,I,H				I							I					
2		G,I,H															
3			G,H														
4				G,I,H	H												
5			I		G												
6						G,I,H		I								I	
7							G,I,H	H	I								
8								G	H								
9									G								
10										G,I,H							
11											G,I,H	H					
12												G					
13													G,I,H				
14														G,I,H			
15															G,I,H	H	
16																G	
17																	G,I,H

Frame Number ⟶

Table 1 shows the outcome of the localization experiment. x axis indicates incoming frame number and on y axis are the topological nodes. G indicates actual ground truth position, I indicates location as determined by SIFT image matching, and H indicates location from the Hidden Markov Model

6 Conclusions and Future Work

The Scale Invariant Feature Transform (SIFT) algorithm, that matches images by extracting interest points invariant to changes in perspective, scale and rotation (and matching the local gradient information around them) was implemented in Matlab. The dimensionality of the SIFT feature vector was reduced from 128 to 20 using the Principal Component Analysis. This image matching algorithm forms the basis for a topological vision-based localization algorithm for a motorized wheelchair in an indoor environment. Images corresponding to topological nodes in the environment (an indoor corridor at the IRIS facility) were captured from the webcam (mounted atop the wheelchair) and stored in a database. In an experiment performed 2 days later, the algorithm localized the wheelchair on the topological map every time a frame was captured as it was driven down the corridor. A Hidden Markov Model used to model the transition probabilities between the nodes improved the robustness of the localization to false image matches.

Possible improvements in the localization algorithm would include a better image normalization procedure (possibly histogram equalization) that makes the algorithm more robust to ambient light intensity. A denser sampling of images (by increasing the number of database images per topological node and also including images taken at different times of day) would improve the accuracy of image matching.

Currently, the map is hand-made and the database generation is done interactively by driving the wheelchair to different physical locations and capturing images. This could be automated. For this, the wheelchair needs to have a navigation algorithm (that would include wall following and obstacle avoidance routines) that would enable it to explore different places and also a map-building algorithm, that would allow it to build an incremental map as it moves through the environment. Lowe, Le and Little [14] report on such a system that calculates camera (and hence robot) ego-motion by matching SIFT features between frames. The system contains 3 cameras that are used to construct 3D world coordinates for each set of matched SIFT features. The SIFT features and their corresponding 3D coordinates serve as landmarks for map-building and tracking.

References

[1] **C. Harris and M. Stephens, "A Combined Corner and Edge Detector," presented at Fourth Alvey Vision Conference, 1988.**

[2] **Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong, "A Robust Technique for Matching Two Uncalibrated Images through the Recovery of Unknown Epipolar Geometry," *Artificial Intelligence*, vol. 78, pp. 87-119, 1995.**

[3] **C. Schmid and R. Mohr, "Local Grayvalue Invariants for Image Retrieval," presented at IEEE Transactions On Pattern Analysis and Machine Intelligence, 1997.**

[4] **D. Lowe, "Object Recognition from Local Scale Invariant Features," presented at International Conference on Computer Vision, 1999.**

[5] **C. Silpa-Anan and R. Hartley, "Localization using an image-map," presented at Australian Conference on Robotics and Automation, Canberra, Australia, 2004.**

[6] **L. Ledwich and S. Williams, "Reduced SIFT Features for Image Retrieval and Indoor Localization," presented at Australian Conference on Robotics and Automation, Canberra, Australia, 2004.**

[7] **S. Se, D. Lowe, and J. Little, "Global Localization using Distinctive Visual Features," presented at International Conference on Intelligent Robots and Systems, EPFL, Lausanne, Switzerland, 2002.**

[8] **D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Internation Journal of Computer Vision*, 2004.**

[9] **K. Mikolajczyk, "Detection of Local Features Invariant to Affine Trasnformations, Ph.D. thesis." Grenoble, France: Institut National Polytechnique, 2002.**

[10] M. Brown and D. G. Lowe, "Invariant Features from Interest Point Groups," presented at British Machine Vision Conference, Cardiff, Wales, 2002.

[11] Y.Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," presented at Computer Vision and Pattern Recognition, 2004.

[12] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-based Vision system for place and object recognition," presented at International Conference on Computer Vision, 2003.

[13] J. Kosecka and F. Li, "Vision Based Markov Localization," presented at 23rd IEEE International Conference on Robotics and Automation (ICRA 2004), New Orleans, USA, 2004.

[14] S. Se, D. Lowe, and J. Little, "Vision-based Mapping with Backward Correction," presented at IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, EPFL, Lausanee, Switzerland, 2002.