# Improved Navigation For Visually Challenged With High Authentication Using a Modified Sift Algorithm

A.Lakshmana Kumar M.E
Department of VLSI Design
Sethu Institute of Technology
*lakshudeepu@gmail.com*

Dr.R.Ganesan M.E.,Ph.D.,
Department of VLSI Design
Sethu Institute of Technology
*ganesanhod@gmail.com*

***Abstract*** **– This project describes the use of SIFT algorithm for the purpose of giving support to Blind and visually impaired. It is a greater challenge for the visually impaired and blind people's to work alone and to satisfy their Needs. The main things that a Blind can need are to identify an Object, Image Recognition, Face recognition. This paper Satisfies the Blind People's Needs by using the SIFT algorithm. SIFT algorithm is mainly based on the object and image retrieval method. Using image processing Tools in MATLAB to implement the SIFT algorithm to the Images. Improve the quality of life for the visually impaired by restoring their ability to self-navigate. Provide the Authenticated Information. Improve the mobility without an external help using Self navigation and Person Identification.**

***Keywords:*** *VGA; SIFT; Difference of Gaussian; scale space; key points; scale invariance; image matching*

## I. INTRODUCTION

As Per the Census Report of the **W**orld **H**ealth **O**rganization, 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 have low vision. About 90% of the worlds visually impaired live in developing countries. 82% of people living with blindness are aged 50 and above. The number of people visually impaired from infectious diseases has greatly reduced in the last 20 years. 80% of all visual impairment can be avoided or cured. It is a greater challenge for the visually impaired and blind people's to work alone and to satisfy their Needs. The main things that a Blind can need are to identify an Object, Image Recognition, Face recognition. This paper Satisfies the Blind People's Needs by using the SIFT algorithm. SIFT algorithm is mainly based on the object and image retrieval method. Using image processing Tools in MATLAB to implement the SIFT algorithm to the Images. Improve the quality of life for the visually impaired by restoring their ability to self-navigate. Provide the Authenticated Information. Improve the mobility without an external help using Self navigation and Person Identification.

## II PROPOSED WORK

Get more authenticated information from training Images. SIFT Algorithm is used. Features are extracted from captured image using SIFT. Resultant Image compared with the Database images and the matched image information is transmitted through voice signal to Get authenticated Information. Block diagram represents the proposed architecture. First process is to capture the image from the image source such as web camera. The captured image should be of high clarity. Then the SIFT algorithm is applied to the captured image. Three octaves levels are used here. Various keypoints are found on apply of the algorithm. We have to apply the SIFT algorithm to the database image also. Finally the keypoints for the both the database image and the captured images are matched, and the matching percentage is calculated. Thus by the matched percentage the authenticated information is transmitted as the voice command through headphones.

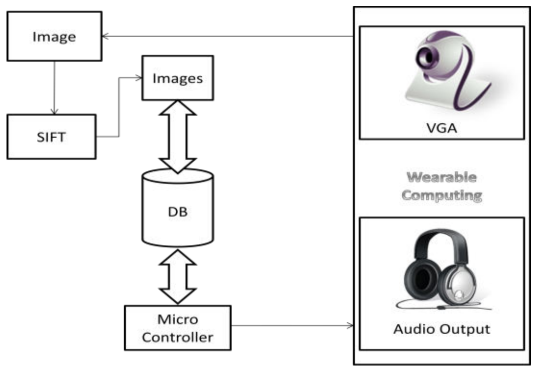


**Fig. 1: Proposed Architecture**

## III. Modified SIFT (Scale Invariant Feature Transform)

In SIFT descriptor DoG detector is used to detect interest points and the extracted regions are described by a vector of dimension 128. The descriptor is normalized by dividing the descriptor vector by square root of sum of the squared components, so that the descriptor becomes illumination invariant. Nowadays many detectors and descriptors algorithms are available for detecting corners edges and regions of interest. A vector is associated with it as a descriptor. The SIFT algorithm by Lowe is explained here which is used to provide primary input to the image matching algorithm. The detected region should have a shape which is a function of the image. To characterize the region invariant descriptor is computed for the extracted region. The Difference of Gaussian function will have strong response along edges, though the location along the edge is poorly determined, as these locations are unstable to small amount of noise. A poorly defined peak in the Difference of Gaussian function will have a large principle curvature across the edge but small along perpendicular direction. Over all scales image locations are found to detect the extrema. Scales of keypoint is used to select the Gaussian smoothed image of closest scale, so that the computations are performed in scale invariant manner. The descriptor computed using these gradient magnitude and orientation at each image sample point is weighted by Gaussian window. A Gaussian weighting function with $\sigma$ equal to one half of the width of the descriptor window is used to assign a weight to the magnitude of each sample point.

**Steps in the Sift Algorithm**

- Constructing a Scale Space.
- Dog Approximation.
- Finding Key points.
- Get rid of bad key points.
- Assigning an Orientation to the Key points.
- Generate Sift Features.

*A Constructing a scale space*

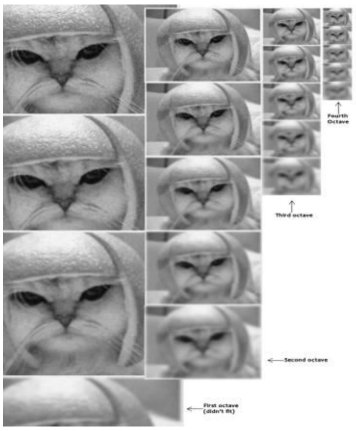


**Fig. 2: Constructing a scale space**

SIFT takes scale spaces to the next level. Take the original image, and generate progressively blurred out images. Then, you resize the original image to half size and generate blurred out images again and keep repeating. The creator of sift suggests that 4 octaves and 5 blur levels are ideal for the algorithm.

If the original image is doubled in size and antialiased a bit (by blurring it) then the algorithm produces more four times more keypoints. The more the keypoints, the better.

Blurring

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \qquad (1)$$

Where * is the convolution operation in x and y ,and

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)}/2\sigma^2 \qquad (2)$$

| octave \ scale → | | | | |
|---|---|---|---|---|
| 0.707107 | 1.000000 | 1.414214 | 2.000000 | 2.828427 |
| 1.414214 | 2.000000 | 2.828427 | 4.000000 | 5.656854 |
| 2.828427 | 4.000000 | 5.656854 | 8.000000 | 11.313708 |
| 5.656854 | 8.000000 | 11.313708 | 16.000000 | 22.627417 |

**Fig. 3: Amount of Blurring**

*B DoG approximation*

This is the stage where the interest points, which are called keypoints in the SIFT framework, are detected. For this, the image is convolved with Gaussian filters at different scales, and then the difference of successive Gaussian-blurred images is taken. Keypoints are then taken as maxima/minima of the Difference of Gaussians (DoG) that occur at multiple scales. Specifically, a DoG image $D(x,y,\sigma)$ is given by

$$D(x,y,\sigma) = L(x,y,k_i\sigma) - L(x,y,k_j\sigma) \qquad (3)$$

where $L(x,y,k\sigma)$ is the convolution of the original image $I(x,y)$ with the Gaussian blur $G(x,y,k\sigma)$ at scale $k\sigma$, i.e.,

$$L(x,y,k\sigma) = G(x,y,k\sigma) * I(x,y) \qquad (4)$$

Hence a DoG image between scales $k_i\sigma$ and $k_j\sigma$ is just the difference of the Gaussian-blurred images at scales $k_i\sigma$ and $k_j\sigma$. For scale space extrema detection in the SIFT algorithm, the image is first convolved with Gaussian-blurs at different scales.The convolved images are grouped by octave and the value of $k_i$ is selected so that we obtain a fixed number of convolved images per octave. Then the Difference-of-Gaussian images are taken from adjacent Gaussian-blurred images per octave.

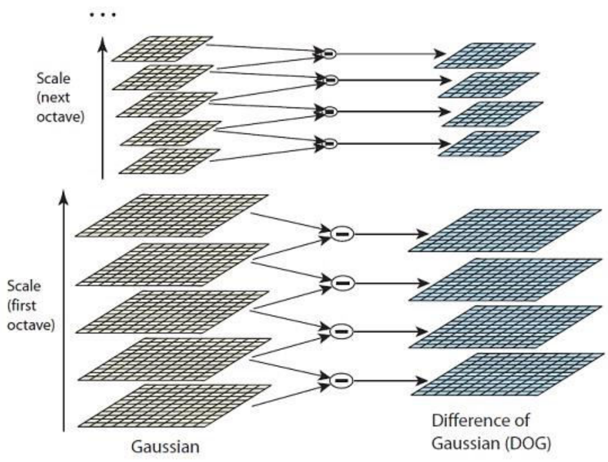


**Fig. 4: DoG approximation**

*C Finding Key Points*

Once DoG images have been obtained, keypoints are identified as local minima/maxima of the DoG images across scales. This is done by comparing each pixel in the DoG images to its eight neighbours at the same scale and nine corresponding neighbouring pixels in each of the neighbouring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint.

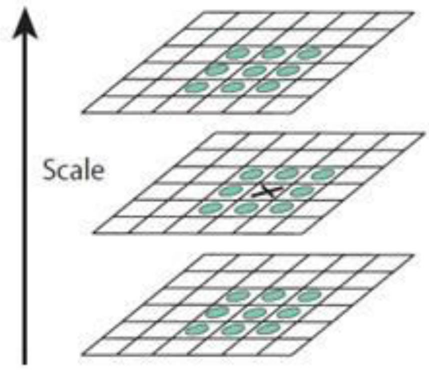


**Fig. 5: Locate maxima/minima in DoG images**

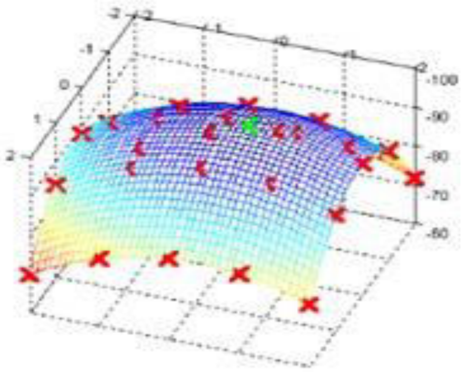

**Fig. 6: Subpixel maxima/minima**

The new approach calculates the interpolated location of the extremum, which substantially improves matching and stability. The interpolation is done using the quadratic Taylor expansion of the Difference-of-Gaussian scale-space function, $D(x, y, \sigma)$ with the candidate keypoint as the origin. This Taylor expansion is given by

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \qquad (5)$$

where D and its derivatives are evaluated at the candidate keypoint and $\mathbf{x} = (x, y, \sigma)$ is the offset from this point.

The location of the extremum, $\hat{\mathbf{x}}$, is determined by taking the derivative of this function with respect to $\mathbf{X}$ and setting it to zero. If the offset $\hat{\mathbf{x}}$ is larger than $0.5$ in any dimension, then that's an indication that the extremum lies closer to another candidate keypoint. In this case, the candidate keypoint is changed and the interpolation performed instead about that point.

Otherwise the offset is added to its candidate keypoint to get the interpolated estimate for the location of the extremum. A similar subpixel determination of the locations of scale-space extrema is performed in the real-time implementation based on hybrid pyramids developed by Lindeberg and his co-workers.

*D Get rid of bad key points*

After scale space extrema are detected (their location being shown in the uppermost image) the SIFT algorithm discards low contrast keypoints (remaining points are shown in the middle image) and then filters out those located on edges. Resulting set of keypoints is shown on last image.

**Fig. 6: Get rid of bad key points**

Scale-space extrema detection produces too many keypoint candidates, some of which are unstable. The next step in the algorithm is to perform a detailed fit to the nearby data for accurate location, scale, and ratio of principal curvatures.The value of the second-order Taylor expansion $D(\mathbf{x})$ is computed at the offset $\hat{\mathbf{x}}$.

If this value is less than $0.03$, the candidate keypoint is discarded. Otherwise it is kept, with final scale-space location $\mathbf{y} + \hat{\mathbf{x}}$, where $\mathbf{y}$ is the original location of the keypoint. we need to eliminate the keypoints that have poorly determined locations but have high edge responses. For poorly defined peaks in the DoG function, the principal curvature across the edge would be much larger than the principal curvature along it. Finding these principal curvatures amounts to solving for the eigenvalues of the second-order Hessian matrix, H

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \qquad (6)$$

The eigenvalues of H are proportional to the principal curvatures of D. It turns out that the ratio of the two eigenvalues, say $\alpha$ is the larger one, and $\beta$ the smaller one, with ratio $r = \alpha/\beta$, is sufficient for SIFT's purposes. The trace of H, i.e., $D_{xx} + D_{yy}$, gives us the sum of the two eigenvalues, while its determinant, i.e., $D_{xx}D_{yy} - D_{xy}^2$, yields the product.

The ratio $\mathrm{R} = \mathrm{Tr}(\mathbf{H})^2 / \mathrm{Det}(\mathbf{H})$ can be shown to be equal to $(r+1)^2/r$, which depends only on the ratio of the eigenvalues rather than their individual values. R is minimum, when the eigenvalues are equal to each other. Therefore the higher the absolute difference between the two eigenvalues, which is equivalent to a higher absolute difference between the two principal curvatures of D, The higher the value of R.It follows that, for some threshold eigenvalue ratio $r_{th}$, if R for a candidate keypoint is larger than $(r_{th}+1)^2/r_{th}$, that keypoint is poorly localized and hence rejected. The new approach uses $r_{th}=10$.

### E Assigning an Orientation to the Key points

Each keypoint is assigned one or more orientations based on local image gradient directions. This is the key step in achieving invariance to rotation as the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.

First, the Gaussian-smoothed image $L(x,y,\sigma)$ at the keypoint's scale $\sigma$ is taken so that all computations are performed in a scale-invariant manner. For an image sample $L(x,y)$ at scale $\sigma$, the gradient magnitude, $m(x,y)$, and orientation, $\theta(x,y)$, are precomputed using pixel differences

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

$$\theta(x,y)=\operatorname{atan2}(L(x,y+1)-L(x,y-1),L(x+1,y)-L(x-1,y))$$

The magnitude and direction calculations for the gradient are done for every pixel in a neighboring region around the keypoint in the Gaussian-blurred image L. An orientation histogram with 36 bins is formed, with each bin covering 10 degrees.

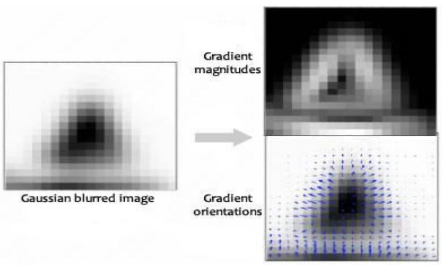


**Fig. 7: Assigning an Orientation to the Key points**

Each sample in the neighboring window added to a histogram bin is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a $\sigma$ that is 1.5 times that of the scale of the keypoint. The peaks in this histogram correspond to dominant orientations.

Each sample in the neighboring window added to a histogram bin is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a $\sigma$ that is 1.5 times that of the scale of the keypoint. The peaks in this histogram correspond to dominant orientations.

### F Generate Sift Features

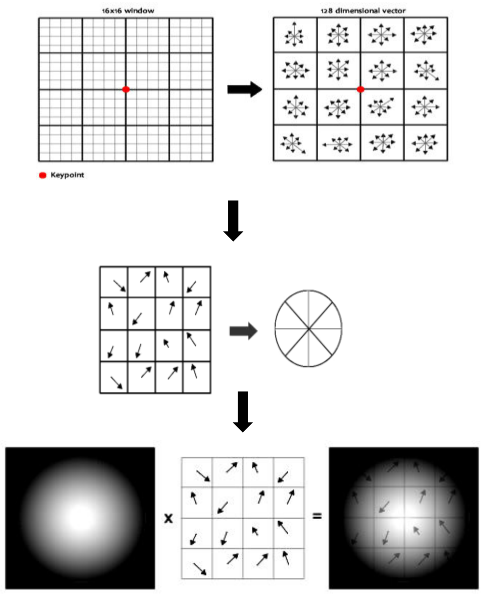


**Fig. 8: Generating Sift Features**

Take a 16×16 window of "in-between" pixels around the keypoint. Split that window into sixteen 4x4 windows. From each 4x4 window you generate a histogram of 8 bins. Each bin corresponding to 0-44 degrees, 45-89 degrees, etc. gradient orientations from the 4x4 are put into these bins. This is done for all 4x4 blocks. Finally, normalize the 128 values you get. To solve a few problems, you subtract the keypoint's orientation and also threshold the value of each element of the feature vector to 0.2 (and normalize again).

## IV. SIMULATION RESULT

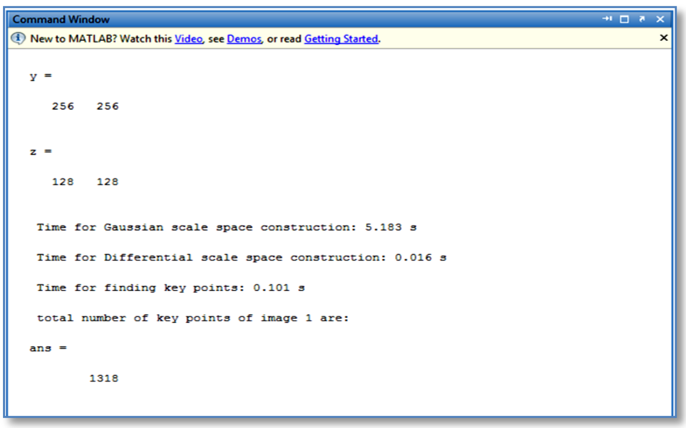


**Fig. 9: Applying sift for database image**

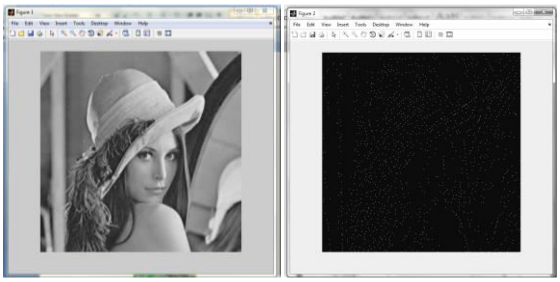

**Fig. 10: Keypoints of database image**

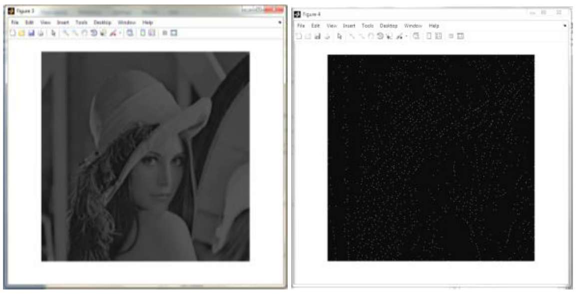

**Fig. 11: Keypoints of Captured image**

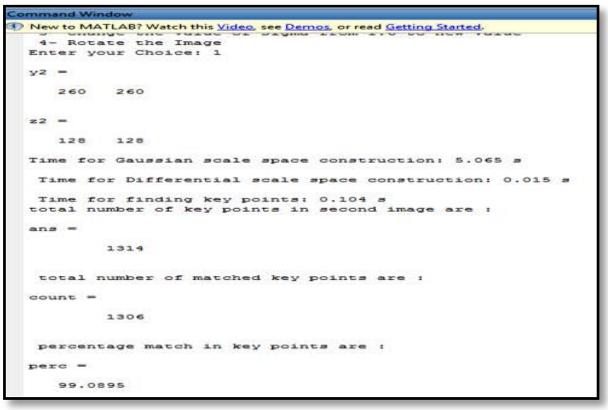


**Fig. 12:Percentage of matches of images**

## V. CONCLUTION & FUTURE WORK

This system will provide a highly authenticate information to the visually impaired. All the images will be stored in the system for the purpose of the matching of the captured images. Still more development have been developing new technology for the visually impaired and blind peoples. In this algorithm I have used only three octaves to get a better keypoints for the perfect matching.

It is a boon for the visually impaired peoples. Apart from this we have to enhance our project in the way to make the thing user friendly. Fully automated navigation system for blind peoples should be imposed. By improving the size of the database it is easy to store many more images so this will help blind people more. More the database improves the accuracy.

## VI. REFERENCE


[1] Shradha Gupta, Neetesh Gupta & Shiv Kumar "*Evaluation of Object Based Video Retrieval using SIFT*" International Journal of Soft Computing and engineering (IJSCE) may 2011.

[2] P. Geetha, Vasumathi Narayanan, "*A Survey of Content-Based Video Retrieval*" Journal of Computer Science 4 (6): 474-486, 2008 ISSN 1549-3636 © 2008 Science Publications.

[3] Arasanathan Anjulan, Nishan Canagarajah "*Object based video retrieval with local region tracking Signal Processing*" Image Communication 22 (2007) 607–621.

[4] Arasanathan Anjulan and Nishan Canagarajah "*Video Scene Retrieval Based on Local Region Features*" ICIP 2006. 1-4244-0481-9/06/2006 IEEE.

[5] Bremner D, Demaine E, Erickson J, Iacono J, Langerman S, Morin P, Toussaint G (2005*). "Output-sensitive algorithms for computing nearest-neighbor decision boundaries"*. Discrete and Computational Geometry 33 (4): 593–604.

[6] D. G. Lowe. *"Distinctive image features from scale-invariant keypoints"* Int. J. Comput. Vision, 60(2):91-110, 2004.

[7] John Eakins & Margaret Graham "*Content-based Image Retrieval*" JISC Technology Applications Programme October 1999.

[8] Lili Nurliyana Abdullah, Shahrul Azman Mohd Noah & Tengku Mohd Tengku Sembok, "*Exploring Video Information: Contents and Architecture*".

[9] Vaclav Hlavac , "*Image Segmentation*" http://cmp.felk.cvut.cz.

[10] http://en.wikipedia.org/wiki/Film_frame.

[11] Rahul Choudhury, EE 368 Project Report Recognizing pictures at an exhibition using SIFT.