FEATURES EXTRACTION IN CONTEXT BASED IMAGE RETRIEVAL

A project report submitted by

ARAVINDH A S

J CHRISTY SAMUEL

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

under the supervision of

Mrs. M. A. P. MANIMEKALAI, M.E.

SCHOOL OF ELECTRICAL SCIENCES
KARUNYA UNIVERSITY
(Karunya Institute of Technology and Sciences)
(Declared as Deemed to be University under Sec-3 of the UGC Act, 1956)
Karunya Nagar, Coimbatore - 641 114, Tamilnadu, India
APRIL 2015

BONA FIDE CERTIFICATE

Certified that this project report "Features Extraction in Context Based Image Retrieval" is the bona fide work of ARAVINDH A S (UR11EC014), and CHRISTY SAMUEL J. (UR11EC026) who carried out the project work under my supervision during the academic year 2014-2015.

Mrs. M. A. P. MANIMEKALAI, M.E
SUPERVISOR
Assistant Professor
Department of ECE
School of Electrical Sciences

Dr. SHOBHA REKH, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Professor
Department of ECE
School of Electrical Sciences

ACKNOWLEDGEMENT

First and foremost, we praise and thank **ALMIGHTY GOD** whose blessings have bestowed in us the will power and confidence to carry out our project. We are grateful to our most respected Founder (late) **Dr. D.G.S. DHINAKARAN**, **C.A.I.I.B, Ph.D.,** and Honourable Chancellor **Dr. PAUL DHINAKARAN**, **M.B.A., Ph.D.** for their grace and blessing.

We extend our thanks to our Honourable Vice-chancellor **Dr. S. SUNDAR MANOHARAN, Ph.D.** and our respected Registrar **Dr. C. JOSEPH KENNEDY, Ph.D.,** for the opportunity given to us to take up the project.

We extend our thanks to **Dr. SHOBHA REKH, M.E., Ph.D., Professor and Head of the Department of ECE** for her encouragement and guidance. We are indebted to our supervisor **Mrs. M.A.P.MANIMEKALAI, M.E. Assistant Professor**, for her valuable support, advice and encouragement.

We would like to thank the project coordinators for their support in all the project reviews. We also thank all the staff members of our department for extending their helping hands to make this project a success.

We would also like to thank all our friends and beloved parents who prayed and helped us during the project work.

Aravindh A.S.

J. Christy Samuel

ABSTRACT

There has been a growing interest in exploiting contextual information in addition to local features to detect and localize multiple object categories in an image. A context model can rule out some unlikely combinations or locations of objects and guide detectors to produce a semantically coherent interpretation of a scene. Our model incorporates global image features, dependencies between object categories, and outputs of local detectors into one probabilistic framework. We demonstrate that our context model improves object recognition performance and provides a coherent interpretation of a scene, which enables a reliable image querying system by multiple object categories. In addition, our model can be applied to scene understanding tasks that local detectors alone cannot solve, such as detecting objects out of context or querying for the most typical and the least typical scenes in a data set.

Key words: image retrieval, context-based extraction, querying system, localization.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS/ ACRONYMS

SURF - SPEEDED UP ROBUST FEATURE

SIFT - SCALE INVARIENT FEATURE TRANSFORM

CHAPTER-1

INTRODUCTION

1.1 Edge Detection:

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in 1D signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to:

i) Discontinuities in depth
ii) Discontinuities in surface orientation,
iii) Changes in material properties and
iv) Variations in scene illumination.

1.2 Objective:

In the ideal case, the result of applying an edge detector to an image may lead to a set of connected curves that indicate the boundaries of objects, the boundaries of surface markings as well as curves that correspond to discontinuities in surface orientation. Thus, applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified. However, it is not always possible to obtain such ideal edges from real life images of moderate complexity.

Edges extracted from non-trivial images are often hampered by fragmentation, meaning that the edge curves are not connected, missing edge segments as well as false edges not corresponding to interesting phenomena in the image – thus complicating the subsequent task of interpreting the image data.

Edge detection is one of the fundamental steps in image processing, image analysis, image pattern recognition, and computer vision techniques.

1.3 Aim and Objective of the Research:

The aim of the project is:

i) To know the working of the SIFT and SURF

ii) Image Properties using SIFT and SURF

iii) To the difference of these both image features

CHAPTER 2

LITERATURE SURVEY

We develop an efficient framework to exploit contextual information in object recognition and scene understanding problems by modeling object dependencies, global image features, and local detector outputs using a tree-based graphical model. Our context model enables a parsimonious modeling of object dependencies, and can easily scale to capture the dependencies of over 100 object categories. The SUN 09 data set presented in this paper has richer contextual information than PASCAL 07, and is more suitable for training and evaluating context models. We demonstrate that our context model learned from SUN 09 significantly improves the accuracy of object recognition and image query results, and can be applied to find objects out of context (The SUN 09 data set and the Matlab implementation of our algorithm). We conclude by discussing some possible extensions of the work presented in this paper. Our location model captures spatial relationships of object categories using Gaussian distributions. While this greatly reduces computational complexity, it does not capture some physical relationships such as a car is supported by a road. In addition, the location model can be improved by encoding different types of interactions or poses among object instances (e.g., person 1 is riding a horse and person 2 is standing next to it), or spatial relationships based on different viewpoints . The tree structure shown in Fig. 6 captures the inherent hierarchy among object categories. For example, most of the objects that commonly appear in a kitchen are descendants of the node sink, and all the vehicles are descendants of road. This suggests that a more intuitive structure for object dependencies could be a hierarchy including some metaobjects (such as a desk area) or scenes (kitchen or street) as nodes at coarser scales. Learning a full hierarchical tree structure with such additional nodes may discover important relationships among objects, metaobjects, and scenes, which is an interesting direction for further research.

A simple form of contextual information is a co-occurrence frequency of a pair of objects. Rabinovich et al. use local detectors to first assign an object label to each image segment, and then adjusts these labels using a conditional random field (CRF). This approach is extended in and to encode spatial relationships between a pair of objects. In spatial relationships are quantized to four prototypical relationships—above, below, inside, and around, whereas in a nonparametric map of spatial priors is learned for each pair of objects. Torralba et al.combine boosting and CRFs to first detect easy objects (e.g., a monitor) and pass the contextual information to detect other more difficult objects (e.g., a keyboard). Tu uses both image patches and their probability maps estimated from classifiers to learn a contextual model, and iteratively refines the classification results by propagating the contextual information. Desai et al.combine individual classifiers by using spatial interactions between object detections in a discriminative manner. Contextual information may be obtained from coarser, global features as well. Torralba demonstrates that a global image feature called "gist" can predict the presence or absence of objects and their locations without running an object detector. This is extended in to combine patch-based local features and the gist feature. Heitz and Koller combine a sliding window method and unsupervised image region clustering to leverage "stuff"such as the sea, the sky, or a road to improve object detection. A cascaded classification model in links scene categorization, multiclass image segmentation, object detection, and 3D reconstruction.

CHAPTER 3

DESIGN METHEODOLOGY

3.1 Description

Image registration is critical task in many applications. To perform image registration/alignment, required steps are: Feature detection, Feature matching, derivation of transformation function based on corresponding features in images and reconstruction of images based on derived transformation function. Accuracy of registered image depends on accurate feature detection and matching. So these two intermediate steps are very important in many image applications: image registration, computer vision, image mosaic etc. This paper presents two different methods for scale and rotation invariant interest point/feature detector and descriptor: Scale Invariant Feature Transform (SIFT) and Speed Up Robust Features (SURF). It also presents a way to extract distinctive invariant features from images that can be used to perform reliable matching between different views of an object/scene

3.2 Process Stages

3.2.1 Types of Edge Detection and Feature Extraction

1. Sobel Edge Detector
2. Canny Edge Detector
3. Prewitt Edge Detector
4. SURF
5. SIFT

3.2.2 Sobel Edge Detector

The Sobel operator, sometimes called Sobel Filter, is used in image processing and computer vision, particularly within edge detection algorithms, and creates an image which emphasizes edges and transitions. It is named after Irwin Sobel, who presented the idea of an "Isotropic 3x3 Image Gradient Operator" at a talk at the Stanford Artificial Intelligence Project (SAIP) in 1968. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high frequency variations in the image. The Kayyali operator for edge detection is another operator generated from Sobel operator.

The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define A as the source image, and Gx and Gy are two images which at each point contain the horizontal and vertical derivative approximations, the computations are as follows:

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A \ \ and \ \ G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \qquad ----\rightarrow 1$$

where * here denotes the 2-dimensional convolution operation.

Since the Sobel kernels can be decomposed as the products of an averaging and a differentiation kernel, they compute the gradient with smoothing. For example, can be written as

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & +1 \end{bmatrix} \qquad ----\rightarrow 2$$

The x-coordinate is defined here as increasing in the "right"-direction, and the y-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

$$G = \sqrt{G_x^2 + G_y^2} \qquad ----\rightarrow 3$$

Using this information, we can also calculate the gradient's direction:

$$\theta = atan2(G_{y,} G_x) \qquad ----\rightarrow 4$$

where, for example, θ is 0 for a vertical edge which is lighter on the right side.

As a consequence of its definition, the Sobel operator can be implemented by simple means in both hardware and software: only eight image points around a point are needed to compute the corresponding result and only integer arithmetic is needed to compute the gradient vector approximation. Furthermore, the two discrete filters described above are both separable:

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \end{bmatrix} \quad ----\rightarrow 5$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \end{bmatrix} \quad ----\rightarrow 6$$

$$G_x = \begin{bmatrix}1\\2\\1\end{bmatrix} * ([-1 \quad 0 \quad -1] * A) \ \ and \ \ G_y = \begin{bmatrix}1\\0\\-1\end{bmatrix} * ([1 \quad 2 \quad 1] * A) \quad ----\rightarrow 7$$

Since the intensity function of a digital image is only known at discrete points, derivatives of this function cannot be defined unless we assume that there is an underlying continuous intensity function which has been sampled at the image points. With some additional assumptions, the derivative of the continuous intensity function can be computed as a function on the sampled intensity function, i.e. the digital image. It turns out that the derivatives at any particular point are functions of the intensity values at virtually all image points. However, approximations of these derivative functions can be defined at lesser or larger degrees of accuracy.

The Sobel operator represents a rather inaccurate approximation of the image gradient, but is still of sufficient quality to be of practical use in many applications. More precisely, it uses intensity values only in a 3×3 region around each image point to approximate the corresponding image gradient, and it uses only integer values for the coefficients which weight the image intensities to produce the gradient approximation.

3.2.3 Canny Edge Detector

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works

Edge detection, especially step edge detection has been widely applied in various different computer vision systems, which is an important technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. Canny has found that, the requirements for the application of edge detection on diverse vision systems are relatively the same. Thus, a development of an edge detection solution to address these requirements can be implemented in a wide range of situations. The general criteria for edge detection includes

1. Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible
2. The edge point detected from the operator should accurately localize on the center of the edge.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

To satisfy these requirements Canny used the calculus of variations – a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

Among the edge detection methods developed so far, canny edge detection algorithm is one of the most strictly defined methods that provides good and reliable detection. Owing to its optimality to meet with the three criteria for edge detection and the simplicity of process for implementation, it becomes one of the most popular algorithms for edge detection.

Process Edge Detection Algorithm of Canny:

The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise

2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

Every step will be described in details as following. The introduction of procedure below is developed based on Prof Thomas Moeslund's lecture note for digital image processing in Indian Institute of Technology.

Non-maximum suppression is an edge thinning technique.

Non-Maximum suppression is applied to "thin" the edge. After applying gradient calculation, the edge extracted from the gradient value is still quite blurred. With respect to criteria 3, there should only be one accurate response to the edge. Thus non-maximum suppression can help to suppress all the gradient values to 0 except the local maximal, which indicates location with the sharpest change of intensity value. The algorithm for each pixel in the gradient image is:

1. Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions.

2. If the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction(i.e, the pixel that is pointing in the y direction, it will be compared to the pixel above and below it in the vertical axis), the value will be preserved. Otherwise, the value will be suppressed.

Double Threshold

After application of non-maximum suppression, the edge pixels are quite accurate to present the real edge. However, there are still some edge pixels at this point caused by noise and color variation. In order to get rid of the spurious responses from these bothering factors, it is essential to filter out the edge pixel with the weak gradient value and preserve the edge with the high gradient value. Thus two threshold values are set to clarify the different types of edge pixels, one is called high threshold value and the other is called the low threshold value. If the edge pixel's gradient value is higher than the high threshold value, they are marked as strong edge pixels. If the edge pixel's gradient value is smaller than the high threshold value and larger than the low threshold value, they are marked as weak edge pixels. If the pixel value is smaller than the low threshold value, they will be suppressed. The two threshold values are empirically determined values, which will need to be defined when applying to different images.

Edge Tracking by Hysteresis

So far, the strong edge pixels should certainly be involved in the final edge image, as they are extracted from the true edges in the image. However, there will be some debate on the weak image pixels, as these pixels can either be extracted from the true edge, or the noise/color variations. To achieve an accurate result, the weak edges caused from the latter reasons should be removed. The criteria to determine which case does the weak edge belongs to, is that usually the weak edge pixel caused from true edges will be connected to the strong edge pixel. To track the edge connection, Binary Large Object-analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels. As long as there is one strong edge pixel is involved in the BLOB, that weak edge point can be identified as one that should be preserved

Improvement on Canny Edge Detection:

While traditional canny edge detection provides relatively simple but precise methodology for edge detection problem, with the more demanding requirements on the accuracy and robustness on the detection, the traditional algorithm can no longer handle the challenging edge detection task. The main defects of the traditional algorithm can be summarized as following:

1. Gaussian filter is applied to smooth out the noise, but it will also smooth the edge, which is considered as the high frequency feature. This will increase the possibility to miss weak edges, and the appearance of isolated edges in the result.

2. For the gradient amplitude calculation, the old canny edge detection algorithm uses center in a small 2*2 neighborhoods window to calculate the finite difference mean value to represent the gradient amplitude. This method is sensitive to noise and can easily detect fake edges and lose real edges.

3. In traditional canny edge detection algorithm, there will be two fixed global threshold values to filter out the false edges. However, as the image gets complex, different local areas will need very different threshold values to accurately find the real edges. In addition, the global threshold values are determined manually through experiments in the traditional method, which leads to complexity of calculation when large number of different images needs to be dealt with.

4. The result of the traditional detection cannot reach a satisfactory high accuracy of single response for each edge- multi-point responses will appear.

Parameters:

The Canny algorithm contains a number of adjustable parameters, which can affect the computation time and effectiveness of the algorithm.

1. The size of the Gaussian filter: the smoothing filter used in the first stage directly affects the results of the Canny algorithm. Smaller filters cause less blurring, and allow

detection of small, sharp lines. A larger filter causes more blurring, smearing out the value of a given pixel over a larger area of the image. Larger blurring radii are more useful for detecting larger, smoother edges – for instance, the edge of a rainbow.

2. Thresholds: the use of two thresholds with hysteresis allows more flexibility than in a single-threshold approach, but general problems of thresholding approaches still apply. A threshold set too high can miss important information. On the other hand, a threshold set too low will falsely identify irrelevant information (such as noise) as important. It is difficult to give a generic threshold that works well on all images. No tried and tested approach to this problem yet exists.

The Canny algorithm is adaptable to various environments. Its parameters allow it to be tailored to recognition of edges of differing characteristics depending on the particular requirements of a given implementation. In Canny's original paper, the derivation of the optimal filter led to a Finite Impulse Response filter, which can be slow to compute in the spatial domain if the amount of smoothing required is important (the filter will have a large spatial support in that case). For this reason, it is often suggested to use Rachid Deriche's infinite impulse response form of Canny's filter (the Canny–Deriche detector), which is recursive, and which can be computed in a short, fixed amount of time for any desired amount of smoothing. The second form is suitable for real time implementations in FPGAs or DSPs, or very fast embedded PCs. In this context, however, the regular recursive implementation of the Canny operator does not give a good approximation of rotational symmetry and therefore gives a bias towards horizontal and vertical edges.

3.2.4 Prewitt Edge Detector

The Prewitt operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Prewitt operator is either the corresponding gradient vector or the norm of this vector. The Prewitt operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical directions and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image. The Prewitt operator was developed by Judith M. S. Prewitt.

Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations, the latter are computed as:

$$G_X = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \ \ and\ \ G_Y = \begin{bmatrix} -1 & -1 & +1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A \qquad ----\rightarrow 8$$

where * here denotes the 2-dimensional convolution operation.

Since the Prewitt kernels can be decomposed as the products of an averaging and a differentiation kernel, they compute the gradient with smoothing. Therefore it is a separable filter. For example, G_x can be written as

$$\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \qquad ----\rightarrow 9$$

The x-coordinate is defined here as increasing in the "right"-direction, and the y-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

$$G = \sqrt{G_x^2 + G_y^2} \qquad \text{-----}\!\!\rightarrow 10$$

Using this information, we can also calculate the gradient's direction:

$$\theta = atan2(G_{y,} G_x) \qquad \text{-----}\!\!\rightarrow 11$$

where, for example, Θ is 0 for a vertical edge which is darker on the right side.

Prewitt operator is used for edge detection in an image. It detects two types of edges:

1. Horizontal edges

2.VerticalEdges

Edges are calculated by using difference between corresponding pixel intensities of an image. All the masks that are used for edge detection are also known as derivative masks. Because as we have stated many times before in this series of tutorials that image is also a signal so changes in a signal can only be calculated using differentiation. So that's why these operators are also called as derivative operators or derivative masks.

All the derivative masks should have the following properties

1. Opposite sign should be present in the mask.
2. Sum of mask should be equal to zero.
3. More weight means more edge detection.

This mask will prominent the horizontal edges in an image. It also works on the principle of above mask and calculates difference among the pixel intensities of a particular edge. As the center row of mask is consist of zeros so it does not include the original values of edge in the image but rather it calculate the difference of above and below pixel intensities of the particular edge. Thus increasing the sudden change of intensities and making the edge more visible. Both the above masks follow the principle of derivate mask. Both masks have opposite sign in them and both masks sum equals to zero. The third condition will not be applicable in this operator as both the above masks are standardize and we can't change the value in them.

3.2.5 SURF

SURF is a detector and a high-performance descriptor points of interest in an image where the image is transformed into coordinates, using a technique called multi-resolution. Is to make a copy of the original image with Pyramidal Gaussian or Laplacian Pyramid shape and obtain image with the same size but with reduced bandwidth. Thus a special blurring effect on the original image, called Scale-Space is achieved. This technique ensures that the points of interest are scale invariant. The SURF algorithm is based on the SIFT predecessor.

Scale-space representation & location of points of interest

The attractions can be found in different scales, partly because the search for correspondences often requires comparison images where they are seen at different scales. The scale spaces are generally applied as a pyramid image . Images are repeatedly smoothed with a Gaussian filter, then, is sub sampled to achieve a higher level of the pyramid. Therefore, several floors or stairs " det H" with various measures of the masks are calculated.

The scale -space is divided into a number of octaves, Where an octave refers to a series of response maps of covering a doubling of scale . In SURF The Lowest level of the Scale- space is Obtained from the output of the 9×9 filters.

Scale spaces are implemented by applying box filters of different size. Therefore, the scale space is analyzed by up-scaling the filter size rather than iteratively reducing the image size. The output of the above 9*9 filter is considered as the initial scale layer, to which we will refer as scale s=1.2 (corresponding to Gaussian derivatives with σ=1.2). The following layers are obtained by filtering the image with gradually bigger masks, taking into account the discrete nature of integral images and the specific structure of or filters. Specifically, this results in filters of size 9*9, 15*15, 21*21, 27*27, etc. In order to localize interest points in the image and over scales, non-maximum suppression in a 3*3*3 neighborhood is applied. The maxima of the determinant of the Hessian matrix are then interpolated in scale and image space with the method proposed by Brown et al. Scale space interpolation is especially important in our case, as the difference in scale between the first layers of every octave is relatively large.

After 3D maxima are looking at (x, y, n) using the cube 3x3x3 neighborhood . From there it is proceed to do the interpolation of the maximum. Lowe rest of the layers of the pyramid to get the DOG (Difference of Gaussian) find images contours and stains.

Specifically, it is entered by variant a quick and Van Gool Neubecker used. The maximum of the determinant of the Hessian matrix in scale and space interpolated image with Brown and Lowe proposed method. The approach of the determinant of the Hessian matrix represents the response of BLOB in the image to the location x. These responses are stored in the BLOB map of responses on different scales. They have the principal feature of repetibility, That means if some point is considered realiable, the detector will find the same point under different perspective (scale, orientation, rotation, etc.).

It has one position (x,y) for each interest point.

Description:

The goal of a descriptor is to provide a unique and robust description of an image feature, e.g. by describing the intensity distribution of the pixels within the neighborhood of the point of interest. Most descriptors are computed thus in a local manner; hence, a description is obtained for every point of interest identified previously.

The dimensionality of the descriptor has direct impact on both its computational complexity and point-matching robustness/accuracy. A short descriptor may be more robust against appearance variations, but may not offer sufficient discrimination and thus give too many false positives.

The next three introducing steps are explained in the following: The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. Then, we construct a square region aligned to the selected orientation and extract the SURF descriptor from it. Finally, features are matched between two images.

The SURF descriptor is based on the similar properties of SIFT, with a complexity stripped down even further. The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. The second step is constructing as square region aligned to the selected orientation, and extracting the SURF descriptor from it. These two steps are now explained in turn. Furthermore, we also propose an upright version of our descriptor (U-SURF) that is not invariant to image rotation and therefore faster to compute and better suited for application where the camera remains more or less horizontal.

Matching

This section details the step back in search of characteristic points that provides the detector. This way it is possible to compare between descriptors and look for matching pairs of images between them. There are two ways to do it:

1. Get the characteristic points of the first image and its descriptor and do the same with the second image . So you will be able to compare the two images descriptor correspondences between points and establish some kind of measure.

2. Get the characteristic points of the first image with the descriptor. Then compare this descriptor with the points of the second image which is believed to be the partner concerned.

3.2.6 SIFT:

Scale-invariant feature transform (or SIFT) is an algorithm in computer vision to detect and describe local features in images. Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving.

For any object in an image, interesting points on the object can be extracted to provide a "feature description" of the object. This description, extracted from a training image, can then be used to identify the object when attempting to locate the object in a test image containing many other objects. To perform reliable recognition, it is important that the features extracted from the training image be detectable even under changes in image scale, noise and illumination. Such points usually lie on high-contrast regions of the image, such as object edges.

Another important characteristic of these features is that the relative positions between them in the original scene shouldn't change from one image to another. For example, if only the four corners of a door were used as features, they would work regardless of the door's position; but if points in the frame were also used, the recognition would fail if the door is opened or closed. Similarly, features located in articulated or flexible objects would typically not work if any change in their internal geometry happens between two images in the set being processed. However, in practice SIFT detects and uses a much larger number of features from the images, which reduces the contribution of the errors caused by these local variations in the average error of all feature matching errors.

SIFT can robustly identify objects even among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to uniform scaling, orientation, and partially invariant to affine distortion and illumination changes. This section summarizes Lowe's object recognition method and mentions a few competing techniques available for object recognition under clutter and partial occlusion

SIFT key points of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of keypoints that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

Features:

The detection and description of local image features can help in object recognition. The SIFT features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint. In addition to these properties, they are highly distinctive, relatively easy to extract and allow for correct object identification with low probability of mismatch. They are relatively easy to match against a (large) database of local features but however the high dimensionality can be an issue, and generally probabilistic algorithms such as k-d trees with best bin first search are used. Object description by set of SIFT features is also robust to partial occlusion; as few as 3 SIFT features from an object are enough to compute its location and pose. Recognition can be performed in close-to-real time, at least for small databases and on modern computer hardware

Scale-invariant feature detection:

Lowe's method for image feature generation transforms an image into a large collection of feature vectors, each of which is invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion. These features share similar properties with neurons in inferior temporal cortex that are used for object recognition in primate vision. Key locations are defined as maxima and minima of the result of difference of Gaussians function applied in scale space to a series of smoothed and resampled images. Low contrast candidate points and edge response points along an edge are discarded. Dominant orientations are assigned to localized keypoints. These steps ensure that the keypoints are more stable for matching and recognition. SIFT descriptors robust to local affine distortion are then obtained by considering pixels around a radius of the key location, blurring and resampling of local image orientation planes

Feature matching and indexing:

Indexing consists of storing SIFT keys and identifying matching keys from the new image. Lowe used a modification of the k-d tree algorithm called the Best-bin-first search method that can identify the nearest neighbors with high probability using only a limited amount of computation. The BBF algorithm uses a modified search ordering for the k-d tree algorithm so that bins in feature space are searched in the order of their closest distance from the query location. This search order requires the use of a heap-based priority queue for efficient determination of the search order. The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of keypoints from training images. The nearest neighbors are defined as the keypoints with minimum Euclidean distance from the given descriptor vector. The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest.

Lower rejected all matches in which the distance ratio is greater than 0.8, which eliminates 90% of the false matches while discarding less than 5% of the correct matches. To further improve the efficiency of the best-bin-first algorithm search was cut off after checking the first 200 nearest neighbor candidates. For a database of 100,000 keypoints, this provides a speedup over exact nearest neighbor search by about 2 orders of magnitude, yet results in less than a 5% loss in the number of correct matches.

Cluster identification by Hough transform voting:

Hough Transform is used to cluster reliable model hypotheses to search for keys that agree upon a particular model pose. Hough transform identifies clusters of features with a consistent interpretation by using each feature to vote for all object poses that are consistent with the feature. When clusters of features are found to vote for the same pose of an object, the probability of the interpretation being correct is much higher than for any single feature. An entry in a hash table is created predicting the model location, orientation, and scale from the match hypothesis.The hash table is searched to identify all clusters of at least 3 entries in a bin, and the bins are sorted into decreasing order of size.

Each of the SIFT keypoints specifies 2D location, scale, and orientation, and each matched keypoint in the database has a record of its parameters relative to the training image in which it was found. The similarity transform implied by these 4 parameters is only an approximation to the full 6 degree-of-freedom pose space for a 3D object and also does not account for any non-rigid deformations. Therefore, Lower used broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times the maximum projected training image dimension (using the predicted scale) for location. The SIFT key samples generated at the larger scale are given twice the weight of those at the smaller scale. This means that the larger scale is in effect able to filter the most likely neighbours for checking at the smaller scale. This also improves recognition performance by giving more weight to the least-noisy scale. To avoid the problem of boundary effects in bin assignment, each keypoint match votes for the 2 closest bins in each dimension, giving a total of 16 entries for each hypothesis and further broadening the pose range.

Model verification by linear least squares:

Each identified cluster is then subject to a verification procedure in which a linear least squares solution is performed for the parameters of the affine transformation relating the model to the image. The affine transformation of a model point [x y]_T to an image point [u v]_T can be written as below

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m1 & m2 \\ m3 & m4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix} \qquad ----\rightarrow 12$$

where the model translation is [tx ty]T and the affine rotation, scale, and stretch are represented by the parameters m1, m2, m3 and m4. To solve for the transformation parameters the equation above can be rewritten to gather the unknowns into a column vector.

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ . & & & & & \\ . & & & & & \\ . & & & & & \end{bmatrix} \begin{bmatrix} m1 \\ m2 \\ m3 \\ m4 \\ tx \\ ty \end{bmatrix} = \begin{bmatrix} u \\ v \\ . \\ . \end{bmatrix} \qquad -----13$$

This equation shows a single match, but any number of further matches can be added, with each match contributing two more rows to the first and last matrix. At least 3 matches are needed to provide a solution. We can write this linear system as

$$A\hat{x} \approx b, \qquad ----\rightarrow 14$$

where A is a known m-by-n matrix (usually with m > n), x is an unknown n-dimensional parameter vector, and b is a known m-dimensional measurement vector.

Therefore the minimizing vector is a solution of the normal equation

$$A^T A\hat{x} = A^T b \qquad ----\rightarrow 15$$

The solution of the system of linear equations is given in terms of the matrix, called the pseudoinverse of A, by

$$\hat{x} = (A^T A)^{-1} A^T b - - - - \rightarrow 16$$

Which minimizes the sum of the squares of the distances from the projected model locations to the corresponding image locations.

Outlier detection:

Outliers can now be removed by checking for agreement between each image feature and the model, given the parameter solution. Given the linear least squares solution, each match is required to agree within half the error range that was used for the parameters in the Hough transform bins. As outliers are discarded, the linear least squares solution is re-solved with the remaining points, and the process iterated. If fewer than 3 points remain after discarding outliers, then the match is rejected. In addition, a top-down matching phase is used to add any further matches that agree with the projected model position, which may have been missed from the Hough transform bin due to the similarity transform approximation or other errors.

The final decision to accept or reject a model hypothesis is based on a detailed probabilistic model. This method first computes the expected number of false matches to the model pose, given the projected size of the model, the number of features within the region, and the accuracy of the fit. A Bayesian probability analysis then gives the probability that the object is present based on the actual number of matching features found. A model is accepted if the final probability for a correct interpretation is greater than 0.98. Lowe's SIFT based object recognition gives excellent results except under wide illumination variations and under non-rigid transformations.

3.2.7 Comparison:

There has been an extensive study done on the performance evaluation of different local descriptors, including SIFT, using a range of detectors. The main results are summarized below:

SIFT and SIFT-like GLOH features exhibit the highest matching accuracies (recall rates) for an affine transformation of 50 degrees. After this transformation limit, results start to become unreliable.

Distinctiveness of descriptors is measured by summing the eigenvalues of the descriptors, obtained by the Principal components analysis of the descriptors normalized by their variance. This corresponds to the amount of variance captured by different descriptors, therefore, to their distinctiveness. PCA-SIFT (Principal Components Analysis applied to SIFT descriptors), GLOH and SIFT features give the highest values.
SIFT-based descriptors outperform other contemporary local descriptors on both textured and structured scenes, with the difference in performance larger on the textured scene.
For scale changes in the range 2-2.5 and image rotations in the range 30 to 45 degrees, SIFT and SIFT-based descriptors again outperform other contemporary local descriptors with both textured and structured scene content.

Introduction of blur affects all local descriptors, especially those based on edges, like shape context, because edges disappear in the case of a strong blur. But GLOH, PCA-SIFT and SIFT still performed better than the others. This is also true for evaluation in the case of illumination changes.

The evaluations carried out suggests strongly that SIFT-based descriptors, which are region-based, are the most robust and distinctive, and are therefore best suited for feature matching. However, most recent feature descriptors such as SURF have not been evaluated in this study.

SURF has later been shown to have similar performance to SIFT, while at the same time being much faster. Another study concludes that when speed is not critical, SIFT outperforms SURF.

Recently, a slight variation of the descriptor employing an irregular histogram grid has been proposed that significantly improves its performance. Instead of using a 4x4 grid of histogram bins, all bins extend to the center of the feature. This improves the descriptor's robustness to scale changes.

The SIFT-Rank descriptor was shown to improve the performance of the standard SIFT descriptor for affine feature matching. A SIFT-Rank descriptor is generated from a standard SIFT descriptor, by setting each histogram bin to its rank in a sorted array of bins. The Euclidean distance between SIFT-Rank descriptors is invariant to arbitrary monotonic changes in histogram bin values, and is related to Spearman's rank correlation coefficient.

Object recognition using SIFT features:

Given SIFT's ability to find distinctive keypoints that are invariant to location, scale and rotation, and robust to affine transformations (changes in scale, rotation, shear, and position) and changes in illumination, they are usable for object recognition. The steps are given below.

First, SIFT features are obtained from the input image using the algorithm described above.

These features are matched to the SIFT feature database obtained from the training images. This feature matching is done through a Euclidean-distance based nearest neighbor approach. To increase robustness, matches are rejected for those keypoints for which the ratio of the nearest neighbor distance to the second nearest neighbor distance is greater than 0.8. This discards many of the false matches arising from background clutter. Finally, to avoid the expensive search required for finding the Euclidean-distance-based nearest neighbor, an approximate algorithm called the best-bin-first algorithm is used. This is a fast method for returning the nearest neighbor with high probability, and can give speedup by factor of 1000 while finding nearest neighbor (of interest) 95% of the time.

Although the distance ratio test described above discards many of the false matches arising from background clutter, we still have matches that belong to different objects. Therefore to increase robustness to object identification, we want to cluster those features that belong to the same object and reject the matches that are left out in the clustering process. This is done using the Hough transform. This will identify clusters of features that vote for the same object pose. When clusters of features are found to vote for the same pose of an object, the probability of the interpretation being correct is much higher than for any single feature. Each keypoint votes for the set of object poses that are consistent with the keypoint's location, scale, and orientation. Bins that accumulate at least 3 votes are identified as candidate object/pose matches.

For each candidate cluster, a least-squares solution for the best estimated affine projection parameters relating the training image to the input image is obtained. If the projection of a keypoint through these parameters lies within half the error range that was used for the parameters in the Hough transform bins, the keypoint match is kept. If fewer than 3 points remain after discarding outliers for a bin, then the object match is rejected. The least-squares fitting is repeated until no more rejections take place. This works better for planar surface recognition than 3D object recognition since the affine model is no longer accurate for 3D objects.

In this journal, authors proposed a new approach to use SIFT descriptors for multiple object detection purposes. The proposed multiple object detection approach is tested on aerial and satellite images.

SIFT features can essentially be applied to any task that requires identification of matching locations between images. Work has been done on applications such as recognition of particular object categories in 2D images, 3D reconstruction, motion tracking and segmentation, robot localization, image panorama stitching and epipolar calibration. Some of these are discussed in more detail below.

3D SIFT-like descriptors for human action recognition:

Extensions of the SIFT descriptor to 2+1-dimensional spatio-temporal data in context of human action recognition in video sequences have been studied. The computation of local position-dependent histograms in the 2D SIFT algorithm are extended from two to three dimensions to describe SIFT features in a spatio-temporal domain. For application to human action recognition in a video sequence, sampling of the training videos is carried out either at spatio-temporal interest points or at randomly determined locations, times and scales. The spatio-temporal regions around these interest points are then described using the 3D SIFT descriptor. These descriptors are then clustered to form a spatio-temporal Bag of words model. 3D SIFT descriptors extracted from the test videos are then matched against these words for human action classification.

The authors report much better results with their 3D SIFT descriptor approach than with other approaches like simple 2D SIFT descriptors and Gradient Magnitude.

CHAPTER 4

RESULT AND DISCUSSSIONS

4.1 Output of Canny Operator:

Fig.4.1 Original image

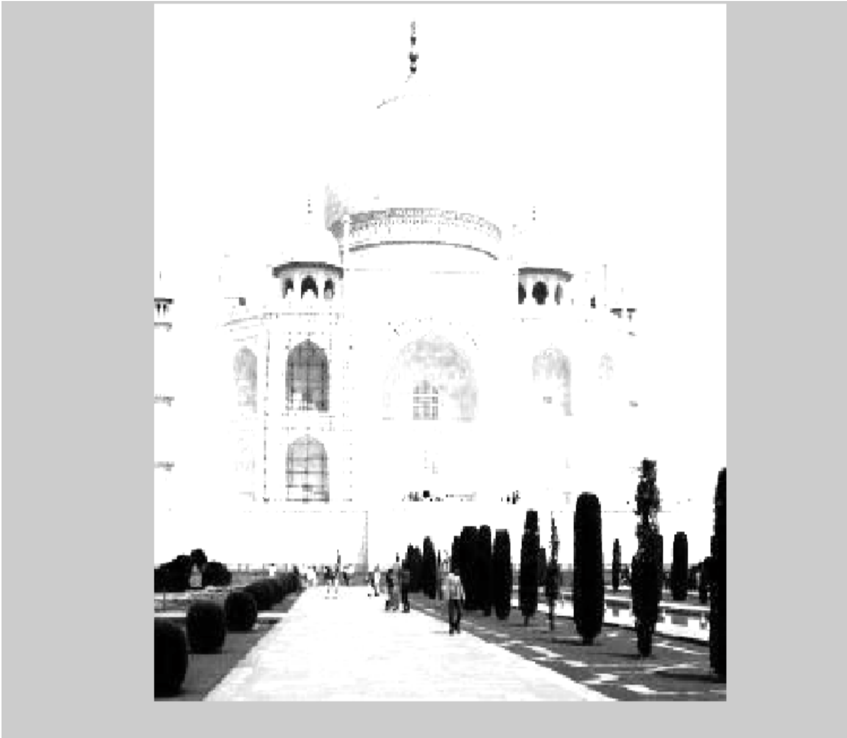

Fig.4.2 Filtered Image

Fig.4.3 Canny Edge Detector

4.2 Output of Sobel Operator:

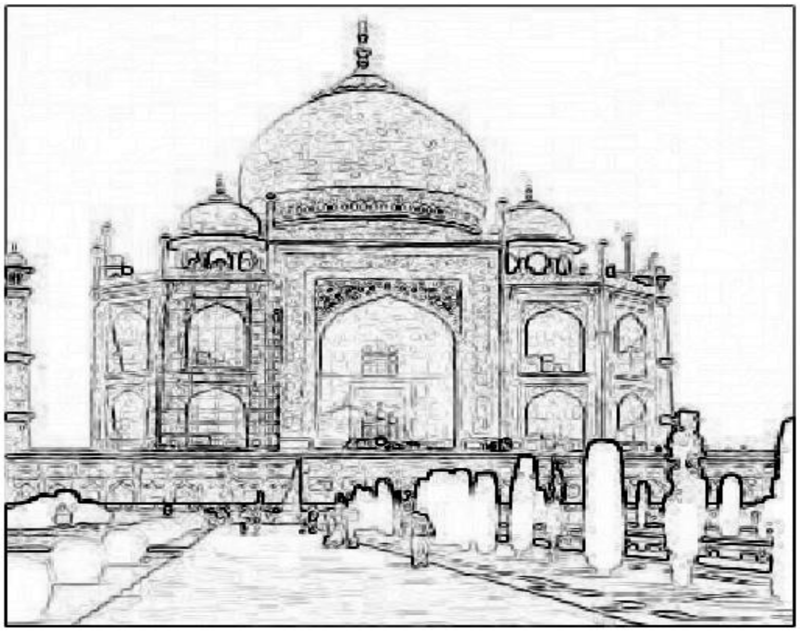

Fig.4.4Sobel Edge Detector

4.3 Output of Prewitt Operator:

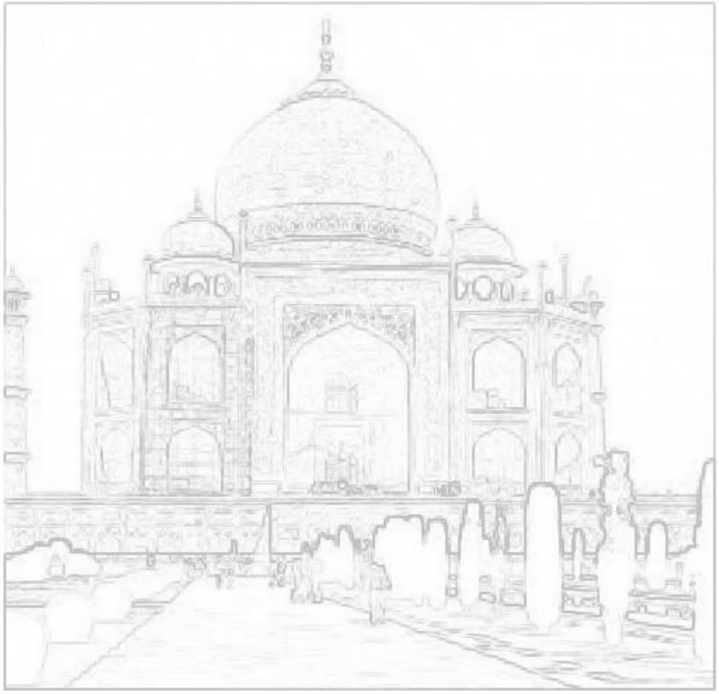

Fig.4.5Prewitt Edge Detector

4.4 Output of SIFT Edge Detection Images:

Fig.4.6 Edges

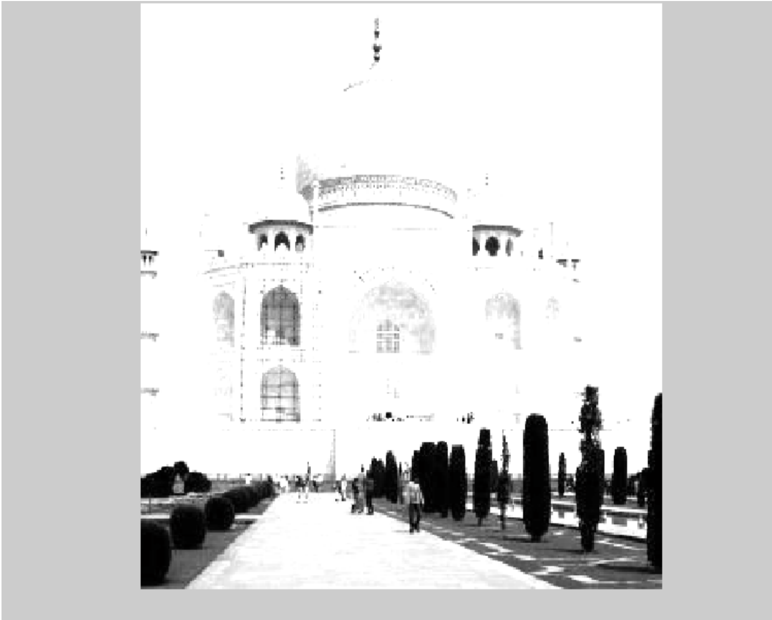

Fig.4.7 Decrease Intensity

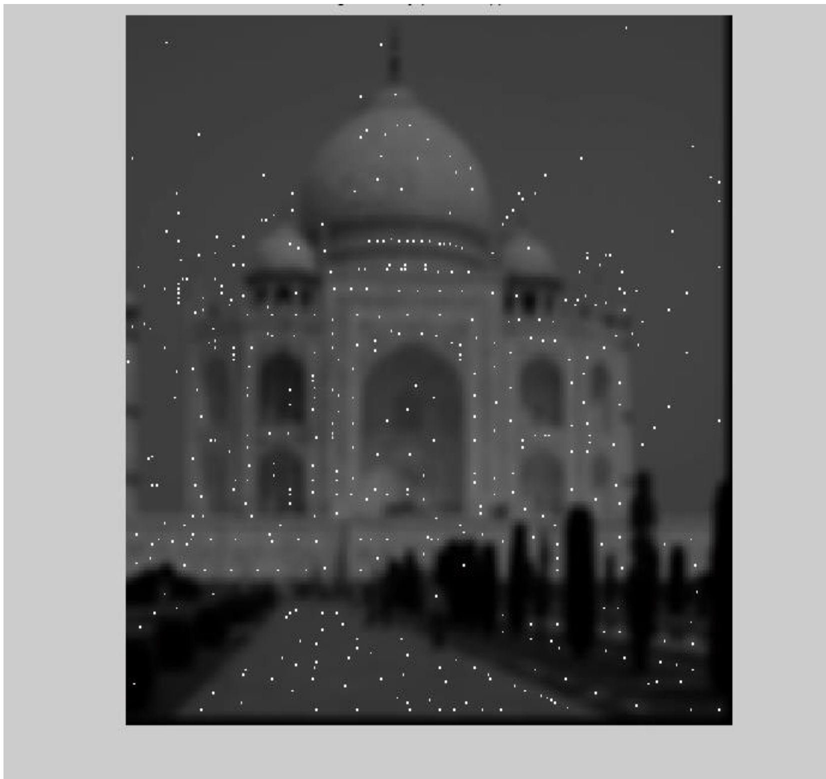

Fig.4.8 Decrease Intensity Edges

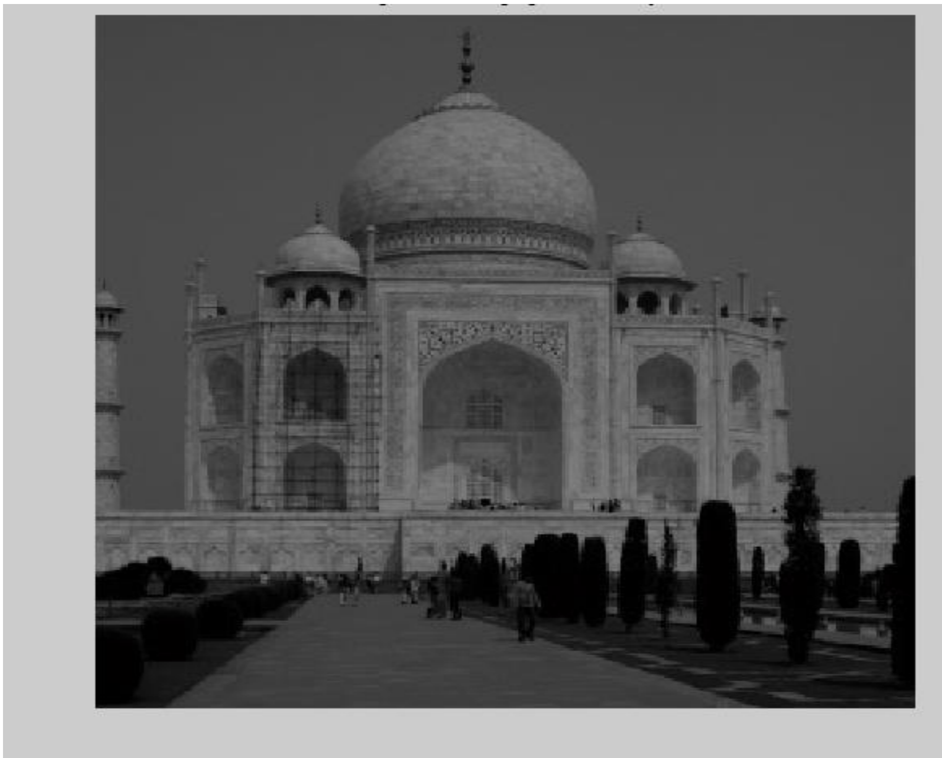

Fig.4.9 Increase Intensity

Fig.4.10 Increase Intensity Edges

Fig.4.11 Rotated Image

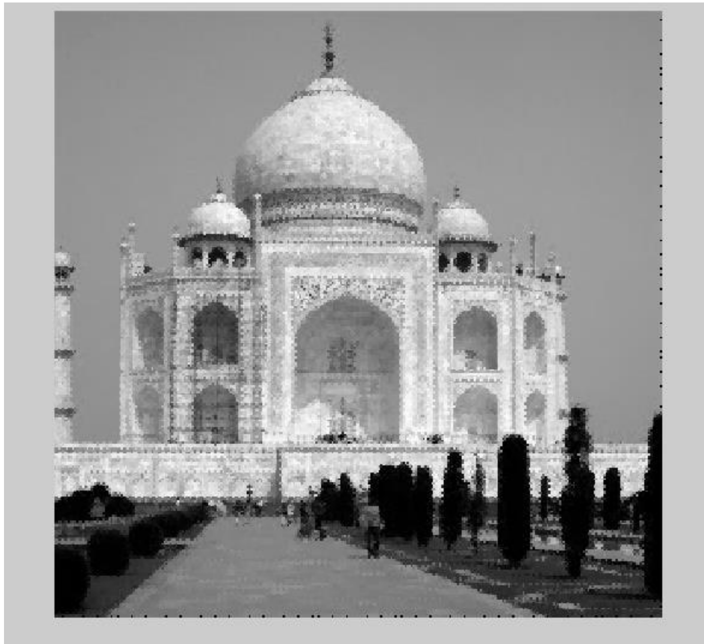

Fig.4.12 Rotate Comparison

Fig.4.13 Rotated Edges

4.5 OUTPUT OF SURF:

Fig.4.14 SURF OUTPUT

CHAPTER 5

CONCLUSION

5.1 Comparison of SURF Output and SIFT Output

Two feature detection methods for image registration. Based on the experimental results, it is found that the SIFT has detected more number of features compared to SURF but it is suffered with speed. The SURF is fast and has good performance as the same as SIFT.

Table.5.1.1 COMPARISON TABLE

ALGORITHM	MATCHING FEATURES	FEATURES MATCHING TIME
SIFT	13	1.543s
SURF	6	0.564s

REFERENCES

1. Myung Jin Choi, Antonio Torralba and Alan S. Willsky, “A Tree-Based Context Model for Object Recognition,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 34

2. N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, 2005.

3. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009.

4. S.K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros, and M. Hebert, “An Empirical Study of Context in Object Detection,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009.

5. M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A.Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” , 2011.

6. P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object Detection with Discriminatively Trained Part Based Models,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627-1645, Sept. 2010.

7. R. Fergus, P. Perona, and A. Zisserman, “Object Class Recognition by Unsupervised Scale-Invariant Learning,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, 2003.

8. C. Galleguillos, A. Rabinovich, and S. Belongie, “Object Categorization Using Co-Occurrence, Location and Appearance,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008.

9. S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, “Multi- Class Segmentation with Relative Location Prior,” Int’l J. Computer Vision, vol. 80, pp. 300-316, 2007.

10. X. He, R.S. Zemel, and M._ A. Carreira-Perpinn˜a´n, “Multiscale Conditional Random Fields for Image Labeling,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, 2004.

11. G. Heitz, S. Gould, A. Saxena, and D. Koller, “Cascaded Classification Models: Combining Models for Holistic Scene Understanding,” Proc. Neural Information Processing Systems, 2008.

12. D. Hoiem, A. Efros, and M. Hebert, “Putting Objects in Perspective,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, 2006.

13. Y. Jin and S. Geman, “Context and Hierarchy in a Probabilistic Image Model,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, 2006.

14. L.-J. Li, R. Socher, and L. Fei-Fei, “Towards Total Scene Understanding: Classification, Annotation and Segmentation in an Automatic Framework,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009.

15. M. Marszałek and C. Schmid, “Semantic Hierarchies for Visual Object Recognition,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2007.

16. J. Porway, K. Wang, B. Yao, and S.C. Zhu, “A Hierarchical and Contextual Model for Aerial Image Understanding,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008.

17. Z. Tu, “Auto-Context and Its Application to High-Level Vision Tasks,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008.

Aravindh A.S., B.Tech.
asaravindh.121193@gmail.com

J. Christy Samuel
jchristysamuel@gmail.com