International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-2, May 2011

Evaluation of Object Based Video Retrieval Using SIFT

Shradha Gupta, Neetesh Gupta, Shiv Kumar

Abstract— **In Video Retrieval system, each video that is stored in the database has its features extracted and compared to the features of the query image. The local invariant features are obtained for all frames in a sequence and tracked throughout the shot to extract stable features. Proposed work is to retrieve video from the database by giving query as an object. Video is firstly converted into frames, these frames are then segmented and an object is separated from the image. Then features are extracted from object image by using SIFT features. Features of the video database obtained by the segmentation and feature extraction using SIFT feature are matched by Nearest Neighbor Search (NNS). In this paper we have evaluated the proposed video retrieval system. The proposed method is better than previous video retrieval methods because it is invariant to illumination changes.**

***Index Terms*— Video retrieval, segmentation, SIFT, Nearest-neighbor search.**

I. INTRODUCTION

Video retrieval has become a prominent research topic in recent years. Research interest in this field has escalated because of the proliferation of video and image data in digital form. The goal in video retrieval is to search through a database to find videos that are perceptually similar to a query video [1]. An ideal video retrieval engine is one that can completely comprehend a given video, i.e., to identify the various objects present in the video and their properties. Advances in data storage and video acquisition technologies have enabled the creation of large video data sets. In this scenario, it is necessary to develop appropriate information systems to efficiently manage these collections. The common approaches use the son called Content-Based Video Retrieval systems [2]. Content based video retrieval system is an active field of research. These systems typically include three steps: video segmentation, feature extraction and feature grouping. Video segmentation algorithms try to divide the video sequences into meaningful subgroups called shots. Most of the existing feature extraction algorithms select one or more key frames as being representative of each shot; feature extraction techniques such as wavelets or Gabor filters are widely used to then extract features from these frames [3]. In the grouping stage, the shot Video segmentation algorithms try to divide the video sequences into meaningful subgroups called shots. Most of the existing feature extraction algorithms select one or more key frames as being representative of each shot; feature extraction techniques such as wavelets or Gabor filters are widely used to then extract features from these frames [3]. In the grouping stage, the shot features are grouped into clusters to represent relevant objects which appear in those shots. In a query problem, features selected from the query region will be compared with existing features for possible matches. However, an object can appear in different imaging conditions (different camera angles, zoom positions, lighting conditions) in different parts of the video and can also be occluded.

In this paper our work is focused on video retrieval using SIFT feature. Video retrieval plays an important role in daily life. Firstly the video is divided into frames, and then frames are divided into images. The object is separated from the image by the segmentation of the image. The segmented object is a part of image. Feature is extracted from the segmented image (object). In these proposed method the features are extracted by using the Scale Invariant Feature Transform (SIFT). SIFT features are used to find the keypoints from the images. SIFT features are invariant to image [4].

Manuscript received March 20, 2011.

Shradha Gupta, Information Technology, RGPV TIT, Bhopal, India, (shraddha20.4@gamil.com).

Prof. Neetesh Gupta, Head & Professor, Department of Information Technology, RGPV TIT, Bhopal, India (Email: gupta.neetesh81@gmail.com).

Prof. Shiv Kumar, Professor, Department of Information Technology, RGPV TIT, Bhopal, India (Email: shivksahu@rediffmail.com).

II. MOTIVATION

Video databases and collections can be enormous in size, containing hundreds, thousands or even millions of videos. The conventional method of video retrieval is searching for a keyword that would match the descriptive keyword assigned to the video by a human categorizer. Currently under development, even though several systems exist, is the retrieval of video based on their content, called ***Content Based Video Retrieval***, ***CBVR***. Motivation for video retrieval is to make it efficient then previous approaches and invariant to illumination changes.

III. BASIC CONCEPT OF VIDEO RETRIEVAL

- **Frames:** Video is divided into **f**rames (images).
- **Feature Extraction:** Features are extracted from the image by using different methods.
- **Matching:** In third step these extracted features are matched from the database videos.

IV. PROPOSED WORK

According to the proposed framework video is divided into frames. A number of frames are generated from the single

video. In the proposed framework we are retrieving video, based on object from the database using SIFT features. The proposed solution for problem is to make the efficient video retrieval. In the proposed solution the video retrieval is done in following steps:

- The video is converted into images.
- These images are segmented using the segmentation algorithm to get the object image.
- Now the features are retrieved from the object image using the SIFT algorithm.
- Last step is the feature matching from the database features by the nearest neighbor algorithm to retrieve the video from the database.

V. METHODOLOGY

A. *Video*

Video data is to mean video as a rich media, not only the video images but also its associated audio. We use video content to denote the information contained in the video data and potentially of interest to the user such as objects, people, motion, static charts, drawings, maps, equations, transparencies and auditorial information. In video production, a shot corresponds to the segment of video captured by a continuous camera recording. Video frame referred to as a static image, is the basic unit of video data [5]. Frame sequence is defined as a set of frame intervals, where a frame interval [i,j] is a sequence of video frames from frame i to j. The individual frames are separated by frame lines. Normally, 24 frames are needed for one second of film. In ordinary filming, the frames are photographed automatically, one after the other, in a movie camera [8]. A common first step for most content-based video analysis techniques available is to segment a video into elementary shots, each comprising a continuous in time and space [10]. These elementary shots are composed to form a video sequence during video sorting or editing with either cut transitions or gradual transitions of visual effects such as fades, dissolves and wipes.Video stream consists of frames, shots, scenes and sequences. Frames are single pictures and the elementary video units. There are 14-25 frames per second, so frame sequences give more meaning than individual frame. Physically related frame sequences generate video **shots.** Shots are segmented based on low level features and shot boundary algorithms can detect shots automatically. Semantically related and temporally adjoining shots are grouped into **scenes**. Scenes may be still small for browsing very long video. It might be necessary to combine related scenes into **sequences** or acts. Fig shows the hierarchical structure of video.

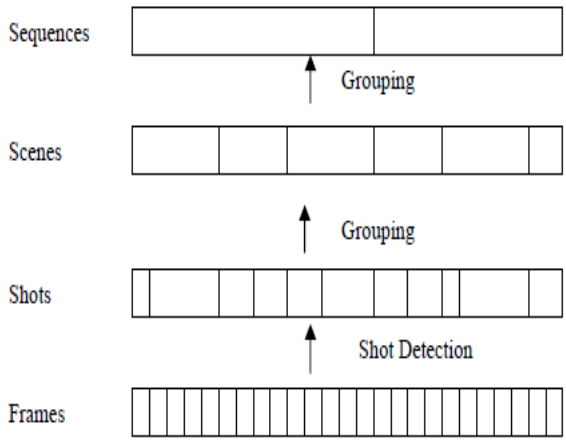


Fig.1 A hierarchical representation of video

B. *Segmentation*

The goal of image segmentation is to cluster pixels into salient image regions, i.e., regions corresponding to individual surfaces, objects, or natural parts of objects. Segmentation is a collection of methods allowing to interpret spatially close parts of the image as objects [6]. Regions (i.e., compact sets) represent spatial closeness naturally and thus are important building steps towards segmentation. Objects in a 2D image very often correspond to distinguishable regions. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in image.

Texture Segmentation

The use of image texture can be used as a description for regions into segments. There are two main types of segmentation based on image texture, region based and boundary based.

Region Based

Region-based segmentation methods attempt to partition or group regions according to common image properties. These image properties consist of:

- Intensity values from original images, or computed values based on an image operator.
- Textures or patterns are unique to each type of region.
- Spectral profiles that provide multidimensional image data.

Boundary Based

Boundary-based methods are often used to look for explicit or implicit boundaries between regions corresponding to different issue types. Fig.2 shows the segmented result of an image. Object is detected from the original image.

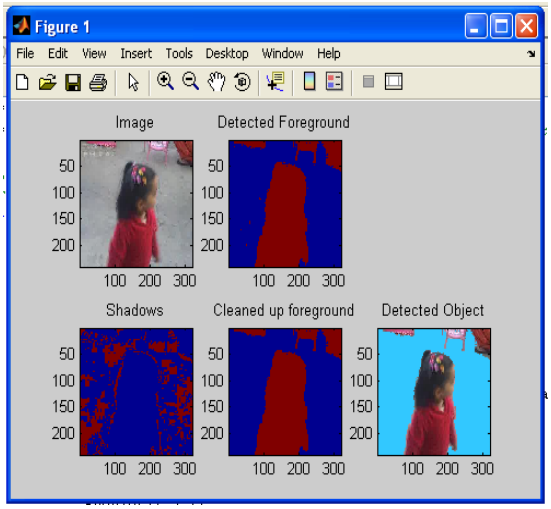


Fig.2 Segmented result of image

C. SIFT (Scale Invariant Feature Transform)

SIFT (Scale Invariant Feature Transform) features are widely used in object recognition. These features are invariant to changes in scale, 2D translation and rotation transformations. SIFT Features however, are of very high dimension and large number of SIFT features are generated from an image [4]. The large computational effort associated with matching all the SIFT features for recognition tasks, limits its application to object recognition problems. Image matching is a fundamental aspect of many problems in computer vision, including object or scene recognition, solving for 3D structure from multiple images, stereo correspondence, and motion tracking. Following are the major stages of computation used to generate the set of image features:

1. **Scale-space extrema detection:** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation. The scale space of an image is defined as a function, $L(x, y, \sigma)$, that is produced from the convolution of a variable-scale Gaussian, $G(x, y, \sigma)$, with an input image, $I(x, y)$:

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

where $*$ is the convolution operation in x and y, and

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

To efficiently detect stable keypoint locations in scale space, we have proposed (Lowe, 1999) using scale-space extrema in the difference-of-Gaussian function convolved with the image, $D(x, y, \sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:

$$D(x, y, \sigma) = (G(x, y, \kappa\sigma) - G(x, y, \sigma)) * I(x, y)$$
$$= L(x, y, \kappa\sigma) - L(x, y, \sigma)$$

There are a number of reasons for choosing this function. First, it is a particularly efficient function to compute, as the smoothed images, L, need to be computed in any case for scale space feature description, and D can therefore be computed by simple image subtraction.

2. **Keypoint localization:** At each candidate location, a detailed model is fit to determine location and scale. Key points are selected based on measures of their stability. Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. This approach uses the Taylor expansion (up to the quadratic terms) of the scale-space function, $D(x, y, \sigma)$ shifted so that the origin is at the sample point:

$$D(\mathrm{X}) = D + \frac{\partial D^{\mathrm{T}}}{\partial \mathrm{X}} \mathrm{X} + \frac{1}{2} \mathrm{X}^{\mathrm{T}} \frac{\partial^2 D}{\partial \mathrm{X}^2} \mathrm{X}$$

where D and its derivatives are evaluated at the sample point and $\mathrm{X} = (x, y, \sigma)^{\mathrm{T}}$ is the offset from this point. The location of the extremum, $\bar{X}$, is determined by taking the derivative of this function with respect to x and setting it to zero, giving

$$\bar{X} = -\frac{\partial^2 D^{-1}}{\partial \mathrm{X}^2} \frac{\partial D}{\partial \mathrm{X}}$$

The function value at the extremum $D(\bar{X})$, is useful for rejecting unstable extrema with low contrast. This can be obtained by substituting the above equations, giving:

$$D(\bar{X}) = D + \frac{1}{2} \frac{\partial D^{\mathrm{T}}}{\partial \mathrm{X}} \bar{X}$$

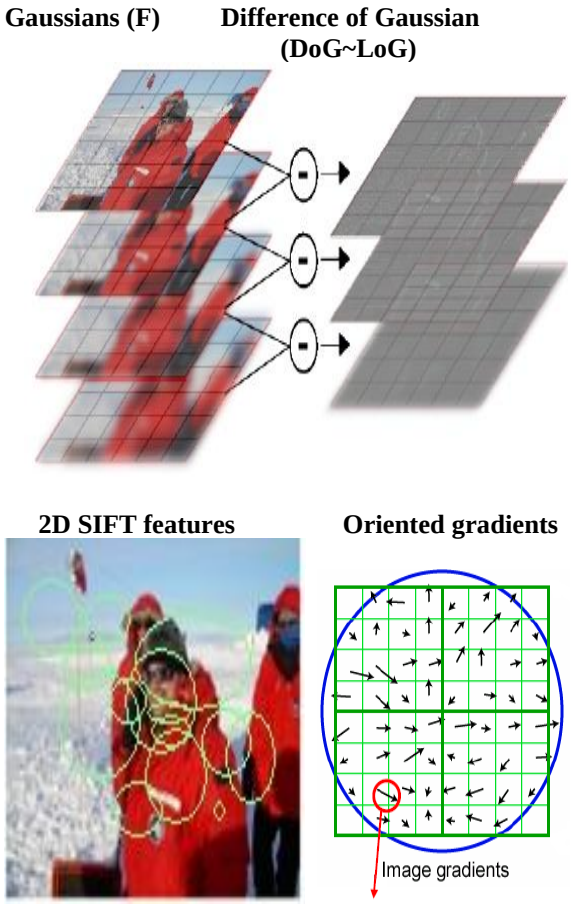
3. Orientation assignment: One or more orientations are assigned to each key point location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations. The scale of the keypoint is used to select the Gaussian smoothed image, L, with the closest scale, so that all computations are performed in a scale-invariant manner. For each image sample, $L(x, y)$, at this scale, the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, is precomputed using pixel differences:

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

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

4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that

allows for significant levels of local shape distortion and change in illumination. The previous operations have assigned an image location, scale, and orientation to each keypoint. These parameters impose a repeatable local 2D coordinate system in which to describe the local image region, and therefore provide invariance to these parameters. The next step is to compute a descriptor for the local image region that is highly distinctive yet is as invariant as possible to remaining variations, such as change in illumination or 3D viewpoint. First the image gradient magnitudes and orientations are sampled around the keypoint location, using the scale of the keypoint to select the level of Gaussian blur for the image. In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation. For efficiency, the gradients are precomputed for all levels of the pyramid. A Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point.



$$m(x,y)=\sqrt{(F(x+1,y)-F(x-1,y))^2+(F(x,y+1)-F(x,y-1))^2}$$
$$\theta(x,y)=\text{atan}((F(x,y+1)-F(x,y-1))/(F(x+1,y)-F(x-1,y)))$$

Fig. 3 Example of SIFT Implementation

D. Selection of local features

The following requirements were key in selecting a suitable local-feature for images used in this project [9]:

a) Invariance: The feature should be resilient to changes in illumination, image noise, uniform scaling, rotation, and minor changes in viewing direction.
b) Highly Distinctive: The feature should allow for correct object identification with low probability of mismatch.
c) Performance: Given the nature of the image recognition problem for an art center, it should be relatively easy and fast to extract the features and compare them against a large database of local features.

E. Matching

Nearest neighbor search

The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of keypoints from training images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. Nearest neighbor search (NNS), also known as proximity search, similarity search or closest point search, is an optimization problem for finding closest points in metric spaces. The problem is: given a set *S* of points in a metric space *M* and a query point $q \in M$, find the closest point in *S* to *q*. In many cases, *M* is taken to be *d*-dimensional Euclidean space and distance is measured by Euclidean distance or Manhattan distance. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor [7]. The k nearest-neighbor classifier is commonly based on the Euclidean distance between a test sample and the specified training samples. Let $\mathbf{x}_i$ be an input sample with p the total number of feature $(j = 1,2,\ldots,p)$ The Euclidean distance between sample $\mathbf{x}_i$ and $\mathbf{x}_l$ ($l = 1,2,\ldots,n$) is defined as features $(x_{i1}, x_{i2}, \ldots, x_{ip})$, n be the total number of input samples ($i = 1,2,\ldots,n$) and p

$$d(\mathbf{x}_i,\mathbf{x}_l)=\sqrt{(x_{i1}-x_{l1})^2+(x_{i2}-x_{l2})^2+\cdots+(x_{ip}-x_{lp})^2}$$

VI. Flow Chart

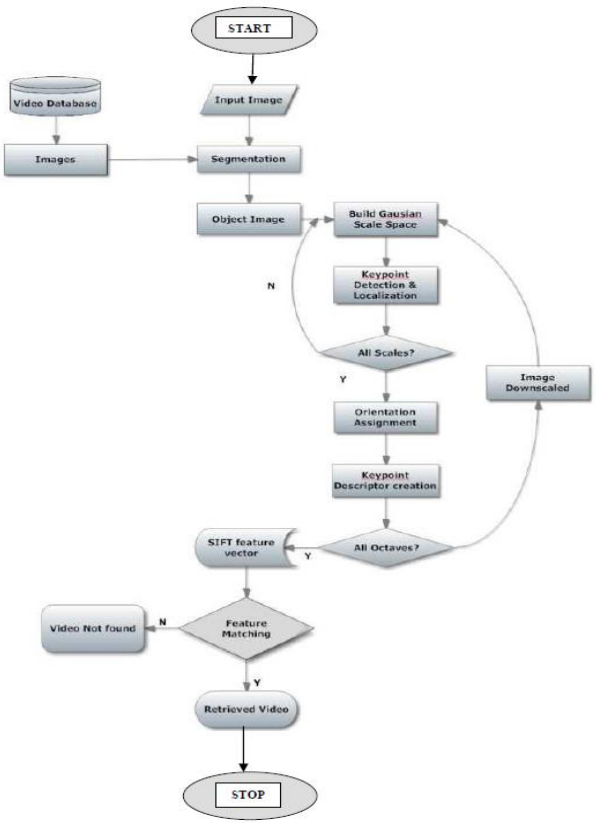


Fig.4 Flow Chart of Proposed work

Flow chart description

1. Take image from the video as input.
2. Separate the object from the image by using segmentation.
3. After detecting object apply SIFT features to get feature vector:
   - Build Gaussian scale space.
   - Keypoint detection & localization. It is checked for all scales, if yes then orientation assignment is done, else again Gaussian scale space is build.
   - Keypoint descriptor creation.
   - This keypoints descriptors are checked for all octaves, if true then SIFT feature vector is generated, else image is downscaled.
4. Video database follow the same steps as described above.
5. SIFT feature vector is matched from the database features. If matched video is retrieved, else not.

VII. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed approach, we set up a database that consists of 20 short videos. The videos containing objects are fan, cow, girl, pen drive, box, toys, etc. The performance of video retrieval is usually measured by the following two metrics:

Precision: In the field of video retrieval, **precision** is the fraction of video that are relevant to the search. A good retrieval system should only retrieve relevant items.

$$precision = \frac{|\{relevant\ videos\} \cap \{retrieved\ videos\}|}{|\{retrieved\ videos\}|}$$

Recall: Recall in video retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved. A good retrieval system should retrieve as many relevant items as possible.

$$recall = \frac{|\{relevant\ videos\} \cap \{retrieved\ videos\}|}{|\{relevant\ videos\}|}$$

Table 1 shows the experimental results of proposed video retrieval system. We calculate the result by the two metrics precision and recall. Fig.5 shows the graph between precision and recall.

Query Image	Detected Object	Actual Retrieved Video	Precision	Recall
			1.00	0.083
			1.00	0.110
			1.00	0.500
			1.00	0.465
			1.00	0.286
			0.96	0.400
			1.00	0.625
			0.92	0.538
			0.94	0.429
			0.98	0.440

Table 1 Experimental results

Results show that the performance of the system is more than 95%. The video retrieval is more efficient than previous approaches because it is invariant to illumination changes. Fig.5 shows the graph between Precision and Recall.

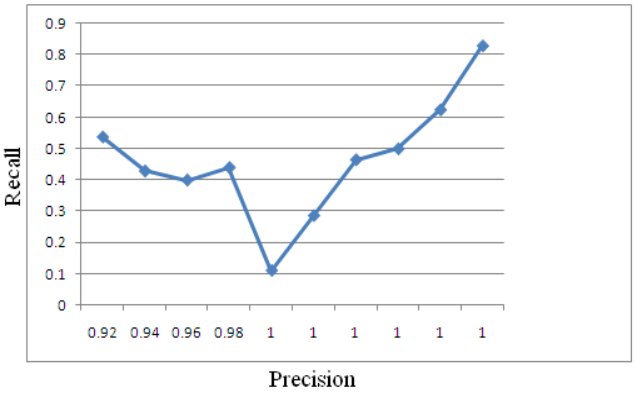


Fig.5 Graph between Precision and Recall

VIII. CONCLUSION

A method for retrieving video containing a particular object, a single image of the object is given as a query. In the

proposed method video retrieval is based on object. The object is detected by the segmentation. SIFT algorithm is applied to the object. The object is invariant to illumination changes. Features extracted by the SIFT are keypoints. These keypoints are matched by the database features by Nearest-Neighbor matching algorithm. The result has being evaluated in the form of precision and recall. This video retrieval system is efficient than existing systems because illumination changes does not affect the system.

REFERENCES

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

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

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

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

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

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

[7] 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.

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

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

[10] 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.