Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 03, March -2018

A SURVEY OF OBJECT VIDEO RETRIEVAL SYSTEMS USING SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

B.Ramu[1], Jugal Kishore Bhandari [2]

[1]*Assistant Professor, Dept. of ECE, GCET, Hyd.,India.*
[2]*Assistant Professor, Dept. of ECE, GCET, Hyd.,India*

Abstract- *Multimedia information systems are increasingly important with the advent of broadband networks, high-powered work stations, and compression standards. Since visual media requires large amounts of storage and processing, there is a need to efficiently index, store, and retrieve the visual information from multimedia database. Content based video retrieval is a proper solution to handle the video data. But because of their huge volumes and high dimensionality, finding a proper way to organize them for efficient search and retrieval becomes a challenging and important task. 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 algorithm. Features of the video database obtained by the segmentation and feature extraction using SIFT algorithm are matched by Nearest Neighbor Search (NNS). Experimental Results confirm the effectiveness and robustness of the algorithm.*

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

I. INTRODUCTION

Along with the rapid development of information technology, more and more multimedia information is available [1], especially for video data. Then a new problem arises: how to get the interested video from the vast amount of video data. Keywords based traditional retrieval method cannot work efficiently anymore. There is an urgent need for effective video retrieval method [2]. Therefore, Content Based Video Retrieval (CBVR) has become a hot research topic in recent years. With the increasing proliferation of digital video contents, efficient techniques for analysis, indexing and retrieval of videos according to their contents have become evermore important. 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. 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. Shot boundaries are typically found by, computing an image-based distance between adjacent frames of the video and noting when this distance exceeds a certain threshold. The distance between adjacent frames can be based on statistical properties of pixels], compression algorithms], or edge differences. The most widely used method is based on histogram differences. If the bin-wise difference between histograms for adjacent frames exceeds a threshold, a shot boundary is assumed. Zhang *et al*. used this method with two thresholds in order to detect gradual transitions. In recent years research has focused on the use of internal features of images and videos computed in an automated or semi-automated way . Automated analysis calculates statistics, which can be approximately correlated to the content features. This is useful as it provides information without costly human interaction

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 key points from the images. SIFT features are invariant to image.

2. 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.

Selection of local features

The following requirements were key in selecting a suitable
Local-feature for images used in this project.

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.

A. Object-based Segmentation

Different classification techniques have been applied for parsing image information and conducting unsupervised classification , object-based classification and fuzzy logic . Traditional pixel based image analysis techniques incorporate mainly intensity information into the workflow process while neglecting the spatial arrangement of pixels and referential data that can be exploited in image classification. The method employed in this section evaluates classification technique based on object-oriented image segmentation approach. We used a toolbox (Developer Life 5.0 (Definiens Inc., Munich, Germany) to perform specific object-oriented tasks, in which pixels are combined to larger objects based on user defined homogeneity criteria (see for specifics), and, as multiple levels of resolutions are generated, a topological relationship between objects on different levels can be defined. The software allowed us to record the required functions in a macro-style program sequence, to customize algorithms and to set parameters. Subsequently, the software was run in a dedicated execution environment directly accessing images from an image database.

The object-oriented technique takes into account not only the spectral and positional characteristics of a single pixel but as well those of the surrounding (contextual) pixels in the image segmentation phase. The result is the creation of varying image object dimensions defined as individual areas with shape and spectral homogeneity . The extracted image objects can provide a greater number of meaningful features for image classification making the image more feature rich in details. Furthermore, objects can also be developed from any spatially distributed variable (e.g., compactness, border, position). Homogeneous image objects are then analyzed using traditional classification algorithms (e.g., nearest-neighbor, enclosed objects, existence of objects). This classification method uses the approach of recognizing important semantic information not only represented in single pixels but in meaningful objects in an image and their contextual and mutual relation. Object-oriented classification assumes that related pixels are actually part of objects, and assigns properties and relationships to the whole object rather than individual pixels. The algorithms behind this classification method utilize spectral, spatial, texture, shape, context and ancillary information to model the feature extraction process. Another key aspect of segmentation are image layers, since image information specific for the modality can reside in particular layers which can then be easily segmented into image objects later to be referenced with corresponding image layers in classification of contextual information.

SIFT (Scale Invariant Feature Transform):

SIFT (Scale Invariant Feature Transform) features are widely used in object recognition. it transforms image data into scale-invariant coordinates relative to local features An important aspect of this approach is that it generates large numbers of features that densely cover the image over the full range of scales and locations. Atypical image of size 500×500 pixels will give rise to about 2000 stable features (although this number depends on both image content and choices for various parameters).The quantity of features is particularly important for object recognition, where the ability to detect small objects in cluttered backgrounds requires that at least 3 features be correctly matched from each object for reliable identification. For image matching and recognition, SIFT features are first extracted from a set of reference images andstored in a database. A new image is matched by individually comparing each feature from the newimage to this previous database and finding candidate matching features based on Euclidean distance of their feature vectors. This paper will discuss fast nearest-neighbor algorithms that can perform this computation rapidly against large databases. 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:

$$\begin{aligned}D(x,y,\sigma)&=(G(x,y,\kappa\sigma)-G(x,y,\sigma))*I(x,y)\\&=L(x,y,\kappa\sigma)-L(x,y,\sigma)\end{aligned}$$

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. **Key point 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 key point 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:

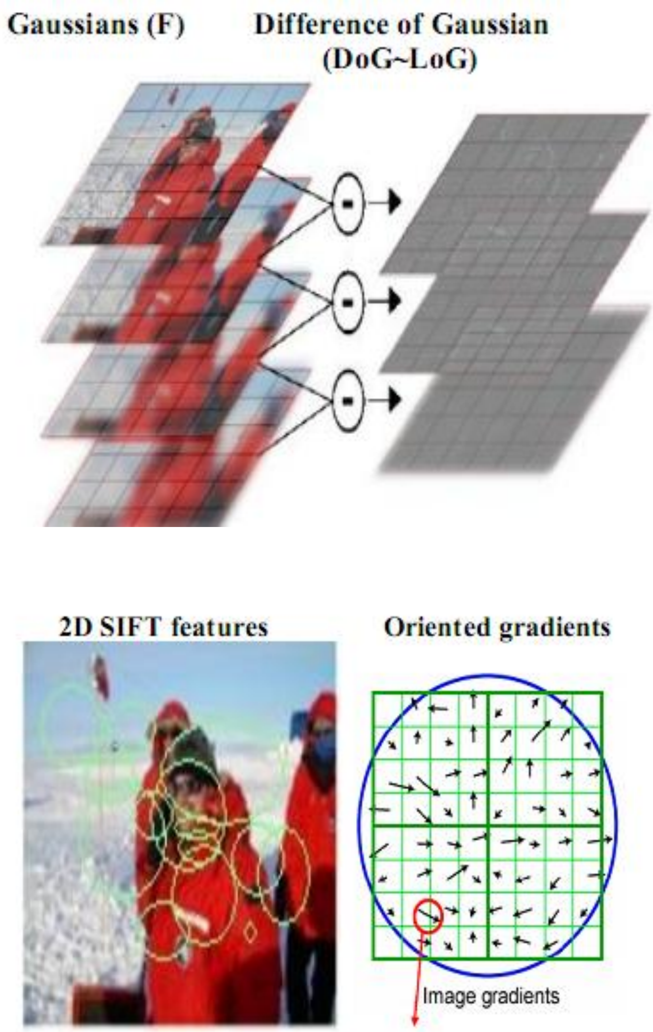
3. Orientation assignment: One or more orientations are assigned to each key point location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations. The scale of the key point 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, □ □(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. **Key point descriptor:**

The local gradient data from the closest smoothed image L(x, y,s) is also used to create the keypoint descriptor. This gradient information is first rotated to align it with the assigned orientation of the keypoint and then weighted by a Gaussian with s that is 1.5 times the scale of the keypoint. The weighted data is used to create a nominated number of histograms over a set window around the keypoint. Typical keypoint descriptors use 16 orientation histograms aligned in a 4x4 grid. Each histogram has 8 orientation bins each created over a support window of 4x4 pixels. The resulting feature vectors are 128 elements with a support window of 16x16 scaled pixels..



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

Fig. 3 Example of SIFT Implementation

Matching:

Given a test image, each one of its key point is compared with key points of every image present in the training database. Euclidean distance between each invariant feature descriptor of the test image and each invariant feature descriptor of a database image is computed at first. However two key points with the minimum Euclidean distance (the closest neighbors) cannot necessarily be matched because many features from an image may not have any correct match in the training database (either because of background clutter or may be the feature was not detected at all in the training images). Instead the ratio of distance between closest neighbors and distance between the second closest neighbors is computed. If the ratio of distances is greater than 0.6, then that match is rejected. Once the comparison is performed with all test images in the database, the title of the database image with the maximum number of matches is returned as the recognized image.

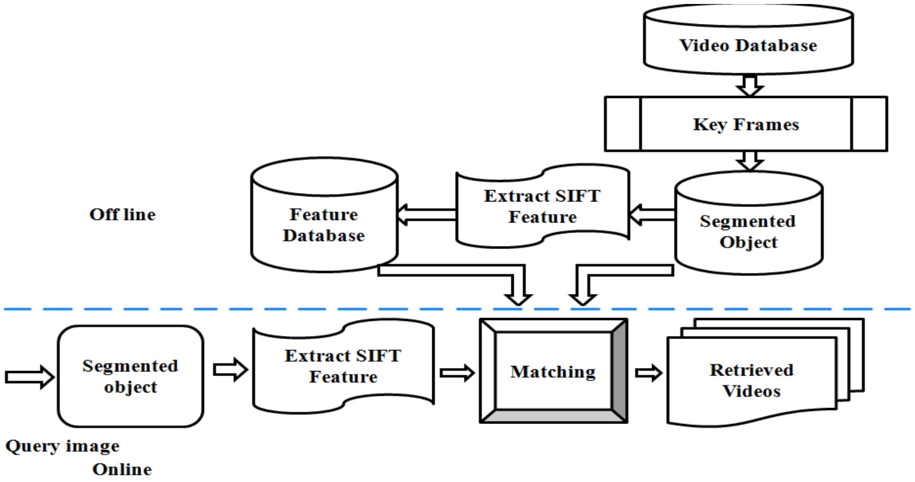


Fig2: block diagram for proposed work

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:

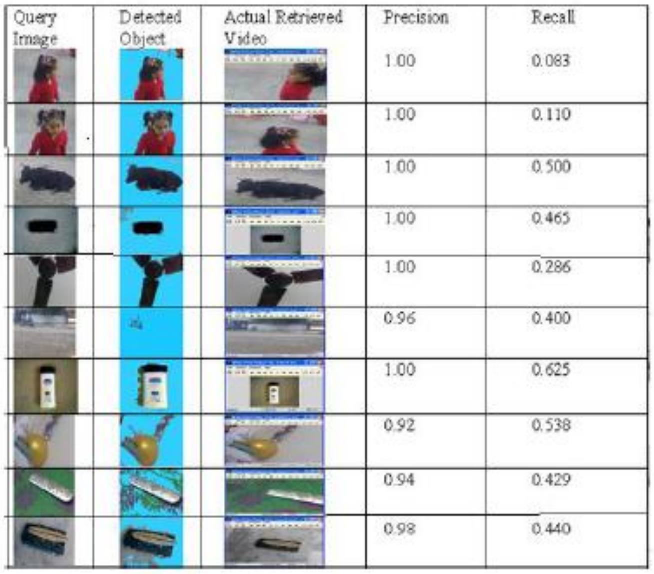
Precision: In the field of video retrieval, **precision** is the fraction of video that are relevant to the search. A good retrieval system should only retrieve relevant items.

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

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.

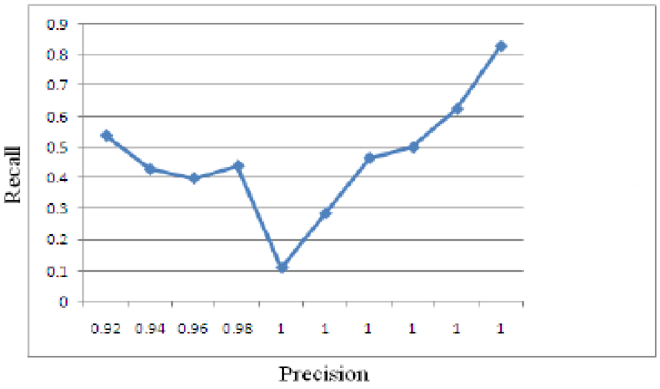


Fig.5 Graph between Precision and Recall

REFERENCES

[1] Weiming hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank, " A Survey on Visual Content-Based Video Indexing and Retrieval," IEEE transactions on systems, man, and cybernetics-part c: applications and reviews 2011, pp. 1-23

[2] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," ACM Trans. Multimedia Comput., Commun. Appl., vol. 2, no. 1, pp. 1–19, Feb. 2006.

[3] Han-ping Gao Zu-qiao Yang, " Content based Video Retrieval using Spatiotemporal Salient Objects," IEEE transactions on Intelligence Information Processing and Trusted Computing 2010, pp 689-692.

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

B.Ramu , received his B.Tech degree in Electronics & communication from JNT University in 2010, M.Tech Degree from JNT University in 2012. He is currently working as Assistant Professor, Dept. of ECE Geethanjali College of Engineering and Technology Autonomous, Secunderabad-501301, T.S. India. His areas of interest Image and video processing

.

Jugal Kishore Bhandari received the B.E. degree in Electronics & communication engineering from Anna University, Chennai, India, in 2007 and Masters of Engineering from Anurag Group of Institutions Formerly known CVSR College of engineering, Ghatkesar, hyd, presently he is working as assistant professor at Geethanjali college of engineering and technology, Hyd. His areas of interest are low power design, fault tolerant system designing and digital electronics.