Video Data Retrieval using Image Color Histogram Technique

D.Saravanan, J.Surendiran

Abstract: ***In this paper, a subspace-based multimedia data-mining framework is proposed for video semantic analysis; specifically Current content management systems support retrieval using low-level features, such as motion, color, and texture. The proposed frameworks achieves full automation via a knowledge-based video indexing and retrieve an appropriate result, and replace a presented object with the retrieval result in real time. Along with this indexing mechanism a histogram-based color descriptors also introduced to reliably capture and represent the color properties of multiple images. Including of this a classification approach is also carried out by the classified associations and by assigning, each of them with a class label, and uses their appearances in the video to construct video indices. Our experimental results demonstrate the performance of the proposed approach.***

Index Terms– video indices, video indexing, histogram

I. INTRODUCTION

The usage of the images has been increasing day by day as they are not only being used for expressing and explanation but also for identification and analysing. They are being used in every field today. Large amounts of image data is being stored in the databases everyday. All the data can only be used efficiently only when the correct data is retrieved successfully. The existing successful systems are text based and are not effective at retrieving the images. For efficient retrieval the image properties are used to retrieve the appropriate images which do not depend on image descriptions. The present day there are only a couple of systems that implement the retrieval mechanism based on content. The proposed system identifies the objects in the images based on the colour distribution and performs the search. A colour histogram is generated and image segmentation is done to obtain the suitable images. Every image is made up of tiny bins called pixels and each pixel is made up of the Red, Green and Blue colours. The system utilizes these values in retrieving the suitable solutions.

1.1 Related Work in Video Indexingz

The evolution of hardware for picture and video media has been remarkable. The trend of multimedia source digitization has accelerated the combining of many multimedia processing techniques, such as storage, editing, and presentation on computers. Accordingly, techniques are expected for new methods of presenting and creating multimedia content. In order to construct an advanced multimedia-processing environment, a huge multimedia database system is required. One of the most important considerations for such a database system is how to reflect the intention of a user in data retrieval.

In a typical system, a user retrieves character data by inputting key words. In other database systems, a user uses numeric values on similar images as the key for retrieval. However, in the retrieval of video data with motion or shapes, the retrieval method of a user is not necessarily in agreement with the particular data to be retrieved even when the database system provides graphic user interface (GUI) tools. This paper describes a retrieval technique based on gestures for video images. By using gestures, a user can input spatiotemporal information intuitively. For example, a gesture can be used to input the motion of an object's direction; the motion of the camera; pan and tilt; the shape of an object; to represent the degree of a scale, and so on. Furthermore, it can objectively match keys and indices because both the motion of gestures and the motion of objects in a video can be obtained automatically. We propose the "Interactive Video Database," which enables the creation and presentation of scenes with gestures by replacing the objects that are extracted. By using this technique, a user can create varied and highly realistic video content via intuitive interaction in real time.

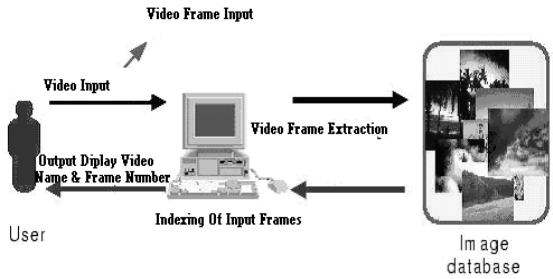


Figure 1 System Architecture

1.2Video Content Modeling

In order to efficiently describe the video content, we decided to borrow a well-known method used for text document analysis named Latent Semantic Indexing [12]. First we detail the adaptation of LSI to our situation and then propose methods to include multiple features and to improve the robustness of LSI in our particular case, i.e modeling of video shots in a large database. Latent Semantic Indexing is a theory and method for extracting and representing the contextual meaning of words by statistical computations applied to a large corpus of text. The underlying idea is that the aggregate of all the word contexts in which a given word does and does not appear provides a set of mutual constraints that largely determines the similarity of meaning of words and sets of words to each other [13]. In practice, we construct the occurrence matrix A of words into documents. The singular value decomposition of A gives transformation parameters to a singular space where projected documents can efficiently be compared.

Revised Manuscript Received on 14 September, 2019.

D.Saravanan, Faculty of Operations & IT, ICFAI Business School (IBS), Hyderabad

J.Surendiran, Professor, HKBKCE, Bangalore.

For video content analysis, a corpus does not naturally exist, however one can be obtained thanks to vector quantification technics. In [14], we presented an approach on single video sequences that relies on k-means clustering to create a corpus of frame-regions. Basically, key-frames are segmented into regions [15] and each region is represented by a set of features like color histogram and Gabor's energies. They are then mapped into a codebook, obtained with the k-means algorithm, to construct the cooccurrence matrix A of codebook elements in video keyframes. Thus each frame is represented by the occurrence of codebook terms. LSI is then applied to the matrix A and provides projection parameters U into a singular space where frame vectors are projected to be indexed and compared. This can be extended to model a set of video sequences; the set can be seen as a unique video where keyframes are the representative frames of shots.

Mathematical operations are finally conducted in the following manner:

- First a codebook of frame – tregions is created on a set of training videos,
- The co-occurrence matrix is constructed:

Let A of size M by N be the co-occurrence matrix of ?M centroids (defining a codebook) into N- key frames (representing the video database).its value at cell (i,j) corresponding to the n8mber of times the region I appears in the frame j.

- Next, it is analyzed through LSA:
  The SVD decomposition gives A = USV^i where
  $$\mathbf{UU^i = VV^i = I,\ L = min\ (M,N)}$$
  $$\mathbf{S \approx diag\ (\sigma_1,\ldots\ldots \sigma_L),\ \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_L}$$
  Then A is approximated by truncating U and V matrices to keep k factors in S corresponding to the highest singular values.
  $$\mathbf{\hat{A} = U_k S_k V_k^t \ with\ S_k = diag(\sigma_1,\ldots\ldots \sigma_k)}$$
- Finally, indexing of a context of A noted c(j) and a new context q is realized as follows:
  $$P_{C(j)} = \text{row j of VS}$$
  $$\mathbf{P_q = q^t\ U_k}$$
- And to retrieve the context q in a database containing indexed contexts P_j, the cosine measure m_c is used to compare elements.
  $$m_c(P_j,q) = P_q.P_j\ /\ \|P_q\|.\|P_j\|$$

the most similar elements to the query are those with the highest value of m_c.

The number of singular values kept for the projection drives the LSA performance. On one hand too many factors are kept, the noise will remain and the detection of synonyms and polysemy of visual terms will fail. On the other hand if too few factors are kept, important information will be lost degrading performances. Unfortunately no solution has yet been found and only experiments allows to find the appropriate factor number.

In the particular situation of video content, many features can be extracted. Three methods were evaluated to consider the multiple features. They are combined at the origin, before the creation of the codebook, or independent codebooks are merged to create a single occurrence matrix, or the LSI is applied to each feature and the similarity measure is modified to combine outputs from each singular space. We retained that equivalent performances were obtained when features were combined just before or after LSI. The latter solution being the most flexible is kept fro our task. Indeed features added without the need to do all computation tasks again. Contrary to the modeling of a single video content, LSI does not reveal as per formant for many videos. The occurrence information in each frame is too weak compared to approximations inherent to the use of the codebook and this effect is further emphasized when many videos are implied. To compensate for codebook instability, we match a region to its k – nearest elements in the codebook. This one to many relationship allows to inject more occurrence information for each key-frame and to deal with the sub-optimality of the codebook. E observe a real improvement when looking for similar frames in the database .

II. 2. THE PROPOSED APPROACH

2.1 Input Video File Processing

The human visual system appears to be capable of temporally integrating information in a video sequence in such a way that the perceived spatial resolution of a sequence appears much higher than the spatial resolution of an individual frame. While the mechanisms in the human visual system that do this are unknown, the effect is not too surprising given that temporally adjacent frames in a video sequence contain slightly different, but unique, information. By Extracting both the spatial and temporal information present in a short image sequence to create a single high-resolution video frame. A novel observation model based on motion compensated sub sampling is proposed for a video sequence. In this Video File processing applications the histogram mechanism is proposed. Which utilize the motion histogram for video retrieval, clustering and classification? The video database consists of 20 video sequences taken from the AVI, MPEG-7 test set. The videos include different videos (e.g., file downloading, Earth Movement), Folder Copy etc. The duration of each video segment varies from 1 to 5 s and there are a total of 50 frames. We expect various types of motion content in the different videos and therefore, they form a suitable data set in which to test the proposed motion histogram.

2. 2 Splitting Of Frame

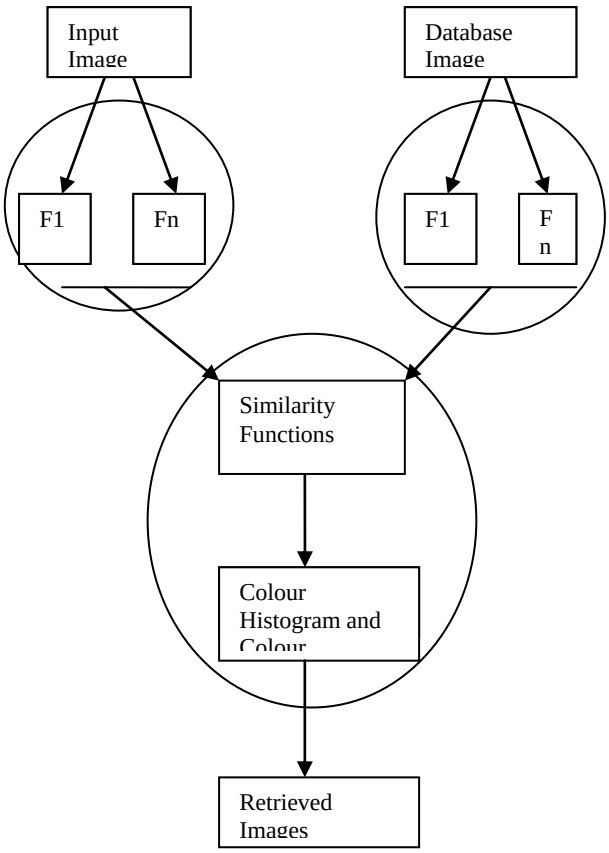
Splitting of frame is the key process and is a simple however effective form of summarizing a long video sequence. The number of key frames used to abstract a shot should be compliant to visual content complexity within the shot and the placement of key frames should represent most salient visual content. Motion is the more salient feature in presenting actions or events in video and, thus, should be the feature to determine key frames. By implementing the model motion patterns in video and a scheme to extract key frames based on this model. The frames at the turning point of the motion acceleration and motion deceleration are selected as key frames. The key-frame selection process is threshold free and fast and the extracted key frames are representative.

2.3 Indexing Of Frame

In order to for the system to fetch the results at a faster rate, there are techniques like catching the most common search or search results and one among them is the Indexing.

For any data retrieval the indexing is treated as an important aspect. The concept of indexing is not new to the world, this is used to catalogue the books and other articles in the libraries and book stores where the books are placed in an order and the reader can go to the right place without going through the whole books. As mentioned earlier there are different types of systems that fetch work based on different algorithms and the retrieved image can be the same query image or a part of the query image or even objects in the image. Sometimes the query can be "retrieve ten most similar images to the given image" which can be called as the Nearest-neighbour. In the existing systems the performance is proportional to the database size. The efficiency of the system should depend on the number of similar images rather the total number of images in the database. [4]

There are two ways of indexing an image in the multimedia databases. One is based on the image features and the other coefficients of the images. The proposed system uses the Indexing frames of images and objects there in are manually identified and described in terms of what they are and what they represent, and in indexing certain features of the images, such as color, are automatically identified and extracted. first method for indexing where the features of the images are calculated and stored in the database along with the image, features like pixel measures, colour histograms etc. These stored values are called Feature Vectors. The same is calculated for the query image. If two images are said to be similar then their feature vectors are similar. The difference between the features vectors finds the distance between the colours. (5)



Where F1 – Fn - Feature Values or vectors from 1 to n

Here feature can be of colour values like the Red, Green and Blue or the values of the generated histogram.

Figure 2.3.1 Indexing of Video Frames

2. 4 Histogram Search

Histogram search algorithms [4], [18] characterize an image by its color distribution or histogram. A histogram is nothing but a graph that represents all the colors and the level of their occurrence in an image irrespective of the type of the image. Few basic properties about an image can be obtained from using a Histogram. It can be used to set a threshold for screening the images. The shape and the concentration of the colors in the histogram will be the same for similar objects even though they are of different colors. Identifying objects in a grey scale image is the easiest one as the histogram is almost similar as the objects have the same colors for same objects. In order for identifying the objects in the images or generating the histogram the system has to obtain the array values.

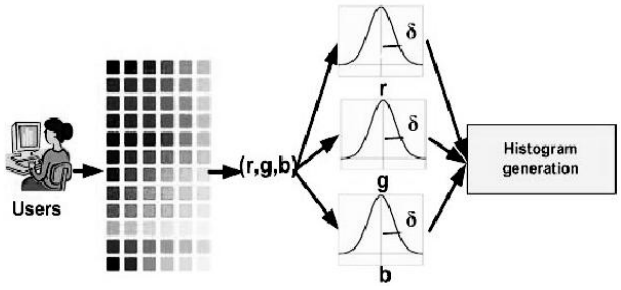


Figure 2.4.1 Color histogram generation process

In general any image contains useful and unwanted information. The system has to differentiate between the both. Consider the below image where the person reading a book is the useful information and the background, people and the market is the unwanted data. The system has to group together the repeated pattern to identify the objects in the image. For example below is given the array for the part of the shirt and this pattern is repeated again

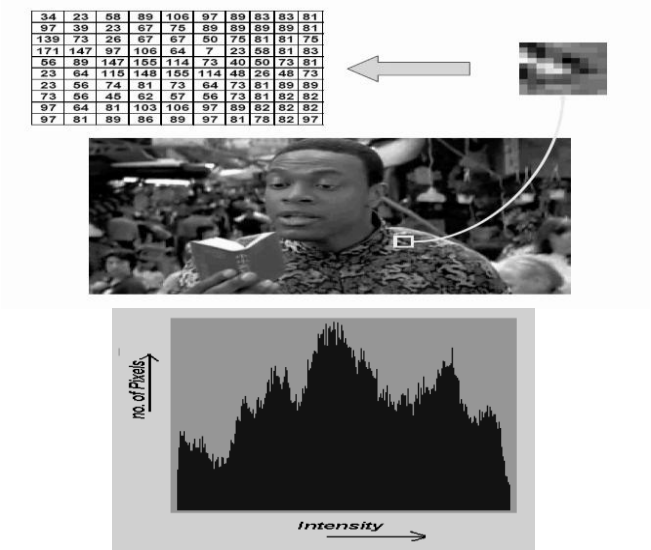


Figure 2.4.2 (Source: Tang, 2007)

Consider the above image where the small part of the person's) shirt is enlarged and the respective representation in the form of the array is given.

The basic concept behind the histogram generation is simple. Each pixel in the image is scanned and the respective color or intensity value is obtained for the pixel.

iColor = (16 * p1[0]) + p1[1] * 4 + p1[2];

Then a graph is generated with total number of pixels against the pixel intensity. An array variable is chosen to store the different intensities and the counter increases for each repeated intensity counting the total number of occurrences of that particular color or intensity.

iHistoArr[iColor] = iHistoArr[iColor] + 1;

An image histogram shows the distribution of pixel intensities within an image. Figure 2.4.3 is an example of an image histogram with amplitude (or color) on the horizontal axis and pixel count on the vertical axis.

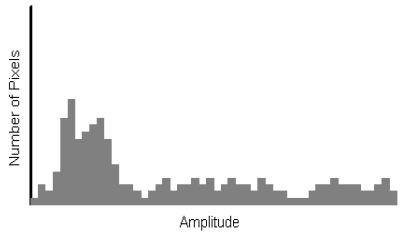


figure :2.4.3 An example of image histogram

2.4.3 Calculation of Image Histogram:

Calculating an image histogram on a sequential device with single thread of execution is fairly easy:

```
for(int i = 0; i < BIN_COUNT; i++)
result[i] = 0;
for(int i = 0; i < dataN; i++)
result[data[i]]++;
```

Listing 1. Histogram calculation on a single-threaded device. (pseudocode).Distribution of the computation process between multiple execution threads is possible. It amounts to :

1) Subdivision of the input data array between execution threads

2) Processing of the sub-arrays by each dedicated execution thread and storing the resulting to a certain number of sub histograms. In some cases it may also be possible to reduce the number of histograms by using atomic operations, but resolving collisions between threads may turn out to be more expensive.

3) Finally all the sub-histograms need to be merged into a single histogram.

When adapting this algorithm to the GPU several constraints should be kept in mind:

Access to the data[ ] array is sequential, but access to result[ ] array is data dependent (random). Due to inherent performance difference between shared and device memory, especially on random patterns, shared memory is the most optimal storage for the result[] array.

On G8x hardware, the total size of the shared memory variables is limited by 16KB.

A single thread block should contain 128-256 threads for efficient execution.

G8x hardware does not have native support for atomic shared memory operations.

III. EXPERIMENTAL RESULTS

Different experiments were performed, one for assessing computational performance, and one assessing robustness with respect to validate the methods we have described, we implemented the components of the video frame based retrieval system and tested with a general purpose image database including about 100 videos. The table given below shows the time taken for Splitting Number of frames from the image database.

Number Of Frames Splitted	Search Time
100	2 Seconds
1000	5 Seconds
10000	14 Seconds

Table 3.1 Average Search Time

These images are stored in JPEG format with size 384 _ 256 or 256 _ 384. The system was written in the VB programming language and compiled on Windows platform. In this section, we describe the training concepts and show indexing results. The table given below shows the number of frames spitted from the videos.

Number Of Frames	Videos
13116	Cartoon
27956	Tennis
71379	News
13254	Movie Song

Table 3.2 Average Search Time For a particular frame

Fig. 3.1. The model input frames of different types of test videos

IV. CONCLUSIONS

In this Research, the need for an efficient video based indexing & Video Frame Based retrieval system is identified and the problems with the existing systems are discussed. In an introduction, we have to discuss the video content based model. A new system is proposed for different types of Video Frames and the structure of the Frames and the way they are indexed and stored inside the database are discussed. Different types of images can be processed using the system. The images in the database are compared with the properties of the Video Frame based on color histogram and appropriate results are displayed to the user. In this search histogram we have to discuss the calculation of histogram and also give the best example of image histogram.

The architecture is designed by studying different existing systems and current research areas. Research in the fields of image processing, segmentation, edge detection, pattern recognition and more is performed to design the system.

REFERENCES

1. Nevenka Dimitrova, Hong-Jiang Zhang, Behzad Shahraray, Ibrahim Sezan, Thomas Huang, and Avideh Zakhor, "Applications of video-content analysis and retrieval," IEEE Multimedia Magazine, vol. 9, no. 3, pp. 42 – 55, July 2002.
2. D.Saravanan, "Improved image retrieval using image noise removal technique" Pak journal of biotechnology, Vol 14(Special Issue-II-2017), Pages 360-362, Dec- 2017.
3. D.Saravanan, "Clustering the irregularity events in intelligence surrounding systems" Int. Journal of pure and applied mathematics, Vol. 119, No.12(2018), Pages 15025-15035, May-2018 (Special Issues) , ISSN:1311-8080.
4. R. Benmokhtar and B. Huet,"Neural network combining classifier based on Dempster-Shafer theory for semantic indexing in video content," International MultiMedia Modeling Conference, vol. 4351, pp. 196–205, Singapore, 2007.
5. D.K. Park, Y.S. Jeon and C.S. Won, "Efficient use of local edge histogram descriptor,"ACM Workshops on Multimedia, pp. 51–54, USA, 2000.
6. M. Rautiainen and T. Seppanen,"Comparison of visual features and fusion techniques in automatic detection of concepts from news video," IEEE International Conference on Multimedia & Expo, The Netherlands, 2005.
7. E. Allwein, R. Schapire, and Y. Singer, "Reducing multiclass to binary : A unifying approach for margin classifiers." Journal of Machine Learning Research, vol. 1, pp. 113–141, 2000.
8. Steyvers, M., Smyth, P., Rosen-Zvi, M., & Griffiths, T.(2004). Probabilistic Author-Topic Models for Information Discovery. The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, Washington.
9. D.Saravanan," Effective video Content Retrieval using image attributes", EAI Endorsed Transactions on energy Web and Information Technologies, Vol5, Issues18, e8, Pages 1-5. June 2018.
10. D.Saravanan," Efficient Video indexing and retrieval using hierarchical clustering techniques", Advances in Intelligence systems and computing, Volume 712, Pages 1-8, ISBN:978-981-10-8227,6, Nov-2018.
11. Tardini, G. Grana C., Marchi, R., Cucchiara, R., (2005). Shot Detection and Motion Analysis for Automatic MPEG-7 Annotation of Sports Videos. In 13th International
12. Conference on Image Analysis and Processing.
13. Witten, I. and Frank, E. (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005
14. Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. Journal of the American Society of Information Science, 41(6):391–407, 1990.
15. Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. Discourse Processes, 25:259–284, 1998.
16. Fabrice Souvannavong, Bernard Merialdo, and Benoˆ Huet. Video content modeling with latent semantic analysis. In Third International Workshop on Content-Based Multimedia Indexing, 2003.
17. P. Felzenszwalb and D. Huttenlocher. Efficiently computing a good segmentation. In IEEE Conference on Computer Vision and Pattern Recognition, pages 98–104, 1998.