

Video Segment Indexing Through Classification and Interactive View-based Query ^{*}

John Chung-Mong Lee[†], Wei Xiong[†], Ding-Gang Shen[‡] and Ruihua Ma[†]

[†]Department of Computer Science
The Hong Kong University of Science & Technology, Hong Kong

[‡]Institute of Optic-Fibre Technology
Shanghai Jiaotong University, Shanghai, China
Email: {cmlee, csxwei, rhma}@cs.ust.hk

Abstract. As video information proliferates, managing video sources becomes increasingly important. Indices must be constructed to allow any future retrieval. We distinguish two categories of indexing: (i) those that are general-purpose and do not make use of domain-specific knowledge, and (ii) those that are application-dependent. In this paper, we present our work in both categories within the VideoBook project. We discuss how to structure video data into shots (physical parts) and clusters (semantic parts). A video partitioning algorithm is described. Its effectiveness and efficiency lies in the use of both statistical and spatial information in the images without, however, having to examine the entire images. To improve the querying efficiency, we propose to investigate in two directions: deriving higher-level indices through classification and providing a method that finds targets of interest through interactive learning. The first technique takes advantage of domain knowledge of underlying applications. The second technique accounts for quantification effect and noise in images and accommodates “learning from negative examples”, resulting into quite good discriminating power. Experimental results are given to demonstrate the effectiveness of our approach.

1 Introduction

With the rapid progress in video technology, large amounts of video sources become available. This availability is not synonymous with *accessibility*. As a matter of fact, traditional text-based methods for video management do not allow easy access to the video sources. The reason for this is twofold. First, video data is not structured, i.e., they have few or no temporal tags. Secondly, indices are basically textual and not rich enough because of the cost required to index video sources. Thus, generally speaking, the solution to this problem lies in structuring video data and associating it with much richer indices. In particular, visual indices may allow a user to access desired video segments by directly

^{*} This research was supported by Sino Software Research Center of Hong Kong University of Science & Technology.

making use of visual cues. Most prototype systems proposed in the literature, including ours, follow this direction [2, 6, 12, 7].

Structuring video data that consists of segmenting the continuous frame stream into physically discontinuous units, generally called *shots*, is a basic operation. In general, these physical units need to be clustered to form more semantically significant units, such as scenes. This so-called *story-based* video structuring has been used in video information browsing systems (eg. [8, 4]). The shots or scenes are described by one or several representative frames, known by the name *key frames* [8, 12]. At the basis of all these are camera break detection and key frame selection. Camera breaks are usually characterized by brusque intensity pattern change between consecutive frames at the boundary. While camera break detection involves only the determination of a threshold, key frame selection is event-driven, which implies that it is subjective and/or context-dependent. Structured video data should be attached with visual indices to allow visual content based retrieval. Thus it comes the problem of knowing what features to use, how to extract them from images, and how to use them for indexing – a key problem in video retrieval. Indices can be of several different levels. Currently, most low-level features found in image processing or computer vision have been attempted as indices. Such features include color, texture, shape, sketch and motion. Success in these areas, however, is not so impressive. In fact, the ways in which a feature is exploited vary from system to system, leading to different effectiveness.

When used as indices for retrieval, low-level features are not very efficient. They act most of the time as constraints in filtering. Also their use implies on-line computations, which tend to be prohibitive whenever the search space becomes big. Higher level features, up to symbolic ones, on the other hand, are quite efficient in retrieval. For example, it is not easy to find scenes containing a dog if one uses features such as color, shape or the like; but it is quite an easy thing if at the indexing stage the scenes are annotated with the term dog, either manually or by an algorithm. It is also well known that the extraction of higher-level features is as difficult as they are efficient. It is, however, sometimes feasible when we make careful use of available context knowledge.

In this paper, we report techniques recently developed at HKUST relating to VideoBook, a video database management system. We first present video structuring (Section 2), especially camera break detection. We also discuss how to do video structuring through semantic clustering. In Section 3, we address the use of domain knowledge for shot classification. In Section 4, we show how to improve querying efficiency through interactive learning. In each of the above sections, we give algorithms and experimental results. Finally, we conclude and discuss future research directions in Section 5.

2 Video Structuring

2.1 Video partitioning using Net Comparison

The partitioning process consists of the detection of boundaries between uninterrupted segments (camera shots) which involve screen time, space or graphic configurations. These boundaries, also known as transitions, can be classified into two categories: gradual and instantaneous. The most common transition is camera breaks.

Several methods such as pairwise comparison, likelihood comparison and histogram comparison have been introduced [5, 10]. These methods have their merits and limitations. The histogram comparison method is insensitive to image movements since it considers intensity/color distribution – a statistical entity – between consecutive images. But it fails if the intensity/color distributions are similar because it ignores spatial information [3]. Both pairwise comparison and likelihood comparison make use of spatial information but the former is too sensitive to image movements and easily causes false alarms, whereas the latter suffers from computational complexity. To overcome these problems and further reduce the computation time, we propose a method, called Net Comparison (NC). It takes advantage of the robustness of the histogram method and the simplicity of the pairwise method by comparing a statistical quantity – the mean value of intensity – along the predefined net lines. Thus only part of the image is inspected.

The algorithm works as follows. First, M points, uniformly distributed over an image, are chosen. Around each of the points a non-overlapped rectangular window is taken, and its mean intensity value \bar{I}_m ($m = 1..M$) is computed. A camera break is declared if the total number of changed regions is larger than a threshold N_{CR} . A region m is said to have changed if

$$|\bar{I}_m^t - \bar{I}_m^{t+\Delta t}| > \Delta I$$

where the superscripts denote two successive frames and ΔI is a predefined threshold.

For the purpose of comparison, we have implemented four other methods for camera break detection: pairwise, likelihood, global histogram, and local histogram. Many experiments have been conducted, on both color and black/white video. It turns out that the proposed method outperforms the others both in accuracy and in speed (for details, see [9]).

2.2 Clusters and key frames

A continuous video is segmented into shots by partitioning. Each shot is represented by or abstracted into one or more frames, commonly called *key frames*. Key frames can serve two purposes: *browsing* and *computation of indices*. In retrieval by browsing, showing shots using key frames as they are may confuse an untrained user rather than help him get the *story*, hence facilitate the retrieval. In light of the table of contents of a book, shots must be organized into several



Fig. 1. Tennis competition shots.

levels of semantic abstraction. We call this *conceptual* or *semantic clustering*. The unit is *cluster*, which is a collection of semantically related shots and/or clusters [4, 1]. Now the question raised is how to find clusters, and to a less extent, how to choose key frames to represent them.

Generally speaking, such a semantic clustering can only be accomplished by a human operator. Nonetheless, in some particular circumstances, automatic clustering is still possible. Section 3 presents an example in which football shooting shots are clustered using the fact that shootings must happen near the goal posts and goal posts can be detected reliably. In [11], Zhang *et al.* use both spatial and temporal structure to classify anchor-person shots and episodes in a news video. Anchor-person shots identification makes use of *persistancy*, which can be applied to other cases. For example, in a conversation sequence, there are limited fixed viewpoints, although shots change frequently. Thus a sequence of this kind may be well summarized using only a few key frames. Another example deals with sport competitions. A coach may be interested only in shots of a competition, not of the public. A shot classification algorithm may be devised (using color histogram comparison, for example) which retains only shots containing the court (Fig.1).

3 Scene Classification for Indexing

As discussed in introduction, low-level image features are not very efficient for retrieval, because they are general, that is, not very discriminant. Further, all queries cannot be constructed on the basis of these indices. On the contrary, indices of higher-levels are much more efficient. Therefore we should provide this

kind of indices as much as possible. However, their derivation poses problems. Manual input is quite time-consuming and hence impractical. Current computer vision technology is not able to do general scene classification or interpretation. The alternative is thus to take advantage of application context whenever this is possible. This will considerably improve retrieval efficiency.

As an example, we have studied the case of football, one of the most popular sports of the time. Shooting is often the most important instant or the moment that people are most interested in, in a football match. Locating such instants may help a sport programme editor to rapidly make a summary of a match or find out all shootings. One way to do this is to identify all segments containing the penalty areas, or more simply, the goal post, because when shootings occur, the goal post is in general in the camera's view. Now we will show how the specific context knowledge allows us to devise a simple algorithm to perform our task.

3.1 Football Goal Detection

A goal post is composed of two vertical bars and a horizontal bar. Due to view-point differences, the horizontal bar is seen as slanted most of the time. On the contrary, no matter what the viewpoint is, the two vertical bars are almost always vertical in the image plane. Thus we model the goal post as (two) vertical bar(s) in the image. One can think of its detection as a simple edge detection and linking problem. In fact, problems due to low resolution, lighting conditions, as well as various background noise (e.g., advertisements), must be taken into account.

The goal posts are painted in white. In the image, a vertical bar is a line (i) composed of flat roof edge points, (ii) of some length, and (iii) of course, vertical. The following procedure detects potential goal post elements.

```

for each pixel  $(i, j)$  in the image
  if  $(i, j)$  is a roof edge
    then mark  $(i, j)$  as potential bar element
    go to  $(i, j + 1)$ .

```

To determine if a (i, j) is a roof edge, we require that there be simultaneously a up-going jump on the left of (i, j) and a down-going jump on the right, i.e.,

$$\begin{aligned}
 I(i, j) - I(i - k, j) &> \Delta I \text{ and} \\
 I(i, j) - I(i + l, j) &> \Delta I, 0 < k, l < \Delta W
 \end{aligned}$$

where ΔI is a threshold and ΔW is a predefined width of 1 or 2 pixels.

All edges so detected are not bar elements. Such edges form a vertical line of some length after being linked together. At the same time, there may be bar elements that are missed due to noise or lighting conditions associated with a vertical bar. Such points are recovered by examining if there are potential bar elements right above and under them. Further, the spatial resolution limit of one pixel is accounted for by allowing one horizontal pixel displacement. Finally, all

vertical lines the length of which is more than a predefined length (depending on the image size) are declared vertical goal posts.

Our algorithm has been applied to the video tape – “94 World Cup, 50 exciting shootings”. It has successfully detected shots that contain goal posts. Fig.2 shows some example images of goal posts detected.

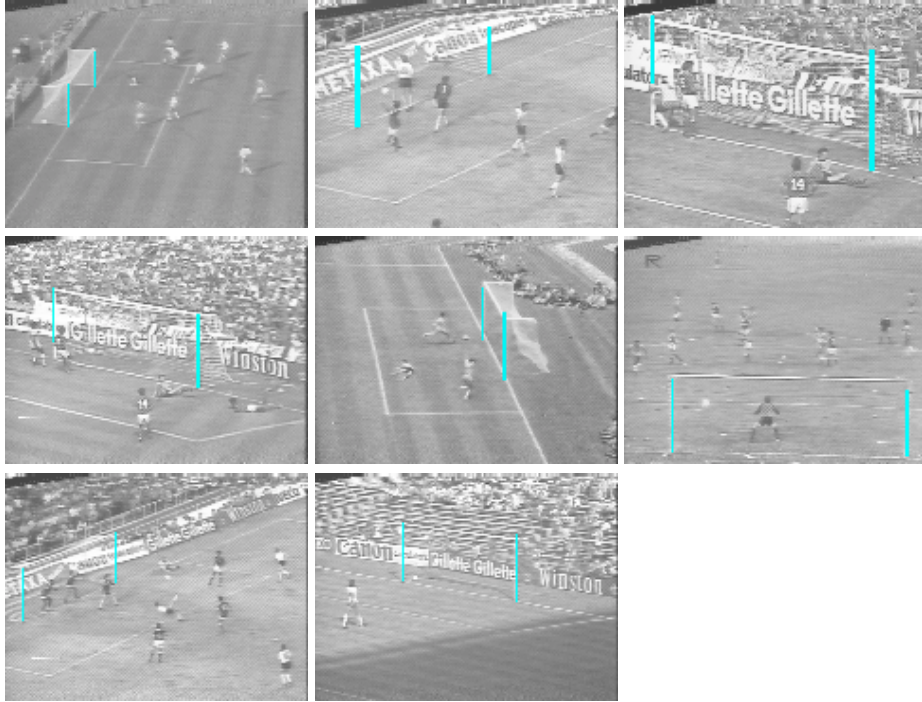


Fig. 2. Localization of vertical bars for goal post detection.

4 Querying through Interactive Learning

For the sake of reducing indexing effort, visual indices will predominantly consist of low-level image features. However, their efficiency for search reduction is limited by their generality. To remedy this, we have proposed to derive symbolic indices using *a priori* knowledge when it is available in Section 3. In this section, we introduce another approach, namely *view-based learning*. It can learn visual features in an effective way and this will allow much more increased discriminating capability. More precisely, instead of using statistical features of the

whole image, we deal with features of user-selected regions or objects. We are interested in locating key frames that contain objects of interest. The outline of this novel approach is as follows.

1. The user chooses one or more key frame(s) which contain objects of interest.
2. The user selects a feature (color, shape, texture, etc.).
3. The user draws a rectangle on the image and tells the system whether the enclosed region is interesting (positive example) or not interesting (negative example). The program learns parameters of the feature from the specified area.
4. The program applies the learned feature parameters on the image.
5. Repeat steps 3 and 4 until the user is satisfied with the result.
6. The program searches for the targets in all of the frames and reports the search result.

Currently, we have implemented the method with color only. The use of other features like shape and texture are under development.

Color segmentation algorithms based on statistical models are basically pixel classification techniques applied to some 3D color space. We adopt the HSV model (Hue, Saturation, Value) to represent and compare the color information. We perceive color as hue, saturation and value (intensity). Hue, H , corresponds to the pure color pigment, saturation S describes the purity of colors (red is highly saturated and pink is unsaturated), and value V contains the relatively bright colors. Because of the perceivable properties in HSV space, it is particularly suitable for the view-based purpose.

In order to account for the quantification effect of the color space as well as slight changes of illumination due to time or viewpoint, we regenerate the learnt color distribution (histogram). The regenerated histogram has the form of a Gaussian distribution function or the sum of such functions with the peaks at the same position(s) as in the learnt one. During learning through negative examples, the value of an entry in the histogram is set to negative whenever the entry appears in a negative example. The resulting histogram is applied to images to classify pixels. This method proves to be robust to small changes in illumination and yet possesses excellent discriminating power. Fig.3 shows some results of our experiments in searching for sofas in the images.

5 Conclusion

In this paper, we have presented our viewpoint on the video data management problem. We consider that the key for efficient retrieval relies on video data structuring and effective indexing. We have described our effort in this direction. Some techniques recently developed in our laboratory are presented. Our video partitioning algorithm (NC) outperforms other existing ones both in accuracy and in speed due to the fact that it uses both statistical and spatial information on the images without, however, having to process the entire image. We have also shown how to take advantage of domain knowledge of underlying



Fig. 3. Object searching with colors.

applications. We have used the knowledge to classify shots allowing derivation of semantic-level indices as well as semantic clustering. We have also proposed a novel approach – interactive learning, to improve the discriminating power of low-level features. The use of one of such features – color, has been studied. The proposed method learns from both positive and negative examples. Furthermore, it accounts for the quantification effect of the color space as well as the slight change of illumination due to time or viewpoint. The resulting algorithm is quite efficient in reducing search space. Experimental results have been provided. For future work, we plan to undertake a thorough investigation into efficient use of low-level image features as well as automatic scene clustering/classification. We will also continue to study interactive learning for retrieval using other features like texture and shape.

References

1. L. S. Huang, C. M. Lee, Q. Li, and W. Xiong. Dynamic object clustering with video database manipulations. 1996. Submitted to the IS&T/SPIE Conf. on Stor-

- age and Retrieval for Image and Video Databases IV, San Jose, CA, February, 1996.
2. T. Kato. Database architecture for content-based image retrieval. In *SPIE Proc. Image Storage and Retrieval Systems*, volume 1662, pages 112–123, 1992.
 3. C. M. Lee and M. C. Ip. A robust approach for camera break detection in color video sequence. In *Proc. IAPR Workshop on Machine Vision Application (MVA '94)*, pages 502–505, Kawasaki, Japan, December 1994.
 4. Q. Li and C. M. Lee. Dynamic object clustering for video database manipulations. In *Proc. IFIP 2.6 Working Conference on Visual Database Systems*, pages 125–137, Lausanne, Switzerland, March 1995.
 5. Akio Nagasaka and Yuzuru Tanaka. Automatic video indexing and full-video search for object appearances. In *IFIP Transactions A-7, Visual Database System II*, pages 113–127, North-Holland, 1992. Elsevier Science Publishers B.V. Edited by E. Knuth and L. M. Wegner.
 6. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Query images by content using color, texture and shape. In *SPIE Proc. Storage and retrieval for image and video databases*, volume 1908, pages 173–186, 1993.
 7. A. Pentland, R.W. Picard, and S. Scaroff. Photobook: Tools for content-based manipulation of image databases. In *SPIE Proc. Storage and retrieval for image and video databases II*, volume 2185, pages 34–46, 1994. Longer version available as MIT Media Lab Perceptual Computing Technical Report No.255, Nov. 1993.
 8. Y. Tonomura. Video handling based on structured information for hypermedia systems. In *Proc. ACM Int'l Conf. on Multimedia Information Systems*, pages 333–344, New York, USA, 1991. ACM Press.
 9. Wei Xiong, C. M Lee, and Man Ching Ip. Net Comparison: A Fast and Effective Method for Classifying Image Sequence. In *IS&T/SPIE Symposium on Storage and Retrieval for Image and Video Databases, San Jose, USA*, 1995.
 10. Hong Jiang Zhang and Stephen W. Smoliar Atreyi Kankanhalli. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.
 11. Hong Jiang Zhang, Yihong Gong, S.W. Smoliar, and Shuang Yeo Tan. Automatic parsing of news video. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 45–54, Boston, MA, USA, 15-19 May 1994.
 12. Hong Jiang Zhang and S. W. Smoliar. Developing power tools for video indexing and retrieval. In *Proceedings of the SPIE vol.2185 (1994), San Jose, CA, USA, 7-8 Feb. 1994.*, pages 140–149, 1994.