MPEG Video Retrieval Using Motion Information

Aiyesha Ma, Raghu Venkatram, Dingguo Chen,
Haiping Song, Ishwar K. Sethi

{ama, rvenkatr, dchen, hsong, isethi}@oakland.edu
Computer Science and Engineering Department
Oakland University
Rochester, MI 48309

1 Introduction

With the mentality "a picture is worth a thousand words" and current compression technology there continues to be an explosion of image data readily available on the internet. Much of this data is in the form of compressed video. Video data can be characterized as an object, or set of objects, that have associated with them some motion information. This motion information is the distinguishing factor between still images and video images.

Despite the additional information presented by motion, in most existing video retrieval systems, video content has traditionally been represented either by simple textual techniques or low-level features such as color, shape, transform domain features and visual summary information based on segmentation, which are not readily applicable in the development of general-purpose video data indexing and retrieval systems [1],[2]. Because of this inattention to motion as a feature, these systems lack the capability for dealing with semantic motions in video clips and temporal events within a scene are not modeled. Powerful intermediate spatiotemporal models combined with semantic meanings are needed to support an effective content-based retrieval system.

Additionally, since much video in in compressed form, by operating in the compressed domain, rather than the uncompressed, the video stream does not need to be fully decompressed. This procedure of operating in the compressed domain directly was popularized by Sethi et. al. in papers such as [3], [4], and [5], among others, in which processing methods were presented for cut detection, feature extraction, and edge detection.

In this paper we propose a general framework video retrieval system that utilizes motion information. Within this framework, we present individual components, such as cut detection and motion segmentation, which operate on a partially decoded stream, rather than the uncompressed video. We show the linking of these components into a fully automated indexing and retrieval system.

2 System Framework

In this design, we break the system into three parts: storage, retrieval, and feedback. Each of these subsystems will be discussed further below. The overall system design is illustrated in Figure 2.

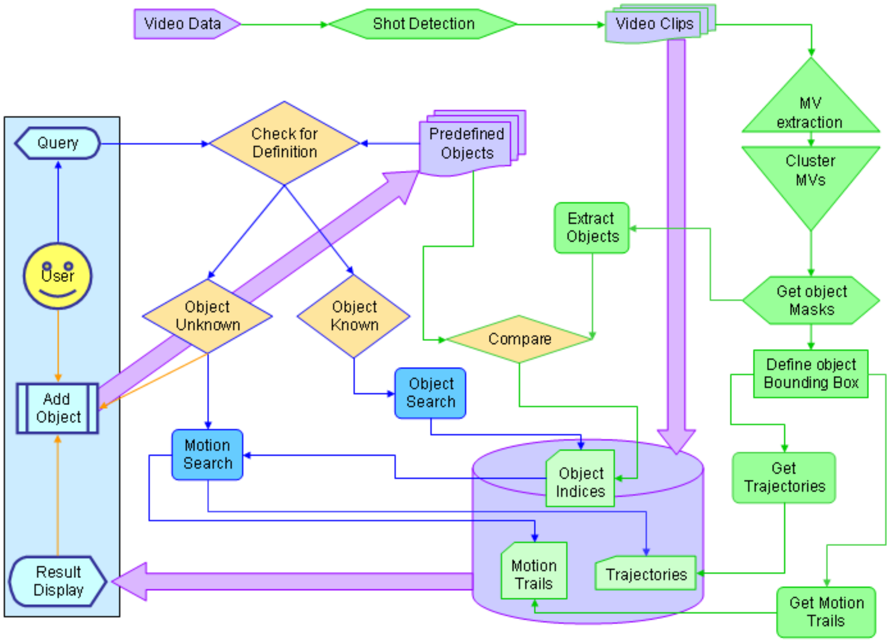


Figure 1: Overall Video Indexing and Retrieval System

2.1 Preprocessing and Storage

The first section of the design pertains to indexing the video. Shot detection is first performed on the raw video. From each clip then, motion information is extracted. This motion information is used to perform object segmentation. Once the objects in the video have been segmented, they are extracted and object recognition is performed by comparing to an initial database of objects. Also, from the segmented objects, motion trajectories and motion trails are acquired from the video sequence. Therefore the video is indexed in two ways: object descriptions, if the object is already in the database, and object motion, from the trajectories and trails.

2.2 Retrieval

In the second portion of the design, the user specifies a query and the relevant results are returned. Since the video is indexed in two methods, the user is able to query on those two features. This method allows queries such as “a car moving left to right,” “a train,” or “an object moving in a spiral.” This methodology can retrieve objects based on motion, even if the object desired does not yet have a prototype. At this point we anticipate a combination of query by example for motion and query by text for objects, but hope to eventually expand the system to allow more freedom in the query specification.

2.3 Feedback

The third section relates to adding prototypes for as of yet unknown objects. So if a user queries for “a cat moving from bottom to top” (jumping), and the cat object is unknown, after results based solely on motion are retrieved, the user can specify which objects in which video clips were cats. These objects will be stored in the database, and the next time the cat object is specified as a query condition the system will be able to retrieve cat objects, rather than all objects.

3 System Components

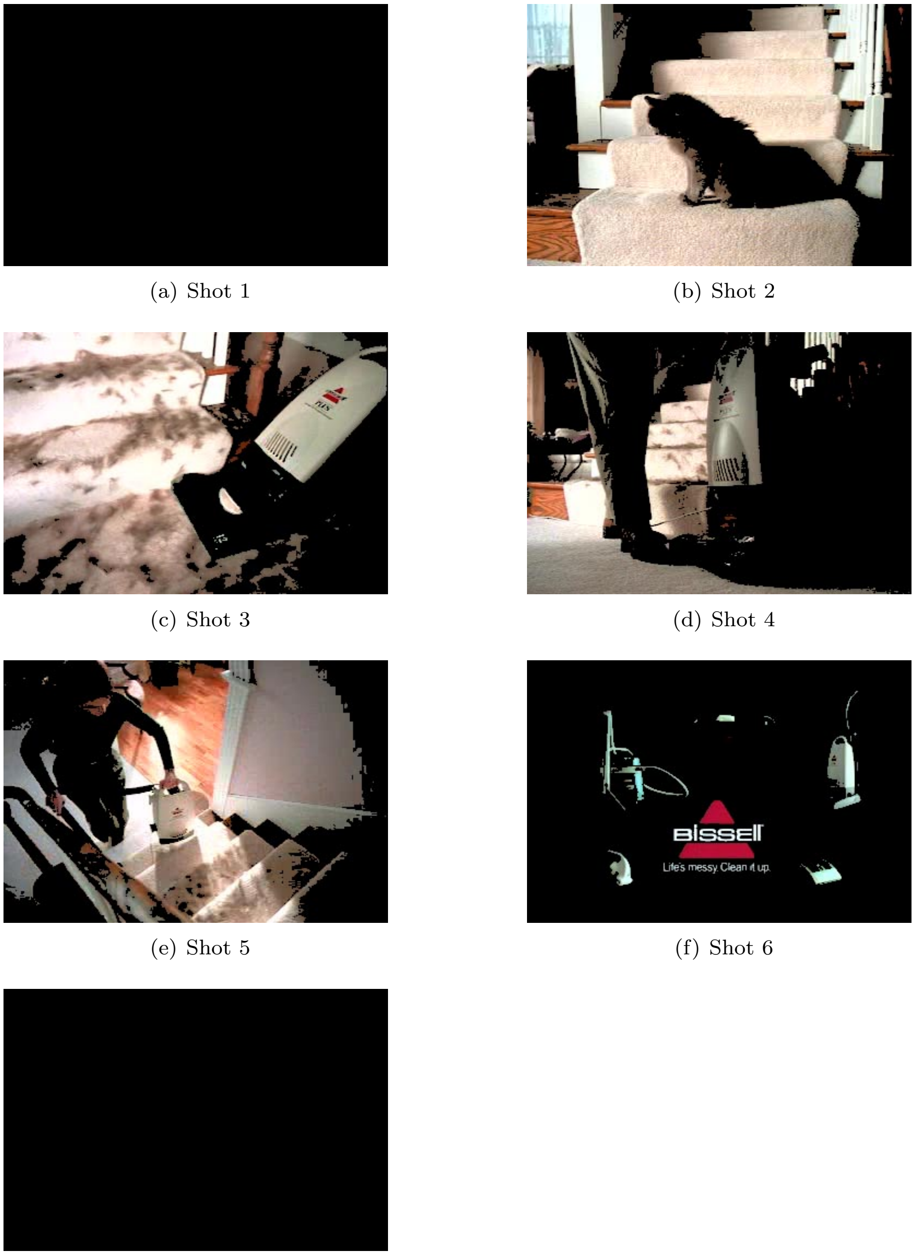
In this section we discuss some of the components of the system. At this point, three components are performed during the indexing stage: shot detection, motion segmentation, and object comparison. The last component, spatiotemporal representation, pertains to motion features and is relevant in both the indexing and retrieval stages. The first two components, shot detection and motion segmentation, operate on the MPEG motion compensation vectors. This allows partial decoding of the MPEG stream rather than full decompression.

3.1 Shot Detection

There are numerous shot detection algorithms published in the literature, and of these there are several that operate on the MPEG streams [6]. A couple methods were tested ([7],[8]). [7] uses the ratio between forward or backward predicted blocks and bidirectional blocks, thereby operating on P and B frames. [8] use a histogram based method that operates on the I frames.

3.2 Motion Segmentation

By assuming that an object has some consistency of motion, that object can be segmented by analyzing the motion vectors between frames. Rather than computing motion vectors or optical flow, we use the precomputed motion compensation vectors encoded in the video stream. The motion segmentation algorithm we present differs from existing methods that operate in the compressed domain([9],[10],[11],[12],[13]) in two ways. First, we plan to use an agglomeration



(a) Shot 1

(b) Shot 2

(c) Shot 3

(d) Shot 4

(e) Shot 5

(f) Shot 6

(g) Shot 7

Figure 2: First frame of each shot in a Bissell commercial using [7]

of motion compensation vectors over several frames to moderate inconsistencies in the field. Although [10] presents a technique to this effect, ours aims to both average the vectors and to provide a denser field at the same time, rather than in two steps. Second, we use a discontinuity detection approach based on [14] rather than the more common vector similarity approach. Our hope is that this approach will be less constraining when handling deformable objects.

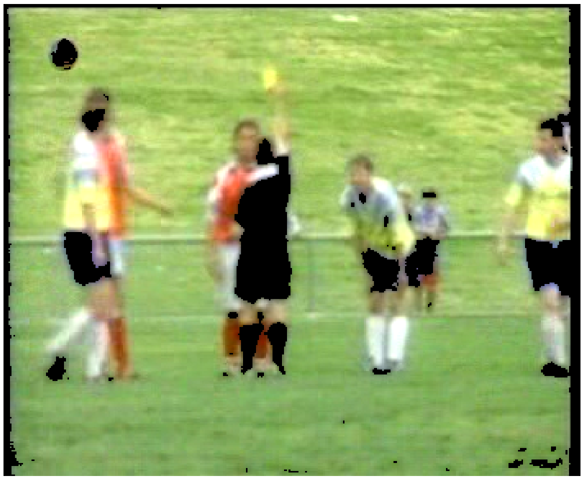


Figure 3: Video Frame: Leftmost two soccer players are walking to the right.

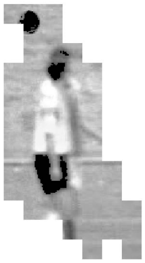


Figure 4: Result from Motion Compensation Vectors, no agglomeration of frames

3.3 Spatiotemporal Representation

We look at two spatiotemporal representations, trajectories and trails [15]. In the trail-based model, an object is tracked by highlighting the area covered by the object's bounding box. The result over several frames is the object's trail image. In the trajectory method, motion representation is more precise; only the center point of the object is used to characterize the position. This method works better when dealing with an object that occupies a large region in the frame. Additionally, the trail model is independent of temporal characteristics; it works well in cases where the object speed is irrelevant.

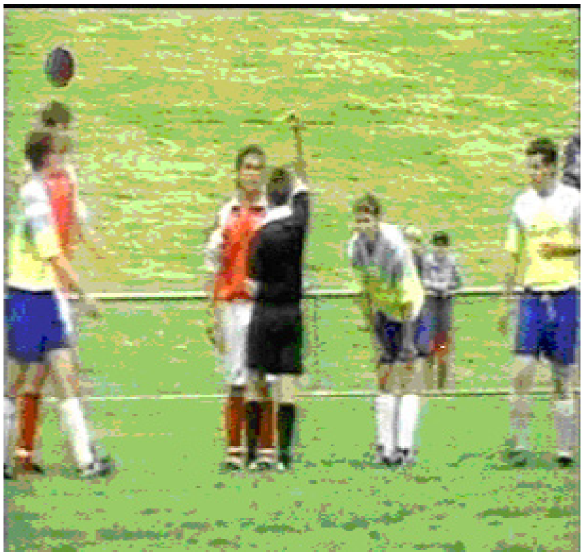


Figure 5: Video Frame: Leftmost two soccer players are walking to the right.

References

[1] P. Bouthemy, "Motion characterization from temporal co-occurrence of local motion-based measures for video indexing," in *Proceedings of International Conference on Pattern Recognition*, August 1998.

[2] D. DeMenthon and D. Doermann, "Video retrieval using spatio-temporal descriptors," in *Proceedings of the eleventh ACM international conference on Multimedia*. ACM Press, 2003, pp. 508–517.

[3] B. Shen and I. K. Sethi, "Direct feature extraction from compressed images," in *Proc. IS&T/SPIE Conf. Storage and Retrieval for Image and Video Databases*, January 1996.

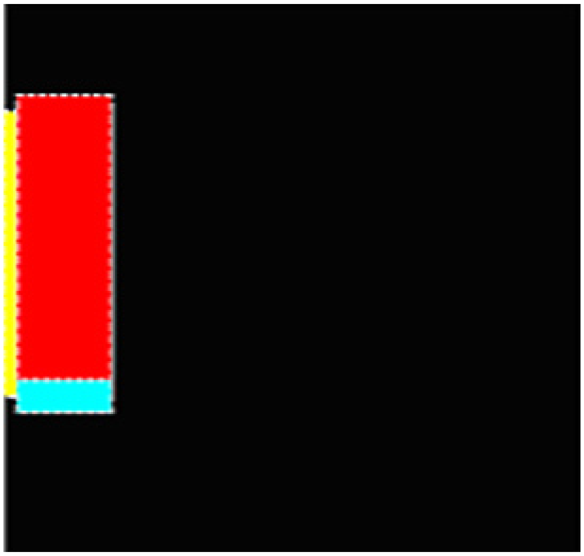


Figure 6: Trail Image Result

Figure 7: Trajectory Representation

[4] N. Patel and I. K. Sethi, "Compressed video processing for cut detection," in *IEE Proceedings - Vision, Image and Signal Processing*, vol. 143, no. 3, October 1996, pp. 315–323.

[5] B. Shen, D. Li, and I. K. Sethi, "Cut detection via edge extraction in

compressed video," in *Visual'97*, December 1997, pp. 149–156.

[6] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey," *SP:IC*, vol. 16, no. 5, pp. 477–500, January 2001.

[7] Y. Haoran, D. Rajan, and C. Tien, "A unified approach to detection of shot boundaries and subshots in compressed video," in *Proceedings International Conf. on Image Processing*, Barcelona, Spain, September 2003.

[8] N. V. Patel and I. K. Sethi, "Compressed video processing for cut detection," in *IEE Proc: Video, Image, and Signal Processing*, 1996.

[9] E. Ardizzone, M. La Cascia, A. Avanzato, and A. Bruna, "Video indexing using mpeg motion compensation vectors," in *Proc. of IEEE International Conference on Multimedia Computing and Systems*, 1999.

[10] R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan, "Video object segmentation: A compressed domain approach," *IEEE Trans. on Circuits and System for Video Technology*, vol. 14, no. 4, April 2004.

[11] F. Porikli, "Real-time video object segmentation for mpeg encoded video sequences," Mitsubishi Electric Research Laboratories [Cambridge, USA], Tech. Rep., March 2004, tR-2004-011.

[12] R. Wang, H. Zhang, and Y. Zhang, "A confidence measure based moving object extraction system built for compressed domain," in *IEEE International Symposium on Cicuits and Systems*, 2000.

[13] C. Chiou and C. Lin, "Compressed-domain video object extraction for content based video retrieval," in *16th IPPR Conference on Computer Vision, Graphics, and Image Processing*, August 2003.

[14] I. K. Sethi, "A general scheme for discontinuity detection," in *Proceedings of International Conference on Pattern Recognition*, vol. 1, Montreal, Canada, July 1984.

[15] S. Dagtas, W. Al-Khatib, A. Ghafoor, and R. Kashyap, "Models for motion-based video indexing and retrieval," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 88 – 101, January 2000.