COMPRESSED DOMAIN OBJECT TRACKING FOR AUTOMATIC INDEXING OF OBJECTS IN MPEG HOME VIDEO

Radhakrishna Achanta, Mohan Kankanhalli*, Philippe Mulhem***

School of Computing*/IPAL-CNRS**, National University of Singapore, Singapore 117543
E-mail: {achanta, mohan, mulhem}@comp.nus.edu.sg

ABSTRACT

Object tracking is of utmost importance for automatic indexing of video content. This work presents an object tracker that operates directly on MPEG compressed data. Motion vectors and Discrete Cosine Transform (DCT) coefficients directly available from the compressed video stream are exploited for the purpose of tracking. Tracking proceeds in two steps: motion vector based tracking in P and B frames within the Groups of Pictures (GOP's), and object identification in I frames. Colour, which is one of the strongest cues for tracking is used for the identification step. Such a system offers speed, simplicity and robustness against occlusion and camera motion, with good intra-shot tracking for shots in excess of 500 frames, as shown in the experimental results.

1. INTRODUCTION

The need to manage video data generated by amateur movie-makers in a fast and easy manner has led to research in recent years into the development of video indexing systems. Indexing of video is done mainly by segmenting it into its various constituent shots, or in a more sophisticated manner, into context-based strata ([6]). In addition however, the content of the shots (objects and events of interest) also needs to be indexed. Objects can only be indexed with the help of tools that can track the object through the sequence of frames. The object tracker presented here does that, serving as the annotation tool for the DIVA project (http://diva.comp.nus.edu.sg:8080). An effective tracker should be able to track objects until the object moves out of the camera capture range, or until it no longer exists in the video (as a result of shot change) despite short occlusions.

2. RELATED WORK

Most of the object trackers deal with the decoded pixel data. Some of these are not suitable for real-time applications as they are computationally intensive. Also most video data is available in the compressed form. So some research has been directed at tracking objects in compressed data. ([8]). Among the cues available ([2]) in the compressed domain data, only motion vectors are used in [4] and [5] for tracking. [9] treats object tracking as a macroblock-linking problem. Motion vectors are used here too, though, in addition, DCT AC coefficient energies of intracoded macroblocks are also used. The work presented in [11] and [12] aims at detection of moving objects, but uses tracking based on motion vectors to segment the moving object(s). In [3] instead of motion vectors, DC difference images and directed graphs are used to track objects. There are two main drawbacks of motion vectors w.r.t tracking: one, they are for coding purposes only, and two, they are absent in intracoded macroblocks and at I frames, where a new GOP starts. [4] and [5] overcome the latter problem of crossing GOP boundaries by performing block matching between the last P frame of the current GOP and the I frame of the following GOP. [9] avoids this potentially computation intensive task by relying on reversed motion vectors of the last P frame of a GOP. [11] and [12] rely on the motion vectors of P frames on either side of the I frame to overcome this problem. These trackers also differ in the way the user interacts with them to define the initial image region belonging to the object. [4] and [5] require the user to select the object by marking each macroblock constituting it, while [3] and [9] perform automatic segmentation.

3. MOTION AND COLOUR BASED TRACKING

Tracking proceeds in two ways after user selects the object(s) by drawing a rectangle around it(them). Within a GOP, motion based tracking, using forward motion vectors of both P and B frames, is performed (unlike [4], [5], [9], [11] and [12] where only P frame motion vectors are used). The advantage is that the object is tracked in both P and B frames of the GOP. GOP boundaries are quite comfortably crossed by using backward motion vectors of the last B frame of each GOP. This is a more reliable method than use of reversed P frame motion vectors ([9] and faster than block matching ([4] and [5]), besides being done completely in the compressed domain.

0-7803-7304-9/02/$17.00

However, due to the limitations of motion vectors, in the absence of any verification for the object being tracked, errors can get introduced in motion vector based tracking. These prove to be cumulative in nature. So at each I frame, colour based tracking is performed, which involves identifying the best image area that matches the original object marked out by the user. For this purpose chrominance DCT values of Cr and Cb in the I frames are used, unlike in [3] and [9] where Y DCT coefficients are used.

3.1 Motion vector based *intra*-GOP tracking:

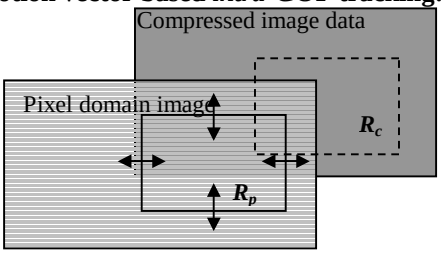

Figure 3.1: Compressed domain equivalent image area

First the compressed domain equivalent area $R_c[(p_1, q_1), (p_2, q_2)]$ of the user chosen pixel image area $R_p[(x_1, y_1), (x_2, y_2)]$ is determined as the best fitting rectangle with macroblock (MB) coordinates given by eq. 3.1.

$$X_c = Quotient\ (\ X_p / 16\ ) + RoundOff \qquad (3.1)$$

for $X_c = p_1, q_1, p_2, q_2$, when $X_p = x_1, y_1, x_2, y_2$ respectively, where $RoundOff$ = 1 if $Remainder\ (\ X_p / 16\ ) > 8$, and 0 otherwise.

$$R_c \leftarrow ShrinkFit\ (\ R_c\ ) \qquad (3.2)$$
$$R_p \leftarrow Translate\ (R_p\ , Mode\ (\ MV\ (R_c)\ )\ ) \qquad (3.3)$$

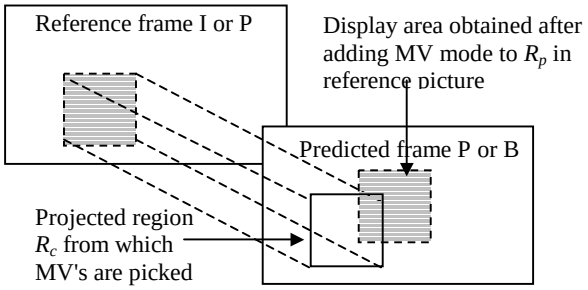

Figure 3.2: Use of forward motion vectors

After this, region R_c of the reference frames (namely, starting I frame of each GOP, and the intermediate P frames), is projected on to the predicted frames (P and B, as the case may be) to get the new R_c (eq. 3.2). *ShrinkFit* simply crops the projected region by one MB width on all sides. *ShrinkFit* is only done if the predicted frame is a B frame. This is followed by obtaining new pixel image display area R_p using eq. 3.3 (fig. 3.2). Here, operator *Translate* (*R,v*) causes a rectangle *R* to shift by a vector *v*, obtained as the mode value of the set of motion vectors in the rectangular area specified in *MV* ( ).

Having found the new R_p, it is now relied upon to find the new R_c again using eq. 3.1. This way tracking within GOP continues until the next I frame is reached, where R_p is comfortably calculated using backward motion vector mode of the most recent B frame. *ShrinkFit* operation is performed (to reduce noisy motion vectors, related to background or of non-uniformly moving parts of a non-rigid body, or otherwise) for B frames because, being further away from the I frame as compared to the P frames, relatively more displacement is likely to occur in them (assuming safely that motion taking place within a span of three or less frames is not more than sixteen pixels). The use of mode of motion vectors further serves to exclude various noisy motion vectors from consideration. It allows treating the set of macroblocks making up the object as a single unit, since tracking them separately might exclude some of those which constitute the object ([4] and [5]). It also makes grouping of motion vectors by thresholding ([11] and [12]) unnecessary. Also, as only regions of interest are tracked, global motion compensation is not necessary, and camera motion gets automatically dealt with (as in the *Cactus*, *Pen and Cup* experiments, fig. 4.2).

3.2 Colour based tracking *inter*-GOP tracking:

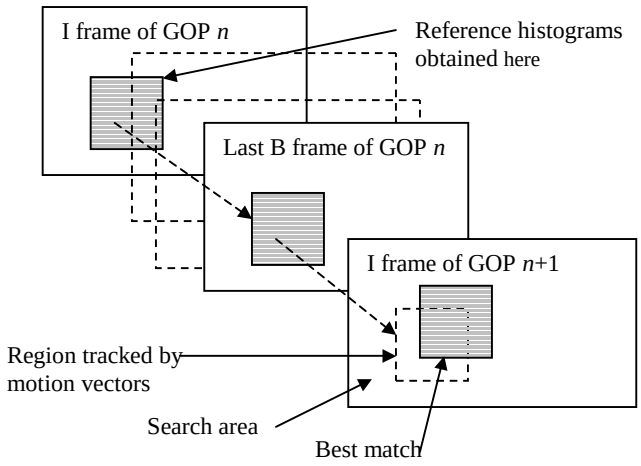

Figure 3.3: Colour based tracking

Home videos are expected to be shot in colour, so use of it in tracking is very advantageous. Colour is a robust cue ([10]), and the sparing use of it in the compressed domain trackers developed so far is surprising. The MPEG standard uses the YCrCb space. Only the Cr and Cb DCT values are used in our work. Since each macroblock in an I frame is treated as made of four blocks of 8x8 pixels, and compressed in the JPEG fashion, the compressed data equivalent rectangle R_c, is different in this case, with the

smallest unit being a block rather than an MB. Coordinates $[(p_1, q_1), (p_2, q_2)]$ are obtained here as:

$$X_i = Quotient\ (\ X_d / 8\ ) + RoundOff \quad (3.3)$$

for $X_c = p_1, q_1, p_2, q_2$, when $X_p = x_1, y_1, x_2, y_2$ respectively, where, *RoundOff* = 1 if *Remainder* (X_p / 8) > 4, and 0 otherwise. At the I frame, the DC value and first eight AC values (since higher frequency AC values beyond the first eight are small or zero usually) are extracted for both Cr and Cb. Histograms with sixteen bins (the number was arrived at after experimentation) are made for each of the eight DCT values for both Cr and Cb. These form the *reference histograms*. Now at every following I frame the best match for this initial object region is found within a search region around the area tracked by the motion vector tracking (figure 3.3). For this, a window of the size of the object is moved one block (eight pixels) at a time, and the *candidate histograms* like the ones made for the reference area are made. Finding the best match involves finding the minimum value of *DiffSum*:

$$DiffSum = \sum_{n\ in\ [1,9]}\ Wt[n]\ (\ |HDiffCr| + |HDiffCb|\ ) \quad (3.4)$$

where HDiffCr and HDiffCb are the histogram bin differences for Cr and Cb values respectively; $Wt[n]$ is the set of weights used {.4, .1, .1, .1, .1, .05, .05, .05, .05} for one DC plus eight AC. The weights are chosen in such a way that the DC value, which most prominently conveys colour information is given maximum weightage, followed by lower frequency AC values, which convey coarse texture or shape information ([1]), in decreasing order of importance in the matching algorithm. Though a lower weightage is given to the AC values than the DC value as in eq. 3.4, it is noteworthy that the AC information proves very useful in distinguishing between two objects with the same colour, but different shapes or textures. This method of identification acts as validation for the motion vector based tracking and helps prevent error accumulation. Size-change of objects (because of camera zoom or otherwise) is also taken care of by performing similar searches with the window sizes smaller and larger on all sides by one block width, w.r.t. the starting window size. This is because in case of a zoom-in, the minimum value of *DiffSum* of a larger window is lesser than that for a smaller window, and vice versa, in case of a zoom-out. Similarly, shape changes are also taken care of by performing the search once with reduced width and reduced height windows. The effectiveness of use of simple histograms, for the AC values for image matching, was proven more effective compared to other methods used for image retrieval in the compressed domain ([7]).

4. EXPERIMENTS AND RESULTS

The tracker was used to track objects in standard test video clips (like *Flower Garden*, *Mobile and Calendar*), and MPEG-7 test-set clips of home videos. Results of tracking shown are in the form of distances of centers of manually drawn (thin lines) and tracked image areas (bold lines) from image origin, in fig 4.1, 4.2, 4.3. For the *Cactus, Pen and Cup* video, the two pens are tracked across a shot change, with camera zoom and pan being the only motion (fig. 4.1). The MPEG-7 test clip showing a girl moving rapidly across a stage is tracked despite partial occlusions, past 500 frames (fig. 4.2). Fig. 4.3 shows results of tracking of a girl in a car in another MPEG-7 test clip that is quite shaky, with the object changing in size and shape. The main causes of errors were the blocky nature of compressed data in I frames and unreliability of motion vectors. These errors were more pronounced when the object was small (or if the video frame itself was small, thus making the objects small, as in case of the MPEG-7 test clips), or the object was non-rigid or was changing a lot in shape and size (girl in car clip).

5. CONCLUSION

Motion vectors present in compressed data are purely for coding purposes, and their use for tracking is not always feasible. Motion vector based tracking errors are corrected by colour based identification done in the compressed domain. Camera pans can be handled with motion vectors while zoom required the use of colour information. Objects can rarely be tracked sharply along their exact boundaries in the compressed domain due to blocky nature of data; they can only be tracked along the MB boundaries unless special approaches are adopted (like depending on R_p to decide R_c as in this paper) or reconstruction of DCT values with one or two pixel shifts, which is being considered for future work. The size of the initial enclosing rectangle affects tracking results; it should ideally enclose as little of the background as possible for good results. Also objects smaller than 2x2 MB size are hard to track, though they can be tracked with some degree of success. Our tracker is more suitable for the specific case of home videos, where shots tend to be long, special effects are fewer and objects of interest tend to occupy large image regions. Future work entails incorporating better object identification methods, so that object may be tracked across shots.

6. REFERENCES

[1] B. Shen, I. K. Sethi, “Direct feature extraction from compressed images” SPIE vol. 2670, Storage and Retrieval for Image and Video Database IV, 1996.

[2] C. S. Won, D. K. Park, S. J. Yoo, “Extracting image features from MPEG-2 compressed stream”, SPIE vol. 3312, pp. 426 – 432, 1997.

[3] H. Chen, Y, Zhan and F. Qi, “Rapid object tracking on compressed video”, *Proc. of Second IEEE Pacific Rim*

Conference on Multimedia, pp. 1066 – 1071, October 2001.
[4] L. Favalli, A Mecocci, and F. Moshetti, "Object tracking and hypermedia links creation in MPEG-2 digital video sequences", *Proc. Of IEEE Int'l Conf. On Circuits and Systems*, pp. IV-261 – 264, 1998.
[5] L. Favalli, A Mecocci, and F. Moshetti, "Object tracking for retrieval applications in MPEG-2", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 3, April 2000.
[6] M. S. Kankanhalli, T. S. Chua, "Video modeling using strata-based annotation" *IEEE Multimedia*, pp. 68 – 74, January – March 2000.
[7] R. S. V. Achanta, "Object tracking in compressed domain MPEG video" *M.Sc. thesis* (in preparation), Dept. of Computer Science, National University of Singapore, 2002.
[8] S. F. Chang, "Compositing and manipulation of video signals for multimedia network video services (parts I-IV), *Ph.D. dissertation*, Dept. EECS, Univ. of California Berkeley 1993.
[9] W. N. Lie and R. L. Chen, "Tracking moving objects in MPEG-compressed videos", *IEEE Intl. Conf. on Multimedia and Expo*, Tokyo, Aug 2001.
[10] Y. B. Lee, B. J. You, S. W. Lee, "A real-time color-based object tracking robust to irregular illumination variation", IEEE 2001
[11] Y. Nakajima, A Yoneyama, H. Yanagihara, M. Sugano, "Moving object detection from MPEG coded data", *SPIE Visual Communications and Image Processing, Vol. 3309*, pp. 988 – 996, 1997.
[12] Y. Nakajima, A Yoneyama, H. Yanagihara and M. Sugano, "Moving object detection and identification from MPEG coded data." *Intl. Conf on Image Processing*, 1999.

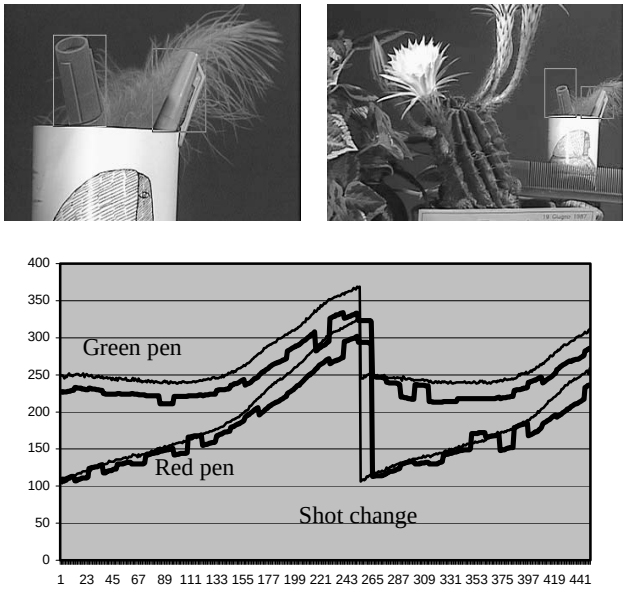

Figure 4.1: *Cactus, pen and cup* video objects and results

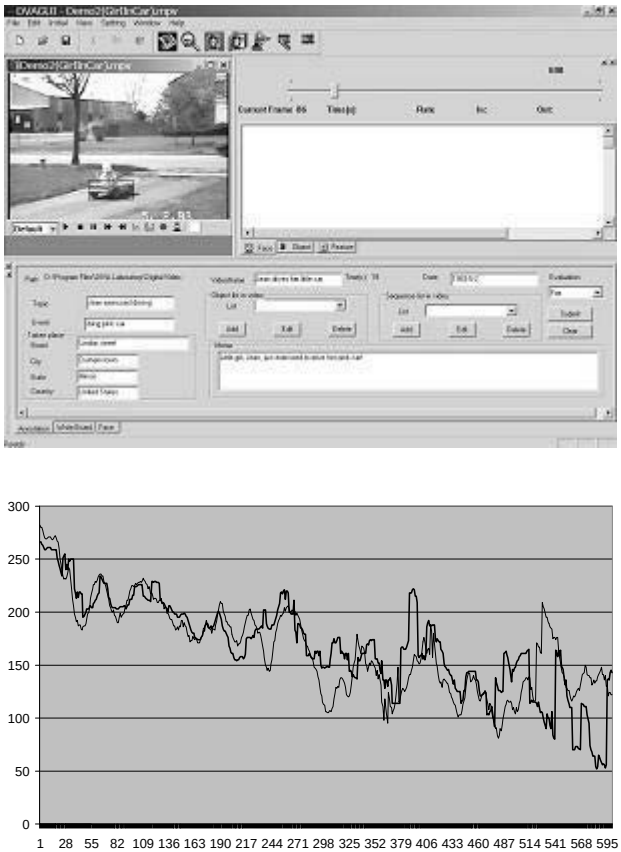

Figure 4.2: Screenshot of the video album interface, with the tracking feature on, showing object (pink car) enclosed in a rectangle in an MPEG 7 test data video clip along with the graph for the tracking results.

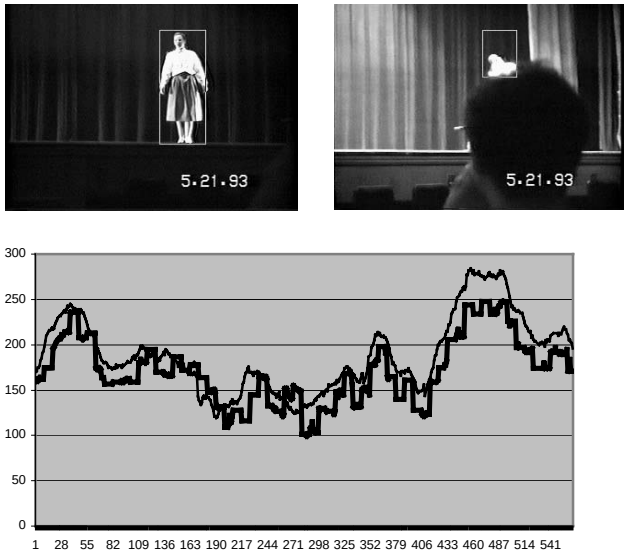

Figure 4.3: Girl dancing across stage and tracking results