VIDEO OBJECT TRACKING IN THE COMPRESSED DOMAIN USING BLOB TRACKING ALGORITHM

S.Kirubakaran[1] M.Vadivelan[2]

[1]PG Scholar, Dept. of Electronics and Communication

[2]ASSISTANT PROFESSOR, Dept. of Electronics and Communication

Dr. S.J.S PAUL MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY [1, 2]

(PONDICHERRY UNIVERSITY)

Kiruba2149@gmail.com

Abstract- In this project video sequences compressed using SPIHT algorithm and moving object in compressed video is identify using hidden markov model and object tracking using Blob tracking algorithm. This project works in the compressed domain and uses only the motion vectors (MVs) and block coding modes from the compressed bit stream to perform tracking. First, the MVs are preprocessed through intra coded block motion approximation and global motion compensation. At each frame, the decision of whether a particular block belongs to the object being tracked is made on the Blob tracking algorithm, which is updated from frame to frame in order to follow the changes in the object's motion.

Index Terms-compressed-domain video object tracking, SPHIT, blob tracking, motion vectors (mv).

I. INTRODUCTION

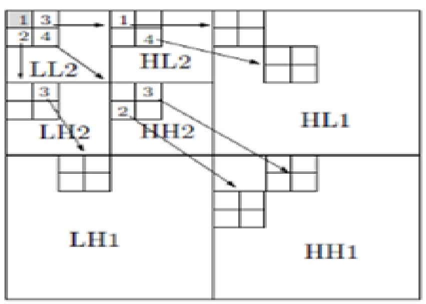
VIDEO-BASED object tracking is one of the challenging problems with a variety of applications, such as video surveillance, video indexing and retrieval, video editing communication, compression, etc. There are two major groups of approaches to segment and track moving objects in a video sequence, distinguished by the domain in which they operate: pixel domain and compressed domain. The former have the potential for higher accuracy, but also require higher computational complexity. In addition, since most video content nowadays only available in the compressed form, decoding is required in order to generate pixel domain information. On the other hand, "compressed-domain" approaches make use of the data from the compressed video bit stream, such as motion vectors. (MVs), block coding modes, motion-compensated prediction residuals or their transform coefficients, etc. In the literature, however, even methods that fully decode the bit stream are sometimes referred to as "compressed-domain" methods, so long as pixel values are not recovered completely for all the frames. The lack of full pixel information often leads to lower accuracy, but the main advantage of compressed domain methods in practical applications is their generally lower computational cost. This is due to the fact that part of decoding can be avoided, a smaller amount of data needs to be processed compared to pixel-domain methods, and some of the information produced during encoding (e.g., MVs) can be reused. Therefore, compressed-domain methods are thought to be more suitable for real-time applications, although some of them are still characterized by high complexity.

A. Prior Work on Compressed-Domain Segmentation and Tracking

An iterative scheme that combines Global Motion Estimation (GME) and macro block (MB) [12] [9] rejection is exploited in to identify moving object blocks, which are then tracked via MB-level tracking. This scheme, however, is not able to segment and track the moving objects whose motion is not sufficiently distinct from the background motion. Foreground objects are identified by applying the background subtraction technique followed by temporal filtering to remove the noise. Afterwards, motion segmentation is performed by Timed Motion History Images approach, and finally, the trajectory is estimated by object correspondence processing. Mean shift clustering is used in to segment moving objects from MVs and partition size in SPHIT bit stream presented an algorithm to track multiple moving objects in SPIHT[8] compressed video based on probabilistic spatiotemporal MB filtering and partial decoding. Their work assumes stationary background and relatively slow-moving objects. MVs over time in order to obtain the salient MVs.

B. Overview of the Proposed Method

In this proposed system presents a moving object in a Video sequence compressed using SPIHT algorithm is given in [11], Video tracking is the process of locating a moving object in time using a camera. An algorithm analyses the video frames and outputs the location of moving targets within the video frame. The role of the tracking algorithm is to analyses the video frames in order to estimate the motion parameters.



These parameters characterize the location of the target. The data from the compressed stream used in the proposed method are the motion vectors and block coding modes. As a result, the proposed method has a fairly low processing time, yet still provides high accuracy. The moving vectors are pre-processed to enhance data images for further processing. Kalman filter produce a statistically optimal estimate of system state..

Motion vectors

In video compression, a motion vector [5]is the key element in the motion estimation process. It is used to represent a macro block in a picture based on the position of this macro block (or a similar one) in another picture, called the reference picture. The SPIHT standard defines motion vector as: motion vector: A two-dimensional vector used for inter prediction that provides an offset from the coordinates in the decoded picture to the coordinates in a reference picture .we consider the frame to be divided into small blocks (4 × 4 in our experiments). Object blocks will be labeled 1, non object blocks 0.

We want to infer the block labels ?t € {0, 1} in frame t, given the labels ?t?1 in frame t ? 1, and the observed motion information t = (?t , ot).

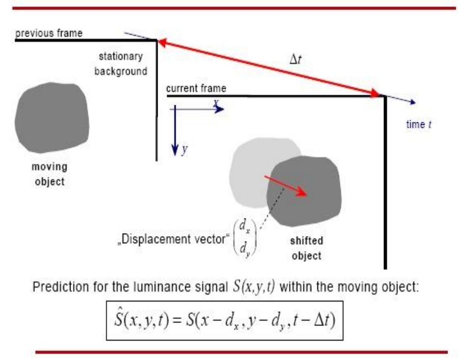


Fig. no 1. Motion-compensated prediction

Block coding Modes

Our tracking algorithm makes use of two types of information from the SPIHT [4] [9] compressed bit stream: block coding mode (partition) information and MVs. Texture data does not need to be decoded in the proposed method. SPIHT defines four basic MB modes: 16 × 16, 16 × 8, 8 × 16, and 8 × 8, where the 8 × 8 mode can be further split into 8 × 4, 4 × 8, and 4 × 4 modes. Since the smallest coding mode (partition) in spiht is 4 × 4, in order to have a

Uniformly sampled MV field, we map all MVs to 4 × 4 blocks .This is straightforward in inter-coded blocks, as well as SKIP blocks where the MV is simply set to zero. However, interpreting the motion in the intra-coded blocks is more involved.

Pre processing

Kalman filter

The Kalman filter,[9] [14] also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. More formally, the Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state The algorithm works in a two-step process.

A. Time Update

Discrete-time Kalman filters begin each iteration by predicting the process's state using a linear dynamics model.
State Prediction: For each time step k, a Kalman filter first makes a prediction ^x□k of the state at this time step:

$$x\^k = Axk - 1 + Buk$$

where ^xk□1 is a vector representing process state at time k-1 and A is a process transition matrix. uk is a control vector at time k, which accounts for the action that the robot takes in response to state xk; B converts the control vector uk into state space. [2] In our model of moving objects on 2D camera images, state is a 4-dimensional vector [x; y; dx; dy], where x and y represent the coordinates of the object's center, and dx and dy represent its velocity. The transition matrix is thus simply.

Hidden Markov Model

Hidden Markov [8] [12]models are widely used in science, engineering and many other areas (speech recognition, optical character recognition, machine translation, bioinformatics, computer vision, finance and economics, and in social science). The Hidden Markov Model (HMM) is a variant of a finite state machine having a set of hidden states, Q, an output alphabet (observations), O, transition probabilities, A, output (emission) probabilities, B, and initial state probabilities The current state is not observable. Instead, each state produces an output with a certain probability (B). Usually the states, Q, and outputs, O, are understood, so an HMM is said to be a triple, (A, B, ?).In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.

Blob tracking

Blobs are detected and the coordinate values of the blobs are obtained. In order to obtain the information to recognize gesture, we must track the movement and status of each blob. Blob tracking is a key procedure of gesture recognition. Gesture in multi-touch is a form of nonverbal communication produced by parts of the body. In multi-touch application, gesture is composed of movements of one or two even more blobs and the touch events. Touch event means the actions of fingers, such as the finger's tapping down and lifting up. The tasks of the blob tracking procedure include the recognition of the corresponding blobs, the detection of touch events, and the transferring from low-level driver to high level applications of events.

Corresponding blob (see Figure 5) is the same finger's blob but in a series of images. To get recognition of the corresponding blobs and detect the touch events are difficult because what can be used are only the discrete coordinates of blobs. We must identify the corresponding blobs from more than one blob in two sequential images.

Corresponding blob analysis

Blob tracking is used to detect the finger's actions and movements. In order to get recognition of the corresponding blobs from many blobs in a series of images, we adopt Minimum Distance First algorithm (MDF). We assume that there are M blobs in the first image and N blobs in the second image. In the first image, all blobs' coordinate values are storied in two array variables (xold*i* ,yold*i*), where 1 ≤□i ≤□M . In the second image, blobs' coordinate values are storied in two array variables (xnew*j* ,ynew*j*), where 1 ≤□j ≤□N . A dynamic two-dimension array D(i,j) is created to store the distance values of each blob. The distance between two blobs can be calculated through equation

$$D(i,j) = \sqrt{(xold - xnew)2} + \sqrt{(yold - ynew)^2}$$

After distance calculation of blobs in two images, the blobs with minimum distance are usually treated as the corresponding blobs. And then, a recursive method is used to qualify the validation of the corresponding blobs.

In all the tests presented here, the sequences were processed at original resolution of 320x240 pixels, with the same parameters for segmentation and tracking (7 pyramid levels). As can be expected, the algorithm gives state-of-the-art real-time performance for low-level blob tracking in scenarios involving large, slow blobs (subject of course to the quality of the segmentation results). Figure 4 shows two frames from one of the PETS 2002 test sequences [2], with tracked blobs and their trajectories (maximum condense path on the tracking graph) over a few frames.

Since the strength of the algorithm is its ability to reliably track not only large slow blobs, but also small fast blobs, it was also tested on more challenging sports scenarios: professional tennis and racquetball. The blobs corresponding not only to the players, but also to the ball, are automatically segmented and tracked, in real-time. Figure 5 is a frame from a tennis video with tracked blobs and

their trajectories (maximum condense path on the tracking graph) over a few frames. The players are relatively slow, so the corresponding displayed trajectories are often very short. The history borer length (5 frames) for the trajectory was set to make the ball most visible without cluttering the scene unnecessarily. The players and the ball are correctly tracked. Occasionally, the ball can be lost. Two cases can be distinguished. First, the ball might not be detected (no blob is produced by the segmentation) or the ball blob is merged with another blob. This is a segmentation limitation. Second, the relationship between two ball blobs in consecutive frames might not be recognized as the most likely at this low semantic level.
in our system. That is, we do not track object parts,

$$\mathrm{Dist}(cp, ci) < T$$

where Dist() function is defined as the Euclidean distance between two points,
which is:

$$\mathrm{Dist}(\mathrm{cp}, \mathrm{ci}) = \sqrt{(\mathrm{xcp} - \mathrm{xci})^2}_{+}\sqrt{(ycp - yci)^2}$$

Since every two objects that are close to each other within a threshold are not necessarily a successful match, in the next step we check the similarity of these two objects to improve correct matching. The criterion for similarity comparison is the size ratio of the objects. Again, this check is motivated by the fact that objects do not grow or shrink too much between consecutive frames. Thus, two objects are classified as similar if they satisfy the following

$$^{sp}/_{si} < \mu \; or \; ^{si}/_{sp} < \mu$$

where si is the size of object Oi and μ is a pre-defined threshold. Checking the objects for size is especially useful if an object in the previous frame splits into a large and a very small region due to inaccurate segmentation. This check eliminates the chance of matching a big region to a small region.

V. CONCLUSION

In this paper, we have presented a novel approach to track a moving object in a SPIHT compressed video. The only data from the compressed stream used in the proposed method are the motion vectors and block coding modes. As a result, the proposed method has a fairly low processing time, yet still provides high accuracy. After the preprocessing stage, which consists of intra-coded block motion approximation and global motion compensation, we employ Blob tracking model to detect and track a moving target. Using this model, an estimate of the labeling of the current frame is formed based on the previous frame labeling and current motion information. The results of experimental evaluations on ground truth video demonstrate superior functionality and accuracy of our approach against other state-of the- art compressed-domain segmentation/tracking approaches. Although our algorithm works well even with fixed parameter values, possibly better performance may be obtained by adaptive tuning, although this would in general increase the complexity.

REFERENCES

1.Video Object Tracking in the Compressed Domain Using Spatio-Temporal Markov Random Fields Sayed Hossein Khatoonabadi and Ivan V. Baji´c, Senior Member, IEEE

2.Shi Y. Q. and Sun H., "Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards," 1st edition, CRC Press, 2000

3.T. Aach, A. Kaup, and R. Mester. Statistical model-based change detection in moving video. Signal Processing, 31:165{180,1993.).

4.A.Jensen, A.laCour- Harbo: Ripples in Mathematics,Springer,2001.

5.A. Said, W.A. Pearlman: A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, 1996

6. Blob Definition in the field of image processing, http://archive.evaluationengineering.com/archive/articles/0806/0806blob_analysis.asp

7.J.Y. Han, "Low-Cost Multi-touch Sensing Through Frustrated Total Internal Reflection." In Proc. UIST2005,ACM Press, 2005, pp. 115-118

8. PETS01: Second IEEE International Workshop on Performance Evaluations of Tracking and Surveillance,December2001.URLhttp://pets2001.visualsurveillance.org.

9. J. G. Allen et al., Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces Conferences in Research and Practice in Information Technology, vol. 36, 2004, pp 1-4

10. G.Welch and G. Bishop. An Introduction to the Kalman Filter, Proceedings of SIGGRAPH 2001, pp 19-24.

11. M. G. Arvanitidou, M. Tok, A. Krutz, and T. Sikora, "Short-term motion based object segmentation," in Proc. IEEE Int. Conf. Multimedia Expo, Barcelona, Spain, Jul. 2011, pp. 1-6.

12. C. Aeschliman, J. Park, and A. C. Kak,"A probabilistic framework for joint segmentation and tracking," in Proc. IEEE Comput. Vis. Pattern Recognit., San Francisco, CA, Jun. 2010, pp. 1371-1378.

13. S. H. Yang, Y. L. Chang, H. C. Chen: A Digital Watermarking Scheme Based on SPIHT coding, IEEE International Conference on Multimedia and Expo (ICME'01),pp.441-444,2001.

14. Salomon D., "Data Compression: the Complete Reference," third edition, Springer, 2004

.