A confidence measure based moving object extraction system built for compressed domain[1]

Roy Wang[1], Hong-Jiang Zhang[2] ,Ya-Qin Zhang[2]
[1]Beckman Institute, University of Illinois at Urbana Champaign, Urbana IL 61801
rwang@uiuc.edu
[2]Microsoft Research Beijing China

ABSTRACT

As the proliferation of compressed video sequences in MPEG formats continues, the ability to perform video analysis directly in the compressed domain becomes increasingly attractive. The availability of motion vectors and pixel values in coded forms can indirectly provide motion and intensity information for object analysis, avoiding the need to re-perform motion estimation. Albeit that the embedded motion field is contaminated with matching modeling errors and measurement errors, we will illustrate several motion field filtering and correction techniques to combat with noisy motion fields. We strike to reconstruct smooth true motion fields with a minimal amount of decoding, reducing computational resource and time requirement. In this paper, we describe the whole moving object extraction system with the general framework and component designs and show their effectiveness with two test sequences.

1. INTRODUCTION

The traditional motion segmentation can be categorized into two types, "direct" and "indirect". Direct methods operate on pixel domain use intensity derivatives to perform motion segmentation directly on the pixels, and indirect methods usually compute optical flow field or motion field then use various clustering methods to group motion into separate distinct groups. Wang and Adelson[6] used a bottom-up approach to perform image layering with motion. More people have refined their techniques [10]. Motion layering in image domain or optical flow domain, due to the number of data points and computational complexity, generally requires large computational resource and time consumption.

In light of the proliferate use of compressed MPEG streams, it increasingly makes video analysis in MPEG domain more attractive. Compressed-domain video possesses several important characteristics attractive for object analysis. First, motion information stored in B, P frames are readily available without incurring cost of re-estimation of motion field. Second, the pixels have been decorrelated and coded in DCT forms, which can indirectly yet readily relay information on image characteristics. On the other hand, the signals are often contaminated with mismatching and quantization errors. Comparably, motion processing in uncompressed image-sequence domain is better suited for accuracy and precision. However the computation in uncompressed domain is often formidable for large video-database. With the pros and cons of compressed domain processing in mind, we establish our goal as to explore the use of compressed and coded motion and pixel information in novel ways to avoid excessive decoding and strike to improve layer-separation accuracy. The bulk of the paper deals with how to accurately estimate and smooth motion field using confident measures based on DCT coefficients, and spatial/temporal continuity of motion. Notice that we do not use any reconstructed pixel-wise information. We try to avoid performing inverse DCT transform, which is an expensive computing process. In a related work in compressed domain, Meng and Chang[8] employ a block count method to estimate parameters in a three-parameter affine global motion model. Then they perform GMC (global motion compensation) to get object mask and perform histogram clustering to deal with multiple objects.

Since MPEG-1 and MPEG-2 encode the bitstream in terms of I-frame, B-frame and P-frame. The B and P frames store the motion information and residues after encoders' motion compensation. The I-frame has no motion values and it stores DCT information of the original frame. Though I-frame provides no motion information, we still could grasp how textured images are and propagate that information to the B, P frames. Some observation made through experiments are that B frames are much more closely placed with their reference frames than P frames in general, thus their motion are more closely matched with true motion vectors. P-frames' motion field, due to larger temporal distance to reference frames, are less reliable. I-frames' DCT coefficients could give us the texture information needed to flag motion reliability. The texture and color information from I-frames can be further propagated into B and P frames by inverse motion compensation.

In light of the limitation imposed by the MPEG domain, we opt to build a system that can perform in near-real time and do not miss objects much but could tolerate false alarms. We realize that the motion fields in MPEG streams are quite prone to quantization errors. On top of that, the encoding steps might have blocks intra-coded or wrongly matched in low-textured areas. Thus, we hope to build a robust system that would generate a confidence measure on the motion field and accurately filter out errors and recover true motion. The goal is to build a fast object index/retrieval tool that makes compressed domain video stream from an unstructured version to a more structured one and allow other component technologies to form a more reliable object filtering later on in a larger system.

This paper is organized as the following. In Section 2 we discuss the overall design of the system as well as some key components.

[1] The work was performed in MSR Beijing, where the 1st author interned in summer 1999.

In section 3, we present the results from several experiments performed on "coast-guard" and "car" sequence. Finally we summarize our approach and findings in section 4.

2. System and Component Design

2.1 Overall Flow

The system derives spatial, temporal, and directional confidence measures from incoming stream, buffering three adjacent frames. Based on the combined confidence score, we perform a hard cut on low confident macroblock to reject those motion vectors that are we believe are very likely to be mismatched by encoders. We then perform one or more linear or non-linear motion filtering operations to repair the holes occurred in the motion field [2][4]. Then the dominant motion is separated out by a recursive least square algorithm to get an object mask as a byproduct. To identify multiple objects, we perform k-means and/or EM clustering based on spatial and motion features. We then track the objects by their location and motion. The eventual goal of the system is to generate the description of objects in a video including appearance time, length, velocity and object shape characteristics.

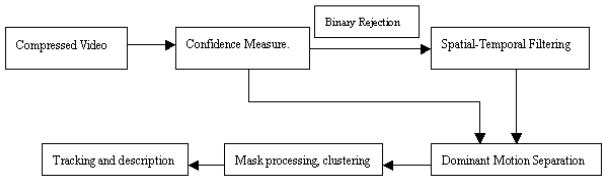


Figure 1. the system diagram with coded stream going in and structured object information coming out, while without pixel decoding.

In section 2.2, 2.3, 2.4, we describe how we measure spatial, temporal and textural confidences and combine them into an overall confidence indictor in 2.5. These are what the 2nd box on top row in the system diagram constitutes. Then 2.6 describes motion filtering process, and 2.7 describes how to use confidence and filtered motion field to perform motion layer separation. 2.8 and 2.9 complete the last stages of the system.

2.2 Spatial Confidence Measure

Discontinuity in motion magnitude and direction comes from several sources including object boundaries and/or motion estimation failure. The magnitude and directional confidence measure produces a spatial confidence score that reflects how current motion vector violates the neighbor-hood smoothness constraint. The measure maps motion discontinuity into a range of confidence score of 0-100 as a probabilistic expression. The same range is used for all other types of confidence measure.

$$Mag\_Confidence_{i,j} = Clip_{0-100}\left(100 - 100 * \left|\frac{M_{i,j} - \sum_{\Theta} M / N}{\sum_{\Theta} M / N}\right|\right),$$

$$Dir\_Confidence_{i,j} = Clip_{0-100}\left(100 - 100 * \left|\frac{D_{i,j} - \sum_{\Theta} D / N}{\sum_{\Theta} D / N}\right|\right), \quad (1)$$

where Θ is the structural element set, either cross structure or 8-neighbor one, that defines neighborhood motion vectors. N is the normalization term, the number of elements in the set Θ. M is motion magnitude and D is the motion direction in gradient. Subscripts indicate block indices.

2.3 Temporal Confidence Measure

This confidence measure is derived from the temporal adjacent neighborhood of a macroblock. This confidence measure comes from temporal neighborhood of the current macroblock based on the intuition that a 'good' motion vector should not have its direction altered in a drastic manner. The confidence measure is just a variation of the directional measure above.

2.4 Texture Confidence Measure

AC coefficients in a DCT transformed macroblock can indirectly provide information on how textured the area of the image is. Low-textured region tends to cause poor encoding matching errors [3]. As B and P frames are residue coded, their texture measures are propagated from the I frames by inverse motion compensation. The resultant macroblock from performing inverse motion compensation could overlap with four other 8x8 DCT blocks in I-frame. We measure the average of the four neighboring blocks' energy as an approximation to true block energy. Texture measure is then based on AC energy computed by grouping AC DCT co-efficient into Horizontal, Vertical and diagonal energy groups then computing the average of the three energy [1].

$$H\begin{bmatrix} 0&0&1&1&1&1&1&1\\ 0&0&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{bmatrix} V\begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0 \end{bmatrix} D\begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&1&1&1&0&0&0&0\\ 0&0&1&1&1&0&0&0\\ 0&0&0&1&1&1&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{bmatrix}$$

$$E = \frac{E_h + E_v + E_d}{3} \quad (2)$$

The top row of equation (2) shows the masking of H(horizontal), V(vertical), and D(diagonal) DCT coefficients. The average energy is the equal-weighted sum of the three. To map the average energy to the confidence range, we employ an energy threshold through experiments as the breakpoint for low and high texture. Anything above the energy threshold is marked as 100% confident on the high textured region. For energy value below the threshold, we measure the ratio of the value to the threshold.

$$\forall E \in (E < E_t), Confidence = \frac{100 * E}{E_t}, \quad (3)$$

$$\forall E \in (E >= E_t), Confidence = 100$$

where E is the average energy value, E_t is a normalizing threshold. A sample frame illustrates the texture relevance.

Figure 2. Left: original frame, Right: low texture regions marked out by confidence processing, where darker regions indicate lower

confidence. The pavement and sky are correctly marked out as low confidence regions.

2.5 Confidence Measures Combination

Combining the four confidence measures, two from the spatial domain, one from temporal, and the last from texture, we form a single score to express our overall confidence to the current motion block. The combination is a weighted sum one, and the weights in our implementation are set to be equally important. Further experimentation is required to identify an empirically optimal weighting function.

2.6 Motion Filtering

Spatial box filtering is performed with a box filter [4], and optionally ensued by a median filter or morphological filter, or neighboring max filter.

The temporal filter coefficients are determined from all three frames' confidence score, three frames' frame types. First, we normalize all motion vectors from B,P frames by their relative distance to their reference frames. Then, the temporal filtering is simply a 1-D weighted sum of motion vectors from the previous, current and next frame's corresponding macroblocks. Suppose a,b,c being the previous, current, and next frame's macroblocks, and MV being their motion vectors. Then filtered MV is the weighted sum of the three, weighted by each one's confidence score, and the frame type. The center frame's weight is further doubled to signify its importance.

$$MV = \sum_{i \in a,b,c} Score_i * MV_i * Frame\_W_i \qquad (4)$$

$$Frame\_W_i = \begin{cases} 0 & i' \; frametype = I \\ 1 & i' \; frametype = P \\ 2 & i' \; frametype = B \end{cases}$$

2.7 Global Motion Compensation

The GM estimation is a recursive least square method that iteratively refines the object mask to estimate three affine parameters, zoom, vertical and horizontal translations[5]. The method is modified by injecting the confidence score for each macroblock. The injection ensures that low confidence of motion vector would contribute less to estimating global motion. The byproduct of this process gives us an object mask where potential multiple moving objects lie in. With no loss of generality, we show the formulation of a four-parameter affine model.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} \qquad (5)$$

where (u,v) is the pixel location at next frame, while (x,y) is the pixel location in current frame. a,b,c,d are the parameters for the model. Let X=[a b c d] be the parameter vector, Y be the observation matrix in next frame, H being the pixel coordinates in pervious frame. We then express the model with equation (6).

$$Y = H * X + N \qquad (6)$$

where Y is a confidence-weighted observation column vector and H is a confidence-weighted coordinate matrix(dimension is 2*number of macroblocks by 4, in this case). N is an additive Gaussian modeling of encoding noise.

$$Y = \begin{bmatrix} W_1 * u_1 \\ W_1 * v_1 \\ W_2 * u_2 \\ W_2 * v_2 \\ \cdots \end{bmatrix}, H = \begin{bmatrix} W_1 * (x_1 & y_1 & 1 & 0) \\ W_1 * (y_1 & -x_1 & 0 & 1) \\ W_2 * (x_2 & y_2 & 1 & 0) \\ W_2 * (y_2 & -x_2 & 0 & 1) \\ .. & & & \end{bmatrix} \qquad (7)$$

(u_i, v_i) are coordinates of corresponding pixel at (x_i,y_i) in next frame.

$X_{LS}=(H^t * H)^{-1} * H^t * Y$ then becomes the ML solution for the particular model. The algorithm takes a generalized EM approach to minimize the statistics of outlier motion vectors. Within every iteration, it calculates the ML solution of X, and compensates the motion field, decides outliers and then derives the statistics of outliers by examining the residue components of the motion field. The notion of pixel in the model is changed to macroblock in the compressed domain. Moving block coordinates in H to Y and performing a subtraction of (x,y) from (u,v) help to incorporate motion vectors in the formulation.

2.8 Object Mask Filtering

With the temporal continuity constraint imposed on object movement, we use temporal alignment of previous several frames' object masks to filter out unwanted segmentation errors introduced at each frame. A temporal median filter is applied to the frames aligned by the motion at each frame.

2.9 Object Clustering

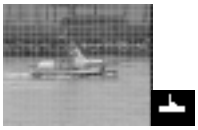
For frames that contain more than one object, we need to perform clustering to separate object mask into multiple parts. One type of clustering is performed first is the K-means clustering on spatial adjacency alone. The clustering works well for objects that are "far" apart. On top of this type of clustering, an EM clustering incorporating the motion model closeness constraint is applied to segment out the objects moving at different direction and speed.

3. Results

We tested the proposed system on two MPEG-2 sequences, "car" and "coastguard". The "coastguard" sequence has a horizontal camera pan while the boat is moving across the water. The "car" sequence has multiple motions including simultaneous panning and zooming. There is a car moving from left to right and the ground is very low-textured.

Frame70 of "car", &mask

Frame100 of "car" &mask



Frame126 “coastguard” Frame299, “coastguard”

Figure 3. sample frames from the sequence.

We construct three cases for testing. Case 1, there is no confidence measure or any type of filtering applied to the sequences. Case 2, the confidence measure is applied but no filtering. Case 3, both confidence measure and filtering are applied. The shape masks reduced to 16x16 blocks for all the frames serve as the ground truth. We measure the mean and standard deviation of the mean object block count, missing blocks and false alarm blocks.

Case/ Sequence	Mean Block Count	Std Block Count	Mean Block Miss	Std Block Miss	Mean False Alarm	Std False Alarm
1/coast	50	18.2	10	9.6	11	11.9
2/coast	55	17.6	8	6.4	14	14.0
3/coast	56	4.5	3	1.8	9	3.7
1/car	83	50.5	10	3.6	67	52.4
2/car	83	49.8	10	3.5	71	53.0
3/car	90	25.2	4	2.8	70	25.8

In both sequences, we observe that the system has successfully reduced the mean and standard deviation of the number of missing blocks and false alarms. One trend is that the confidence measure and subsequent rejection could cause higher false alarm while reducing and keeping the missing block counts. This is due to that the global motion estimation with confidence injection assumes lower probability of the rejected region as background, thus higher probability of the rejected region as foreground. Further constraints on other features or a priori object knowledge would significantly reduce the number of false alarms. As we said before, we could tolerate false alarms while opting for lower misses.

Current implementation of the system is not based on the mpeg decoder but on motion and DCT extracts from the sequence. It thus requires much unnecessary file I/O processing. The speed of the current system is only bout 0.5 sec/frame for the CIF size on a Pentium III 450Mhz. The achievable speed is expected to be much faster when the system is transferred to be on top of a mpeg decoder.

4. Summary

Compared with object analysis in conventional domain, the compressed domain segmentation provides a tradeoff between speed and accuracy. Faced with contaminated motion field and transformed subset of pixel information, our proposed technique and system aim to improve on motion layer separation accuracy with minimal amount of decoding. We incorporate several confidence measures in our object analysis to have some probabilistic knowledge on reliability of encoded motion to their ‘true’ motion counterparts. The confidence measures then motion filtering have straightforward computation. We believe, with some careful implementation design, that the system could achieve much faster speed than our current implementation. Work ahead includes incorporating other features such as color into object segmentation and tracking. A scheme for signaling heavy-noised frames is also desirable to allow the system to skip those frames.

5. Acknowledgement

We want to thank Horace Meng, Prof. Shi-fu Chang for their help in software development. Also thank goes to Drs. Li Jin and S.Y. Kung and reviewers for their invaluable comments.

6. REFERENCES

[1] Hualu Wang, Shi-Fu Chang “ A highly efficient system for automatic face region detection in MPEG video, *IEEE Transaction on circuits and systems for video technology*, vol 7. No 4. August 1997

[2] Yao Wang, Qin-Fan Zhu “Error Control and concealment for video communication: A review”, *Proceedings of the IEEE*, vol 86, no 5. May 1998

[3] Roberto Castagno, Touradj Ebrahimi, Murat Kunt “Video segmentation based on multiple features for interactive multimedia applications” *IEEE Transaction on circuits and systems for video technology*, vol 8. No 5. September 1999

[4] Sohail Zafar, Ya-Qin Zhang, “Predictive block matching motion estimation for TV coding” *IEEE transactions on broadcasting* vol37. no3. sept. 1991

[5] Roy Wang, Thomas Huang “Fast camera motion anaysis in MPEG domain” *International Conference on Image Processing, 1999*

[6] John Wang, Edward Adelson “Representing moving images with layers” *IEEE Transaction on Image Processing* Vol 3. No5. September 1994

[7] Thomas Meier and King N. Ngan. “Automatic Segmentation of moving objects for video object plane generation”, *IEEE Transaction on circuits and systems for video technology vol 8 No 5*. Septemeber 1998

[8] Jianhao Meng, Shih-Fu Chang. “Tools for compressed-domain video indexing and editing”, *SPIE Conference on Storage and Retrieval for Image and video database*, vol 2570, Feb. 1996..

[9] K.W. Chun, K.W. Lim, H.D. Cho and J.B. Ra “An adaptive perceptual quantization algorithm for video coding” *IEEE Transaction on Consumer Electronics*, 39(3) Aug, 1993.

[10]Georgi D. Borshukov, Gozde Bozdagi, etl. “Motion Segmentation by Multistage Affine Classification” *IEEE Transactions on Image Processing*, vol 6. No. 11 November 1997