Object Tracking in Compressed Video with Confidence Measures

Lan Dong[1], Imad Zoghlami[2], Stuart C. Schwartz[1]

[1]Dept. of Electrical Engineering, Princeton University, {ldong, stuart}@ princeton.edu
[2]Real-Time Vision & Modelling, Siemens Corporate Research Inc., imad.zoghlami@siemens.com

ABSTRACT

In this paper, a novel robust tracking algorithm in compressed video is proposed. Within the framework of video compression standards, we consider how to accurately estimate motion of an object by utilizing motion vectors available in compressed video together with derived confidence measures. These confidence measures are based on DCT coefficients, spatial continuity of motion and texture measure of the object. We perform tracking directly on the compressed data and also consider tracking of an object with image scale change. In order to achieve robust tracking, we develop a system which enables us to detect object appearance change such as illumination change and occlusion by exploring the confidence measures derived above. Preliminary results indicate that our tracking algorithm works well with a variety of video sequences.

1. INTRODUCTION

Visual tracking has been an area of intensive research in the field of computer vision. With the advent of emerging multimedia standards like MPEG, it has become essential to develop a system that performs visual tracking in a computationally efficient manner. Tracking in the compressed domain is of interest since working in the compressed domain brings a lot of advantages. As discussed in Wang, Zhang and Zhang [1], motion information stored in B, P frames of compressed video are readily available without incurring the cost of re-estimation of the motion field. Secondly, the pixels have been de-correlated and coded in DCT form, which can indirectly, yet readily relay information on image characteristics.

Motion vectors have been widely used for tracking purposes in compressed domain as they readily provide motion information [1]-[3]. But there are also problems using them. Although motion vectors embed rich motion information among frames, they were originally designed to minimize the matching error for coding purposes; thus they do not always indicate true motions [4]. Secondly, error from motion vectors due to absence of any verification for the object being tracked (i.e., ground truth) is easily accumulated and typically results in loss of tracker. Thirdly, the appearance of an object to be tracked is always changing, caused by occlusion, illumination, size change or etc., in which cases, the use of motion vectors for tracking is not always feasible.

With the pros and cons of compressed domain processing in mind, we have as our goal to explore the use of compressed motion and pixel information in novel ways to avoid excessive decoding and simultaneously to improve tracking accuracy.

In this paper, we develop a system which can achieve robust tracking of an object. The robustness consists of two aspects: we try to accurately estimate and smooth the motion field by a confidence measure based on DCT coefficients, spatial continuity and texture of object. Our system utilizes the idea found in [1] for designing the confidence measures and we further explore these measures to determine object size and to detect object change (occlusion, illumination change, etc.), which is the other aspect of the robustness of our system. If object change is detected, we use the next I frame, the reference for each GOP (Group of Pictures), to determine the position of the tracked object. The proposed object tracking system tries to achieve robustness and efficiency at the same time as we avoid performing an inverse DCT transform, which is an expensive computing process.

The rest of the paper is organized as follows. The description of the approach is discussed in Sec.2. The experimental results are presented in Sec.3, followed by Sec.4, with discussion and conclusions.

2. ROBUST SYSTEM AND COMPONENT DESIGN

2.1 System overview

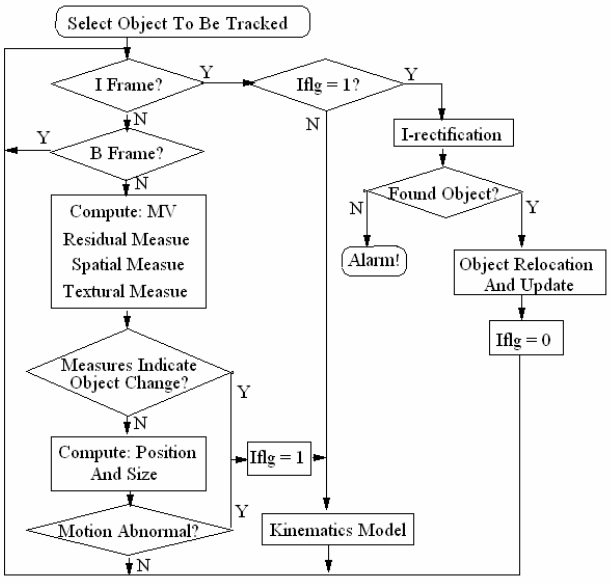


Fig 1 Block Diagram of System (Iflg is the I-rectification indicator)

Given a MPEG-2 video clip as input, we perform tracking of an object which is manually selected at the first frame of the video. Initial selection can be automated in future versions. In the

following P frames, leaving B frames unprocessed, the motion vectors (MV) *hitting* the object, together with their derived residual, spatial and textural confidence measures are then used for deciding the new position and size of the object. Simultaneously, by exploring these three confidence measures, we can detect change of object (occlusion or global illumination change or else). When there is no object change detected, we continue with this process until the next I frame occurs, at which point we use a kinematics model to predict the position of the object. If we have detected object change or the motion trajectory of the object is abnormal, we use kinematics model to predict the position in the current frame and call for I rectification to find the object when the next I frame occurs. After the object is relocated and updated, we repeat the above process in the new GOP. If the object can not be found in this I frame, we call for human interaction. The overall process can be seen in Figure1. The detailed process will be given in the following sections.

2.2 Residual Confidence Measure

In the encoding process, for the P frame, a temporal prediction is first employed and then the inter-frame prediction error is DCT encoded [5]. The prediction error is the difference between the current block and the motion-compensated reference block, which can be modeled as Gaussian noise for each pixel in the block. Then we can consider the sum-square of the prediction error of the whole block as a probability measure of the current block matching the reference block. In other words, if the sum-square of the prediction error is big, the probability of the current block matching the reference one is small. This is used as a Residual Confidence Measure for the motion vector in our method. We define it as: $wt\_1 = \exp(-\frac{\sum err^2}{K}) = \exp(-\frac{\sum err_{DCT}{}^2}{K})$ where K is the variance of the error.

The second equation comes from Parseval's relation that the total energy of the signal is not changed by a DCT transformation. Thus, this weight is easy to get from the coded data without decoding.

2.3 Spatial Confidence Measure

It is believed that motion vectors of macro-blocks over the rigid object area remain spatially unchanged just as the optical flow field has some smoothness constraints [6]. We want to favor the MV which is smooth within its neighborhood and to penalize the MV of sharp motion discontinuities which may frequently occur near an object boundary or in an occluded area. Our spatial confidence measure is a score that reflects how the current motion vector violates the "neighborhood smoothness" constraint. It is defined as: $wt\_2 = \exp(-\left| \mathbf{mv} - med\{\text{neighbor of } \mathbf{mv}\} \right|)$ where $med\{\cdot\}$ is a median filter.

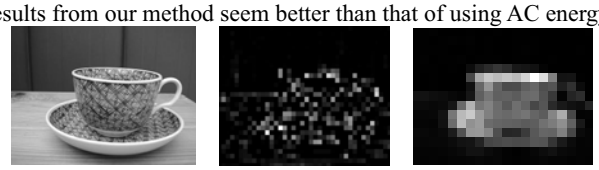
2.4 Textural Confidence Measure

From the encoding standard, the motion vector is a temporal prediction which aims at minimizing the temporal redundancy or matching error. It is not necessarily the true motion of the macro-block. For example, the motion vectors of the homogeneous region (e.g. the background of Fig. 2) will be random, which provides no information about how the object moves. Therefore, we want to find a measure which will be related to the homogeneity of the texture and which indicates how much confidence we have that the motion vector is the true motion.

In [1], the authors use the AC energy of each block as a textural confidence measure. This works well for a constant region but we have found it does not work well for homogenous regions which have a rich pattern and, thus, high AC energy.

Our textural confidence measure is defined in the following way: for each macro-block in the object, we search in a small area centered on it for the best match macro-block according to the mean absolute difference (MAD), which is the most popular block matching criterion in MPEG. We then use the MAD of the best match, i.e. the minimal MAD-- minMAD for each macro-block as our confidence measure. It is clear that the bigger the minMAD, the more distinguishable the macro-block is with its neighborhood, the less probable that the motion vector points to the mistaken block, the more reliable its corresponding motion vector being the true motion. The equation is: $wt\_3 = \alpha \cdot \text{minMAD}$ where α is a normalization coefficient.

Our idea in defining the textural confidence is to explore how the MV is formed and to view the root cause of the uncertainty of the MV before we try to design a confidence measure. Figure 2 shows a sample of textural confidence measure where darker regions indicate lower confidence. The cup in the figure has a rich pattern but is still homogenous; thus it should have low confidence. Results from our method seem better than that of using AC energy.



(a) (b) (c)

Fig 2: (a) original image (b) Textural measure by AC (c) Textural measure by minMAD

The textural confidence of an object is calculated at the first frame when the user selects the object and saved for future use. As it is related to only the object texture, there is no need to compute it in each new frame.

2.5 Motion Prediction Based on 3 Measures

Combining these three measures, we obtain a single score to express our overall confidence in the current motion vector. As we don't know which weight is more important for now, the weights in our implementation are set to be equally important based on our preliminary simulation. That is: $w = \frac{1}{3}wt\_1 + \frac{1}{3}wt\_2 + \frac{1}{3}wt\_3$ However, further experimentation is required to optimize the weights of the confidence measures.

We extract the MV from the MPEG data stream. For each non-skipped slice macro-block in the current P frame, we have a motion vector that points to a macro-block which is most similar with the current one among the nearby macro-blocks in the previous reference frame. For the skipped macro-blocks, we consider the motion vector having zero-length. We ignore the intra-coded macro-blocks.

With MV and associated confidence, we are ready to compute the position of the object. For now, we assume the object is fixed in size. For the motion vector which points into the object region, we call it a *hit* and denote it as $\mathbf{v}_i$. In Schonfeld and Lelescu [3], they collect and average all the *hits* (motion vector which points to the object region) to get the motion of the object. This procedure does not work well when the MV does not indicate true motion. In our algorithm, we combine the confidence measure w with each *hit* vector to estimate the motion of the object, $\mathbf{d}$, as:

$$\mathbf{d} = -\frac{1}{w_1 + \ldots + w_n}\sum_{i=1}^{n} w_i \mathbf{v}_i$$ where n is the number of *hits* in the reference picture, including relevant skipped macro-blocks.

Figure 3 illustrates how we compute the motion from *hits*. The new position of the object in the current frame, $\mathbf{p}_{cur}$, based on the previous position $\mathbf{p}_{ref}$ is: $\mathbf{p}_{cur} = \mathbf{p}_{ref} + \mathbf{d}$

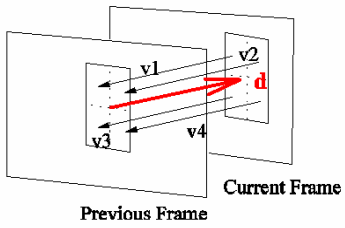


Fig 3: Use *hits* for object motion

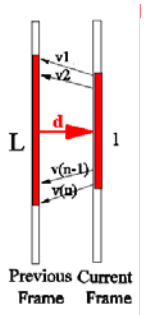
The proposed method is superior in finding the exact object position when compared with block based tracking method as in [1], [2]. This is because for the block based method, objects can rarely be tracked sharply and with precision along their exact boundaries due to the blocky nature of the data. But as we compute object motion from the MV, which is pixel-wise accurate, our object motion is also pixel-wise accurate and thus we are able to find the precise position of the object.

2.6 Object Size Change

Position and size scale change are the most commonly seen motion of objects in a video. Above, we have already given how to get the position of the fixed size object using the weighted sum of motion vector. There remains the problem to get the new position and size of the object when there is object scale change.

We outlined the idea of proof in the following that in the ideal case (all the MV's are 100% confident), the position can still be determined by the average of motion vectors and the scale by the variance of them. For simplicity, we consider the 1-D object since it is straightforward to generalize to the 2-D case. Assume the object has length l in the current frame and length L in the reference frame (See Figure 4). Then, we will have n *hit* motion vectors: $v(1), v(2), \ldots, v(n-1), v(n)$, the average of which gives the motion ($\mathbf{d}$) of object.

Assume those *hit* motion vectors average to zero (if not zero, shift by the mean) in the y-direction and are all the same in the x-direction (Figure 4). It is easy to get that, after some calculation, the variance the vectors as indicated in the figure.



$$\mathrm{var}(V) = K(n)(L-l)^2$$

where

$$K(n) = \begin{cases} \frac{2}{n}\left(\frac{n}{8} - \frac{n(n-2)}{12(n-1)}\right) & \text{if } n \text{ is even} \\ \frac{2}{n}\left(\frac{n-1}{8} - \frac{(n-1)(2n-1)}{24(n-1)}\right) & \text{if } n \text{ is odd} \end{cases}$$

Fig 4: MV's for object with size change (the solid part in the frame indicates an object) and variance for the vectors

Therefore, the new length l can be determined as:

$$l = \begin{cases} L + \sqrt{\frac{\mathrm{var}(V)}{K(n)}} & \text{if y - direction of } v(1) \sim v(n) \text{ change from positive to negtive} \\ L - \sqrt{\frac{\mathrm{var}(V)}{K(n)}} & \text{if y - direction of } v(1) \sim v(n) \text{ change from negtive to positive} \end{cases}$$

For the general non-ideal case, the object position and size can be determined in the same way except that we will use the MV together with its confidence measure to derive the weighted average and variance.

2.7 Detection of Object Change

One of the biggest problems with MV tracking is that in a video sequence, it is very often the case that the object appearance changes, even for a rigid object. The causes include global illumination change, object being occluded, object turning around, scene switch, etc. Change of object appearance will result in abnormal motion vector behavior and thus loss of track. Therefore, it is very important that we use a robust tracking method that can detect change of object and relocate the object as soon as possible.

In our method, we explore the behavior of the MV and the weights developed for the confidence measures to detect object appearance change.

First, when there appear many intra-coded macro-blocks in the object region, we declare object change. Intra-coded macro-block often indicates that there is no 'matching' macro-block in the reference frame. When there are not enough matching macro-blocks in the current frame, we consider the object appearance may have changed.

Second, when there are many macro-blocks whose residual measure wt_1 falls below some threshold, we claim the object has changed. If the residual measure of the macro-block falls below some threshold, that is equivalent to saying that the prediction error is big and the probability of the macro-block being the true match of the reference one is small.

Third, when there are many macro-blocks whose value $f = wt\_2^{wt\_3}$ falls below some threshold, we announce the object change. As we discussed before, when the spatial measure wt_2 is small, it indicates that the motion vector in the object region is not consistent with its neighbor. This is usually caused by object change. But sometimes, it is not true. For example, for a homogenous object region, the motion vector can be random and thus the spatial measure is small. However, it is not caused by object change but rather by 'matching criteria' during encoding and we should not label these macro-blocks as changed. Therefore, we should consider the spatial measure together with the textural measure wt_3 to decide object change. If a function is used to present with how much confidence an object has not been changed, it should have two properties: for fixed textural measure, the bigger the spatial measure, the more confident the object has not been changed; for fixed spatial measure, the smaller the textural measure, the more confident the object has not been changed. As $f = wt\_2^{wt\_3}$ has the above properties, it can be used in our object change test, i.e. if f is less than some threshold, we claim that macro-block has changed; when number of changed macro-blocks is big, we claim object change.

When any one of the above three test is satisfied, we announce an object change. We use the kinematics model instead of computing from the MV for the object position in the current frame and call for I-Rectification in the following I frame to relocate the object.

2.8 I Frame Rectification

I frame rectification is very commonly used in MV tracking algorithms to ensure tracking precision, as in [2], [3]. If we decode the coming I frame into the pixel domain, there are numerous existing methods to do I-Rectification as it then becomes a common tracking problem in the computer vision area. However, it is a potentially computationally intensive task to decode the whole frame. In our system, we use the DCT-based compressed domain tracking method proposed in [7]. If the object is found, the tracker continues to track using the algorithm discussed above. If the object is not found, the system calls for human interaction.

Besides detection of object change, there are other cases for which I-Rectification is needed. One case arises when the object motion becomes bigger than the search window during encoding, and the motion vector will not be big enough to predict the motion. In this case, the patterns of the motion vectors will be similar with that of object change and it fails any of the above three tests. Another case is when we see an abnormal motion trajectory of the object from the kinematics model. In addition, in order to ensure a balance between tracking precision and efficiency, we also call for I-Rectification when there is no I-Rectification for a great number N_0 of frames (e.g. N_0=1000).

3. EXPERIMENTAL RESULTS

The experiments are designed to test the reliability of tracking with the MV and the detection of object change. Initially, we simply use template matching in a local search window for I-Rectification. For the kinematics model, we use linear prediction from the previous two tracking positions: $\mathbf{x}_i = \mathbf{x}_{i-1} + (\mathbf{x}_{i-1} - \mathbf{x}_{i-2})$ for simplicity. All the videos are encoded with 24 frames in a GOP, the structure of which is: IBBPBBPBBPBBPBBPBBPBBPBB. The tracking performance is shown in Table 2. False Alarm and Miss Detection are used to test the reliability of the detection of object change while Average % overlap BB, which is the average percentage of the ground truth bounding box (BB) has overlapped the tracker's bounding box, is used to see how well MV tracking performs.

Table 2 Tracking performance of different videos

Sequence	Green	Red	Train	Son	Girl	Walk
# of P/I Frames	80	80	97	17	56	74
False Alarm	0	3	0	0	0	2
Miss Detection	0	0	1	0	0	1
Avg%overlap BB	94	81	95	95	98	78

The first two sequences are the tracking results from a well-known sequence *cactus*. We try to track the size varying red pen and the green pen (Figure 5) respectively.

The third one is from another well-known sequence *Mobile and Calendar*. It works very well for tracking the train until it gives out continuous alarms at the end where the train is occluded.

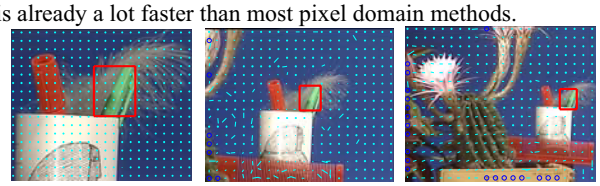
The fourth sequence has sudden scene change in it. We try to track the face of the boy at the first frame while the whole scene suddenly changes at the 30th frame (Figure 6). Our tracker gives out an alarm immediately and calls for human interaction when in the following I frame, the tracker can not find the object face.

The fifth sequence has gradual illumination change in it and in the sixth sequence the person makes two turnarounds. Our tracker has detected the first turnaround as an object change but misses the second one. One interesting thing to see is that when the person is turning around, the bounding box first becomes larger and then becomes smaller. It tries to fit the person as the image of the person changes from bigger and then to smaller as she turns. This is achieved by the size prediction feature in our algorithm, which has again been shown to work well.

Because of space limits, only two results are shown here in Fig 5 and 6. More results can be accessed at:
http://www.princeton.edu/~ldong/research/MV_tracking/

These sequences were chosen to reveal a variety of common difficulties and different object changes in tracking tasks. Our algorithm appears to overcome many of these difficulties and achieves good performance.

On a 1 GHz PIII processor, our implementation can process about 1600 P frames per second with the object size of 96x64 pixels. The C++ program is not optimized in terms of speed though it is already a lot faster than most pixel domain methods.



Frame 3 Frame 123 Frame 228

Figure 5 Sample tracking results for green pen

Frame 3 Frame 27 Frame 30

Figure 6 Sample tracking results for son

4. CONCLUSION

This paper has presented a novel robust tracking system using MV and associated residual, spatial and textural confidence measures that are derived from compressed video. The tracker is able to track objects with varying size. In order to achieve robust tracking, the three confidence measures are also used to detect object change. I-Rectification is called for when object change is detected.

Besides fine tuning the parameters used in this system, there are additional problems to be considered. We found using the variance of the MV to track the object is sensitive to noise. This is especially so when we try to track non-rigid objects where the size prediction is easy to fail. Among our future modifications are to make size prediction more stable and robust to noise.

REFERENCE

[1] R. Wang, H.J. Zhang, Y.Q. Zhang, "A confidence measure based moving object extraction system built for compressed domain" Proc. of ISCAS 2000 Geneva, Vol. 5, pp.21 – 24, 2000

[2] L. Favalli, A. Mecocci, F. Moshetti, "Object tracking for retrieval applications in MPEG-2", IEEE Trans. Circuits and Systems for Video Tech., Vol. 10, Issue 3, pp.427 – 432, 2000

[3] D. Schonfeld, D. Lelescu, "VORTEX: video retrieval and tracking from compressed multimedia databases", Proceedings of ICIP98, vol.3, pp. 123-127, 1998

[4] R. Achanta, M. Kankanhalli, P. Mulhem, "Compressed domain object tracking for automatic indexing of objects in MPEG home video", Proc of ICME, Vol. 2, pp. 61 – 64, 2000

[5] M. Tekalp, "Digital Video Processing", Signal Processing Series, Prentice Hall, 1995

[6] B.K.P. Horn, B.G. Schunck, "Determining optical flow", Artificial Intelligence, Vol. 17, pp. 185-203, 1981

[7] L. Dong, S.C. Schwartz, "DCT-Based Object Tracking in Compressed video", to appear in ICASSP 2006