A FEATURE TRACKING ALGORITHM USING NEIGHBORHOOD RELAXATION WITH MULTI-CANDIDATE PRE-SCREENING

Yen-Kuang Chen, Yun-Ting Lin, and S.Y. Kung

Princeton University

ABSTRACT

Tracking of features in video sequences has many applications. Conventionally, the minimum displaced frame difference (referred to as DFD or residue) of a block of pixels is used as the criterion for tracking in block-matching algorithms (BMA). However, such a criterion often misses the true motion vectors, due to many practical factors, e.g. affine warping, image noises, object occlusion, lighting variation, and existence of multiple minimal DFD. Our goal in this paper is to find motion vectors of the features for object-based motion tracking, in which (1) any region of an object contains a good number of blocks, whose motion vectors exhibit certain consistency; (2) only true motion vectors for a few blocks per region are needed. Hence, we propose a new tracking method: (1) At the outset, we disqualify some of the reference blocks which are considered to be unreliable to track. (2) We adopt a multi-candidate pre-screening to provide some robustness in selecting motion candidates. (3) Assuming the true motion field is piecewise continuous, we determine the motion of a feature block by consulting all its neighboring blocks' directions. This allows a chance that a singular and erroneous motion vector may be corrected by its surrounding motion vectors (just like median filtering). Our method is also designed for tracking more flexible affine-type motions, such as rotation, zooming, sheering, etc. Finally, the performance improvement over other existing methods is demonstrated.

1. INTRODUCTION

Tracking of features in video sequences has many useful applications in scene segmentation [3, 9], image analysis for security purposes [5, 8], object-based video coding, video synthesis, computer vision, etc. The minimum displaced frame difference (DFD) of a block of pixels criterion are widely used in motion compensated video coding, such as MPEG I and II. However, such a conventional criterion often misses the true motion vectors, due to many practical factors, e.g. affine warping, image noises, object occlusion, lighting variation, and existence of multiple minimal DFD [11, 12]. Our goal here is aimed at finding the true motion vectors of the features.

Just like BMA, a frame is segmented into blocks (8×8, 16×16, etc.) in our scheme. Unlike BMA, not all of the blocks will be tracked, since many of them do not contain sufficient prominent features, making them reliable to track. For object-based coding and segmentation applications, the major emphasis is not placed on the number of *feature blocks* (FBs) to be tracked (i.e. quantity) but on the reliability of the FBs we choose to track (i.e. quality). This means that we can afford to be more selective in our choice of FBs. So a natural first step is to eliminate those unreliable or unnecessary FBs. For example, if a block does not contain any prominent texture feature, then it is very likely to be confused by its adjacent blocks due to image noise. It is advisable to exclude such a FB from the pool of "valid" FBs, saving/improving the tracking effort/result [1, 14]. To avoid tracking such homogeneous blocks, we must take into account the variance of the blocks. If the block's variance of intensity is small, the block is considered to be low-confidence and will be disqualified. In other words, only blocks with variance exceeding a certain threshold will be considered.

2. OUR FEATURE TRACKING METHOD

After the prominent FBs are properly identified, the main results of this paper lies in a new algorithm to determine the true motion vectors. It is well known that the conventional BMA which outputs the motion vectors with the minimal residues might work well for a residue-oriented encoder, such as MPEG I & II. However, one must recognize the fact that the **true** motion vector does not always yield minimal residue. Likewise, the minimum residue solution does not necessarily deliver the true motion vector. Fortunately, this difficulty may be effectively circumvented by the following two practical observations:

1. The true motion vector, while not the absolute minimum, is very likely one of the *multiple minima*, according to the residue criterion. This observation leads to a multi-candidate pre-screening so as to keep track of all possible true motions for every FB. This will be addressed in Section 2.1.
2. The true motion vector is the absolute minimum if cost criterion can be modified properly. This leads to a neighborhood relaxation score function, as discussed in Section 2.2.

2.1. Multi-Candidate Pre-screening

Recall that while the true motion vector may not be the absolute minimum, we should at least make a finalist list among several *minima*. Hence, a multi-candidate pre-screening becomes necessary so that a more inclusive record on possible motion vectors could be maintained, preventing (1) the true motion vector from being eliminated, and (2) the wrong motion vectors from being accepted in the early phase.

At the outset, two kinds of thresholds can be used to qualify or disqualify the candidates. One is called *lower residue threshold*. As long as the residue is less than this threshold, the motion vector will be automatically **accepted** as a possible candidate, cf. Figure 1(a). In contrast, we also propose a so-called *upper residue threshold*. If the residue exceeds this threshold, the motion vector will be automatically **eliminated** from the candidate pool, cf. Figure 1(b).

After this initial stage, we now adopt a slightly more sophisticated scheme to select the final candidates. Since we want to provide some robustness in selecting candidates, certain noise margin

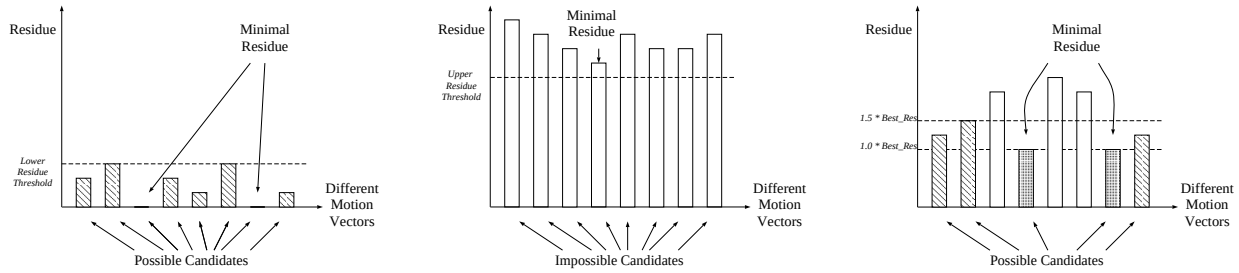

Figure 1: Multi-Candidate Pre-screening.

must be allowed so as to admit a proper number of possible candidates, cf. Figure 1(c). In order to maintain some kind statistical consistency from blocks to blocks, we apply the same level of relative tolerance to all blocks. The motion vectors yielding no more than 1.5 folds of the minimal residue will be admitted into the candidate pool, where about 90% of true motion vectors are kept.

2.2. Neighborhood Relaxation Score Function

In order not to lose the true motion vectors in the first place, we use a multi-candidate pre-screening to record all the possible motions of every FB. On the other hand, this raises another question: "how to decide which one of them is the most likely true motion vector?"

Imposing Neighborhood Sensitivity Into Cost Function
It is observed that true motion field are piecewise continuous [1, 2, 4, 6, 10, 11, 13, 14, 15]. In a video sequence, the pixels of the same moving object are expected to move in a consistent way. Assuming translational motion only, the blocks associated with the same object should share exactly the same motion. At least there should be a good degree of motion similarities between the neighboring blocks. Therefore, the motion vector can be more robustly estimated if the global motion trend of an entire neighborhood is considered, as opposed to that of one feature block itself [6, 10]. This enhances the chance that a singular and erroneous motion vector may be corrected by its surrounding motion vectors [15]. For example, assume that there is an object moving in certain direction and a tracker fails to track its central block due to noise, but successfully track the boundary blocks. With neighborhood-sensitivity, the true motion of the central block could be recovered.

Therefore, instead of considering each feature block individually, we determine the motion of a feature block (say, $B_{i,j}$) by moving all its neighboring blocks ($\mathcal{N}(B_{i,j})$) with it in the same direction. A score function is introduced as the following [14]:

$$\begin{aligned} score(B_{i,j}, \vec{v}) &= DFD(B_{i,j}, \vec{v}) \\ &\quad + \sum_{B_{k,l} \in \mathcal{N}(B_{i,j})} (W(B_{k,l}, B_{i,j}) \times DFD(B_{k,l}, \vec{v})) \quad (1) \\ &= \text{image force (external energy)} \\ &\quad + \text{constraint forces (internal energy)} \end{aligned}$$

where $W(B_{k,l}, B_{i,j})$ is the weighting function which will be explained momentarily. The final solution can be obtained by finding the minimizing motion vector

$$motion\ of\ block(i,j) = \arg\left(\min_{\vec{v}}\ score(B_{i,j}, \vec{v})\right)$$

where $\vec{v}$ should be one of the possible candidates which are recorded by the multi-candidate pre-screening.

The central block's residue in the score function is called *image force* which is similar to the external energy function of SNAKE [7]. On the other hand, the neighbors' residue in the score function is called *constraint forces* which reflect the influence of neighbors, corresponding to the internal energy function of SNAKE.

In Equation (1), not all the neighboring blocks have the same weighting factors. In fact, they can be made to be dependent on several factors, such as the **distance** to the central block, the **confidence** of the blocks, the color/texture **similarity** between $B_{i,j}$ and $B_{k,l}$, etc. In terms of the distance, the weighting function could be a Gaussian-like function which puts higher emphasis in the central area. In terms of the confidence, it is practical to have low confidence blocks guided by their higher confidence neighbors [1]. In general, the larger variance a block has, the more confidence of feature tracking on that block. (An obvious example is that homogeneous blocks would not yield a high confidence.) Therefore, the weighting function must take into account the block variance. In addition, because different objects usually have different color/texture characteristics, we set $W(B_{k,l}, B_{i,j})$ to be proportional to the color/texture similarity between $B_{i,j}$ and $B_{k,l}$, so as to reduce the weights of the neighborhoods which contain different objects.

Neighborhood Relaxation for Non-Translational Motion
The above approach will be inadequate for non-translational motion, such as object rotating, zooming, and approaching [12]. For example, in Figure 2(b), assume an object is rotating counterclockwise. Because the Equation (1) assumes the neighboring blocks will move in the same translational motion, it may not adequately model the rotational motion. Since the neighboring blocks may not have uniform motion vectors, a neighborhood relaxation formulation is needed:

$$\begin{aligned} score(B_{i,j}, \vec{v}) &= DFD(B_{i,j}, \vec{v}) + \\ &\quad \sum_{B_{k,l} \in \mathcal{N}(B_{i,j})} (W(B_{k,l}, B_{i,j}) \times \min_{\vec{\delta}} DFD(B_{k,l}, \vec{v}+\vec{\delta})) \quad (2) \end{aligned}$$

where a small $\vec{\delta}$ is incorporated to allow some local variations of motion vectors among neighboring blocks. Again, the motion vector is obtained as

$$motion\ of\ block(i,j) = \arg\left(\min_{\vec{v}}\ score(B_{i,j}, \vec{v})\right)$$

As shown in Figure 2, the variation afforded to every block can be used to fine-tune the motions to accommodate the non-translational effect. This in principle can track more flexible affine-type motions, such as rotating, zooming, sheering, etc.

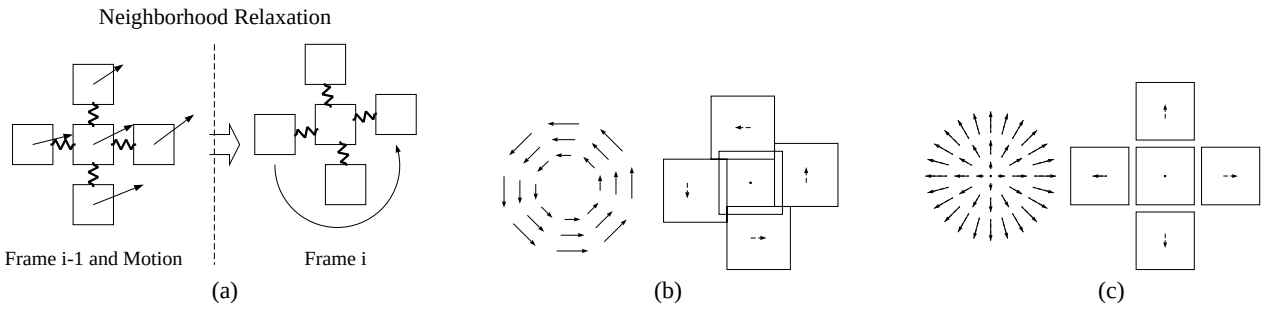

Figure 2: (a) Neighborhood relaxation will consider the global trend in object motion as well as provide some flexibility to accommodate non-translational motion. Local variations $\vec{\delta}$ among neighboring blocks, cf. Equation (2), are included in order to accommodate other (i.e. non-translational) affine motions such as (b) rotation, and (c) zooming/approaching.

2.3. Neighborhood Relaxation Algorithm with Multi-Candidate Pre-screening

The multi-candidate pre-screening must be applied before the minimization of the score function so that the minimal solution will be selected only from a restricted set of candidate solutions. The reasons for doing so are two fold: One obvious reason is to achieve computational saving by reducing the search space. More importantly, the second reason is due to an observation originally made by Anandan [1]: If there are discontinuities in the motion fields, such as those at the boundaries of different moving objects, then the neighborhood-sensitive score function might not yield a true motion vector. This can be explained as below. Recall that the validity of the neighborhood-sensitive score function is based on the smoothness of the motion fields. If there exist discontinuities, however, then there may be risk that some neighbors belonging to an alien object may have undue influence toward the selection of the final solution (since the neighbor's residues are part of the score function). Therefore, the score function could yield an apparently impossible motion vector, if the search space is allowed to expand all the vector space. This is especially vulnerable for the FBs which lie close to object boundaries. The multi-candidate pre-screening stage serves the purpose of reducing such a risk, since the central block retains some control over the final solution.

3. EXPERIMENT RESULTS

Figure 3 shows our simulation results with two consecutive frames, (a) & (b), and the corresponding blocks tracked by our method, (c) & (d). The performance comparison with other trackers is illustrated in Figure 4. Figure 4(b) shows that the conventional full-search block-matching algorithm is obviously poorest, with spurious vectors scattered around in homogeneous regions. Significant improvement is observed (Figure 4(c)) by applying a minimum threshold on variance to trim down the unreliable blocks. However, there are still noticeable tracking errors on the object boundaries (e.g. above and below the left-book). Figure 4(d) shows the result by our approach. The neighborhood relaxation has clearly helped regularize the motions of the block in the same region, e.g. the lower-left corner. The multi-candidate pre-screening has succeeded in eliminating many wrong motion vectors. Still, we have observed minor tracking errors on points along a long edge (e.g. below the left-book). A possible solution (currently under our study) is to adopt Anandan's scheme [1], in which the confidence (in $W(\cdot)$ of Equation 2) takes into account the cross-correlation between vertical/horizontal components of (1) the motion vector and (2) image features. Its purpose is that the motion of a block which has low confidence in vertical movement can be guided by its neighbors which has higher confidence in vertical movement.

The tracked results could subsequently used for motion-based scene segmentation. An algorithm based on the principal component coordinate transformation on the *feature track matrix* (formed by the tracked FBs) can be applied for separating image layers or moving objects [9]. As shown in Figure 5, (a) our tracker captures the true motion vectors. (b) Feature blocks with different object-based motions tend to form separate clusters on the principal component space. (c) The frame can thus be divided into different layers (segments) characterized by a consistent motion. Finally, (d) a fairly accurate motion-compensated frame can be obtained.

4. REFERENCES

[1] P. Anandan, "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *Int'l J. of Computer Vision*, vol. 2, no. 3, pp. 283–310, 1989.

[2] J. L. Barron, D. J. Fleet, and S. S. Beacuchemin, "*Systems and Experiment* Performance of Optical Flow Techniques," *Int'l J. of Computer Vision*, vol. 13, no. 1, pp. 43–77, 1994.

[3] Y.-K. Chen and S. Y. Kung, "A Multi-Module Minimization Neural Network for Motion-Based Scene Segmentation," in *IEEE Workshop on Neural Networks for Signal Processing*, (Kyoto, Japan), 1996.

[4] G. de Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, "True-Motion Estimation with 3-D Recursive Search Block Matching," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, no. 5, pp. 368–379, 1993.

[5] L. Dreschler and H.-H. Nagel, "Volumetric Model and 3D Trajectory of a Moving Car Derived from Monocular TV Frame Sequences of a Street Scene," *Computer Graphics and Image Processing*, no. 20, pp. 199–228, 1982.

[6] F. Dufaux and F. Moscheni, "Motion Estimation Techniques for Digital TV: A Review and a New Contribution," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 858–876, 1995.

[7] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[8] M. Klima, P. Dvořák, P. Zahradnik, J. Kolář, and P. Kott, "Motion Detection and Target Tracking in a TV Image for Security Purposes," *Proceedings of IEEE Annual Int'l Carnahan Conference on Security Technology*, pp. 43–44, 1994.

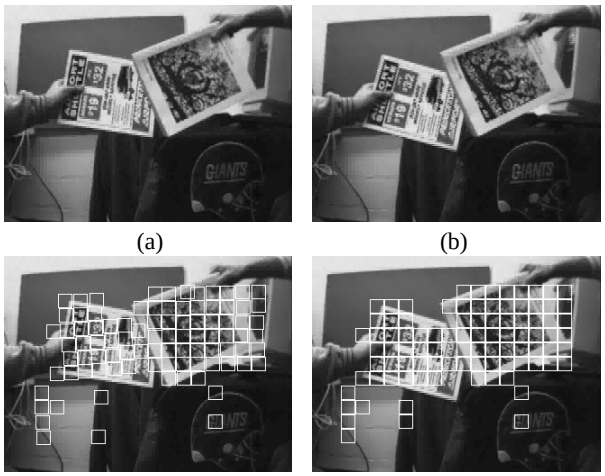

Figure 3: (a) and (b) show 2 consecutive frames of 2 rotating books amid a panning background, (c) and (d) show the tracked FBs by our approach.

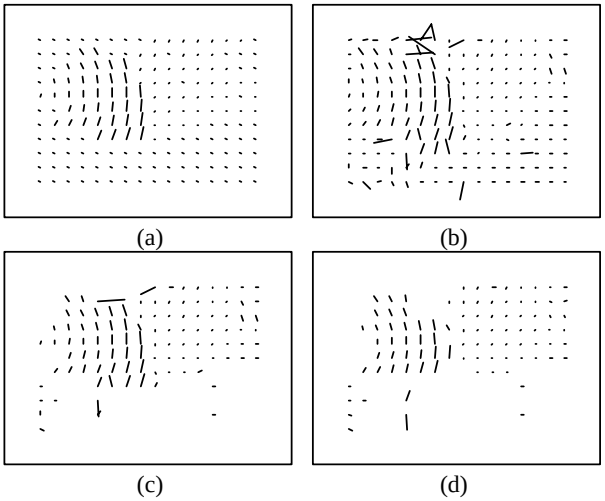

Figure 4: (a) shows the true motion field corresponding to Figure 3. (b) the motion vectors by conventional full-search block-matching algorithms. (c) the motion vectors by conventional BMA with variance threshold applied. (d) the motion vectors by our approach.

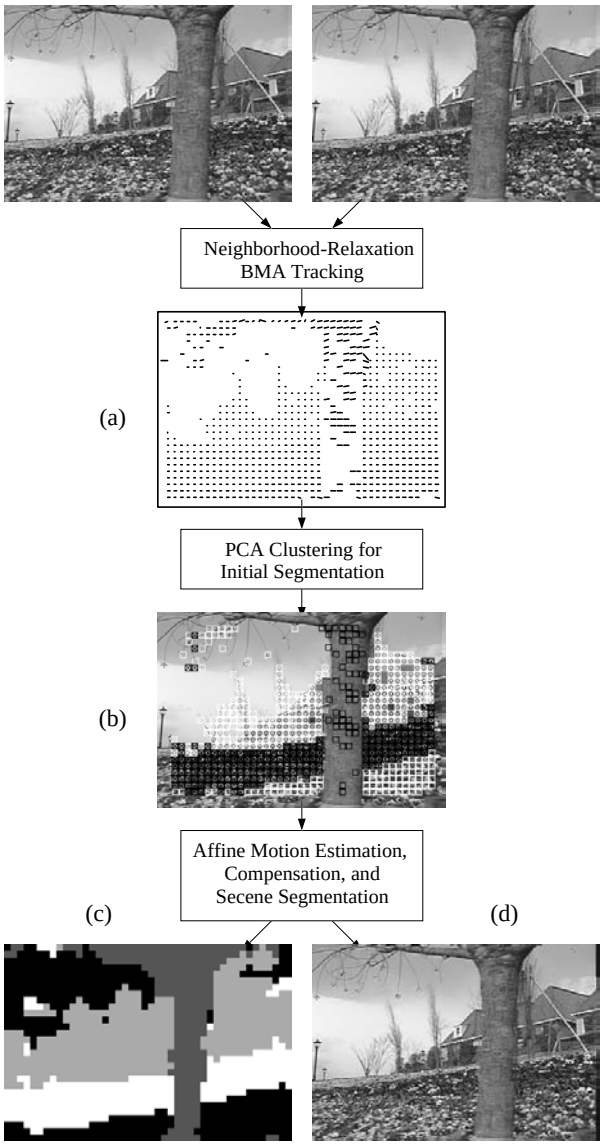

Figure 5: A flow chart of a motion-based segmentation by principal component analysis (PCA) clustering method. First, several frames of a video sequence are input to the feature tracker. After (a) the tracking of moving features is obtained, PCA is used to separate the tracked features into (b) several motion clusters. Finally, affine motion estimation and affine motion compensation test for each clusters are applied so that (c) the scene is segmented and (d) affine reconstructed image is obtained.

[9] S. Y. Kung, Y.-T. Lin, and Y.-K. Chen, "Motion-Based Segmentation by Principal Singular Vector (PSV) Clustering Method," *Proceedings of ICASSP '96*, pp. 3410–3413, 1996.

[10] J.-B. Lee and S.-D. Kim, "Moving Target Extraction and Image Coding Based on Motion Information," *IEICE Trans. Fundamentals*, vol. E78-A, no. 1, pp. 127–130, 1995.

[11] M. T. Orchard, "Predictive Motion-Field Segmentation for Image Sequence Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, no. 1, pp. 54–70, 1993.

[12] J. M. Rehg and A. P. Witkin, "Visual Tracking with Deformation Models," *Proceedings of IEEE Int'l Conference on Robotics and Automation*, pp. 844–850, 1991.

[13] V. Seferidis and M. Ghanbari, "Generalized Block-Matching Motion Estimation Using Quad-Tree Structured Spatial Decomposition," *IEE Proc.-Vis. Image Signal Process*, vol. 141, no. 6, pp. 446–452, 1994.

[14] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: a Factorization Method–Part 3, Detection and Tracking of Point Features," Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, 1991.

[15] J. Y.-A. Wang and E. H. Adelson, "Representing Moving Images with Layers," *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp. 625–638, 1994.