An Efficient Moving Object Detection and Description

Tao Xia[1], Chaoqiang Liu[2], Hui Li[2]
Centre for Wavelets, Approximation and Information Processing,
National University of Singapore[1]
Temasek Laboratories and Centre for Wavelets, Approximation and Information Processing,
National University of Singapore[2]

***Abstract*— In this paper, a new algorithm of moving objects detection and description is proposed. Based on the analysis of projection of the 3D motion of objects, the information of motion field is exploited to make moving object detection more efficient. The discontinuities of motion vector field on the boundaries of moving objects enable us to detect the moving objects blocks in which the potential boundaries of the moving objects locate. A further refinement of the boundary of moving objects and efficient descriptor for moving objects are proposed as well. The simulation results show that each moving object is successfully identified.**

***Index Terms*— Moving object detection, motion vector field, perspective projection.**

I. Introduction

Moving object detection techniques have been studied extensively for such purposes as video content analysis as well as remote surveillance. There are many ways to track the moving object [1]-[6]. Most of them use the frame differences to analysis the moving object and obtain object boundary, which may be quite time consuming. In order to reduce the computation cost, a more efficient way is expected. From this point of view, a new moving object detection algorithm is proposed to exploit information of motion field. As motion vectors introduce the discontinuities on the boundaries of moving objects, basic idea here is to use the discontinuous points in the motion field as the boundary of the moving objects.

II. Moving Object Detection Algorithm

The process of proposed moving object detection algorithm is presented in Figure. 1. At the first stage, block-based motion estimation is used to obtain the coarse motion vectors, the vectors for each block, where the central pixel of the block is regarded as the key point. These motion vectors are used to detect the boundary blocks, which contain the boundary of the object. Later on, the linear interpolation is used to make the coarse motion field a dense motion field, by this way to remove the block artifacts. This property can also be used to detect whether the motion field is continuous or not. This refined dense motion field is used to define detail boundaries in each boundary block. At last, moving object is detected and coded.

This algorithm could be divided into two parts, one is the motion field study and the other is object detection method, which will introduced in following sections.

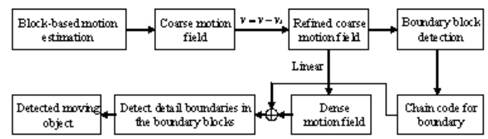

Fig. 1. process of moving object detection algorithm

III. Motion Vector Field of Rigid Objects

As the ideal imaging model, perspective projection model is adopted in this paper. Under this model, motion vector corresponding the 3-D motion of rigid objects have following properties.

Proposition 1: Define $f(\mathbf{P}) = m_{\mathbf{P}}/P_z$, where $m_{\mathbf{P}}$ and P_z are the 3-D motion and the $z-$ coordinate of point $\mathbf{P}$ respectively, if for three points $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ locating in a surface of an object,

1) $\mathbf{P}_3 = \lambda \mathbf{P}_1 + (1-\lambda)\mathbf{P}_2$ and,
2) The surface containing $\mathbf{P}_1, \mathbf{P}_2$ and $\mathbf{P}_3$ is a plane surface. and,
3) $f(\mathbf{P}_1) = f(\mathbf{P}_2) = f(\mathbf{P}_3)$,

then

$$\mathbf{v}_{\mathbf{P}_3} = \lambda \mathbf{v}_{\mathbf{P}_1} + (1-\lambda)\mathbf{v}_{\mathbf{P}_2} \tag{1}$$

where $\mathbf{v}_{\mathbf{P}}$ is the motion vector in image plane associated with 3-D point $\mathbf{P}$.

This proposition shows that for any three points in a line, if condition 2) and 3) are satisfied. Then their associated motion vectors are linear interpolated as well. The condition 3) means the motion of all along the z-direction should be same. These two conditions almost hold for arbitrary 3-D objects. It is reasonable to assume the motion in same objects should be same. In addition, it is also convenient considering the connected objects with same motion as one moving object. The equation (1) shows that under these constraints, the motion vector associated with the points in one moving object should be linear interpolated. That will introduce the following algorithms.

IV. Moving Object Detection Algorithms

In this section, object detection algorithms are introduced step by step as detecting boundary blocks, writing chain codes, and detecting boundaries in the boundary blocks, where the boundary block is defined as a block containing boundary of a certain moving object.

To simplify the moving object boundary detection, some assumptions are drawn:

1) At most two objects' boundaries in one block;
2) Any object cannot be hold in one block;
3) The boundary of moving object is a simple closed curve.

The idea of proposed moving objects detection is to exploit the motion vector detected in video compression as the clue for the moving objects and then refine the moving objects boundaries. This scheme will speed up the moving object detection significant.

A. Boundary Block Detection

At first, we will detect the blocks containing moving object boundaries by using the information of the motion vector field generated by video compression. The motion vector derived by motion estimation is the common motion vector for all pixels in one block. However, we can assume it is the motion vector of the critical points locating in the centroid of each block. As mentioned above, pixels in the same object should satisfy the condition of linear interpolation. Hence the critical points in one moving objects should satisfy this condition. When this condition is violated, the discontinuity of motion vector field exist and imply the existence of the boundary in associated block.

Then the algorithm detecting blocks containing moving objects'boundaries is as follows:

Algorithm 1: **moving object block detection**

1) Scan along horizontal direction, for three consecutive critical points $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, which are central points of each block and satisfying $2\mathbf{P}_2 = \mathbf{P}_1 + \mathbf{P}_3$, if $2\mathbf{v}_{\mathbf{P}_2} \neq \mathbf{v}_{\mathbf{P}_1} + \mathbf{v}_{\mathbf{P}_3}$, then $\mathbf{P}_2$ might be a boundary block.
2) Scan along vertical direction, for three consecutive critical points $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, which are central points of each block and satisfying $2\mathbf{P}_2 = \mathbf{P}_1 + \mathbf{P}_3$, if $2\mathbf{v}_{\mathbf{P}_2} \neq \mathbf{v}_{\mathbf{P}_1} + \mathbf{v}_{\mathbf{P}_3}$, then $\mathbf{P}_2$ might be a boundary block. Now the possible boundary blocks are selected, for further verification, another condition is provided.
3) For six consecutive critical points $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$, $\mathbf{P}_5$, $\mathbf{P}_6$, which are in a same line, and $\mathbf{P}_3$, $\mathbf{P}_4$, $\mathbf{P}_5$ are possible boundary blocks, while $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_6$ are not. The central block $\mathbf{P}_4$ is the boundary block.

The above algorithm does not guarantee the moving objects boundary of closed simple curve. Thus we the following algorithm to find the simple closed curve corresponding the boundary of moving objects.

Then we describe the boundary blocks by chain code. 8-neighboring connectivity is used here. The number 0 to 7 denotes neighboring blocks. Initial search starts from block 0, then round the target block, from block 1 to block 7. During the search, once a neighboring block is found as a boundary block, then the search will stop and the neighboring block will be regarded as current target block to start 8 neighboring blocks search iteratively. The searching algorithm applied to blocks is as follows:

Algorithm 2: **Forming the closed curve of boundary blocks**

1) Let $m = 0$, Scanning from the left top corner block of the blocks in image to find the first boundary block
2) The first boundary block of mth object is denoted by $\mathbf{B} = \mathbf{B}_0 = (i, j)$, initialize the current direction as horizontal right $c = 4$; The boundary blocks of moving object m, is denoted by $\mathbf{S}_m = \{(i, j), c\}$
3) Define the 8-neighboring blocks of $\mathbf{B}$ as $\{\mathbf{B}_{c_t}\}$ where $c_t = (c - 3 + t) \mod 8$, $t = 0, ..., 7$, and $\mathbf{B}_{c_t}$ denotes the block searching from $\mathbf{B}$ along direction c_t.
4) Searching the next boundary block of $\mathbf{B}$ in the set $\{\mathbf{B}_{c_t} | t = 0, \cdots, 7\}$ to find the next boundary block. Once it is found, it is defined as $\mathbf{B}_1$ and $c = c_t$ is the current chain code.
5) If $\mathbf{B}_1 \neq \mathbf{B}_0$, then $\mathbf{B} = \mathbf{B}_1, \mathbf{S}_m \cup \{c_t\} \to \mathbf{S}_m$, and $c_t \to c$; go to the step 3)
6) $m + 1 \to m$, searching the first boundary block for the mth object in remain blocks. If it is successful, then go to the step 2);
7) End.

The chain code of mth moving object is $\mathbf{S}_m = \{(i, j), c_1, c_2, c_3, ..., c_j\}$, where (i, j) means the first boundary block is (i, j)th block in the image.

B. Detection of the detail boundaries of moving objects

To simplify, we use the line segment in one block to approximate the actual boundary of moving objects in that block. In the other words, we use polygon to approximate the moving objects boundary. Therefore, each boundary block obtained in algorithm 2 is replaced by a line segment. That will be the algorithm of detecting detail boundary of moving objects.

Assume there are M moving objects $\Psi = \{\Psi_i | i = 1...M\}$. Please pay attention here, in the following contents, the subscript i, j of $\mathbf{B}_{i,j}$ does not denote the position of block $\mathbf{B}$, instead, i denotes the ith moving object and j denotes the jth boundary block of ith moving object.

As mentioned above, each moving object can be represented by a sequence $\Psi_i = (\mathbf{B}_{i,0}, d_{i,1}, d_{i,2}, ..., d_{i,L^i})$, where $\mathbf{B}_{i,0}$ is the index of the initial boundary block and $\mathbf{B}_{i,0}(x, y)$ is used to denote its location, $d_{i,j}$ is the chain code from block $B_{i,j-1}$ to block $B_{i,i}$, i.e., $\mathbf{B}_{i,j} = N_b(\mathbf{B}_{i,j-1}, d_{i,j}), j = 1, ..., L^i$, and the chain code is defined clockwise in the 8-connected. The intensity function of the frame k is f_k and an indicator function will be generated for every boundary block.

Algorithm 3: **Fine moving objects boundary detection**

1) $i = 1, \Psi' = \phi$.
2) $\Psi'_i = \phi, j = 1$.
3) $\mathbf{B}'_{i,j} = N_b(\mathbf{B}_{i,j-1}, d_{i,j} - 1)$,suppose $\mathbf{P}_{i,j}\mathbf{Q}_{i,j}$ is the common edge of $\mathbf{B}_{i,j}$ and $\mathbf{B}'_{i,j}$.
4) If motion vector $\mathbf{v}(\mathbf{B}_{i,j}) = \mathbf{v}(\mathbf{B}'_{i,j})$
   $g(\mathbf{P}) = |f_k(\mathbf{P} + \mathbf{v}(\mathbf{B}'_{i,j})) - f_{k-1}(\mathbf{P})| - |f_k(\mathbf{P} + \mathbf{v}(\mathbf{B}_{i,j})) - f_{k-1}(\mathbf{P})|, \forall \mathbf{P} \in \mathbf{B}_{i,j}$
   else
   $g(\mathbf{P}) = |f_k(\mathbf{P} + \mathbf{v}(\mathbf{B}'_{i,j})) - f_{k-1}(\mathbf{P})| - T_0, \forall \mathbf{P} \in \mathbf{B}_{i,j}$
   where T_0 is predetermined threshold.
5) Generate indicator function
   $$I(\mathbf{P}) = \begin{cases} 1, & \text{if } g(\mathbf{P}) > 0 \\ 0, & \text{if otherwise} \end{cases}, \forall \mathbf{P} \in B_{i,j}$$
   .
6) If $I(\mathbf{P}_{i,j})I(\mathbf{Q}_{i,j}) = 1$, go to step 9).
7) $$\mathbf{P}^0_{i,j} = \begin{cases} \mathbf{P}_{i,j}, & \text{if } I(\mathbf{P}_{i,j}) = 0 \\ \mathbf{Q}_{i,j}, & \text{if otherwise} \end{cases}$$
8) Regarding $\mathbf{P}^0_{i,j}$ as the initial point, to find the first pixel whose indicator is nonzero at the boundaries of $\mathbf{B}_{i,j}$ along the counter clockwise direction to replace $\mathbf{P}_{i,j}$, and find the first pixel whose indicator is nonzero at the boundaries of $\mathbf{B}_{i,j}$ along the clockwise direction to replace $\mathbf{Q}_{i,j}$.
9) $\psi'_i = \psi'_i \cup \{(d_{i,j}, \mathbf{P}_{i,j}, \mathbf{Q}_{i,j})\}$
10) If $j < L^i, j = j + 1$ then go to step 3.
11) $\Psi'_i = \Psi'_i \cup \{\psi'_i\}$
12) if $i < M, i = i + 1$ then go to step 2.
13) End.

Remarks:

1) Ψ' is the set for the detail boundaries of all moving objects.
2) ψ'_i is a closed detail boundary of ith moving object.
3) The outside neighbor of boundary block $\mathbf{B}_{i,j}$ is block $\mathbf{B}'_{i,j}$ (as depicted in Figure. 2).
4) In indicator function I for block, in which 1 denotes the pixel in moving objects and 0 denotes pixel in outside region.
5) The position of endpoints $\mathbf{P}_{i,j}, \mathbf{Q}_{i,j}$ in step 9) can be represented by arc length coordinate which require 4 bits only.

V. Simulation Results

The simulation results are presented as shown in Figure. 3, Figure. 4 and Figure. 5. As shown in Figure. 3, the connected outline is the object boundary in the motion field, which is also the chain code. The short white lines are the motion vectors for each block, while the grey short line surrounded by short white lines denotes for the block which is not boundary block. As shown in Figure. 4 and Figure. 5, the results are moving objects in previous frame and current frame separately.

As displayed in the figures, the moving object can be tracked and the vision effect is good.

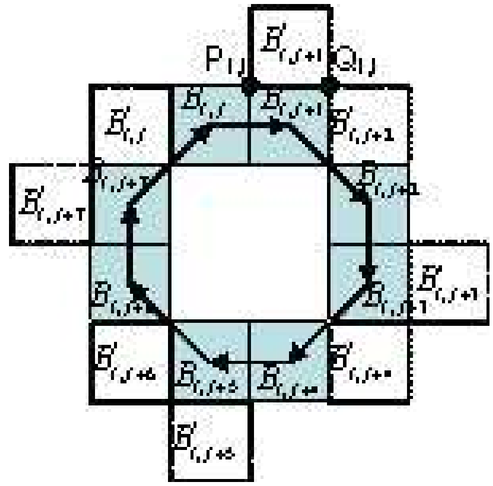

Fig. 2. Outside block $B'_{i,j}$

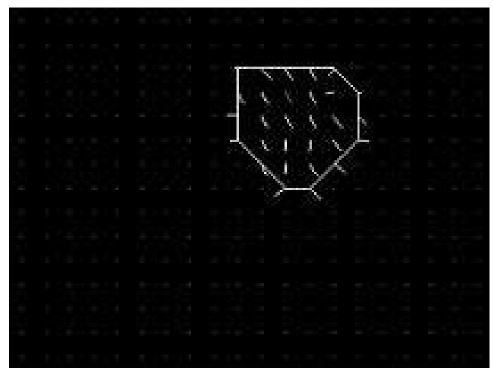

Fig. 3. Boundary block detection

VI. Conclusion

In short, a more efficient moving object detection algorithm is suggested for video compression. First of all, the coarse motion vectors are obtained through block-based motion estimation. Then these motion vectors are used to detect the boundary blocks. Later on, a dense motion field is formed by linear interpolation to define detail boundaries in each boundary block. At last, moving object is detected and coded. Simulation results show that moving object could be identified correctly with good vision effect.

Acknowledgment

The authors would like to thank DSTA funding for support of the programme ”Wavelets and Information Processing”.

References

[1] H.Y. Zhang and B.Z. Yuan, “A multi-resolution image matching algorithm for moving object detection by wavelets”, *ISSIPNN Proceedings*, vol. 1, pp. 276-279, April, 1994.
[2] D. Li, “Moving objects detection by block comparison”, *Electronics, Circuits and Systems*, vol. 1, pp. 341-344, Dec, 2000.
[3] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, “Statistic and knowledge-based moving object detection in traffic scenes”, *IEEE Proceedings. Intelligent Transportation Systems*, pp. 27-32, Oct, 2000.

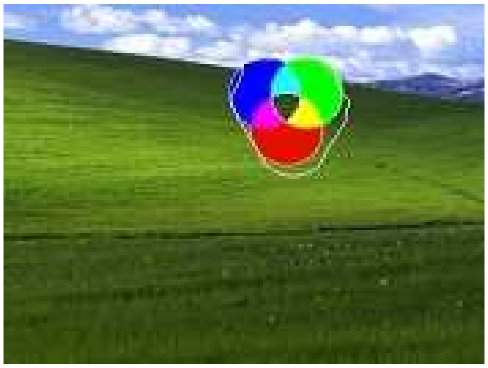

Fig. 4. Moving object in previous image

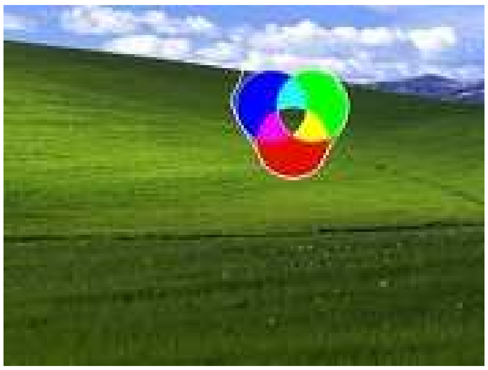

Fig. 5. Moving object in current image

[4] J.M. Odebez and P. Bouthemy, “Detection of multiple moving objects using multiscale MRF with camera motion compensation”, *ICIP 1994*, vol. 2 , pp. 257-261, Nov, 1994.

[5] Y.K. Jung, K.W. Lee and Y.S. Ho, “Content-based event retrieval using semantic scene interpretation for automated traffic surveillance”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, pp. 151-163, Sep, 2001.

[6] R. Montoliu and F. Pla, “Multiple parametric motion model estimation and segmentation”, *ICIP 2001*, vol. 2, pp. 933-936, Oct, 2001.

[7] T. Xia, “Wavelets-based colour image sequence compression”, PHD Thesis, School of Mathematical Science, Peking University, 1997.