Correspondence Search Algorithm for Feature Tracking with Incomplete Trajectories

Jong Myeon Jeong, Young Shik Moon
Dept. of Computer Science & Engr., Hanyang University
1271, Sa-1 dong, Ansan-City Kyunggi-do, 425-791, Korea
Tel : +82-345-400-5196, Fax : +82-345-419-1162
E-mail : {jmjeong, ysmoon}@cse.hanyang.ac.kr

Abstract : The correspondence problem is known to be difficult to solve because false positives and false negatives almost always exist in real image sequences.

In this paper, we propose a robust feature tracking algorithm considering incomplete trajectories such as entering and/or vanishing trajectories. We solve the correspondence problem as the optimal graph search problem, by considering false feature points and by properly reflecting motion characteristics. The proposed algorithm finds a local optimal correspondence so that the effect of false feature points can be minimized in the decision process. The time complexity of the proposed graph search algorithm is given by $O(mn)$ in the best case and $O(m^2n)$ in the worst case, where m and n are the number of feature points in two consecutive frames. The proposed algorithm can find trajectories correctly and robustly, which has been shown by experimental results.

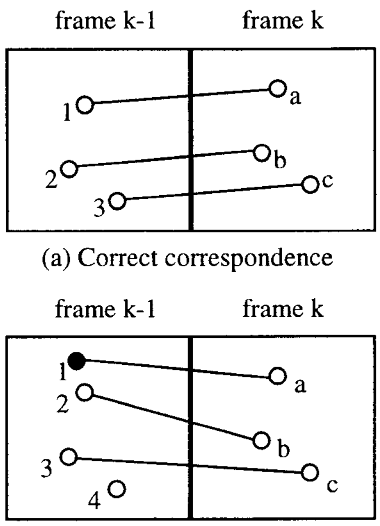
I . Introduction

Given a sequence of image frames, the problem of finding the correspondence of feature points is an important domain of motion analysis in numerous applications. However, the correspondence problem is known to be difficult to solve because false positives and/or false negatives almost always exist in real image sequences. To produce trajectories correctly, feature tracking algorithms must handle the self-initialization, incomplete trajectories, distortion of feature points, huge amount of feature points, etc.. However, most existing feature tracking algorithms[1]-[7] cannot handle all of these issues.

In this paper, we propose a robust feature tracking algorithm considering incomplete trajectories such as entering and/or vanishing trajectories. The trajectories of feature points can be initialized and tracked automatically using attributes of feature points.

II. Correspondence search problem

The feature sets extracted from given sequences almost always include the false positives and false negatives. These false positives and false negatives have an effect on the correspondence search procedure. Consequently,



(b) Correspondence error caused by a false positive

Fig. 1 Errors in correspondence search problem the correspondences may include some false matching results. If there exists only one false correspondence in the matching procedure, the error propagates to the subsequent matching procedure as shown in Fig. 1, where the black dot represents a false feature point.

Ⅲ. Feature vector and matching measure

In this paper, corner points in each image are selected as the feature points, where a corner point is defined as the point whose variation of the edge angle is a local minimum in its neighborhood as shown in Fig. 2. The attributes of a feature point including spatial coordinate, vertex angle, vertex orientation and motion vector can be represented by a 6-dimensional feature vector as given in equation (1).

$$X_k^i = [x_k^i, y_k^i, p_k^i, h_k^i, d_k^i, m_k^i,]' \qquad (1)$$

where X_k^i is the i-*th* feature point in the k-*th* frame, (x_k^i, y_k^i) is the spatial coordinate, p_k^i and h_k^i are the vertex angle and vertex orientation and d_k^i and m_k^i are the direction and the speed of motion for the i-*th* feature point in the k-*th* frame.

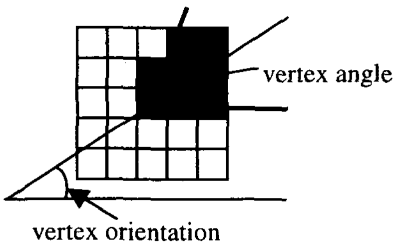


Fig. 2 Definition of feature point

We use the weighted Euclidean distance between two feature vectors as a matching measure as given in equation (2).

$$D_k^{ij} = (X_{k-1}^i - X_k^j)' W (X_{k-1}^i - X_k^j) \qquad (2)$$

where D_k^{ij} is the Euclidean distance between i-*th* point in the (k-1)-*th* frame and the j-th point in k-*th* frame that is closest to the i-*th* point in the (k-1)-*th* frame and W is the weight matrix.

W is adjusted so that the attributes with small variance give more emphasis on the correspondence decision. The weights are automatically updated in order to properly reflect the motion characteristics, as described in [7].

Ⅳ. Feature tracking algorithm

To decide the correspondence between feature points in two frames, most existing feature tracking algorithms select two points that minimize or maximize the matching measure. But, as stated in Ⅱ, the correspondence selection scheme that simply minimizes or maximizes the matching measure is not appropriate in real applications, where numerous false feature points may exist. In order to solve the correspondence problem in the presence of false feature points, we construct a weighted bipartite graph *G(N, E)* as shown in Fig. 3. In the graph, *G(N, E)*, the nodes are feature points in two consecutive frames and the weighted edges are the Euclidean distances between two feature points. Fig. 3 shows an example of bipartite graph for Fig. 1(b). In Fig. 3, the weights of edges are the Euclidean distances between two points in Fig. 1(b). We consider the correspondence search problem as a graph search problem to find the node set that has the minimum weighted edges. The proposed graph search algorithm can be described as the following pseudo code.

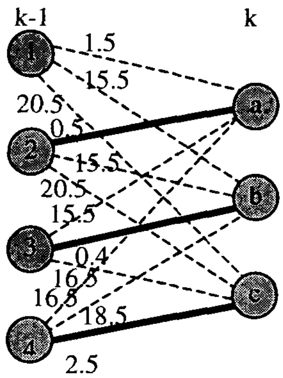


Fig. 3 A correspondence search algorithm using weighted bipartite graph

```
WHILE ( there exists a node in the graph) BEGIN
  Step 1. Initialization
        1.1 Eg     arbitrary value
        1.2 Pe     arbitrary value that is different
             from Eg
        1.3 Nd     arbitrary node that exists in the
             frame that contains small number of
             feature points
  Step 2. Optimal correspondence search
      WHILE ((Pe     Eg) and (there exists an edge
        in the graph)) BEGIN
        2.1 Pe     Eg
        2.2 Eg     the minimum weight among the
            weighted edges of Nd
        2.3 Nd     the node that is connected to Nd
            with the minimum weighted edge in
            step 2.2
      END WHILE;
  Step 3. Eliminate the selected corresponding nodes
        from the graph
      IF (Pe = Eg) THEN BEGIN
        3.1 Select the two nodes connected by Eg
            as the optimal corresponding nodes
        3.2 Eliminate the two nodes connected to
            Eg from the graph
        3.3 Eliminate all the edges of nodes
            connected to Eg from the graph
      END
      ELSE BEGIN
        3.4 Terminate correspondence search
            procedure
      END IF;
END WHILE;
```

The proposed algorithm finds a local optimal correspondence so that the effect of false feature point can be minimized in the decision process.

On the other hand, the unmatched feature point in the current frame that was not assigned to any trajectory can create a new trajectory, referred to as an entering trajectory. Also, the feature points that were assigned to trajectories in previous frames may not be assigned to trajectories in current frame because their trajectories disappear. To handle these cases, we propose the following decision criteria.

1) **Extension of existing trajectories** : Extend the trajectories to the correspondences between the feature points that were assigned to trajectories in (k-1)-*th* frame and the feature points in the k-*th* frame. Extended trajectories and feature points are excluded in further process.

2) **Vanishing trajectories** : Among the feature points that were assigned to trajectories in (k-1)-*th* frame, the points with no corresponding point in k-*th* frame are vanishing trajectories.

3) **Entering trajectories** : Find the corresponding points in the k-*th* frame for the feature points that were not assigned to any trajectory in the (k-1)-*th* frame. If we can find a correspondence in this step, it is a new trajectory, i. e. entering trajectory.

The time complexity of the proposed graph search algorithm is given by $O(mn)$ in the best case and $O(m^2n)$ in the worst case, where m and n are the number of feature points in two consecutive frames.

Ⅴ. Experimental results

The proposed algorithm has been implemented and tested on a Pentium Ⅱ MMX 300 Mhz PC. Fig. 3 shows the results of corner extraction for the initial 8 frames of box sequence[8]. As shown in Fig. 4, there exist many false positives and false negatives which have serious effects on the correspondence search. Fig. 5 shows the correspondence of feature points in the consecutive frames. It is noted that the errors marked by arrows do not propagate to the subsequent frames. Fig. 6 shows the trajectories of feature points. In Fig. 6(a), the trajectory marked by arrow ① is an incorrect trajectory and the trajectory marked by arrow ② is a correct trajectory. In the subsequent tracking procedure as shown in Fig. 6(b)-(g), the incorrect trajectory is not extended further but the correct trajectory is fully extended. In Fig. 6(c), the trajectory marked by arrow ③ is an entering trajectory, which is correctly tracked in the subsequent frames as shown in Fig. 6(d)-(g). In Fig. 6(e), the trajectory marked arrow ④ is a vanishing trajectory, which is not extended further in the subsequent frames. It has been shown that the proposed algorithm can find trajectories correctly and robustly.

Ⅵ. Conclusions

In this paper, we proposed robust algorithm for tracking feature points with incomplete trajectories. The weighted Euclidean distance is employed to determine the correspondence between two points. The weights are automatically updated in order to properly reflect the motion characteristics. For tracking feature points with incomplete trajectories, we represent the feature points in consecutive frames as s weighted bipartite graph. Then we consider the correspondence search problem as the weighted bipartite graph search problem. Proposed method can track feature points robustly, which has been shown by experimental results.

Acknowledgement

This work was supported by the Brain Korea 21 Project

References

[1] I. K. Sethi and R. Jain, "Finding Trajectories of Feature Points in an Monocular Image Sequence," *IEEE Trans. PAMI.*, Vol. 9, No. 1, pp. 56-73, 1987.

[2] V. Salari and I. K. Sethi, "Feature Point Correspondence in the Present of Occlusion," *IEEE Trans. PAMI.*, Vol. 12, No. 1, pp. 87-91, 1990.

[3] K. Rangarajan and M. Shah, "Establishing Motion Correspondence," *CVGIP : Image Understanding*, Vol. 54, No. 1, pp. 56-73, 1991.

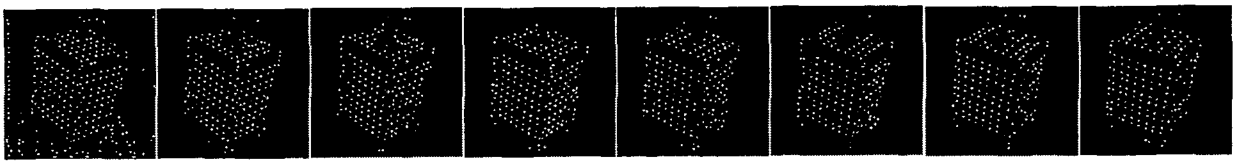
[4] S. Krishnan and D. Raviv, "2D Feature Tracking Algorithm for Motion Analysis," *Pattern Recognition*, Vol. 28, No. 8, pp. 1103-1126, 1995

[5] R. Mehrotra, "Establishing Motion-based Feature Point Correspondence," *Pattern Recognition*, Vol. 31 No. 1, pp. 23-30, 1998.

[6] D. Chetverikov and J. Verestoy, "Tracking Feature Points : A New Algorithm," *Proc. ICPR 98*, pp. 1436-1438, 1998.

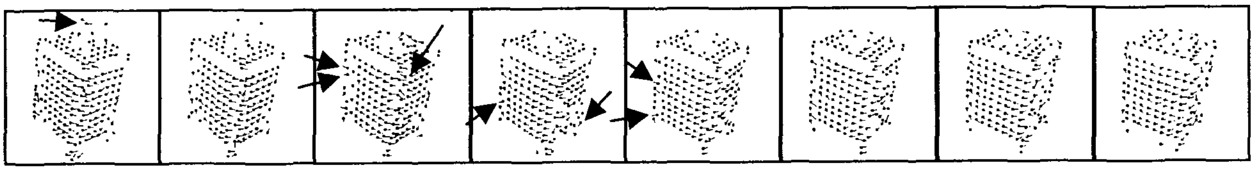
[7] Jong Myeon Jeong, Young Shik Moon, "A Feature Tracking Algorithm Using Adaptive Weight Adjustment," Journal of the IEEK., Vol. 36-S, No. 11, pp.68-78, 1999

[8] http://periscope.cs.umass.edu/~vislib/Motion/box/images.htm



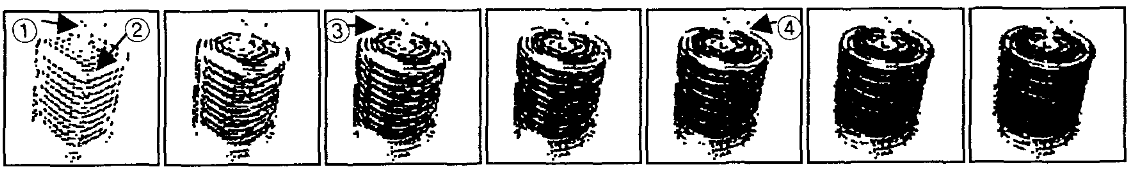
(a) frame 1 (b) frame 2 (c) frame 3 (d) frame 4 (e) frame 5 (f) frame 6 (g) frame 7 (h) frame 8

Fig. 4 Extracted feature points in the box sequence



(a)frame 1-2 (b)frame 2-3 (c)frame 3-4 (d)frame 4-5 (e)frame 5-6 (f)frame 6-7 (g)frame 7-8 (h)frame 8-9

Fig. 5 The correspondence of the feature points in the consecutive frames



(a)frame 1-4 (b)frame 1-8 (c)frame 1-12 (d)frame 1-16 (e)frame 1-20 (f)frame 1-24 (g) frame 1-28

Fig. 6 The trajectories of the feature points in the box sequence