

# 점 계열 유사도에 기반한 모션 대응 알고리즘

## (A Motion Correspondence Algorithm based on Point Series Similarity)

엄기열<sup>†</sup>      정재영<sup>\*\*</sup>      김문현<sup>\*\*\*</sup>  
 (Kiyeol Eom)      (Jaeyoung Jung)      (Moonhyun Kim)

**요약** 본 논문에서는 점 계열 유사도에 기반하여 모션 대응에 대한 휴리스틱 알고리즘을 제안한다. 점 계열은 x좌표를 기준으로 하여 오름 차순으로 정렬된 일련의 점들의 리스트이다. 본 연구에서는 국부 근접성에 기반하여 현재 프레임 전 프레임의 점들을 클러스터링 하고, 각 그룹에 있는 점들의 순서를 변경해 가면서 여러 개의 가상 점 계열들을 구성하며, 가상 점 계열은 근접 제약조건에 근거한 유사도를 이용하여 현재 프레임의 점 계열과 정합한다. 국부적 애매함을 해결하기 위해 두 개 점 계열간의 가장 긴 부분 문자열을 찾는 방법이 전체적인 정보로 사용된다. PETS2009과 CAVIAR데이터 집합들과 같은 다양한 영상 이미지 시퀀스에 대해 본 연구의 모션 대응 알고리즘은 90%이상의 정확도를 보여준다.

**키워드** : 모션대응, 객체 추적, 계열 유사도, 궤도, 클러스터링, 최장 공통 부분열

**Abstract** In this paper, we propose a heuristic algorithm for motion correspondence based on a point series similarity. A point series is a sequence of points which are sorted in the ascending order of their x-coordinate values. The proposed algorithm clusters the points of a previous frame based on their local adjacency. For each group, we construct several potential point series by permuting the points in it, each of which is compared to the point series of the following frame in order to match the set of points through their similarity based on a proximity constraint. The longest common subsequence between two point series is used as global information to resolve the local ambiguity. Experimental results show an accuracy of more than 90% on two image sequences from the PETS 2009 and the CAVIAR data sets.

**Key words** : motion correspondence, object tracking, series similarity, trajectory, clustering, longest common subsequence

## 1. Introduction

Establishing motion correspondence is an impor-

tant research topic in computer vision applications, such as motion analysis, object tracking and surveillance [1,2]. In motion correspondence, the goal is to track points that are moving in a sequence of frames. It is a challenging problem, especially in situations where the number of points in each frame is different due to various reasons. Local methods are efficient but have difficulty determining locally ambiguous regions with densely moving point sets. In order to resolve these ambiguities, global methods use additional information provided from the point sets, such as geometrical relationships [3] or a global motion model [4]. In the global motion model, the motion smoothness constraint is imposed not only on individual points, but also on the complete set of points [4].

\* 본 연구는 건설교통부 도시철도표준화 2단계연구개발사업의 연구비지원 (07도시철도표준화A01)에 의해 수행되었습니다.

† 학생회원 : 성균관대학교 정보통신공학부  
eomkiyol@empal.com

\*\* 종신회원 : 동양대학교 컴퓨터정보전학과 교수  
jyjung@dyu.ac.kr

\*\*\* 종신회원 : 성균관대학교 정보통신공학부  
mhkim@ece.skku.ac.kr

논문접수 : 2009년 11월 19일

심사완료 : 2010년 2월 13일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제37권 제4호(2010.4)

Sethi and Jain [5] propose an iterative algorithm to find the trajectories of points and solve the correspondence problem through correspondence considered multiple frames. Rangarajan and Shah [6] proposed a proximal uniformity constraint, using a gradient based optical flow for establishing the initial correspondence without using an iterative optimization procedure. Veenman *et al.* [4] present a framework for motion modeling and the greedy optimal assignment (GOA) tracker which is efficient in handling detection errors and occlusion. Shafique and Shah [7] present a bipartite graph theoretical formulation of the point correspondences over multiple frames and a sliding window technique to reduce the time complexity of the overall algorithm.

In this paper, we propose a new global method using the order of precedence among points with respect to the x coordinates. Our proposed algorithm tries not only to match the points maximally in locally ambiguous regions, but also to maintain the order of the matched points. The degree of order maintenance in the matched points is measured by the length of the longest common subsequence (LCSS) used for the evaluation of similarity between the time series [8,9]. This measurement method is based on the Fast Time Series Evaluation (FTSE) which can be used to evaluate such threshold value techniques, including the LCSS and the Edit Distance on Real Sequence (EDR) [8].

The rest of this paper is organized as follows: Section 2 provides the definition and the notion for the Point Series. We then present a technique that groups points into P-regions, in which two or more points are potentially interchangeable in Section 3. Section 4 describes a method that determines all of the possible arrangements in the P-region. Section 5 presents the point correspondence method in the P-regions, and the performance of our algorithm is evaluated in Section 6. Lastly, our conclusions are remarked upon in Section 7.

## 2. The Point Series

We define point series  $S$  as an ordered sequence of points, where the order of a point is determined by sorting the points in an ascending order of the x-coordinate value of each point. The order of a

point in a point series is regarded as the time index of a sample in a time series.

$$S: \langle s_1, s_2, \dots, s_d \rangle, 1 \leq i < d, s_i.x \leq s_{i+1}.x \quad (1)$$

where  $s_i.x$  is the x-coordinate of the  $i$ th point  $s_i$ .

Consider a point series with  $m$  points constructed from a frame. Owing to object motion, some adjacent points of the series may interchange their orders in the point series found in the consecutive frame. Considering the maximum velocity constraint, we cluster the points in a point series into several small groups. Each group represents a set of points which can switch their precedence order in the following frame. For each group, we construct a potential point series, i.e. the candidates of the point series possibly appearing in the following frame. Each series is compared with the real point series, which is the constructed point series of the following frame.

## 3. Grouping the Points

For each point of a  $(t-1)$ th frame, we construct a bounding box centered at the point with a side length of  $2\epsilon$  in the x and y directions, where  $\epsilon$  is a predefined maximum velocity of a point between two consecutive frames. If the adjacent bounding boxes overlap, then the center points of their boxes may swap their orders in the next point series. We call the union of these connected bounding boxes the P-region. All the points in a P-region become a group used to construct a potential point series.

To construct the P-regions efficiently, the following three steps are performed. First, all the points of a group are projected onto the x axis and clustered into several groups. If two consecutive points are within the distance of  $2\epsilon$  in the x-direction, they belong to the same group. Second, a group with two or more points is split into smaller groups if the y-directional distance between the adjacent points exceeds  $2\epsilon$ . The above process is performed for the y axis in a similar fashion. Finally, a group is split into different groups so that each group consists of a set of points whose bounding boxes overlap each other. Each group in this step is a set of points belonging to the same P-region.

An example of this is shown in Fig. 1. First, a set of points,  $\{①, ②, ③, ④, ⑤, ⑥, ⑦, ⑧\}$ , from the  $(t-1)$ th

time frame is split into two different groups, {①,②,③} and {④,⑤,⑥,⑦,⑧} since the x-directional distance between ③ and ④ is larger than  $2\epsilon$ . Second, the group {①,②,③} is split into {①} and {②,③} because of the y-directional distance between ① and ②. Similarly, the group {④,⑤,⑥,⑦,⑧} is split into {④,⑥,⑦,⑧} and {⑤} because of the y-directional distance between ⑤ and ⑧. Finally, the group {④,⑥,⑦,⑧} is split into {④,⑥,⑦} and {⑧} since their bounding boxes are not connected. The connected bounding boxes for each group represent a P-region.

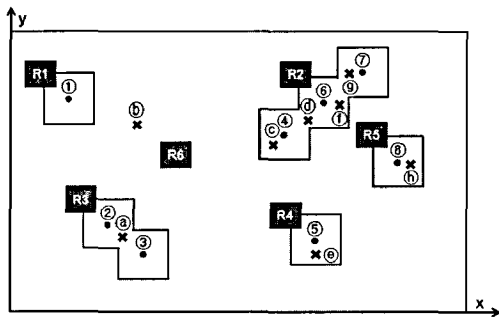


Fig. 1 An example of the P-region corresponding process

#### 4. Assigning Points to the P-regions

Let us assign each point of the next frame to P-regions generated at the grouping stage according to the (x, y) coordinates of the point. As shown in Fig. 1, the situations in the P-regions are categorized as the following three types. The p and q in p:q denote the numbers of points from two adjacent frames at times t-1 and t, respectively, in a P-region

**Type 1:** p:0 This situation occurs when the p points at time t-1 were not detected in the next frame due to a detection miss or a scene-out. R1 in Fig. 1 is an example of this case. Tracking of these points should be terminated.

**Type 2:** 0:q This is a situation in where q new points were detected in a frame at time t because of a false detection or a scene-in. R6 with point ⑥ in Fig. 1 is an example of this case. Tracking of these points should be initiated.

**Type 3:** p:q This is a composite case in that the situations of Types 1 and 2 occur simultaneously.

The remaining P-regions in Fig. 1 are examples of this. In this case, we construct a set of potential point series by permuting the p points of the t-1 frame in the P-region. If a potential point series has a pair of points which cannot be switched due to their excessive distance, the series is ignored.

For the points from a previous frame in a P-region,  $G = \{V_1, V_2, \dots, V_p\}$ , a set of the potential point series, V, is constructed. Each series consists of p points.

$$V = \{V_k = \langle \dots, y_p, \dots, y_1, \dots \rangle | 1 \leq k \leq p!, \|V_k\| = p\} \quad (2)$$

where  $1 \leq i < j \leq p, v_i, v_j \in G, v_i \neq v_j, v_i x - v_j x \leq 2\epsilon$ .

A real point series, R, i.e. the point series of the t<sup>th</sup> frame, with q points in the same P-region is constructed as:

$$R = \langle r_1, \dots, r_i, \dots, r_j, \dots, r_q \rangle \quad (3)$$

where  $1 \leq i < j \leq q, r_i \neq r_j, r_i x \leq r_j x$ .

The set of the potential point series constructed from the group of points {④,⑥,⑦} located in the R2 region in Fig. 1 consists of three point series:  $\langle \textcircled{4}, \textcircled{6}, \textcircled{7} \rangle, \langle \textcircled{4}, \textcircled{7}, \textcircled{6} \rangle, \langle \textcircled{6}, \textcircled{4}, \textcircled{7} \rangle$ . The remaining series,  $\langle \textcircled{6}, \textcircled{7}, \textcircled{4} \rangle, \langle \textcircled{7}, \textcircled{4}, \textcircled{6} \rangle, \langle \textcircled{7}, \textcircled{6}, \textcircled{4} \rangle$ , are discarded because the distance between ④ and ⑦ exceeds the bounds of  $2\epsilon$ . The real point series for the R2 region becomes  $\langle \textcircled{c}, \textcircled{d}, \textcircled{f}, \textcircled{g} \rangle$

#### 5. Corresponding Two Point Series

Each potential point series,  $V_k$ , is recursively compared with a real point series, R, to find the LCSS based on the proximity constraints. Two points in each series are regarded as matched, i.e. the same in the LCSS problem, if their Euclidean distance is within the bound of  $\epsilon$ . The LCSS gives us the maximum pairs of matching points

$$P_k = \{(v'_i, r'_i) | 1 \leq i \leq l_k, v'_i \in V_k, r'_i \in R, |\overline{v'_i r'_i}| \leq \epsilon\}$$

along with their length  $l_k$ . The  $v'_i, r'_i$  are the i<sup>th</sup> matching points between the potential point series and the real point series, respectively. Next, we find out the potential point series,  $V_{\max}$ , that have a maximum length  $L_{\max}$  among the entire potential point series.

$$l_{\max} = \max_{V_k} \{l_k \text{ of } LCSS(V_k, R)\} \quad (4)$$

The maximum pairs of the matching points of

$V_{\max}$  are denoted as  $P_{\max}$ . If several potential point series exist with the same maximum length  $L_{\max}$ , we select from them the  $V_{\max}$  which  $P_k$  maximizes with the heuristic function found in (5). Eq. (5) is developed to measure the average motion smoothness of the matching pairs.

$$f(P_k) = \frac{w_1 * \sum_{i=1}^k \min\{|\overline{p_i v'_i}|, |\overline{v'_i r'_i}|\}}{l_k \sum_{i=1}^k \max\{|\overline{p_i v'_i}|, |\overline{v'_i r'_i}|\}} + \frac{w_2 * \sum_{i=1}^k \min\{\text{ang}(\overline{p_i v'_i}), \text{ang}(\overline{v'_i r'_i})\}}{l_k \sum_{i=1}^k \max\{\text{ang}(\overline{p_i v'_i}), \text{ang}(\overline{v'_i r'_i})\}} \quad (5)$$

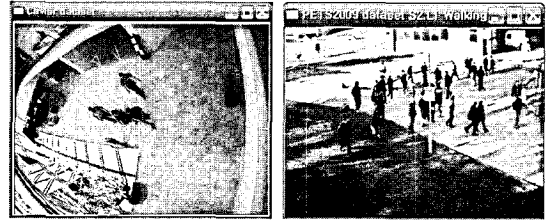
where  $p_i$  is a point in the  $(t-2)_{\text{th}}$  frame corresponding with point  $v'_i$ , and  $\text{ang}(\overline{pq})$  denotes an angle between  $\overline{pq}$  and the x axis. The first and the second term measure the average speed smoothness and the direction smoothness, respectively.  $w_1, w_2$  are the corresponding weights. For  $P_{\max}$ , (5) computes and checks whether the value is greater than the threshold  $Th$  for an acceptable motion change.

Finally, if the value of the function is greater than  $Th$ , the matched points in  $P_{\max}$  are regarded as *corresponding*. Otherwise, (5) is computed repeatedly with an updated  $P_{\max}$  as a subset of the current  $P_{\max}$  by excluding the pair with the maximum motion change until the condition is satisfied.

## 6. The Experimental Results

We performed our experiments on two image sequences. The first dataset, named "Crowd of four people meet, walk and split", is from the CAVIAR project web site (provided with ground truth) [10]. It contains various conditions, such as the scene-in and scene-out of 4 persons, and their crossing of each other. The second dataset, from Pets2009, includes many occlusions due to the number of human beings in the image sequence [11]. In this, the ground truth is manually obtained by pointing at the center of each moving person for all of the frames. Also, we made several artificial datasets by adding random noises to represent falsely detected objects to each image sequence at the rate of 10~50% to simulate detection errors. Sample images of the two original datasets are shown in Fig. 2.

To evaluate the performance, the ratio of the



(a) Caviar dataset  
(488 frames)

(b) Pets dataset  
(50 frames)

Fig. 2 The two datasets

correctly corresponding pairs (CCPR), the ratio of the correctly detected tracks (CDTR), and the average ID change (AIDC) [12] are measured. The CCPR is the ratio of the number of correctly established correspondences to the total number of true correspondences for all frames.

$$CCPR = \frac{\sum_{j=1}^M \sum_{i=2}^N CC_{ij}}{TC} \quad (6)$$

where  $CC_{ij}$  is 1 if the correspondence between the points of the  $i-1$  frame and the points of the  $i$  frame for the  $j$  track is correct.  $TC$  is the total number of true correspondences,  $N$  is the total number of frames, and  $M$  is the total number of tracks.

The CDTR is the ratio of the number of correctly detected tracks to the total number of true tracks in the image sequence, as shown in (7) and (8). A track is regarded as a correctly detected track if it satisfies the condition that the CDTR for each track is larger than the predefined threshold ( $Th$ ) which in our experiments is set to 90%. This measure is used to get the ratio of the valid track to the total track.

$$CDTR = \sum_{j=1}^M I_j / M \quad (7)$$

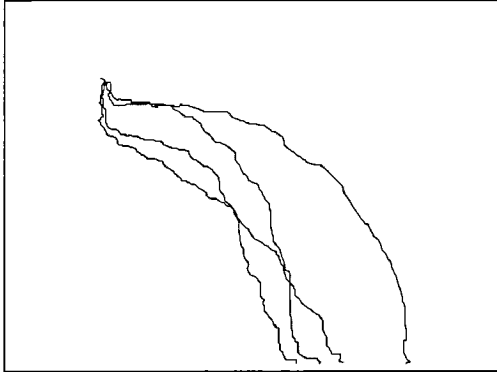
$$I_j = \begin{cases} 1, & \frac{\sum_{i=2}^N CC_{ij}}{TC_j} > Th \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where  $TC_j$  is the total number of true correspondences for the  $j$  track.

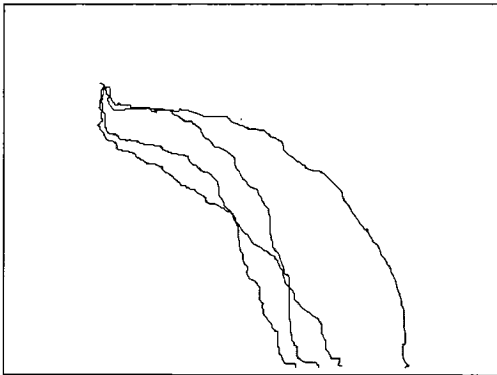
The Average of ID Change (AIDC) is the ratio of the number of ID changes to the total number of true correspondences for all tracks. This means that the IDs for all the correspondences are changed as

Table 1 The Experimental Results for the Caviar and Pets datasets

Noises	Caviar dataset			Pets dataset			
	0%	25%	50%	0%	10%	20%	30%
CCPR	100%	96%	95%	96%	95%	93%	91%
CDTR	100%	100%	100%	100%	96%	91%	89%
AIDC	0%	0.1%	0.1%	1%	2.2%	2.3%	2.5%



(a) the true trajectories



(b) the calculated trajectories  
Fig. 3 The CAVIAR dataset

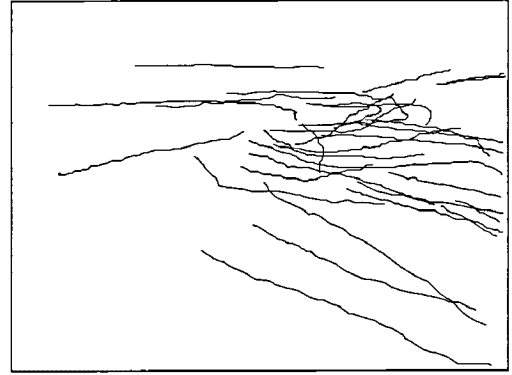
much as the AIDC on average.

$$AIDC = \sum_{j=1}^M IDC_j / (M * (N - 1)) \quad (9)$$

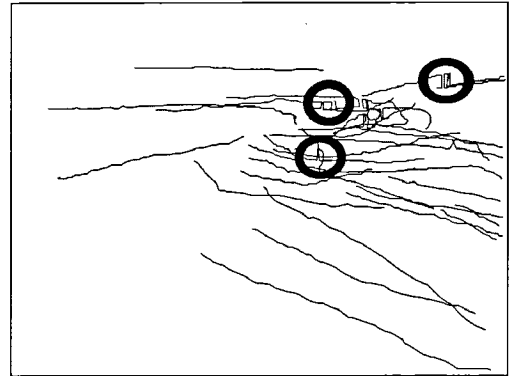
where  $IDC_j$  is the total number of the ID changes for the  $j$  track.

We let function parameters  $w_1/l_k = 0.5$ ,  $w_2/l_k = 1$  and  $\epsilon=10$ .

The results are shown in Table 1. Fig. 3 shows the true trajectories of the ground truth (GT) and the calculated trajectories of the ST for the CAVIAR dataset without noise. Fig. 4 shows the



(a) the true trajectories



(b) the calculated trajectories  
Fig. 4 The PETS dataset

true trajectories of the GT and the calculated trajectories of the ST for the PETS dataset without noise. The four trajectories seen in Fig. 3(a) and Fig. 3(b) for the CAVIAR dataset without noises are the same, but the trajectories seen in Fig. 4 for the PETS dataset without noises show a difference. The thick circles in Fig. 4(b) show the regions where the ids are changed and the point correspondences between the points are incorrect.

## 7. Conclusion

We proposed a motion correspondence algorithm which uses global information based on the precedence order of points in the x-coordinates. According to the maximum velocity constraint of a point, we divide the overall points into several groups in which the points may swap their orders. Two point series extracted from consecutive frames are evaluated to locate the LCSS in order to acquire the

maximum matching pairs. If there are two or more LCSSs with the same length, we select the one that satisfies the motion smoothness constraint. We tested several data sets with random noises added at the rate of 0~50%. The experimental results show that the proposed algorithm is efficient and robust for finding motion correspondences even when high level noises are added.

### References

- [1] J. Y. Jung, M. H. Kim, "Automatic feature point extraction and tracking in image sequences with the multiple moving objects," *Journal of KIISE*, vol.25, no.3, pp.562-573, Mar. 1998. (in Korean)
- [2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol.38, no.4, pp.1-45, 2006.
- [3] J. H. Kao, J. H. Chuang, M. L. Hsieh, "Finding point correspondence using local similarity and global constraint under insignificant scaling and roll," *Electronics Letters*, vol.42, no.23, pp. 1340-1341, Nov. 2006.
- [4] C. J. Veenman, M. J. T. Reinders, E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.23, no.1, pp.54-72, Jan. 2001.
- [5] I. Sethi, R. Jain, "Finding Trajectories of Feature Points in a Monocular Image Sequence," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.9, no.1, pp.56-73, Jan. 1987.
- [6] K. Rangarajan, M. Shah, "Establishing motion correspondence," *Conference Vision Graphics Image Process*, vol.54, no.1, pp.56-73, July 1991.
- [7] K. Shafique, M. Shah, "A Noniterative Greedy Algorithm for Multiframe Point Correspondence," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.27, no.1, pp.51-65, Jan. 2005.
- [8] M.D. Morse, J. M. Patel, "An Efficient and accurate method for evaluating time series similarity," *In the ACM SIGMOD Int. Conf. on Management of data*, pp.569-580, 2007.
- [9] L. Chen, M. T. Ozsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," *ACM*, pp.491-502, 2005.
- [10] CAVIAR project, <http://homepages.inf.ed.ac.uk/rbf/CAVIAR>
- [11] PETS 2009, <http://www.cvg.rdg.ac.uk/PETS2009/>
- [12] F. Yin, D. Makris, S. Velastin, "Performance Evaluation of Object Tracking Algorithms," *In 10th IEEE International Workshop on PETS2007*, pp. 17-24, Oct. 2007.



염 기 열

1995년 세종대학교 응용통계학과 졸업  
2006년 성균관대학교 정보통신대학원 정보통신공학과 공학석사. 2006년~현재 성균관대학교 정보통신공학부 박사과정. 관심분야는 컴퓨터비전, 패턴인식, 퍼지 추론, 인공지능 등임



정 재 영

1989년 성균관대학교 정보공학과 졸업  
1993년 성균관대학교 정보공학과 공학석사. 1997년 성균관대학교 정보공학과 공학박사. 1997년~현재 동양대학교 컴퓨터정보전학과 부교수. 관심분야는 컴퓨터비전, 패턴인식, 영상압축 등임



김 문 현

1978년 서울대학교 전자공학과 졸업. 1980년 한국과학기술원 전기 및 전자공학과 공학석사. 1988년 Univ. of Southern California 공학박사. 1988년~현재 성균관대학교 정보통신공학부 교수. 관심분야는 컴퓨터비전, 패턴인식, 인공지능 등임