DETECT COHERENT MOTIONS IN CROWD SCENES BASED ON TRACKLETS ASSOCIATION

Yi Zou, Xu Zhao, and Yuncai Liu*

Key Laboratory of System Control and Information Processing
Department of Automation, Shanghai Jiao Tong University

ABSTRACT

Coherent motion is a very common motion pattern in crowded scenes. Coherent Filter is a very effective and robust tool to detect coherent motions based on point trajectories, the performance of coherent filter depends on point trajectories' property. In this work, we present a two-stage strategy to extract dense, accurate and long-term point trajectories from crowded scenes. The method includes a tracklets acquisition procedure and a tracklets association procedure. We use LDOF tracker to acquire dense tracklets, and then formulate tracklets association as a linear assignment problem (LAP). Experiments conducted on challenging crowd datasets show that our trajectories are very suitable for detecting coherent motions in crowded scenes.

Index Terms—Crowded scenes, coherent motions, point tracker, tracklets association

1. INTRODUCTION

In crowded scenes, it's difficult to detect and track individuals [1-3]. Therefore, many recently works detect motion patterns in crowded scenes based on feature point trajectories [4-7]. Coherent motions are very common in crowded scenes, which describe individuals' collective movement in crowd. Coherent Filter [13] is a very effective and robust tool to detect coherent motions based on point trajectories. Coherent filter seeks points' invariant neighbors during several frames, and connects these invariant neighbor points to build coherent groups. The number and reliability of invariant neighbors depends on the point trajectories, apparently, we hope the point trajectories to be accurate, dense, and long-term.

The most widely used point tracker is the Kanade Lucas Tomasi (KLT) tracker [8], which is used to obtain motion trajectories by a good deal of the existing works. KLT tracker tracks sparse feature points across the frames based on LK optical flow. KLT tracker stops point tracking when the image patches around the tracked points vary too much, and periodically adds new feature points in gaps among existing feature points to compensate for the lost points. Because of the property of LK optical flow, KLT tracker is fit to track a small number of feature points which have sufficient structure. However, for those points which have insufficient structure, KLT tracker would result in huge tracking errors. In other word, one cannot extract both accurate and dense point trajectories in crowded scenes by KLT tracker. Sand's particle video (PV) [9] is another scheme to track dense points across the frames based on dense variational optical flow. PV tracker stops tracking the particles having high post-optimization errors, and adds new particles in gaps between existing particles. The biggest difference between PV tracker and other point trackers is that: after predicting particle positions by optical flow, PV tracker optimizes particle positions by measuring appearance consistency along the particle trajectories and distortion between the particles. In crowded scenes, objects are very small and have similar appearances. The optimization process of PV tracker would lead to extra tracking errors. Sundaram and Brox [10] proposed a method to track dense points based on large displacement optical flow (LDOF), a kind of dense variational optical flow. LDOF tracker stops tracking points when the tracked points are occluded or on motion boundaries, and periodically adds new points in gaps between existing points. According to Sundaram's experiment results, LODF tracker can provide more dense and accurate trajectories than the other two mentioned above, but we find that the tracked points by LDOF tracker are prone to lost faster due to the clutter of the crowded scenes.

In this paper, we propose a two-stage strategy to extract accurate, dense, and long-term point trajectories. In stage 1, we use LDOF tracker to track points across the frames. In theory, we can track any point by LDOF. However, in practice, the points with more structure are chosen in prior. In crowed scenes, many tracked feature points are lost across a few frames because of occlusion or instability of optical flow, but some of them would be detected and tracked again in a few frames later. We call these raw trajectories obtained by LDOF tracker as tracklets. Intuitively, tracklets which belong to the same point should be connected into a single trajectory. In stage 2, we integrate the tracklets belonging to the same point into a single trajectory. We divide the set of tracklets into several overlapped subsets. In every subset, we consider whether any two tracklets should be linked or not. We formulate this problem as a linear assignment problem (LAP), and provide a pairwise cost function composed of the tracklets' length of time, likelihood of motion model and similarity of appearance. Finally, we associate the linked tracklets in the whole video to acquire refined trajectories.

In sum, the main contributions of this work can be summarized as follows. Firstly, we proposed a two-stage strategy to extract accurate, dense, and long-term point trajectories. Our trajectories are longer than LDOF in temporal dimension, and are denser and more accurate than KLT. Secondly, we integrate our trajectories into coherent filter, the experiments validated that the coherent filter based on our trajectories performs better than others. In fact,

This research has been partially supported by China 973 Program (2011CB302203) and NSFC grants (61273285, 61375019). * indicates corresponding author.

our trajectories are also suitable for other works generally on motion pattern mining.

2. EXTRACTING DENSE POINT TRACKLETS

We adopt LDOF tracker to track points in crowded scenes, and select appropriate points to represent moving objects.

2.1. Points Selection

Generally speaking, points with certain structure in image correspond to the specific parts of objects in real world, such as heads, shoulders, etc. For this sake, we prefer to select points which have more structure. Similar to [11], the selection process goes as follows:

1). Compute the gradient matrix G and its minimum eigenvalue λ_m at every pixel in the image I.

2). Compute the average value λ_{avg} of λ_m over the whole image I.

3). Set zero to the image pixels that have a λ_m value smaller than a percentage (50% in our experiments) of λ_{avg}.

4). Retain the local maximum pixels (w_n*w_n neighborhood, w_n is the Gaussian window's length), and sort the subset of nonzero pixels in descending order.

5). Select the pixels from the subset sequentially whose minimum distance to any existing points are larger than a given threshold distance (1.5*w_n in our experiments).

6). Filter out the points whose magnitudes of velocity are smaller than a given threshold V_{thr} (0.5 pixels in our experiments).

The biggest difference between our points selection and KLT tracker's points selection is in the step 3, where we can retain the pixels that have a λ_m value bigger than half (or smaller) of λ_{avg}, while KLT tracker would drop the pixels that have a λ_m value smaller than two times (or bigger) of λ_{avg}. Obviously, we can flexibly select points in various degree of density for different applications.

2.2. Points Tracking

Suppose a point is at (x_t, y_t) in frame t, and (x_{t+1}, y_{t+1}) in frame t+1, then LDOF tracker propagates the point by:

$$(x_{t+1}, y_{t+1}) = (x_t, y_t) + (B * V_t)\big|(x_t, y_t) \tag{1}$$

Where V_t denotes LDOF field from frame t to frame t+1, B denotes bilinear interpolation in LDOF at (x_t, y_t). One can track any point across the frames according to Eq. (1). The tacking process however should be terminated when LDOF may cause large tracking errors.

2.3. Points Pruning

LDOF tracker defines two situations to stop tracking point: the tracked point is occluded, or the tracked point is on boundaries of motion. Specifically, if the tracked point does not satisfy the following two criteria, we stop tracking it.

$$\left|V_f + V_b\right|^2 < 0.01\left(\left|V_f\right|^2 + \left|V_b\right|^2\right) + 0.5 \tag{2}$$

$$\left|\nabla u_f\right|^2 + \left|\nabla v_f\right|^2 < 0.01\left|V_f\right|^2 + 0.002 \tag{3}$$

Eq. (2) prevents the tracked point from occlusion, where V_f denotes LDOF from frame t to frame t+1, V_b denotes LDOF from frame t+1 to frame t. Eq. (3) prevents the tracked point from motion boundaries, where ∇ denotes gradient, and $V_f = (u_f, v_f)$.

3. TRACKLETS ASSOCIATION & COHERENT MOTION DETECTION

LDOF tracker can track dense points across the frames, but the tracked points are usually lost frequently in crowded scenes. We notice that many tracked feature points are lost across a few frames, but some of them will be detected and tracked again in a few frames later. Intuitively, tracklets belonging to the same feature point should be merged into a single trajectory. In this section, we introduce how to integrate these tracklets into a single trajectory.

3.1. Tracklets Association

Framework. Our goal is to answer two questions: whether a tracklet should be continued by another one, and which tracklet should be chosen as the posterity. For the sake of computational cost, we adopt a greedy strategy in temporal dimension. We divide the set of tracklets into several overlapped subsets.

$$Set = \left(Subset_1, Subset_2, ...\right) \tag{4}$$

$$Subset_i = \{Fset_i, Pset_i\} \tag{5}$$

Fset is the set of tracklets to be continued, and Pset is the set of the candidates of posterity. $Fset_i$ contains all tracklets which end in the period from frame $(t_s+(i-1)*w_t)$ to frame (t_s+i*w_t-1). $Pset_i$ contains all tracklets which start in the period from frame $(t_s+(i-1)*w_t)$ to frame $(t_s+i*w_t-1+t_g)$.

Where t_s denotes the beginning frame of the whole video, w_t denotes the time window length, t_g denotes the maximum time gap we permit between two tracklets to be connected. We set $w_t = t_g =$ 10 frames in our experiments.

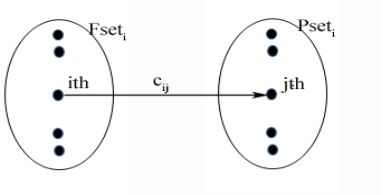

Figure 1. Match tracklets from Fset to Pset

As shown in Fig. 1, we can formulate the matching problem as a linear assignment problem. In any subset, we have:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij}$$
$$s.t. \begin{cases} \sum_j x_{ij} = 1; \\ \sum_i x_{ij} = 1; \\ x_{ij} \in \{0,1\} \end{cases} \tag{6}$$

Where c_{ij} denotes the cost of connecting the i_{th} tracklet in Fset with the j_{th} tracklet in Pset, x_{ij} denotes the linkage between the i_{th} tracklet in Fset and the j_{th} tracklet in Pset.

We can achieve the optimal match by minimizing the sum of linkage cost in Eq. (6). Notice that, the linkages whose costs are bigger than a threshold c_{trd} would be cut. Finally, we associate the linked tracklets in the whole video to acquire refined trajectories.

Cost Function. We define a pairwise cost function by measuring the tracklets' length of time, likelihood of motion model and similarity of appearance.

$$\cos t(p,q) = \begin{cases} -P(p,q), & if\ tsq > tep \\ 0, & else \end{cases} \tag{7}$$

Where p denotes a tracklet in Fset, q denotes a tracklet in Pset. Eq. (7) constrains that, tracklet p should not be continued by tracklet q which starts earlier than p ends.

$$P(p,q) = \left(P_{len}(p) * P_{len}(q)\right) * P_{appr}(p,q) * P_{motion}(p,q) \tag{8}$$

Eq. (8) measures the possibility of the linkage between tracklet p and tracklet q. P_{len} is a scale factor that encourages the longer tracklets to connect, because the short tracklets often correspond to the noise points.

$$P_{len}(p) = sqrt\left(\max\left(\min\left(t_{len}(p)/m_{len}, 2\right), 0.5\right)\right) \tag{9}$$

Where t_{len} denotes the time length of tracklet, m_{len} denotes the average time length of all tracklets. To prevent the extreme cases, we restrict P_{len} in $[\sqrt{0.5}, \sqrt{2}]$.

Due to the fact that objects in crowded scenes always occupy very few pixels, we use the average gray value of points on a tracklet to represent the tracklet's appearance.

$$P_{appr}(p,q) = 1 - \left|gray(p) - gray(q)\right| / 255 \tag{10}$$

We suppose that acceleration of a point is constant and small in a short period. Assume that tracklet p and tracklet q belong to the same feature point, we can predict point p's position at the time when q starts:

$$X_{pred}(p) = X_{end}(p) + 0.5 * \left(V_{end}(p) + V_{start}(q)\right) * t\_gap \tag{11}$$

Where X_{end} denotes the position of tracklet's end point, V_{end} denotes the velocity of tracklet's end point, V_{start} denotes the velocity of tracklet's start point, t_gap denotes the time gap between p and q. $0.5*(V_{end}(p)+V_{start}(q))$ is the estimate of average velocity in the gap.

If tracklet p and q belong to the same feature point, the predicted position of p and the position of q's start point will be very close. So we have:

$$P_{pos}(p,q) \sim \mathrm{N}\left(dist\left(X_{pred}(p), X_{start}(q)\right), \sigma_{dist}\right) \tag{12}$$

The velocity should not change too much from p to q, we measure the similarity of two velocity as:

$$S(p,q) = \frac{\left(V_{end}(p) \cdot V_{start}(q)\right)}{\max\left(\left\|V_{end}(p)\right\|^2, \left\|V_{start}(q)\right\|^2\right)} \tag{13}$$

$$P_{vol}(p,q) \sim \mathrm{N}\left(1 - S(p,q), \sigma_{vol}\right) \tag{14}$$

$$P_{motion} = 2 * (P_{pos} * P_{vol}) / (P_{pos} + P_{vol}) \tag{15}$$

Where N denotes Gaussian distribution, σ denotes standard deviation of Gaussian distribution. In our experiment, we set σ_{pos} = 0.9 * Wn, σ_{vol} = 0.9. The possibility function of motion model will prefer constant velocity trajectories which are widespread in nature.

3.2. Coherent Motions Detecting

Coherent Filter (CF) proposed by Zhou [13] is used to detect coherent motions, which widely exist in crowd events. CF seeks points' invariant neighbors during several frames, and connects these invariant neighbor points to build the coherent groups.

For point j, if point k keeps belonging to the K nearest neighbors of point j from frame t to frame t+d, and their velocity correlations are higher than a threshold, point k is an invariant neighbor of point j at frame t. Then build a connectivity graph, where vertices are points and edges are defined by invariant neighborhood of points. Finally, incoherent motions are identified as the connected components of the graph.

There are two parameters in coherent filter: K and d. The spatial dimension of the neighborhood K is usually set to 10. The temporal dimension of neighborhood d has a great influence on the invariant neighbors. Generally, more robust invariant neighbors are obtained when d is larger. Unfortunately, the temporal length of point trajectories is always short, and the number of invariant neighbors would be very small, when a large d is chosen. Therefore, d is set to 10 frames in our experiments to balance the reliability and number of invariant neighbors.

4. EXPERIMENTS

4.1. Dataset and Experimental Settings

To test our approach, we conduct experiments on three challenging datasets: UCF_Crowds dataset, Grand Center station dataset and SJTU_Crowds dataset [12]. UCF_Crowds dataset is collected by the University of Central Florida. It contains several videos of crowded scenes in nature. Most videos in UCF dataset are very short and some of them just last for several seconds. Grand Center station dataset is a half hour long video collected from the New York's Grand Center station, which is at a resolution of 480×720. SJTU_Crowds dataset is designed by Shanghai Jiao Tong University, which includes various motion patterns of crowd, such

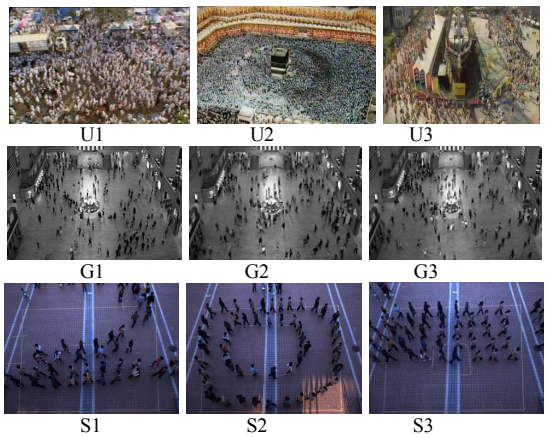

Figure 2. Dataset selected for our experiments

as merging, splitting, intersecting, circular motion, etc. Most videos are about 2 minutes long and with a resolution of 964×1280.

For an illustrative purpose, we select three video clips from each dataset to test our method, as shown in Fig. 2. Scenes of the first row are selected from UCF_Crowds dataset, each scene contains a 60-frame image sequence. Scenes of the second row are selected from Grand Center station dataset, each scene contains a 180-frame image sequence, 6 frames per second. Scenes of the third row are selected from SJTU_Crowds dataset, each scene contains a 60-frame image sequence, 6 frames per second.

According to density of crowd and resolution of image, we set w_n in section 2.1 as 3 pixels, 5 pixels, and 7 pixels for the scenes of the three rows. We select more points for the denser scene, which corresponds to a smaller value of w_n. To obtain accurate trajectories, we set c_{trd} in section 3.1 as -0.3 in our experiments.

4.2. Tracklets Association Evaluation

To validate the effectiveness of our tracklets association, we compare our method with LDOF tracker. The feature scales are set as same in the three algorithms. Table 1 shows a quantitative comparison of trajectory numbers with various lengths.

TABLE I. NUMBER OF TRAJECTORIES WITH VARIOUS LENGTHS

	Number of trajectories					
	Scene U1		Scene U2		Scene U3	
Length	LDOF	Ours	LDOF	Ours	LDOF	Ours
46~60	60	**228**	29	**184**	56	**226**
31~45	127	**245**	79	**139**	108	**154**
16~30	398	**481**	**316**	223	**372**	249
1~15	**4535**	2485	**1576**	636	**1506**	601
	Scene G1		Scene G2		Scene G3	
Length	LDOF	Ours	LDOF	Ours	LDOF	Ours
91~180	3	**35**	3	**63**	3	**42**
61~90	19	**146**	19	**135**	25	**108**
31~60	194	**486**	207	**448**	160	**436**
1~30	**8516**	4693	**7846**	4187	**7343**	4092
	Scene S1		Scene S2		Scene S3	
Length	LDOF	Ours	LDOF	Ours	LDOF	Ours
46~60	10	**17**	46	**92**	8	**20**
31~45	23	**35**	47	**54**	18	**33**
16~30	73	**96**	100	**105**	56	**65**
1~15	**853**	576	**1053**	599	**642**	376

UNIT OF LENGTH IS FRAME.

As shown in Table 1, our method obtains more long trajectories (longer than half of total length) and less short trajectories (shorter than 25% of total length) than LDOF tracker. This is because the tracklets association process merges tracklets belonging to the same feature point into a single trajectory. In general, long trajectories correspond to the paths in the scenes, which are very useful for us to understand the scenes. Our method can acquire twice as many long trajectories as LDOF tracker in most of experimental scenes. It means that, our method can provide more information on crowded scenes than LDOF tracker. But, the tracklets association process would lead to extra tracking errors more or less. One can adjust c_{trd} to make a tradeoff between length and accuracy of trajectories for different applications.

4.3. Coherent Motions Detecting Evaluation

To evaluate the performance of our trajectories on coherent motions detecting, we adopted coherent filter (CF) to detect the coherent motions based on the three trajectories: our trajectories, GKLT trajectories and LDOF trajectories. In coherent filter, the spatial dimension of the neighborhood K is set to 10 points, the temporal dimension of neighborhood d is set to 10 frames.

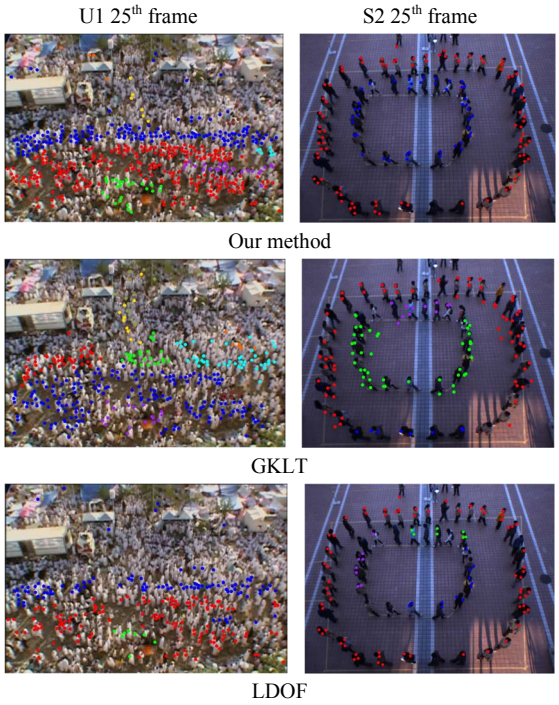

Figure 3. Coherent motions detected by coherent filter[13] based on three kind of trajectories.

In Fig. 3, the left column was scene U1, where people went through a market along several paths and the right column was scene S2, where people walked along two opposite direction circles. The first row is coherent groups based on our trajectories. The second and the third row are based on KLT and LDOF trajectories, respectively. As can be seen, the coherent groups based on our trajectories were closer to the ground truth than others. In the third row, we could see that, the coherent motion points based on LDOF trajectories are markedly less than ours. This is because the LDOF trajectories are always shorter than ours, which leads to less number of invariant neighbors. In the second row, we could see that, the coherent motion points based on KLT trajectories are not as smooth as the results based on our trajectories. This is because the KLT trajectories are less accurate than ours, which leads to less reliability of invariant neighbors. To sum up, our method can provide dense, accurate and long-term trajectories in crowded scenes, which is very suitable for coherent motions detecting.

5. CONCLUSION

Studying on several famous point trackers, we propose a new method to achieve dense, accurate, and long-term trajectories in crowded scenes. Our tracker has been shown effective to take dense, accurate and long-term trajectories in crowded scenes. It is very suitable for coherent motions detection.

6. REFERENCES

[1] Ali, S. and Shah, M., "Floor fields for tracking in high density crowd scenes," *European Conf. on Computer Vision*, vol. 2, pp. 1–14, 2008.

[2] Ali, S., "Taming crowded visual scenes," *Ph.D. thesis*, *University of Central Florida*, 2010.

[3] Zhao, T. and Nevatia, R., "Tracking multiple humans in crowded environment," *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 406-413, 2004.

[4] Zhou, B. Wang, X. and Tang, X., "Understanding collective crowd behaviors: Learning a Mixture model of Dynamic pedestrian-Agents," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2871–2878, 2012.

[5] Mehran, R. and Moore, B. and Shah, M., "A streakline representation of flow in crowded scenes," *European Conf. on Computer Vision,* pp. 439–452, 2010.

[6] Brostow, G. and Cipolla, R., "Unsupervised Bayesian Detection of Independent Motion in Crowds," *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 594–601, 2006.

[7] Zhou, B. Wang, X. and Tang, X., "Random Field Topic Model for Semantic Region Analysis in Crowded Scenes from Tracklets," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3441-3448, 2011.

[8] Shi, J. and Tomasi, C., "Good features to track," IEEE Conf. on Computer Vision and Pattern Recognition, pp. 593–600, 1994.

[9] Sand, P. and Teller, S., "Particle video: Long-range motion estimation using point trajectories," IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2195–2202, 2006.

[10] Sundaram, N. and Brox, T. and Keutzer, K., "Dense point trajectories by GPU-accelerated large displacement optical flow," *European Conf. on Computer Vision*, pp. 438–451, 2010.

[11] Bouguet, J., "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm," *Intel Corporation, Microprocessor Research Labs*, 2000.

[12] Wang, C. Wu, Z. and Zhao, X. and Liu, Y., "Motion Patterns Analysis in Crowded Scenes based on Hybrid Generative-Discriminative Feature Maps," *IEEE Conf. on Image Processing*, pp. 2837-2841, 2013.

[13] Zhou, B. and Wang, X. and Tang, X., "Coherent Filtering: Detecting Coherent Motions from Crowd Clutters," *European Conf. on Computer Vision*, vol. 2, pp. 857–871, 2012.

[14] Robert, C., "Multitarget data association with higher-order motion models," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1744-1751, 2012.