Detecting Global Motion Patterns in Complex Videos

Min Hu, Saad Ali, Mubarak Shah
Computer Vision Lab, University of Central Florida
{*mhu,sali,shah*}*@eecs.ucf.edu*

Abstract

Learning dominant motion patterns or activities from a video is an important surveillance problem, especially in crowded environments like markets, subways etc., where tracking of individual objects is hard if not impossible. In this paper, we propose an algorithm that uses instantaneous motion field of the video instead of long-term motion tracks for learning the motion patterns. The motion field is a collection of independent flow vectors detected in each frame of the video where each flow is vector is associated with a spatial location. A ***motion pattern*** *is then defined as a group of flow vectors that are part of the same physical process or motion pattern. Algorithmically, this is accomplished by first detecting the representative modes (sinks) of the motion patterns, followed by construction of* ***super tracks****, which are the collective representation of the discovered motion patterns. We also use the super tracks for event-based video matching. The efficacy of the approach is demonstrated on challenging real-world sequences.*

1. Introduction

The traditional approach for activity analysis in a video sequence consists of following steps: i) detection of all the moving objects that are present in the scene; ii) tracking of the detected object; and, iii) analysis of the tracks for event/activity detection. This standard processing pipeline works well in a low density scene where reliable trajectories of moving objects can be obtained which eventually facilitates the detection of typical motion patterns as well. However, in real-world situation the assumption of low density does not always hold. For instance, videos depicting events such as marathons, political rallies, city center etc., usually contain hundreds of objects. Over the years, little attention has been paid to analyze videos of these situations especially in terms of learning the activity models and motion patterns hidden in these crowded scenes.

To deal with videos of these challenging settings, we propose a new method to learn the typical motion patterns using only the *global* motion flow field, instead of long-term trajectories of moving objects. Here, the motion flow field is a set of independent flow vectors representing the instantaneous motion present in a frame of a video. Such instantaneous motion information is readily available in any situation as it is not effected by the density of objects. The motion flow field is obtained by first using the existing optical flow methods to compute the optical flow vectors in each frame, and then combining the optical flow vectors from all frames of the video into a single *global* motion field. This global motion field does not contain any temporal information as the flow vectors from all the frames are merged into a single field without maintaining the information about the video frames they came from. Next, from the global motion flow field, we extract the representative modes, which are called the sinks, for each motion pattern. The process of detecting the sinks is referred to as the *sink seeking process*. After extracting the sinks and sink paths, they are grouped into several clusters, each corresponding to a motion pattern present in the video. To collectively represent the motion pattern, a single *super track* is generated from the sink paths.

Related Work: Learning of motion paths or patterns by clustering trajectories of moving objects has been attempted before in the literature. For instance, Grimson *et al.* [12] used the trajectories of moving objects to learn the motion patterns which are then used for abnormal event detection. Johnson *et al.* [5] used neural networks to model motion paths from trajectories. While in [3], trajectories were iteratively merged into a path. Similarly, Wang *et al.* [9] used a trajectory similarity measure to cluster trajectories where each clusters was representing a specific dominant activity. Porikli *et al.* [1] represented the trajectories in the HMM parameter space for activity analysis. Vaswani *et al.* [10] modeled the motion of all the moving objects performing the same activity by analyzing the temporal deformation of the "shape" which was constructed by joining the locations of the objects in each frame. These above mentioned methods are based on long-term tracks of moving objects and therefore are only applicable to low density

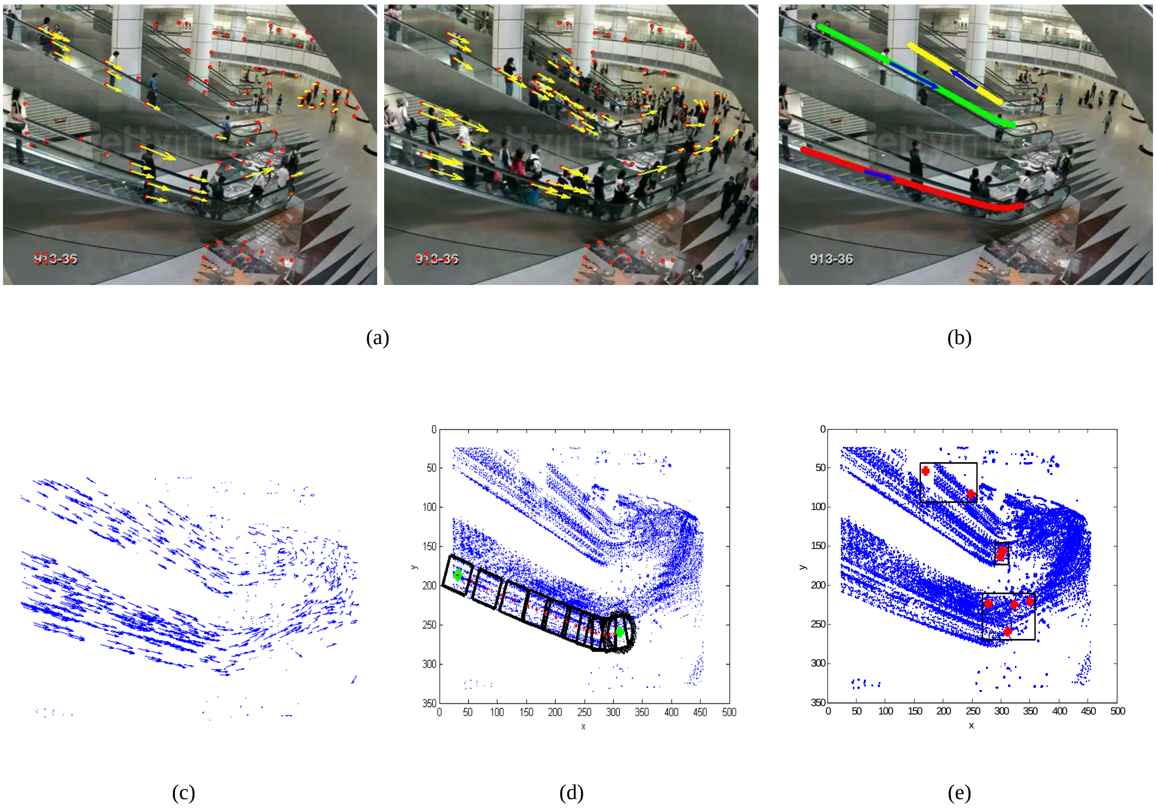


Figure 1. Elevator video: (a) flow vectors (yellow arrows) detected at the corresponding frames #1, #101; (b) detected super tracks; (c) the motion flow field; (d) a sink seeking process; (e) sink clustering.

scenes. In contrast, we are proposing a new method to detect motion patterns in challenging crowded scenes where long-term tracks of moving objects are not available or not reliable. In trajectory analysis, sinks are defined as the endpoints of paths and can be learned from the start and end points of the trajectories [2, 4]. However, fragmented trajectories resulting from occlusions or tracking failures will result in false sinks. To detect sinks in this case, Stauffer [6] defined a transition likelihood matrix and iteratively optimized the matrix for the estimation of sources/sinks. Wang *et al.* [9] estimated the sinks using the local density velocity map in a trajectory clustering. In this paper, the sinks are defined as the end points of the sink paths. They are the modes of motion patterns and define the number of distinct motion patterns.

2. Global Motion Flow Field Generation

Given an input video, for each frame we use the existing methods to compute sparse optical flow (instantaneous velocities) using the interest points ([8]) or dense optical flow for all pixel ([11]) in each frame. Consider a point i in the given frame. Its flow vector, Z_i, includes the location, $X_i = (x_i, y_i)$, and the velocity, $V_i = (v_{x_i}, v_{y_i})$, i.e., $Z_i = (X_i, V_i)$. Note that, these flow vectors do not necessarily belong to foreground objects and no time order or object labels are associated with them. In case, trajectories are available but not reliable, e.g., broken trajectories, then the flow vectors can be obtained directly from these fragmented pieces of trajectories.

All the flow vectors computed from all the frames of the given video then constitute the global motion flow field representing the instantaneous motion field of the video. This flow field may contain thousands of flow vectors and it is computational expensive to apply sink seeking process to such a large amount of data. Moveover, these flow vectors always contain redundant information and noise. Therefore, the flow vectors belonging to the background can be considered as noise as they contain little motion information. To achieve this, we first apply a threshold on the velocity magnitude to remove the flow vectors that have little motion information. Next, we use Gaussian ART (see [13]) to reduce the number of flow vectors from thousands to hundreds. The reduced number of flow vectors still maintain the geometric structure of the flow field, and, therefore, do not effect the results of detecting motion patterns. Fig. 1 shows example flow vectors and corresponding motion flow field.

Sink Seeking: Suppose $\{Z_1, Z_2, \cdots, Z_n\}$ is the motion flow field where $Z_i = (X_i, V_i)$. The states of the sink seeking process of each point, i, are defined as, $\widetilde{Z}_{i,t} = (\widetilde{X}_{i,t}, \widetilde{V}_{i,t}), t = 1, 2, ...,$ and computed using:

$$\widetilde{Z}_{i,1} = Z_i \quad , \qquad \widetilde{X}_{i,t+1} = \widetilde{X}_{i,t} + \widetilde{V}_{i,t}, \qquad (1)$$

$$\widetilde{V}_{i,t} = \frac{\sum_{n \in Neighbor(\widetilde{X}_{i,t})} V_n W_{t,n}}{\sum_{n \in Neighbor(\widetilde{X}_{i,t})} W_{t,n}}. \qquad (2)$$

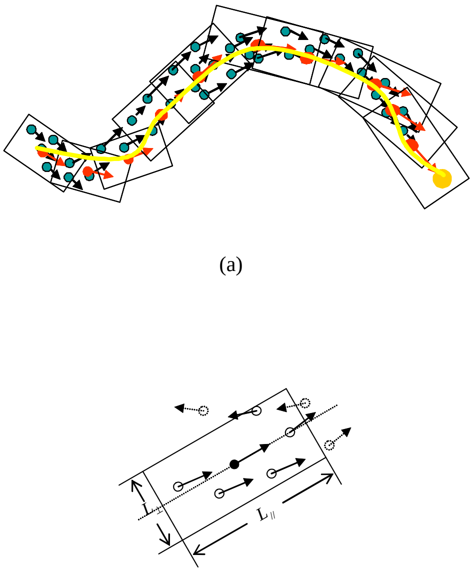
The above equations states that the new 'position' of a point depends only on its location and velocity at the last state. While the new 'velocity', $\widetilde{V}_{i,t+1}$, depends not only on the previous velocity but also on the observed velocities of its neighbors. See Fig. 2(b) which shows the motion trend of group of points in a local neighborhood. In this paper, we employ the kernel based estimation similar to the mean shift approach [14] to incorporate this neighborhood effect using following equation:

$$W_{t,n} = \exp\left(-\left\|\frac{\widetilde{V}_{i,t-1} - V_n}{h_{t-1}}\right\|^2\right), \qquad (3)$$

where h_{t-1} is the bandwidth. Note that, in the mean shift tracking [14], the *appearance* of pixels in a small neighborhood around the object is used to determine the location of the object in the next frame. In our approach, we use *the location and the velocity* of neighboring points in the global flow field to determine the next location. The pictorial description of the sink seeking process is presented in Fig. 2(a).

3. Super Track Extraction

After the sinks are obtained the next task is to cluster the sinks and determine their corresponding sink paths. The clustering algorithm starts by initializing the sink cluster set to an empty set. It takes each sink and attempts to match it with all existing clusters. If a match is found, the sink is assigned to the matched cluster. Otherwise a new cluster is initialized with the current sink as its center. Clusters with a small number of sinks are often caused by the background or noise, and, therefore, are discarded. Formally, given a sink $Z_i^* = (X_i^*, V_i^*)$ associated with a sink path $P_{Z_i^*}$, and a cluster C_k, the sink-cluster distances are given by: i) $D_x(Z_i^*, C_k) = \max_{Z_j^* \in C_k} \|X_i^* - X_j^*\|$, ii) $D_v(Z_i^*, C_k) = \min_{Z_j^* \in C_k} \frac{<V_i^*, V_j^*>}{\|V_i^*\|\|V_j^*\|}$, iii) $D_p(Z_i^*, C_k) = \max_{Z_j^* \in C_k} \mathit{HausdorffDist}(P_{Z_i^*}, P_{Z_j^*})$.



(a)

(b)

Figure 2. Sink seeking process for a given point. (a) sink seeking (red: the states of the flow vector in the sink seeking process, orange: the sink, rectangles: sliding windows, yellow: the sink path); (b) sliding window (solid circle: the flow vector under consideration; rectangle: sliding window; hollow circles: neighboring points; dotted circles: non-neighboring points).

Here all metrics are based on comparison between the given sink Z_i^* and the other sink Z_j^* in the cluster C_k. The first metric measures whether the given sink Z_i^* is spatially close to the cluster C_k or not. The second metric measures the similarity of their directions, and the third measures the Hausdorff distance between their corresponding sink paths represented by $P_{Z_i^*}$ and $P_{Z_j^*}$ respectively. These three metrics ensure that two flow vectors involved in a similar motion pattern have similar sinks and sink paths. Following the clustering of sinks, for each cluster a super track is extracted as the sink path with the maximum arc length to represent the corresponding global motion pattern (see Fig. 1).

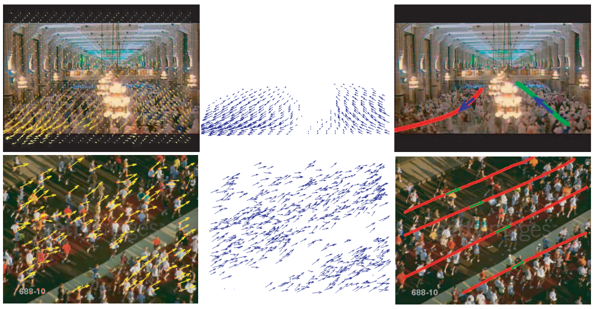


Figure 3. Generating super tracks for crowd videos. Left Col: Extracted flow vectors (yellow arrows). Center Col: The motion flow field. Right Col: Detected super tracks.

Super Track Matching: Each super track may represent motions of several different objects (people, cars etc), since they are generated using global flow field of the whole video. Therefore, super tracks are different form the traditional object tracks representing the locations of a single object in different frames. Super track can be used in video matching since they can effectively reduce the problem of multi-object multi-event video matching to the problem of matching two sets of super tracks. Consider two videos X and Y, and assume X and Y respectively have n and m super tracks $\{x_i\}_{i=1,2,\ldots,n}$ and $\{y_j\}_{j=1,2,\ldots,m}$. We first define the similarity between two super tracks x_i and y_j as $p(x_i, y_j) = \frac{(w_i+w_j)\exp\{-d(x_i,y_j)\}}{\sum_{i,j}(w_i+w_j)}$, where $d(x_i, y_j)$ is the shape distance computed by performing the dynamic time warping of the directional vectors of x_i and y_j (see [7] for details), and w_i is the reliability weight associated to each track x_i, which is given by $w_i = \frac{ArcLength(x_i)}{\sum_{k=1}^{n} ArcLength(x_k)}$. To find the best matching between two groups: $\{x_i\}_{i=1,2,\ldots,n}$ and $\{y_j\}_{j=1,2,\ldots,m}$, we use maximum bipartite graph matching to achieve where each super track is a node in the bipartite graph. The weight of an edge between two nodes is given by the above equation. Given a bipartite graph $G = (V, E)$, a matching M is a subset of E such that for any two different members $e, e' \in M$, $e \cap e' = \oslash$. The maximum weight matching is the one that maximizes the sum of the weights.

4. Experiments

Two classes of videos are considered for the experiments which are i) Crowd, and ii) Aerial videos. These videos contain groups of people and vehicles moving mostly in an unconstrained setting in the presence of shadows and severe occlusions.

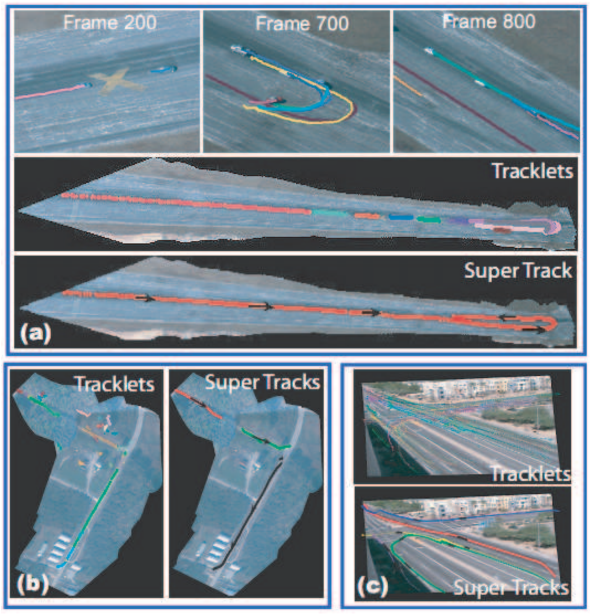


Figure 4. Super tracks in aerial video. (a) Top: Initial tracking results where 6 cars generated 16 broken tracklets. Middle: Trajectories superimposed on the video mosaic. Bottom: Correctly generated single super track. (b) Left: Flow vectors superimposed on the mosaic. Right: Three super tracks. (c) Top: Flow vectors. Bottom: Five super tracks.

Crowd Videos: Fig. 1 shows a crowded scene of a supermarket where crowds of people go up and down through three escalators. Here, we used KLT to extract initial flow vectors, and correctly generated three super-tracks corresponding to the motion patterns of three escalators. Fig. 3 shows results on two other challenging sequence containing dense crowd. In Fig. 3(top-row), the crowd of pilgrims is moving in two opposite directions. The pilgrims are wearing clothes of similar color and are occluded by each other, which makes it very hard to detect and track individual persons. By processing this video through our proposed method, we generated two super tracks which correctly correspond to the two motion patterns: pilgrims going up and pilgrims going down. Fig. 3(bottom-row) demonstrates the strength of our method on a sequence of an outdoor scene containing crowd and shadows. In this case several super tracks were extracted from the motion flow field. Again they correctly correspond to the running routes and the direction of motion.

Aerial Videos: The aerial videos were taken from DARPA's VIVID data set. Here, the main challenge is to resolve the issue of broken trajectories resulting from the limited field of view and occlusion of objects due to terrain features. Initial tracklets were generated using mean-shift tracker in motion compensated imagery. The point flows are then extracted from these tracklets. The first result is shown in Fig. 4(a) where super track is extracted from the video showing a group of cars making a U-turn. In this video, six vehicles move on a highway in a convoy form, but only three or four of them are captured by the camera at any time. Some cars disappear for more than 100 frames and then reappear which results in trajectories which are broken into many tracklets. It is very difficult for a tracking based approaches to detect the motion pattern from these broken trajectories. In contrast, our method obtains the flow vectors from these tracklets and does not use the labels of objects, and, therefore, does not require a complete trajectory. By applying our algorithm, we are able to generate one super track representing the motion patterns hidden in the 16 tracklets of this sequence. Two more results are shown in Fig. 4(b) and (c).

Super Track Matching: We also tested the proposed method for super track based video matching using the VIVID data set consisting of 21 videos. Given a query video, the super tracks were generated using the proposed method. The super tracks of the query video were then compared with the super-track of each video in the database. Fig. 5 illustrates the video matching results for the sequence shown at the top which is an IR video. In this video, there was a group of cars making "S-turns" (see first row in Fig. 5). Fig. 5 shows the three videos with the greatest similarity to the query video. Note that even though there are multiple groups of objects in these three videos and only one group in the query video, all of them contain the same motion pattern i.e. the S-turn. Despite the imperfect tracking and the variability in path shapes, our method successfully matched the videos with the query video.

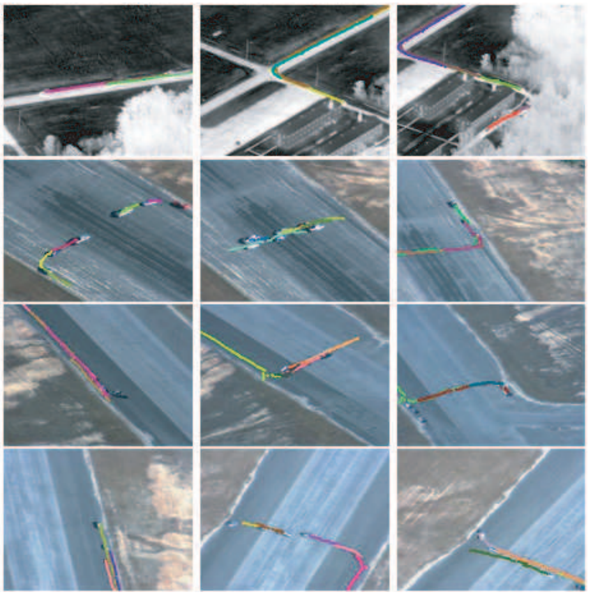


Figure 5. First Row: Frames #500, #1000, #1130 of the query video. Second to Fourth Row: Most similar videos with scores of 0.88, 0.76 and 0.75 respectively.

5. Conclusions

We have proposed a new method based on instantaneous motion information, to detect typical motion patterns for dense crowded scenes. This is achieved by proposing a new construct called 'super track'.

Acknowledgements: This research was funded by the US Government VACE program.

References

[1] F. M. Porikli, *Trajectory Pattern Detection by HMM Parameter Space Features and Eigenvector Clustering*, ECCV, 2004.

[2] D. Makris and T. Ellis, *Automatic Learning of an Activity-based Semantic Scene Model*, AVSBS, 2003.

[3] D. Makris and T. Ellis, *Path Detection in Video Sequence*, IVC, Vol. 30, 2002.

[4] S. McKenna et al., *Learning Spatial Context from Tracking Using Penalised Likelihood Estimation*, ICPR, 2004.

[5] N. Johnson et al., *Learning the Distribution of Object Trajectories for Event Recognition*, IVC, 14, 1996.

[6] C. Stauffer, *Estimating Tracking Sources and Sinks*, Event Mining Workshop, 2003.

[7] M. Vlachos et al., *Rotation Invariant Distance Measures for Trajectories*, SIGKDD, 2004.

[8] B. D. Lucas and T. Kanade, *An Iterative Image Registration Technique with an Application to Stereo Vision*, IJCAI, 1981.

[9] X. Wang et al., *Learning Semantic Scene Models by Trajectory Analysis*, ECCV, 2006.

[10] N. Vaswani et al., *Activity Recognition Using the Dynamics of the Configuration of Interacting Objects*, CVPR, 2003.

[11] R. Gurka et al., *Computation of Pressure Distribution Using PIV Velocity Data*, Workshop on Particle Image Velocimetry, 1999.

[12] W. E. L. Grimson et al., *Using Adaptive Tracking to Classify and Monitor Activities in a Site*, CVPR, 1998.

[13] J. R. Williamson, *Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps*, Neural Netw., 1996.

[14] D. Comaniciu et al., *Mean Shift: A Robust Approach Toward Feature Space Analysis*, PAMI, 24(5). 2002.