Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 2 (2014) 195 – 200

Transportation Research Procedia

www.elsevier.com/locate/procedia

The Conference on Pedestrian and Evacuation Dynamics 2014 (PED2014)

Identifying sources and sinks and detecting dominant motion patterns in crowds

Sultan Daud Khan [a,*], Giuseppe Vizzari [a], Stefania Bandini [a]

[a] *Complex Systems and Artificial Intelligence Research Center, University of Milano-Bicocca, viale Sarca 336/14, 20126 Milano, Italy*

Abstract

The construction of origin-destination matrixes is a necessary activity for most significant applications of pedestrian simulation and it can be fruitfully supported by computer vision techniques. Pedestrians, in videos taken from fixed cameras, tend to appear and disappear at precise locations (doors, gateways or edges of the scene): we refer to locations where pedestrians appear as sources (potential origins) and the locations where they disappear as sinks (potential destinations). In this paper we propose an original technique to identify these points and characterize dominant pedestrian flows. The paper presents the defined technique and it discusses its application in a real-world scenario.

Peer-review under responsibility of Department of Transport & Planning Faculty of Civil Engineering and Geosciences Delft University of Technology

Keywords: computer vision; dominant flow detection; pedestrian and crowd analysis; hierarchical clustering; optical flow

2010 MSC: 68T45

1. Introduction

Urbanization is increasingly leading to crowding situations generating potential issues for architectural planning and even public safety. Pedestrians and crowd studies, related both to the synthesis and automated analysis of the implied dynamics, are growingly investigated as means to support decision makers, as discussed by Challenger et al. (2009). Simulation studies require high quality data, as argued by Bandini et al. (2014) that call for an integrated approach including both analysis and synthesis to provide valid and useful results: automated analysis techniques employing computer vision approaches to tackle these problems are therefore ever more urgent. One of the activities that could be fruitfully automated is the construction of the so-called origin-destination (O-D) matrixes. This kind of characterization of pedestrian behavior in the analyzed scenario is a necessary condition for a proper initialization of simulation systems in realistic conditions (see, e.g., a relevant work by Lee et al. (2001)). Due to the growing availability of cameras, especially in the premises of train or subway stations, videos for performing this kind of analysis are ever more available and this represents an interesting opportunity to investigate.

Pedestrians in videos taken from fixed cameras tend to appear and disappear at precise locations, such as doors, gateways or edges of the scene. We refer to locations where pedestrians appear as *sources* (potential origins) and

* Corresponding author. Tel.: +39-02-64487865; fax: +39-02-64487839.
E-mail address: sultan.khan@disco.unimib.it

Peer-review under responsibility of Department of Transport & Planning Faculty of Civil Engineering and Geosciences Delft University of Technology
doi:10.1016/j.trpro.2014.09.030

the locations where they disappear as *sinks* (potential destinations). Intuitively, detecting sources and sinks implies detection and tracking: this kind of approach was adopted by Stauffer (2003), which analyses low density situations and essentially relies on the performance of the tracking algorithm. However, in crowded scenes, where the number of objects is often in the order of hundreds, these tasks usually fail due to (i) the variable and potentially low number of pixels per object and (ii) frequent and severe occlusions related to the constant interaction among the objects (pedestrians) in a crowded scene. Research in this area has therefore instead assumed that raw data about pedestrian paths can be noisy or unreliable: Nedrich and Davis (2010), for instance, employ a so-called weak tracking system and they aggregate raw *tracklets* through a mean-shift clustering technique allowing them to identify entry and exit zones in the scene. More recently, research has focused on gathering global motion information at higher level, often based on optical flow analysis. In this paper, we propose an algorithm in which a scene is overlaid by a grid of particles initializing a dynamical system defined by optical flow, as discussed by Solmaz et al. (2012). Time integration of the dynamical system over a segment of the video provides particle trajectories (tracklets) that represent motion patterns in the scene for a certain time interval associated to the analysed segment. We detect sources, sinks and main flows in the segment (for sake of brevity sometimes we will refer to this information as segment local *track*) by analyzing motion patterns followed by clusters of tracklets, obtained using an unsupervised hierarchical clustering algorithm, where the similarity is measured by the Longest Common Sub-sequence (LCSS). We also detect and analyse dominant motion patterns representing multiple flows to further characterize detected origins and destinations. Finally, local segment information is combined to achieve a global set of tracks identifying sources and sinks and characterizing the flow of pedestrians connecting them. The overall framework is summarized in Fig. 1(a) and exemplified in Fig. 1(b). The following Sections will introduce the various steps of the process and they will describe its application to a video taken in a real-world case study in which several tens of perspective students reach the site where an admission test to the University was being held, which is discussed by Federici et al. (2014). A discussion of the achieved results and future works will end the paper.

2. Optical flow computation

The first step for the overall sources and sinks identification and motion flows characterization, given a video sequence, is to divide the input video into N number of segments s, each containing K frames. Next, we compute optical flow field between two consecutive frames of every segment. We employ the method proposed by Brox et al. (2004) where gray value constancy, gradient constancy, smoothness, and multi-scale constraints were used to compute highly accurate optical flow. Consider a feature point i in frame associate to time t of a segment: its flow vector $Z_{i,t}$ includes its location $X_{i,t} = (x_{i,t}, y_{i,t})$ and its velocity vector $V_{i,t} = (v_{x_{i},t}, v_{y_{i,t}})$, i. e. $Z_{i,t} = (X_{i,t}, V_{i,t})$ where θ_i is the angle or direction of V_i, where $0° \leq \theta \leq 360°$. Then $\{Z_1, Z_2, \ldots, Z_k\}$ is the motion flow field of all the foreground points of an image. We can thus initialize a continuous dynamical system in which the velocity of a point at time t is essentially related to the optical flow in the same point:

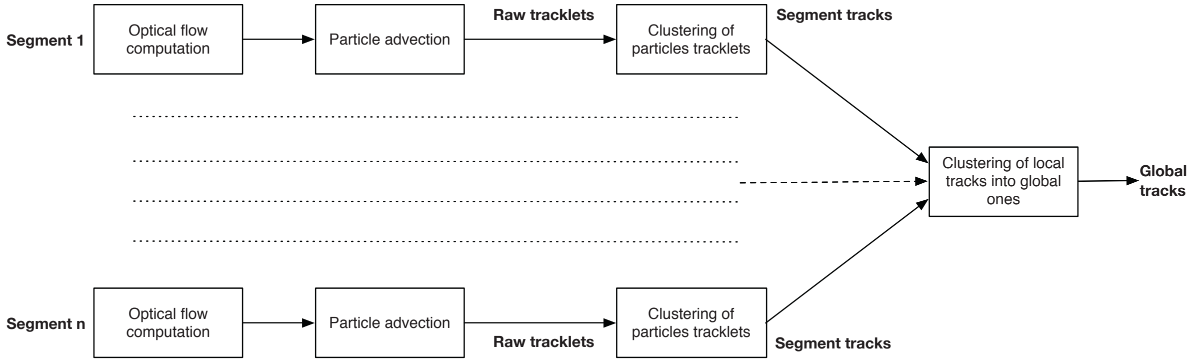
$$V_{i,t} = F(X_{i,t}) \tag{1}$$

3. Particle advection

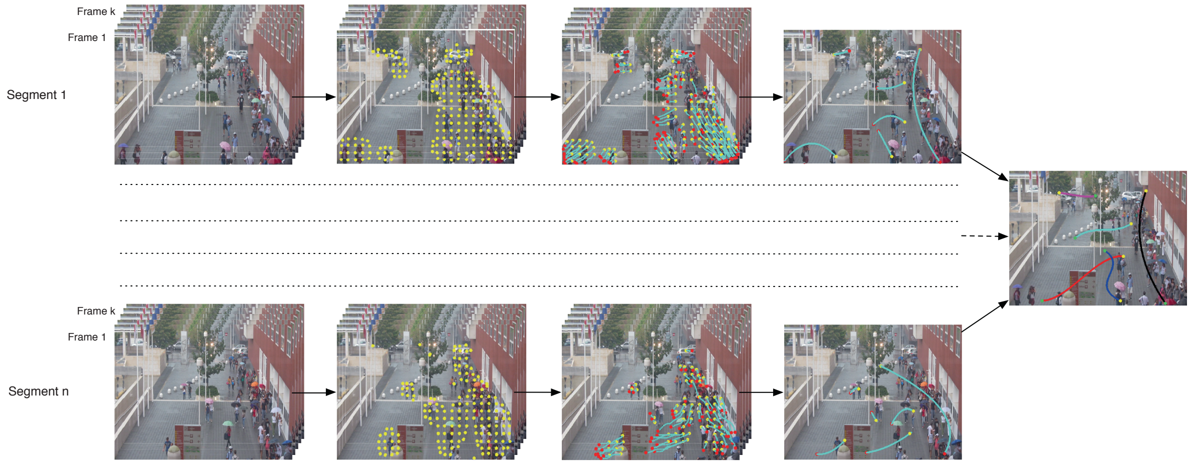
The next step is to advect grid of particles over the optical flow field, that corresponds to the time interval 1 to T for each segment. We launch a grid of particles over the first optical flow field of every segment and each initial position of the particle represents the source point. Ideally, the grid should have the same resolution of the base frame, but the computational cost would be extremely high. The trade-off choice we adopted it to overly particles on the points which have non-zero optical flow which means, due to equation 1, non-zero velocity, i.e. foreground points. An example of this mesh of particles placed over the flow field and their corresponding trajectories is provided in Fig. 1(b). The new position $X_{(t+1)}$ of a particle at time $t + 1$ is computed by numerically solving the system of equations 1 by using this approximation:

$$X_{(t+1)} = F(X_{(t)}) + X_{(t)} \tag{2}$$

During forward integration, a pair of flow maps, ψ_x and ψ_y, are maintained for every grid of particles. These flow maps indicate the relation between the initial position of each particle to its later position obtained after the advection



(a) Block diagram of the proposed approach.



(b) Sample frames of the application of the approach to the case study.

Fig. 1. The proposed framework for sources and sinks detection.

process. The flow maps integrate motion over longer durations of time. The flow map ψ_x keeps track of how x coordinate is changing, and similarly, ψ_y keeps track of y coordinate. As a result of this evolution of particles, small tracklets are generated as shown in Fig. 1(b), where the initial point (source point) marked with yellow and the last point, where the particle stops (sink point) marked with red. The tracklet ends either when the analyzed sequence is finished or in specific conditions, like an occlusion or the merging into a different flow.

Similar tracklets could be generated using Kanade-Lucas-Tomasi (KLT) as done by Cheriyadat and Radke (2008) and by the already cited Stauffer (2003), or by SIFT features tracking, as discussed by Battiato et al. (2007). However, these techniques are based on feature extraction and they may not able to capture whole motion information of a scene, as argued by Wang et al. (2013).

4. Longest Common Sub-Sequence

Tracklets achieved as a result of the previous step represent a relatively raw kind of information, plausibly about the movement of an individual, and they can be fruitfully grouped to provide a more concise description of overall flows in the scene. In the next step, therefore, we cluster tracklets that are spatially close to each other and have similar direction of motion. This procedure requires a similarity metric. A survey of different similarity measures for trajectory clustering is reported by Zhang et al. (2006): Euclidean and Dynamic Time Warping (DTW) are more sensitive to noise whereas Longest Common Sub-Sequence is efficient for series of unequal lengths and it is more robust to noise and outliers than DTW, as discussed by Vlachos et al. (2002) and by Cheriyadat and Radke (2007).

The key idea of LCSS is to match two time-series of tracklets by not considering all points of the tracklets, that can, to a certain extent, have different lengths. The following procedure allows verifying to which extent two trajectories can be considered similar (matching, according to a certain similarity measure) and therefore what is the longest portion they have in common. Let T_1 and T_2 represent two tracklets with size n and m respectively: $T_1 = \{(x_t, y_t), t = 1, ..., n\}$ and T_2 having analogous structure but m elements; with $T_1(i)$ we denote (x_i, y_i) with $0 \leq i \leq n$ and analogously for T_2. We compute the similarity among two tracklets by recursively finding a matching M between portions of these trajectories using a dynamic programming procedure that we will only briefly introduce for sake of space. Two constants are needed, respectively Φ controlling matching sequences in time and Ω which is the spatial matching threshold. Formally the matching matrix M comparing T_1 and T_2 can be computed recursively as follows:

$$M_{i,j} = \begin{cases} 0, & \text{if } i \text{ or } j \text{ are } 0 \\ 1 + M_{i-1,j-1}, & \text{if } \| T_1(i) \text{ - } T_2(j) \|_2 < \Omega \text{ and } | i - j | < \Phi \\ max(M_{i-1,j}, M_{i,j-1}), & \text{otherwise} \end{cases}$$

The similarity measure between two tracklets T_1 and T_2 is therefore $S(T_1, T_2) = \frac{LCSS(T_1,T_2)}{min(n,m)}$, where $LCSS$ is the number of matching points between T_1 and T_2, according to the above matching matrix.

5. Particle's Trajectories Clustering

The classical supervised clustering algorithms can not be used as the number of flows (and therefore desired clusters) are unknown. Therefore, we propose a novel hierarchical clustering algorithm, based on the following procedure.

1. We sort the tracklets in descending order on the basis of their length. Let L = $\{T_1, T_2, \ldots, T_k\}$ represent sorted list of tracklets and $\{N_1, N_2, \ldots, N_k\}$ represents length of tracklets. Tracklets with length lower than a fixed threshold are removed from the list, because considered noise.
2. We set up a list of tracklets to be considered L_T, initially the complete list of tracklets excluding T_1; we also set up a list of clusters L_C, initially containing a cluster associated to the first tracklet T_1 (the longest one) that is also used as initial cluster center;
3. We select shortest tracklet from the list, say T_s, and compare it with the centers of all clusters present in L_C using longest common sub-sequence to compute similarity measure. If the similarity measure is greater than a threshold, then tracklet T_s is assigned to the currently considered cluster, and the center is updated by using K^{th} order least square polynomial regression. Otherwise, a new cluster is inserted in L_C, tracklet T_s is assigned to it and set as its center. We delete the tracklet T_s from the list L_T after assignment to a cluster.
4. We repeat the previous step until L_T is not empty.

We employ this clustering procedure in two different phases: (i) when analyzing tracklets inside a sequence, to achieve local tracks within the related time interval; (ii) when combining local tracks into the global one, describing the motion pattern of the total length of the video. In the first phase, as exemplified in Fig. 1(b), the clustering aggregates a number of raw tracklets to generate a lower number of more significant tracks; in the second phase, as shown in Fig. 2, it combines portions of tracks into longer ones.

In particular, we can see that:

- the identified tracks are relatively robust to occlusions: for instance, the pole and tree in the middle of the scene and the signboard at the bottom do not prevent the identification of flows behind these obstacles; locally, the occlusion can cause the loss of particles but the overall integration procedure is often able to compensate if the particles appear at a later stage;
- tracks can be associated to a weight representing the number of clustered tracklets so they can be characterized according to how many pedestrians follow that macro-trajectory;
- sources and sinks are identified in a robust way, nonetheless, a commonsense form of spatial reasoning correlating the implied flows could further improve the results: for instance, two of the identified flows moving

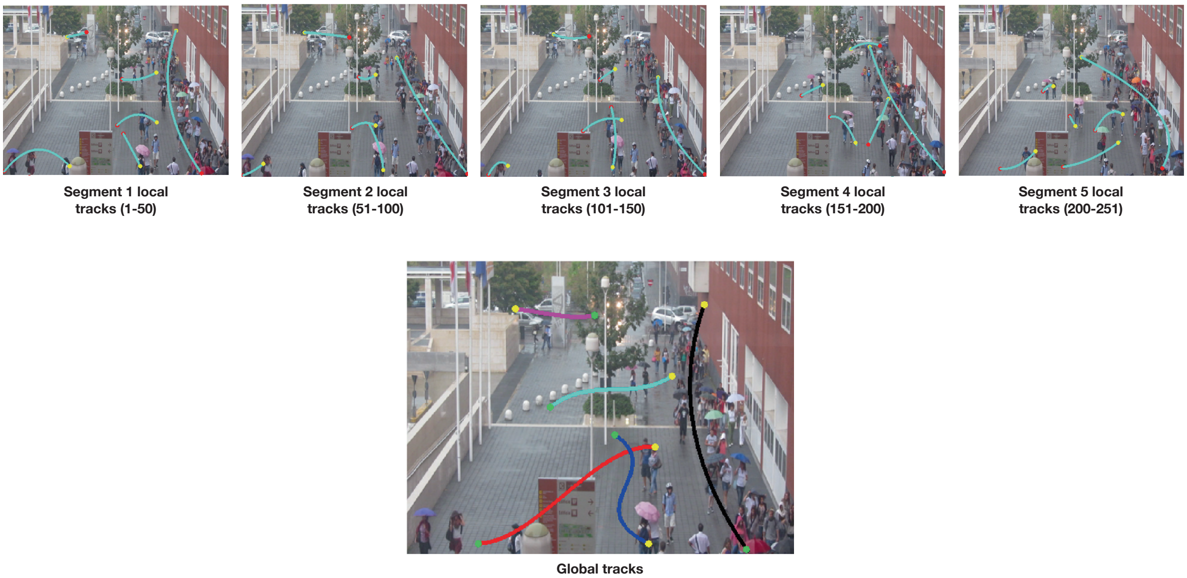


Fig. 2. Clustering of local tracks into global one in the case study.

towards the bottom-left part of the scene actually start very close to intermediate points of the rightmost flow. Actually, these flows are generated by this large flow whose source is located in the upper-right part of the scene. Heuristic rules for the identification of bifurcations, junctions or crossings could be easily defined and they are object of future works.

6. Conclusions and future developments

In this paper we proposed an algorithm for the automated detection of starting points (sources), dominant motion patterns and ending points (sinks) of pedestrian flows in videos taken from fixed cameras. The procedure employs a grid of particles initializing a dynamical system defined by optical flow, a high level global motion information. The dynamical system is integrated through time and this process provides particle trajectories (tracklets) that represent motion patterns in segments of the scene. Sources and sinks are detected and characterized by analyzing motion patterns followed by clusters of tracklets, obtained using a hierarchical unsupervised clustering algorithm, where the similarity is measured by the Longest Common Subsequence metric. Local segment information are combined to achieve a global set of traces identifying sources and sinks, and characterizing the flow of pedestrians connecting them. Current results are encouraging and future works are aimed at, on the one hand, completing the process with a commonsense spatial reasoning step or other semantical processing to correlate identified tracks, identifying bifurcations, junctions and crossings. In general, this kind of approach can support the identification and understanding of crowd behavior and it represents a starting point for multi-camera approaches.

Acknowledgements

This work was partly supported by the ALIAS project (“Higher education and internationalization for the Ageing Society”), funded by Fondazione CARIPLO.

References

Bandini, S., Gorrini, A., Vizzari, G., 2014. Towards an integrated approach to crowd analysis and crowd synthesis: A case study and first results. Pattern Recognition Letters 44, 16–29.

Battiato, S., Gallo, G., Puglisi, G., Scellato, S., 2007. Sift features tracking for video stabilization, in: Image Analysis and Processing, 2007. ICIAP

2007. 14th International Conference on, IEEE. pp. 825–830.
Brox, T., Bruhn, A., Papenberg, N., Weickert, J., 2004. High accuracy optical flow estimation based on a theory for warping, in: Computer Vision-ECCV 2004. Springer, pp. 25–36.
Challenger, R., Clegg, C.W., Robinson, M.A., 2009. Understanding Crowd Behaviours: Guidance and Lessons Identified. Technical Report. University of Leeds.
Cheriyadat, A.M., Radke, R.J., 2007. Automatically determining dominant motions in crowded scenes by clustering partial feature trajectories, in: Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on, IEEE. pp. 52–58.
Cheriyadat, A.M., Radke, R.J., 2008. Detecting dominant motions in dense crowds. Selected Topics in Signal Processing, IEEE Journal of 2, 568–581.
Federici, M., Gorrini, A., Manenti, L., Vizzari, G., 2014. An innovative scenario for pedestrian data collection: The observation of an admission test at the university of milano-bicocca, in: Weidmann, U., Kirsch, U., Schreckenberg, M. (Eds.), Pedestrian and Evacuation Dynamics 2012. Springer International Publishing, pp. 143–150.
Lee, J., Lam, W., Wong, S., 2001. Pedestrian simulation model for hong kong underground stations, in: Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE, pp. 554–558. doi:10.1109/ITSC.2001.948719.
Nedrich, M., Davis, J., 2010. Learning scene entries and exits using coherent motion regions, in: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Chung, R., Hammoud, R., Hussain, M., Kar-Han, T., Crawfis, R., Thalmann, D., Kao, D., Avila, L. (Eds.), Advances in Visual Computing. Springer Berlin Heidelberg. volume 6453 of *Lecture Notes in Computer Science*, pp. 120–131.
Solmaz, B., Moore, B.E., Shah, M., 2012. Identifying behaviors in crowd scenes using stability analysis for dynamical systems. IEEE Trans. Pattern Anal. Mach. Intell. 34, 2064–2070.
Stauffer, C., 2003. Estimating tracking sources and sinks, in: Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on, pp. 35–35. doi:10.1109/CVPRW.2003.10036.
Vlachos, M., Kollios, G., Gunopulos, D., 2002. Discovering similar multidimensional trajectories, in: Data Engineering, 2002. Proceedings. 18th International Conference on, IEEE. pp. 673–684.
Wang, H., Kläser, A., Schmid, C., Liu, C.L., 2013. Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision 103, 60–79.
Zhang, Z., Huang, K., Tan, T., 2006. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes, in: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, IEEE. pp. 1135–1138.