PARTICLES CROSS-INFLUENCE FOR ENTITY GROUPING

Paolo Rota, Habib Ullah, Nicola Conci, Nicu Sebe, Francesco G.B. De Natale

DISI - University of Trento (Italy)

ABSTRACT

In this paper we propose a novel approach to detect and track moving entities in wide surveillance video. Considering the wide area covered by the camera, which makes the detection and tracking of humans, as well as the classification of their motion a complex task and resource consuming, we adopt a particle-based approach to highlight particles of interest and group them based on their motion properties. A cross-influence matrix is computed at the particle level identifying the relevant areas of the video, and pruning static particles and outliers. Based on the motion features of the particles marked as interacting with their neighbors, a learning procedure based on an MLP neural network is implemented, in order to create consistent groups, representing the moving entities to be tracked over time. The method has been tested on two publicly available datasets with different resolutions and motion characteristics.

Index Terms**—** Particle tracking, entity influence, social interactions.

1. INTRODUCTION

Understanding human activities through vision-based technologies has been a challenging and attracting research area since the beginning of the past decade [1, 2]. Thanks to the increase in performance of detection and tracking algorithms, the research focus has shifted towards the provisioning of a higher level of interpretation of the visual scene, which includes the identification of semantically meaningful events, such as, for example, role detection [3], people grouping [4], as well as crowd assemblage and crowd flow analysis [5] to name a few.

Rota et al. [6] propose a framework to detect social interactions exploiting the so-called proxemics cues. Starting from the relative displacement, and the speed and orientation of the involved subjects, interactions are defined as normal and abnormal. In [7] the authors exploit the contextual information as a relative contribution of distances and orientations in relation to a reference subject considered as an *anchor*. This information is then processed using a Spatio-Temporal Volume representation combined with Random Forests, to infer simple individual actions. In the framework of activity recognition, the authors in [8] propose a dual approach (top-down and bottom-up) to classify human activities jointly exploiting multiple detectors information and higher level behavioral features, also based on context. In the work by Lan et al. [9], the authors analyze the contextual information in a sport events so as to infer individual and group behaviors relying on the overall game situation.

However, in case the number of moving subjects in the scene becomes dense, and the chance of incurring in severe occlusions increases, detectors and trackers are likely to fail, and more generic approaches that analyze the motion flow should be exploited, as it commonly occurs in crowd motion analysis [10]. Although in this way the notion of *person* is neglected, due to the absence of an explicit detector, it is still possible to estimate, for example, the density of people, and the aggregation points in the monitored environment. This turns out to be an efficient pre-processing step for any further and more detailed analysis.

In this paper our objective is thus to propose a hybrid method to overcome the known limitations related to detectors, using a particle based approach, and inferring the presence of moving entities by observing the consistency of the motion features of particles over time. Our approach starts by considering each particle as a single entity, and, as such, with its intrinsic and extrinsic characteristics. The former are related to the motion of the particle itself, the latter refer to the influence that each particle has over the neighboring particles. These features, modeled as a Markovian process to preserve the time coherence, are then fed into a Multi-Layer Perceptron (MLP) neural network, which goal is to learn the motion properties of the particles and form coherent groups of entities sharing similar motion properties.

The main contributions of this paper are:

- the motion information is acquired through a model that combines the particles cross-influence and self-influence.
- the model we propose does not rely on pre-stored templates, and only considers motion features, ensuring fast computation also in very high resolution video;

The paper is structured as follows. In Section 2 we introduce the particle based method and the cross-influence computation; in Section 3 we describe the learning phase and the grouping structure, and in Section 4 we present the experimental results obtained on the recently released high definition UCLA and BIWI datasets. Conclusions remarks are discussed in Section 5.

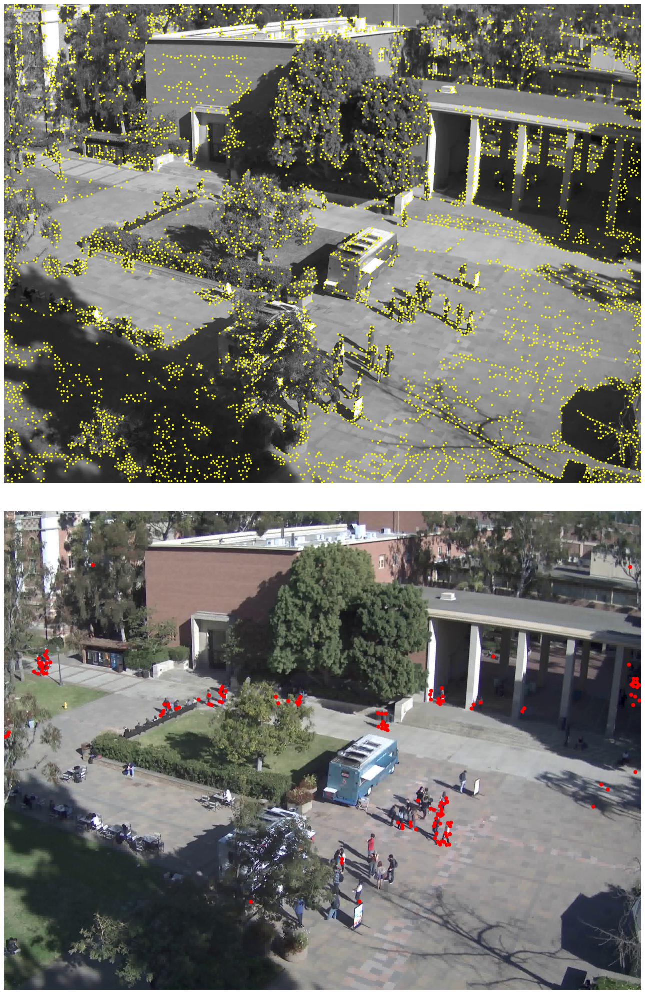


Fig. 2. An example of particle initialization (top) and after pruning (bottom).

2. PARTICLES MUTUAL INFLUENCE

As mentioned in Section 1, the first step of our method relies on particles dynamic properties. In our model we assume that each particle corresponds to an entity and has attractive and repulsive forces upon other particles surrounding it. Under this hypothesis, each particle can be classified not only on the basis of its own motion characteristics, but also in relation to the context, in this case provided by its neighbors.

As proposed in [11], the state of an entity $c \in C$ at time t, and defined as $h_t^{(c)}$ can be derived using a Markovian assumption, making it a direct temporal consequence of the state in the preceding temporal observation $(t-1)$. The influence among particles can be expressed by the so-called influence matrix. The influence matrix is a row stochastic matrix of dimension $C \times C$ where C is the number of the particles (entities) present in the given time window. In order to compute a single value of the matrix (see Fig.1) we assume that each particle relates with the others according to a Gaussian distribution, therefore the influence of c_i on c_j is computed as in Eq. (1), where d is the Euclidean distance between c_i and c_j.

$$R^{(c_i,c_j)} = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{d(c_i(t),c_j(t-1))}{2\sigma^2}} \tag{1}$$

As can be seen in Eq. (1), the motion information is already comprised by the formula, distance is computed indeed between the paticle c_i at time t and the particle c_j at time $t-1$. Particles are then classified in two states (h), *grouped* (G) or *alone* (A) according to the model in Eq. (2) and Eq. (3).

$$S\left(h_t^{(c_i)} \Rightarrow G\right) = \sum_{c_j \in \{1 \ldots C\} \wedge c_j \neq c_i} R^{(c_i,c_j)} \times P(G_t/h_{t-1}^{(c_j)}) \tag{2}$$

$$S\left(h_t^{(c_i)} \Rightarrow A\right) = R^{(c_i,c_i)} \times P(A_t/h_{t-1}^{(c_i)}) \tag{3}$$

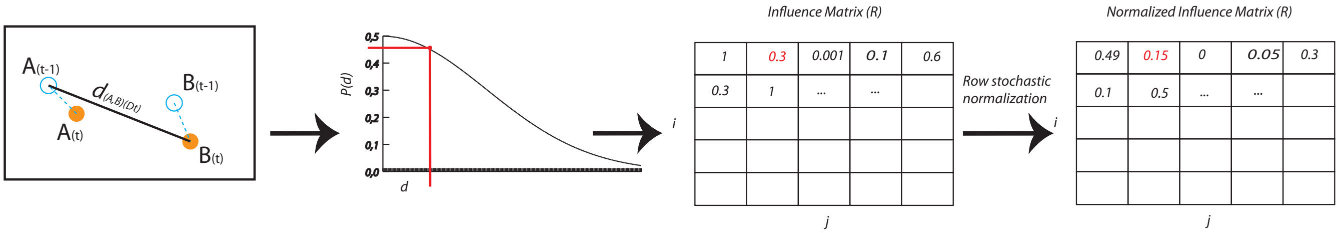
In the equations above, R represent the social influence matrix. The highest value of score S will dictate the state of the current particle in the current time window. For the purpose of this work just the *grouped* particles are considered relevant and passed to the feature extraction step described in Section 3.1. Particles marked as *alone* are instead pruned and not considered in the further processing steps. A visual example of pruned particles annotated in red is shown in the second row of Fig. 2.

In our work, particles are generated through the Good-Features-To-Track algorithm, and tracked by the Lucas-Kanade optical flow. The influence matrix is computed at discrete steps at the end of each tracking period.

3. ENTITY GROUPING

3.1. Feature Extraction

The objective of the features extraction process is to identify low-level information relative to the particles interaction. Features are extracted only for the particles obtained from the mutual influence model. In our approach we have selected the average distance among the particles and their density as two representative elements to infer the interaction among particles. In fact, proximity, which is partially exploited also in the influence model measures the instantaneous relationship among neighboring entities. At the same time, the higher the density of the particles, the higher the chance for them to interact.



Influence Matrix (R)

1	0.3	0.001	0.1	0.6
0.3	1	...	...	

Normalized Influence Matrix (R)

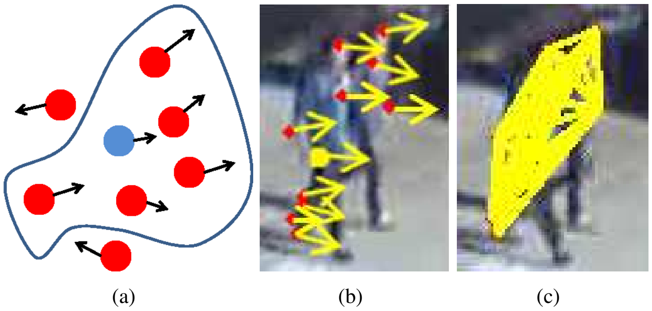
0.49	0.15	0	0.05	0.3
0.1	0.5	...	...	

Fig. 1. Computation of the particles influence matrix.

For both features, orientation is used as a prior, meaning that particles are considered in the same group, only if their relative offset in terms of direction of motion fall in a predefined range. For the purpose of visualization, in Fig. 3 (a), a set of synthetic entities are shown where a reference entity, annotated in blue, is grouped with the neighboring entities annotated in red. On the contrary, two entites are not included in the same group since their orientations do not conform to the orientation of the reference entity. Moreover, a reference entity annotated in yellow and neighboring entities annotated in red are shown in Fig. 3 (b). These entities constitute a group, as shown in (c), according to the compliance in terms of density and mutual distances with the reference entity.

3.2. Classification

In order to properly weight the features we have selected for entity grouping, we have trained a feedforward multi-layer perceptron (MLP) neural network. The motivation for exploiting MLP is in its substantial ability, through backpropagation, to resist to noise, and the dexterity to generalize. To group the particles from the preceding stage of the mutual influence model, the average distance of a reference particle with its neighbors is accumulated and averaged. A particle is only considered for grouping with a reference particle if its relative orientation is compliant with the orientation of a reference particle. The density and average distance of the reference particle are fed as an input to the MLP.



(a) (b) (c)

Fig. 3. Entities grouping. Synthetic example of moving entities (a), moving entities obtained from the particles mutual influence model (b) and grouping implemented according to the motion and density features (c).

The output y is obtained by the propagation of input x through the hidden layers as in Eq. (4), where y^0 is an input vector.

$$y^0 \xrightarrow{W^1,\, b^1} y^1 \xrightarrow{W^2,\, b^2} \ldots. \xrightarrow{W^L,\, b^L} y^L \tag{4}$$

In MLP networks, there are $L+1$ layers of neurons, and L layers of weights. During the training stage, the weights W and biases b are updated so that the actual output y^L becomes closer to the desired output d. For this purpose, a cost function is defined as in Eq. (5).

$$E\left(W, b\right) = \frac{1}{2}\sum_{i=1}^{n_l}(d_i - y_i^L)^2 \tag{5}$$

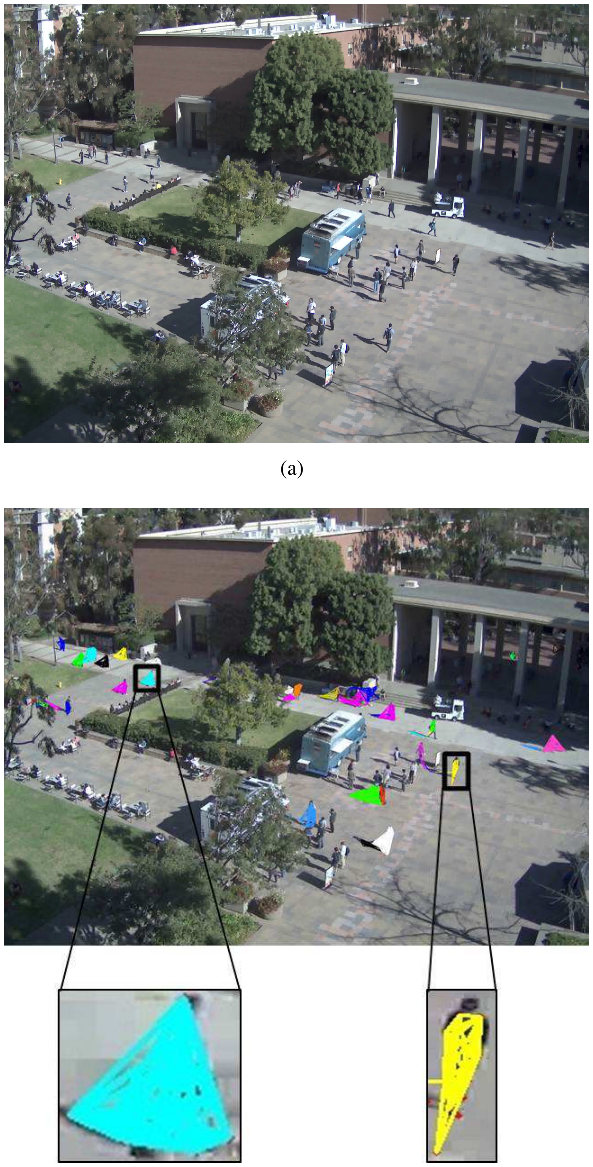
The cost function measures the squared error between the desired and actual output vectors. The backpropagation is gradient descent on the cost function in Eq. (5). Therefore, during the training stage, weights and biases are updated according to Eq. (6) and Eq. (7).

$$\Delta W_{ij}^l = -\eta \frac{\partial E}{\partial W_{ij}^l} \tag{6}$$

$$\Delta b_{ij}^l = -\eta \frac{\partial E}{\partial b_{ij}^l} \tag{7}$$

The learned weights and biases are used to predict groups from the inputs during testing stage.

In Fig. 4, the tracked groups of entities are shown. Two groups, annotated in cyan (left) and yellow (right) respectively, are zoomed and shown in the third row of Fig. 4. Primarily, entities are snipped with the particles mutual influence model and propagated over a predefined temporal window to associate them in groups in consonance with the features. At the same time, these groups are then mapped to a new set of snipped entities, with mutual influence model, which are then tracked over the same temporal window and the re-association process is repeated over time.



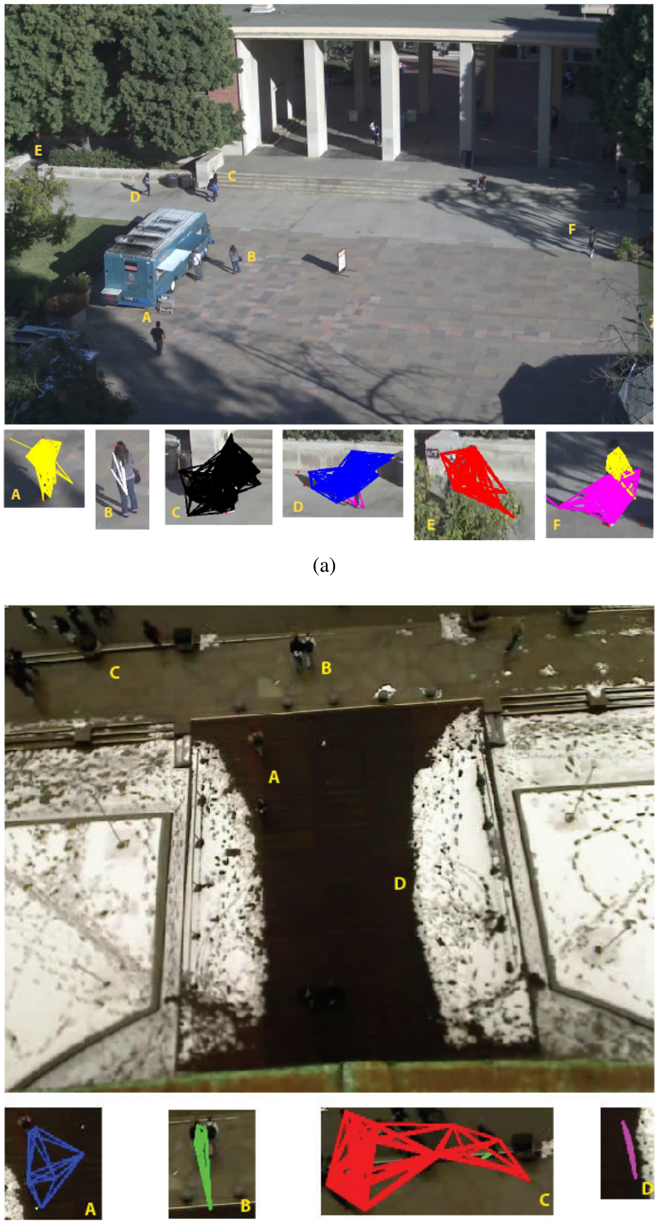
(a)

(b)

Fig. 4. Input frame (a), entities grouping with the zoom on two sample groups (b).

4. RESULTS

For the experiments, we have considered two datasets: the UCLA Courtyard dataset [8] and the BIWI Walking Pedestrians dataset [12]. The UCLA dataset consists of two distinct scenes from a wide top/side viewpoint of a courtyard at the UCLA campus. The dataset comprises of a 106-minute video, 30 fps, and 2560 x 1920 resolution. The dataset presents human activities including walking, talking, riding-skateboard, riding-bike and driving car. The BIWI dataset includes 2 videos at the resolution of 640 x 480, 25 fps. Here we have considered only the ETH sequence because of the exclusive presence of pedestrians.



(a)

(b)

Fig. 5. Particle influence and entity grouping. Results obtained on the UCLA dataset (a) and on the BIWI dataset (b). For visibility, labels are super-imposed on the original frame and the corresponding grouped entities are zoomed in lower row.

For the influence model, we have configured the following parameters: the length of the time window (time lapse between two observations in the influence model) has been set to 45 frames and the standard deviation of the Gaussian in Eq. (1) has been set to 0.8. These parameters are kept constant for both sequences to demonstrate the capability of generalization.

The neural network has been configured considering one input layer, two hidden layers and one output layer. The input layer consists of two neurons, each hidden layer consists of three neurons, and a single neuron is allocated to the output layer. The configuration of neural network in terms of number of layers and number of neuron does not affect the performance significantly. To extract the input features, the relative orientation with a reference particle is set to ± 30 degrees. Furthermore, the distance threshold from the reference particle is set to 80 pixels. For the purpose of training, we exploited 1000 training samples, where each sample is a vector of two observations namely; average distance and density of particles.

The obtained results are displayed in Fig. 5. The first image shows an example of the method applied on the UCLA dataset, where we can notice a very clear group composition, especially for zones A, D and E. In zone B the number of particles is not dense as in the previous cases, but still grouping is possible since the distance and density features of the entities are sufficient for the neural network. In zone F, instead, two groups have been detected instead of a single one; this can be ascribed to the severe shadowing, in which the pedestrian is located where, in fact, features of the entities are mainly segmented into two groups.

The quality of the results is also confirmed by the ETH sequence, where the groups are well defined (zones A, B, and C). However, in this case a few mistakes are also present (zone D). This is most probably connected to the limited resolution and the compression artifacts.

The UCLA dataset have much better results in terms of grouping not only because of the resolution but also because the bird eye view is less accentuated and the illumination conditions are considerably better.

5. CONCLUSION

In this paper, we have proposed a method to detect and track moving entities using a particle-based approach. According to the mutual influence model, particles are analyzed on the basis of their dynamic properties. For this purpose, we have extracted the average distance and density features for the particles sharing similar orientation. The obtained features are then exploited to train a multi-layer perceptron neural network, using the back propagation algorithm. Experimental results on two benchmark datasets, UCLA and BIWI, have demonstrated that our method can be efficiently used to detect and track moving entities present in the scene at a low computational burden, thus saving precious resources for any further processing step, compared to more traditional approaches based on detectors.

6. REFERENCES

[1] N. M. Oliver, B. Rosario, and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *PAMI, IEEE Transactions on*, vol. 22, no. 8, pp. 831–843, 2000.

[2] S. Hongeng, R. Nevatia, and F. Bremond, "Video-based event recognition: activity representation and probabilistic recognition methods," *Computer Vision and Image Understanding*, vol. 96, no. 2, pp. 129–162, 2004.

[3] A. Fathi, J. K. Hodgins, and J. M. Rehg, "Social interactions: A first-person perspective," in *CVPR*. IEEE, 2012, pp. 1226–1233.

[4] S. Pellegrini, A. Ess, and L. Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," *ECCV*, pp. 452–465, 2010.

[5] B. Solmaz, B. E. Moore, and M. Shah, "Identifying behaviors in crowd scenes using stability analysis for dynamical systems," *PAMI, IEEE Transactions on*, vol. 34, no. 10, pp. 2064–2070, 2012.

[6] P. Rota, N. Conci, and N. Sebe, "Real time detection of social interactions in surveillance video," in *ECCV. Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*. Springer, 2012, pp. 111–120.

[7] W. Choi, K. Shahid, and S. Savarese, "Learning context for collective activity recognition," in *CVPR*. IEEE, 2011, pp. 3273–3280.

[8] M. R. Amer, D. Xie, M. Zhao, S. Todorovic, and S. C. Zhu, "Cost-sensitive top-down/bottom-up inference for multiscale activity recognition," in *ECCV*, 2012.

[9] T. Lan, L. Sigal, and G. Mori, "Social roles in hierarchical models for human activity recognition," in *CVPR*. IEEE, 2012, pp. 1354–1361.

[10] H. Ullah and N. Conci, "Crowd motion segmentation and anomaly detection vis multi-label optimization," in *ICPR workshop on Pattern Recognition and Crowd Analysis*, 2012.

[11] W. Pan, W. Dong, M. Cebrian, T. Kim, J. H. Fowler, and A. S. Pentland, "Modeling dynamical influence in human interaction: Using data to make better inferences about influence within social systems," *Signal Processing Magazine, IEEE*, vol. 29, no. 2, pp. 77–86, 2012.

[12] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *ICCV*, 2009, pp. 261–268.