Deep Trajectory Representation-Based Clustering for Motion Pattern Extraction in Videos

Jonathan Boyle, Tahir Nawaz, and James Ferryman
Computational Vision Group, Department of Computer Science, University of Reading, UK
{j.n.boyle | t.h.nawaz | j.m.ferryman}@reading.ac.uk

Abstract

We present a deep trajectory feature representation approach to aid trajectory clustering and motion pattern extraction in videos. The proposed feature representation includes the use of a neural network-based approach that uses the output of the smallest hidden layer of a trained auto-encoder to encapsulate trajectory information. The trajectory features are then fed into a mean-shift clustering framework with an adaptive bandwidth parameter computation to yield dominant trajectory clusters. The corresponding motion patterns are extracted based on a distance minimization from the clusters' centroids. We show the effectiveness of the proposed approach on challenging public datasets involving traffic as well non-traffic scenarios.

1. Introduction

A trajectory is defined as an evolution of target states over time estimated by a video tracker on the image plane. The use of trajectory information could aid in several tasks [10, 16, 4, 22, 17, 13] including that of extracting motion patterns of moving targets in a scene [7, 12]. Indeed, several approaches exist for representing trajectory information in the form of features. Some approaches encode trajectory information in the *spatio-temporal domain*. A simple spatio-temporal feature involves the use of start and end points [12] that could be useful to distinguish between short and long trajectories. The directional distance computed using start and end points, the mean of trajectory points, the average target velocity, the indices corresponding to the three peaks of the directional histogram along a trajectory, the two dominant eigen vectors, and the quadratic polynomial coefficients have all been used as trajectory features as well [1]. A combination of directional information and the polynomial coefficients is also employed to encode trajectory information [25]. Compared to the feature extraction in spatio-temporal domain, the feature extraction in the *frequency domain* has been shown to be more effective for trajectory clustering and motion pattern extraction [7, 12]. For example, the first few Discrete Fourier Transform (DFT) coefficients are used to encode trajectory information in [7]. Trajectory information is also represented in the form of a non-parametric distribution of the Discrete Wavelet Transform (DWT) trajectory coefficients [12]. More recently, the focus has shifted towards the use of a different class of features that employ *deep learning* approaches for representing optical-flow-based dense trajectories [19, 14] or estimated target trajectories [6, 17]; however, these approaches have not been aimed at trajectory clustering and motion pattern extraction. Therefore, the needs remains to investigate and study effective deep feature representations in the context of trajectory clustering and pattern extraction.

Trajectory features are often fed into clustering stage for extracting motion patterns [1, 7, 12]. A hierarchical clustering approach was proposed for separately clustering trajectories of persons and vehicles in order to identify motion patterns using a point-based trajectory representation [8]. A framework was presented for trajectory clustering and pattern extraction that involved fusion of the results of multiple spatio-termporal features [1]. A Dynamic Dual Hierarchical Dirichlet Processes based approach for motion pattern learning was introduced that used target positional and directional information as trajectory features [20]. A Kernal Density Estimation-based model was proposed to extract motion patterns using a point-based trajectory representation by analyzing long-term tracking information [16]. The Random Field Topic model was introduced for clustering tracklets (short-term trajectories) to then learn motion patterns using a point-based feature representation [26]. Another method [24] also used tracklets in order to learn short-range motion patterns based on a computed motion map. A trajectory clustering and motion pattern learning framework was proposed that used DFT coefficients to represent trajectory information using Dirichlet Process Mixture Model (DPMM) [7]. An approach to extract trajectory patterns was presented in [12] that adopted a DWT-coefficient-based trajectory feature representation using the adaptive mean-shift clustering framework of [1]. Recently, an incremental tra-

978-1-5386-2939-0/17/$31.00

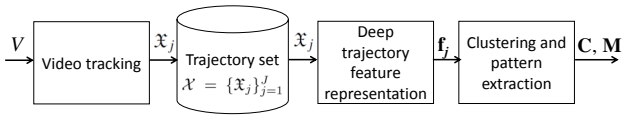

Figure 1. Proposed trajectory analysis framework invloving video tracking to estimate trajectories followed by a deep trajectory feature representation and clustering for motion pattern extraction. V: video sequence; $\mathcal{X}$: set of trajectories; $\mathfrak{X}_j$: trajectory j; $\mathbf{f}_j$: feature vector for $\mathfrak{X}_j$; $\mathbf{C}$: set of clusters; $\mathbf{M}$: set of motion patterns.

jectory clustering approach was introduced for activity pattern extraction in a non-parametric Bayesian framework [2]. There exists methods that do not rely on the use of trajectories but are based on the use of local motion information for motion pattern extraction [23, 21, 15, 9]; they are however more appropriate for short-term motion patterns [26, 9].

In this paper, we propose the use of an auto-encoder based deep trajectory feature representation for performing effective trajectory clustering to enable extraction of the key motion patterns in videos. We demonstrate the usefulness of the proposed method by quantitatively evaluating and comparing its performance with existing approaches on four challenging publicly available datasets.

This paper is organized as follows. Section 2 provides a formal definition of the problem, and Section 3 describes the proposed method. This is followed by an experimental validation in Section 4, and conclusions in Section 5.

2. Problem Definition

Let $\mathcal{X}$ be a set of trajectories estimated by a tracker in a video sequence, V: $\mathcal{X} = \{\mathfrak{X}_j\}_{j=1}^{J}$, where J is the number of estimated trajectories. $\mathfrak{X}_j$ is the estimated trajectory for target j: $\mathfrak{X}_j = (X_{k,j})_{k=k_{start}^j}^{k_{end}^j}$, where k_{start}^j and k_{end}^j are the first and final frame numbers of $\mathfrak{X}_j$, respectively. $X_{k,j}$ is the estimated state of target j at frame $k : k = 1, \ldots, K$ with K as the total number of frames in V. $X_{k,j} = (x_{k,j}, y_{k,j})$, where $(x_{k,j}, y_{k,j})$ denotes at frame k the position of target j on the image plane. The analysis of trajectories ($\mathcal{X}$) aids in identifying the motion patterns of moving objects (people, vehicles) in a scene. This could involve applying trajectory clustering on $\mathcal{X}$ in the feature space $\Psi_{\mathbf{f}}$ to produce a set of clusters: $\mathbf{C} = \{C_n\}_{n=1}^{N}$. A cluster C_n is then associated to a motion pattern, M_n, that represents a spatio-temporal trend of moving objects in a scene.

3. Proposed Trajectory Analysis Framework

This section describes the proposed framework for trajectory clustering and motion pattern extraction that is based on a deep trajectory feature representation (Fig. 1). We first describe the deep feature representation for trajectories in Sec. 3.1 that is followed by the trajectory clustering in Sec. 3.2.

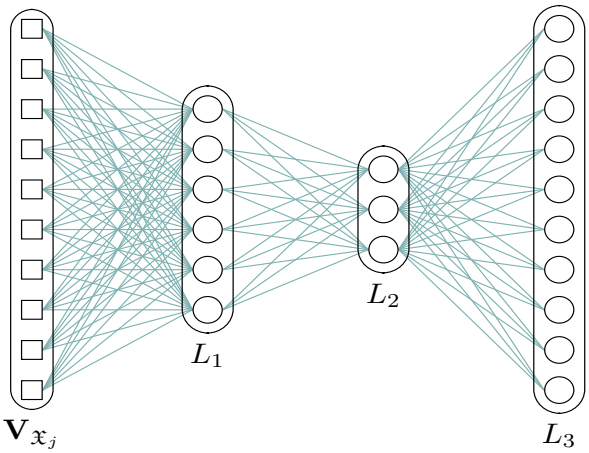

Figure 2. Proposed feature representation uses a neural network-based approach that employs the output of the smallest hidden layer (L_2) of a trained auto-encoder to represent trajectory information. $\mathbf{V}_{\mathfrak{X}_j}$: vectorization of the data of the jth trajectory ($\mathfrak{X}_j$); L_1: first hidden layer; L_3: output layer.

3.1. Deep Trajectory Feature Representation

In order to generate the feature vector $\mathbf{f}_j$ for a trajectory $\mathfrak{X}_j$, an auto-encoder (also known as autoassociator or diabolo network) [3] is first trained to reproduce the set of trajectories $\mathcal{X}$. Once a network has been trained, the output of its smallest layer is used as the feature vector $\mathbf{f}j$. A separate network has to be trained for each dataset due to the varying length of the input vectors described above. A network consisting only of fully-connected layers can not handle input vectors of varying sizes without introducing further methods of normalization than described above; because of this, and the difference in scenes of different datasets, it was opted to use separate networks.

3.1.1 Training the Auto-Encoder

An auto-encoder is an arrangement of a neural network where the output, once trained, is an estimation of the input vector that is provided (Fig. 2). Outputs of any of the layers in a trained network can be used as a representation of the input, due to the ability of the rest of the network to reproduce the input vector. In this case, the input is the vectorization of the data of $\mathfrak{X}_j$ and is denoted as $\mathbf{V}_{\mathfrak{X}_j}$:

$$\mathbf{V}_{\mathfrak{X}_j} = [x_{k,j}, y_{k,j}]_{k=k_{start}^j}^{k_{end}^j}. \quad (1)$$

A neural network is a combination of small units called neurons that are built up into multiple layers. The type of neuron used in this paper is based on the McCulloch-Pitts neuron [11]; this multiplies a single-dimensional input vector

Table 1. The hidden layer sizes of auto-encoder for different datasets.

Dataset	Input	L_1 Size	L_2 Size
Traffic Junction	3212	321	160
Parking Lot	5244	524	262
Students003	5746	574	287
Train Station	3024	302	151

with a weight vector summed with a bias node, and outputs a single value:

$$y_{l,n} = \sum_{i}^{I_l} (w_{i,n}\mathbf{x}_{l,i}) + b_n, \tag{2}$$

where $y_{l,n}$ donates the output pre-activation-function y of the neuron n in layer l. I_l is the length of the input vector X_l for a particular layer, w_i and $\mathbf{x}_{l,i}$ is a value in the weight vector and input vector, respectively, and b is the bias value. The weight vector is initialized to random values. A sigmoidal function is used for the activation function:

$$Y_{l,n} = \frac{1}{1 + e^{-y_{l,n}}}, \tag{3}$$

where Y is the output of the neuron. The neurons are then placed alongside each other as a layer. The output of a layer l can be described by the following equation:

$$L_l = [Y_{l,1}, ..., Y_{l,I_l}], \tag{4}$$

where the layer L_l is a vector containing all of the outputs of the neurons in the layer. The next layer is then provided with the output of the previous as it's input vector (known as a fully-connected layer), excluding the first layer ($l = 1$) for which the input vector is the vectorized trajectory $\mathbf{V}_{\mathfrak{x}_j}$ rather than a previous layer:

$$\mathbf{X}_l = \begin{cases} \mathbf{V}_{\mathfrak{x}_j}, & \text{if } l = 1; \\ L_{l-1}, & \text{if } l > 1. \end{cases} \tag{5}$$

The architecture of this type of network contains two stages: an encoding and decoding stage. For encoding, the number of neurons in each layer decreases so that the dimensionality of the original input vector is effectively reduced when passed through the network. The decoding stage is the opposite: a set of layers incrementing in size up to the original length of the input vector. Both of these stages consist of one or more layers. As mentioned above, this method uses pre-defined scales based on the length of the original feature vector to determine the number of neurons in each layer in the encoding stage. Two layers are used: the first is 10% the length of the vector, the second 5% of the vector. Only one layer is used in the decoding stage of the same size as the original vector. These scales are static across all of the datasets used in this study. The values of layer sizes are listed in Table 1.

3.1.2 Feature Extraction

All the information necessary to reproduce the input vector is provided in the output values of any layer in a trained auto-encoder network; the smallest layer at the end of the encoding stage provides the most reduced dimensionality, in this case the last hidden layer (a layer that is not the final layer in the network). These can be used as an effective representation of the input vectors passed through the network, due to their property of being able to reproduce the original trajectory. In this case, the feature vector used is the output of layer 2, when trajectory j is used as the input vector into layer 1:

$$\mathbf{f}_j = L_2. \tag{6}$$

3.2. Trajectory Clustering

The extracted features, $\mathbf{f}_j$, are fed into a clustering framework producing a set of clusters $\mathbf{C}$. In this regard, we used the adaptive mean-shift clustering algorithm of [1] that does not require knowledge of the number of clusters *a priori*. For an extracted cluster, C_n, the corresponding motion pattern, M_n, is identified by a trajectory based on a distance minimization from the cluster's centroid [12]. The qualitative results for the extracted clusters and motion patterns using the proposed method are shown in Fig. 3 on all of the datasets used in this study.

4. Experimental Validation

This section presents the experimental validation of the proposed method describing the datasets in Sec. 4.1, the evaluation criteria in Sec. 4.2, and the results in Sec. 4.3.

4.1. Datasets

We show the effectiveness of the proposed framework by evaluating and comparing its performance with exisiting approaches on four challenging publicly available datasets (Table 2). The first two are related to traffic monitoring and called Traffic Junction [12] and Parking Lot [12]. Traffic Junction offers a busy junction scenario with vehicles moving in varying directions as well as people walking across and alongside roads. Parking lot presents a multi-row car park scenario with vehicles and people targets. Both Traffic Junction and Parking Lot are recorded from a mobile aerial platform. For both datasets we use the real trajectories with induced camera motion already compensated as made available by the authors [12].

For a greater generalization of the proposed approach, we also evaluated it on two non-traffic datasets: Students003 [18] and Train Station [27]. Students003 offers a highly crowded scene with people walking around in an outdoor open area. Train Station also offers an open area

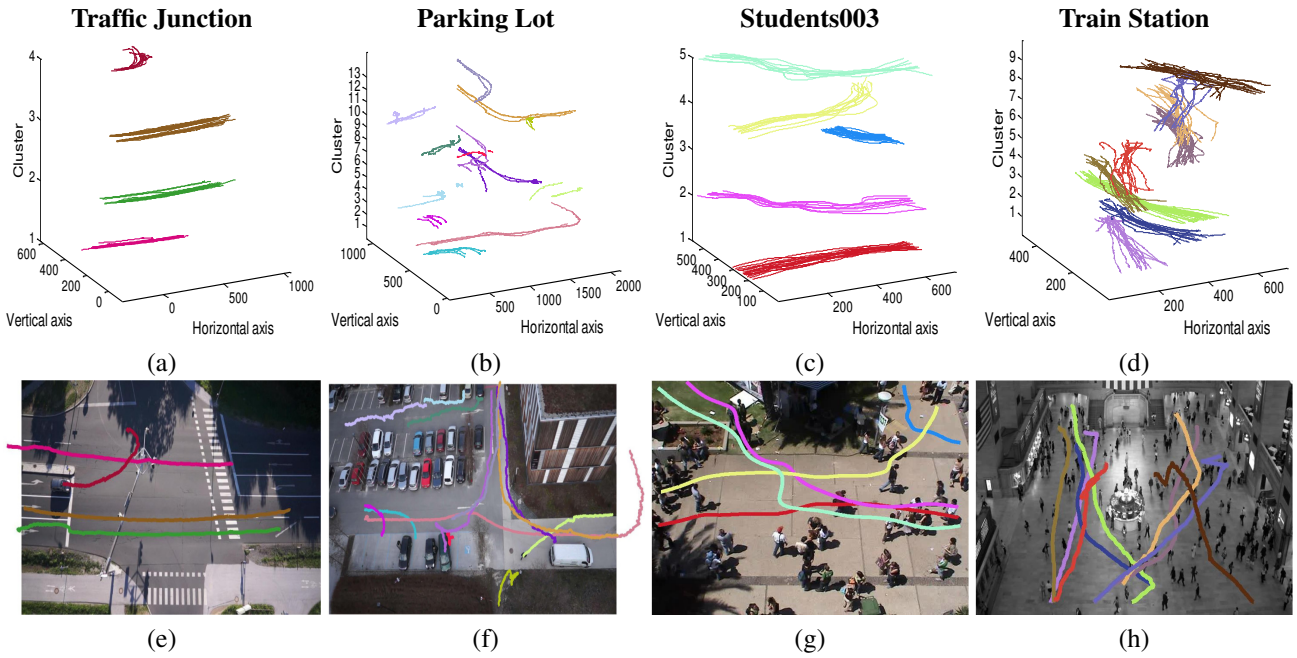

Figure 3. Color coded qualitative results obtained using the proposed method on four datasets: Traffic Junction, Parking Lot, Students003, Train Station. (a-d) For each dataset, trajectory clusters are shown on different planes along z-axis. (e-h) For each dataset, the corresponding extracted motion patterns are overlayed on original frames (e-h).

that is highly crowded with people moving inside a train station. Both Students003 and Train Station are recorded from a top-down(ish) fixed camera. On Students003 and Train Station we use the available trajectories by respective authors [18, 27]. For Train Station we use only longer trajectories (length>600) [12] because the processing of shorter trajectories (tracklets) is not within the scope of the proposed method.

Note that, while training the auto-encoder, each dataset is split into a training set (70%) and a testing set (30%). Only the training set for each dataset is seen during the training process. Due to the random initialization of the networks weight vectors, 30 networks were trained per dataset with the same training/testing set. The results from the training process are provided in terms of the average of the l_2-norms between the data in each original trajectory and the respective activations of the output layer (L_3) of the corresponding best network (Table 3). The network is trained for 40000 epochs using stochastic gradient descent on the training set, with the learning rate at 0.5 and stepping the learning rate every 500 epochs by 0.9. The training loss used is the cross-entropy loss.

Table 2. Summary of the datasets used in the study.

Dataset	Frame size	Number of frames	No. of trajectories	Frames per second
Traffic Junction	540 × 960	16154	236	30
Parking Lot	1080 × 1920	9517	54	30
Students003	576 × 720	5405	417	25
Train Station	480 × 720	46009	762	23

4.2. Evaluation Criteria

We evaluate trajectory clustering by using an accuracy measure (A) that provides the assessment by quantifying the concentration of trajectories having the same ground-truth cluster label and the highest proportion in each cluster, and then averaging it over all of the clusters [7]. For all datasets we use the provided ground truth cluster labeling by [12]. Additionally, we use the precision (P) and recall (R) measures to assess the extracted motion patterns. P provides the assessment by penalizing the correct (true positive) patterns with respect to incorrect (false positive) patterns. R provides the assessment by penalizing the correct (true positive) patterns with respect to missed (false negative) patterns. If an extracted pattern belongs to a ground-truth cluster, it is deemed correct. To account for the randomness of the clustering algorithm [1], on each dataset the mean A, P and R scores are computed for five runs [12].

Table 3. Average of the l_2-norms between the data in each original trajectory and the respective activations of the output layer (L_3) of the corresponding best network for all datasets.

Dataset	Average l_2-norm
Traffic Junction	2.7711
Parking Lot	10.4947
Students003	4.2008
Train Station	2.9301

Table 4. Evaluation results of the trajectory clustering and motion pattern extraction on all datasets based on A, P and R for different approaches: DFTfeat, MULTfeat, DWTfeat, and DEEPfeat (the proposed method).

Method	Traffic Junction			Parking Lot			Students003			Train Station		
	A	P	R	A	P	R	A	P	R	A	P	R
DFTfeat [7]	0.67	0.67	0.27	0.64	0.48	0.53	0.41	**0.90**	0.40	0.32	**0.60**	0.18
MULTfeat [1]	0.70	0.40	0.33	0.56	0.63	0.33	0.51	0.60	0.28	0.33	0.35	0.18
DWTfeat [12]	0.88	0.52	**0.50**	**0.89**	**0.65**	**1**	**0.90**	0.58	0.51	0.82	0.45	0.50
DEEPfeat (proposed)	**1**	**0.80**	**0.50**	0.72	0.38	0.83	**0.90**	0.64	**0.60**	**0.84**	0.47	**0.56**

4.3. Results

We evaluate and compare the performance of the proposed deep trajectory feature (here referred to as 'DEEPfeat') in the clustering and motion pattern extraction framework with 1) an approach that uses DFT coefficients of x- and y-coordinates to represent trajectory information [7] (referred to as 'DFTfeat'), 2) a method that uses multiple spatio-temporal trajectory features [1] (here referred to as 'MULTfeat'), and 3) an approach that represents trajectory information using a non-parametric distribution of DWT coefficients of x- and y-coordinates [12] (here referred to as 'DWTfeat'). Table 4 lists the evaluation results for DFTfeat, MULTfeat, DWTfeat and DEEPfeat.

The results demonstrate that DEEPfeat outperforms other approaches in terms of R on all datasets (except on Parking Lot where it is second best to DWTfeat), which shows it has generally missed the lowest number of motion patterns (Table 4). Likewise, in terms of A, DEEPfeat has again shown the best performance on all datasets except on Parking Lot where it is again the second best after DWTfeat. This means that the extracted clusters using DEEPfeat are mostly more meanignful and accurate than the remaining methods. Moreover, based on P, DEEPfeat shows a mixed performance: on Traffic Junction, it has achieved the best performance; on Students003 and Train Station, DEEPfeat is the second best after DFTfeat; and on Parking Lot, DEEPfeat stands the last in terms of the performance. The reason of the bad performance of DEEPfeat on Parking Lot is likely due to a comparatively smaller number of available trajectories, whereas deep learning approaches generally work better in the presence of a large enough amount of data [5].

5. Conclusions

This paper presented an approach that involved the use of of a deep feature representation to perform trajectory clustering for motion pattern extraction in videos. The proposed feature representation is based on an auto-encoder based neural network that employs the output of the smallest hidden layer of a trained auto-encoder. We evaluated and compared the proposed method with existing approaches on four challenging real public datasets.The results particularly show the superior performance of the proposed method in terms of (mostly) the highest clustering accuracy and the highest recall across all datasets.

Future work could involve investigating other methods for training neural networks as well as testing system's performance with deeper (larger number of hidden layers) neural networks within the proposed framework.

References

[1] N. Anjum and A. Cavallaro. Multi-feature object trajectory clustering for video analysis. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(11):1555 – 1564, 2008.

[2] V. Bastani, L. Marcenaro, and C. S. Regazzoni. Online nonparametric bayesian activity mining and analysis from surveillance video. *IEEE Trans. on Image Processing*, 25(5):2089–2102, 2016.

[3] Y. Bengio. Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

[4] S. Calderara, A. Prati, and R. Cucchiara. Mixtures of von mises distributions for people trajectory shape analysis. *IEEE Trans. on Circuits and Systems for Video Technology*, 21(4):457–471, 2011.

[5] X.-W. Chen and X. Lin. Big data deep learning: Challenges and perspectives. *IEEE Access*, 2:514–525, 2014.

[6] Z. Deng, A. Vahdat, H. Hu, and G. Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *Proc. of CVPR*, Las Vegas, NV, June 2016.

[7] W. Hu, X. Li, G. Tian, S. Maybank, and Z. Zhang. An incremental dpmm-based method for trajectory clustering, modeling, and retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(5):1051–1065, May 2013.

[8] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, September 2006.

[9] P.-M. Jodoin, Y. Benezeth, and Y. Wang. Meta-tracking for video scene understanding. In *Proc. of the IEEE Conf. on Advanced Video and Signal Based Surveillance*, Krakow, Poland, August 2013.

[10] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Trans. on Systems, Man and Cybernetics - Part B*, 35(3):397–408, June 2005.

[11] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

[12] T. Nawaz, A. Cavallaro, and B. Rinner. Trajectory clustering for motion pattern extraction in aerial videos. In *Proc. of ICIP*, Paris, October 2014.

[13] L. Patino, T. Nawaz, T. Cane, and J. Ferryman. Pets 2017: Dataset and challenge. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, Hawaii, July 2017.

[14] M. S. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. In *Proc. of CVPR*, Boston, MA, June 2015.

[15] I. Saleemi, L. Hartung, and M. Shah. Scene understanding by statistical modeling of motion patterns. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, June 2010.

[16] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(8):1472–1485, August 2009.

[17] T. Shu, S. Todorovic, and S.-C. Zhu. CERN: Confidence-energy recurrent network for group activity recognition. In *Proc. of CVPR*, Honolulu, Hawaii, July 2017.

[18] J. Sochman and D. C. Hogg. Who knows who - inverting the social force model for finding groups. In *Proc. of Int. Conf. on Computer Vision Workshops*, Barcelona, November 2011.

[19] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proc. of CVPR*, Boston, MA, June 2015.

[20] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson. Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. *IJCV*, 95(3):287–312, December 2011.

[21] X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(3):539–555, March 2009.

[22] X. Wang, K. Tieu, and W. E. L. Grimson. Correspondence-free activity analysis and scene modeling in multiple camera views. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(1):56–71, January 2010.

[23] S. Wu, B. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[24] B. Yang and R. Nevatia. Multi-target tracking by outline learning of non-linear motion patterns and robust appearance models. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[25] T. Zhang, H. Lu, and S. Z. Li. Learning semantic scene models by object classification and trajectory clustering. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, FL, June 2009.

[26] B. Zhou, X. Wang, and X. Tang. Random field topic model for semantic region analysis in crowded scenes from tracklets. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3441–3448, 2011.

[27] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.