Correspondence-Free Multi-Camera Activity Analysis and Scene Modeling

Xiaogang Wang Kinh Tieu W. Eric L. Grimson

Computer Science and Artificial Intelligence Lab, MIT,
77 Massachusetts Avenue, Cambridge, MA, 02139, USA
{xgwang, tieu, welg}@csail.mit.edu

Abstract

We propose a novel approach for activity analysis in multiple synchronized but uncalibrated static camera views. We assume that the topology of camera views is unknown and quite arbitrary, the fields of views covered by these cameras may have no overlap or any amount of overlap, and objects may move on different ground planes. Using low-level cues, objects are tracked in each of the camera views independently, and the positions and velocities of objects along trajectories are computed as features. Under a generative model, our approach jointly learns the distribution of an activity in the feature spaces of different camera views. It accomplishes two tasks: (1) grouping trajectories in different camera views belonging to the same activity into one cluster; (2) modeling paths commonly taken by objects across camera views. To our knowledge, no prior result of co-clustering trajectories in multiple camera views has been published. Advantages of this approach are that it does not require first solving the challenging correspondence problem, and the learning is unsupervised. Our approach is evaluated on two very large data sets with $22,951$ *and* $14,985$ *trajectories.*

1. Introduction

In visual surveillance, a key task is to monitor activities in the scene. In many surveillance systems, especially for far-field settings, objects are first detected and tracked. The activity of an object is then treated as sequential movements along its trajectory. Many approaches [13, 9, 17, 16, 7] have been proposed to cluster or classify trajectories of objects into different activities. They used the spatial proximity between a pair of trajectories, measured in different ways, for clustering. Since activities are often closely related to the structures of the scene, the models of paths commonly taken by objects can be learnt from clusters of trajectories [9, 10, 2, 16, 7].

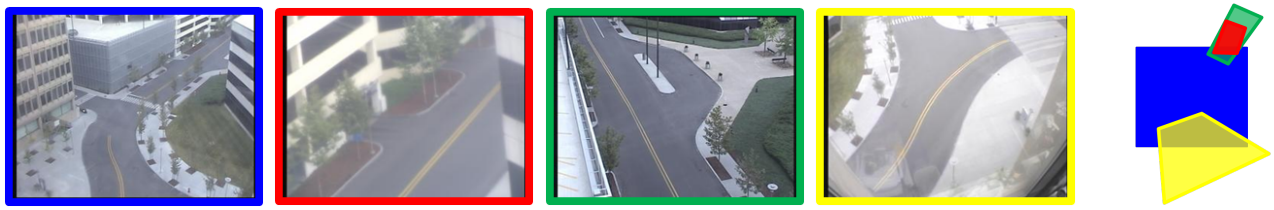
All these clustering and modeling approaches assumed a single camera view whose visible area is finite and limited by the structures of the scene. In order to monitor activities in a wide area video streams from multiple cameras have to be used. Because of the structures of the scene, the distribution and configuration of these cameras could be quite arbitrary. The camera views may have any combination of large, little, or even no overlap. The objects in the views may move on one or more ground planes. Analyzing activities over such a multi-camera network is quite challenging. A natural way of doing multi-camera surveillance is to first infer the topology of camera views [11, 15], solve the correspondence problem [8, 14, 12, 6], stitching the trajectories of the same object in different camera views into a complete long trajectory, and then analyze the stitched trajectories using the same approaches developed for a single camera view. However both inferring the topology of camera views and solving the multi-camera correspondence problem are notoriously difficult especially when the number of cameras is large and the topology of the cameras is arbitrary.

We propose an approach to group trajectories in different camera views and belong to the same activity into one cluster and to model the paths of objects across camera views. They are jointly learnt under a generative model, that is completely unsupervised and does not require the correspondence problem to be solved in advance. The cameras are static and synchronized but do not have to be calibrated. The fields of view covered by these cameras may have no overlap or any amount of overlap. Examples of multi-camera settings are shown in Figure 1.

We briefly explain several basic concepts used in this paper. There are paths in the physical world. Objects move along these paths and thus have different moving patterns, which are called activities. A path may be observed in multiple camera views and has spatial distributions in these views. A trajectory, which only records the positions of an object, is a history of the movement of an object in a camera view. The points on trajectories are called observations. In

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

(a) Parking lot scene



(b) Street scene

Figure 1. Camera views and their topology in two data sets, a parking lot scene and a street scene. When the topology of camera views is plotted, the fields of cameras are represented by different colors: blue (camera 1), red (camera 2), green (camera 3), yellow (camera 4). However, our approach does not require the knowledge of the topology of the camera views in advance.

this work, trajectories are clustered into different activities, based on their spatial distributions and moving directions. A cluster of trajectories is often related to a path. The scene of a camera view is quantized into small cells. When an object moves around, it connects two cells far apart in a camera view by its trajectory. Our generative model is based on some simple, general assumptions on the spatial and temporal features related to activities: (1) cells located on the same path are likely to be connected by trajectories; (2) trajectories passing through the same path belong to the same activity; (3) it is likely for trajectories of the same object observed in different cameras views to be on the same path in the real world and belong to the same activity.

In our approach, a network is first built by connecting trajectories that are in different camera views and whose temporal extents are close. Then a generative model, in which different kinds of activities have distributions in low-level feature spaces of different camera views, is built. A trajectory is treated as a set of observations that belong to different activities. The smoothness constraint of the trajectory network requires that two neighboring trajectories connected by an edge have similar distributions over activities. Trajectories are clustered according to the assigned major activities among their observations. The distributions of activities over feature spaces in different camera views model the semantic regions of paths across camera views. We show results on two data sets, each of which has four cameras. The views and topology of these cameras are shown in Figure 1.

2. Related Work

Many similarity-based trajectory clustering methods have been proposed. A comparison of different similarity measures can be found in [17]. The spatial extents can be estimated from trajectory clusters [2, 9, 16, 7]. They assumed that all of the trajectories are observed in a single camera view. In order to extend these approaches to multiple camera views, trajectories observed in different camera views have to be stitched together.

Considerable work has been done to solve the challenging correspondence problem in multiple camera views. Lee et al. [8] and Stauffer and Tieu [14] calibrated multiple camera views using tracking data from moving objects. They assumed that camera views had significant overlap and that objects moved on the same ground plane. Lee et al. [8] assumed that the topological arrangement of cameras was known. Stauffer and Tieu [14] could automatically infer it, but with high complexity ($O(N^2)$ where N is the number of cameras).

When the camera views are disjointed or their overlap is small, the appearance of objects is often used as a cue to correspondence [5, 6, 4]. This is a very challenging problem and not well solved yet. The appearance of objects may significantly change because of different cameras' settings and different poses of objects. Many objects, such as cars or persons, have similar appearance, confusing correspondence. In far-field settings, objects may only cover a few pixels, making comparison difficult. Other approaches [11, 15] inferred the topology of disjoint camera views using the transition time between cameras.

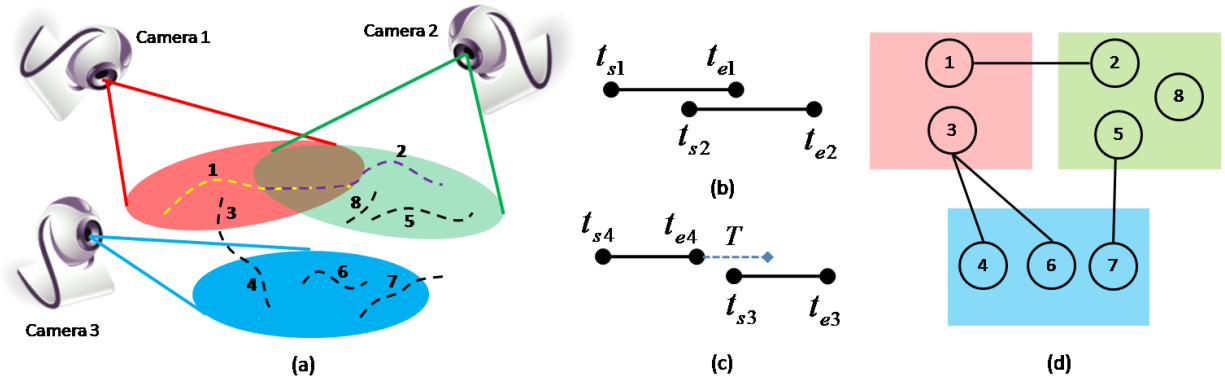


Figure 2. An example of building a network connecting trajectories in multiple cameras. (a) Trajectories in three camera views. (b) The temporal extents of trajectories 1 and 2. (c) The temporal extents of trajectories 3 and 4. (d) The trajectory network. See text for details.

Even given similarities between trajectories observed in different camera views, solving the correspondence problem is still difficult because of the large search space, especially when there are many trajectories and cameras. In general, if there are more than two cameras, the problem is *NP* hard in the number of trajectories [3].

Our approach does not require a solution to the correspondence problem. It has fewer constraints on the topology of camera views and the number of cameras.

3. Feature Space

Objects are tracked in each camera view independently using the Stauffer-Grimson tracker [13]. A trajectory is treated as a set of observations. The locations and moving directions of observations are computed as features and quantized to visual words according to a codebook of its camera view. In each camera view, the space of the view is uniformly quantized into small cells and the velocity of objects is quantized into several directions. A global codebook concatenates the codebooks of all the cameras. Thus the word value of an observation i is indexed by (c_i, x_i, y_i, d_i) in the global codebook. c_i is the camera in which i is observed. (x_i, y_i) and d_i are the quantized coordinates and moving direction of observation i in camera c_i. The set of visual words on the trajectory are modeled as exchangeable (i.e., the distribution is invariant to a permutation of the observations). Although quite simple, the position and velocity features can distinguish many different activity patterns especially in far-field settings.

4. Trajectory Network

A network is built connecting trajectories observed in multiple camera views based on their temporal extents. Each trajectory is a node in the network. Let t_{si} and t_{ei} be the starting and ending time of trajectory i. Let T be a positive temporal threshold. If trajectories a and b are in different camera views and their temporal extents are close,

$$(t_{sa} \leq t_{sb} \leq t_{ea} + T) \vee (t_{sb} \leq t_{sa} \leq t_{eb} + T), \quad (1)$$

then a and b will be connected by an edge in the network. This means that a and b are likely to be the same object. There is no edge between two trajectories observed in the same camera view. An example can be found in Figure 2. As shown in (a), the views of cameras 1 and 2 overlap and are disjoint with the view of camera 3. Trajectories 1 and 2 observed by cameras 1 and 2 correspond to the same object moving across camera views. Their temporal extents overlap as shown in (b), so they are connected by an edge in the network as shown in (d). Trajectories 3 and 4 observed in cameras 1 and 3 correspond to an object crossing disjoint views. Their temporal extents have no overlap but the gap is smaller than T as shown in (c), so they are also connected. Trajectories 3 and 6, 5 and 7 do not correspond to the same objects, but their temporal extents are close, so they are also connected in the network. A single trajectory 3 can be connected to two trajectories (4 and 6) in other cameras.

5. Generative Model

In this section, we will describe our generative model which clusters trajectories in different camera views into activities. Our work is related to topic models, such as LDA [1], which was used for word-document analysis. These topic models assume that a document is a mixture of topics and cluster words that often co-occur in the same documents into one topic. In our domain, documents are trajectories, words are observations, and topics are activities. Each activity has a distribution over locations and moving directions in different camera views, and models a path commonly taken by objects. If two word values, which are indices of locations and moving directions, often co-occur on the same trajectories, they are on the same path. Trajectories passing through the same paths belong to the same

activities. In previous topic models, documents are generated independently. However, we assume that if two trajectories in different camera views are connected by an edge on the network, which means that they may correspond to the same object, they tend to have a similar distribution over activities. Thus the distributions of an activity (the path of objects) in different camera views can be jointly modeled.

Let M be the number of trajectories in the data set. Each trajectory j has N_j observations. Each observation i on trajectory j has a visual word value w_{ji} which is a index of the global codebook. Observations will be clustered to one of the K activity categories. Let z_{ji} be the activity label of observation i in trajectory j. Each activity k has a multinomial distribution ϕ_k over the global codebook, which is a concatenation of codebooks of multiple camera views. So an activity is modeled as distributions over space and moving directions in multiple camera views. Each trajectory has a random variable θ_j which is the parameter of a multinomial distribution over K activities.

The joint distribution of these variables is given by

$$
\begin{aligned}
&p(\{\phi_k\},\{\theta_j\},\{z_{ji}\},\{w_{ji}\}|\alpha,\beta,\gamma)\\
=&p(\{\theta\},\{z_{ji}\}|\alpha,\gamma)p(\{\phi_k\}|\beta)p(\{w_{ji}\}|\{z_{ji}\},\{\phi_k\})\\
\propto&\prod_{j=1}^{M}\prod_{k=1}^{K}(\theta_{jk})^{\alpha-1}\prod_{\{j_1,j_2\}\in E}\prod_{k=1}^{K}(\theta_{j_1k})^{\gamma\cdot n_{j_2k}}(\theta_{j_2k})^{\gamma\cdot n_{j_1k}}\\
&\prod_{k=1}^{K}\mathrm{Dir}(\phi_k;\beta)\prod_{j=1}^{M}\prod_{i=1}^{N_j}\left(\theta_{jz_{ji}}\cdot\phi_{z_{ji}w_{ji}}\right) \qquad (2)\\
=&\prod_{j=1}^{M}\left[\frac{\prod_{k=1}^{K}\Gamma(\alpha+\gamma\sum_{j'\in\Omega_j}n_{j'k})}{\Gamma(K\cdot\alpha+\gamma\sum_{j'\in\Omega_j}\sum_{k=1}^{K}n_{j'k})}\right.\\
&\left.\mathrm{Dir}(\theta_j;\alpha+\gamma\sum_{j'\in\Omega_j}n_{j'1},\ldots,\alpha+\gamma\sum_{j'\in\Omega_j}n_{j'K})\right]\\
&\prod_{k=1}^{K}\mathrm{Dir}(\phi_k;\beta)\prod_{j=1}^{M}\prod_{i=1}^{N_j}\left(\theta_{jz_{ji}}\cdot\phi_{z_{ji}w_{ji}}\right) \qquad (3)
\end{aligned}
$$

$\mathrm{Dir}(\cdot;\cdot)$ is a Dirichlet distribution. If two trajectories are connected by an edge on the network, they are neighbors. E is the set of pairs of neighboring trajectories. Ω_j is the set of trajectories that are neighbors of j. In this generative model, observation i in trajectory j samples its activity label z_{ji} from a discrete distribution parameterized by θ_j of trajectory j. Then it samples its word value w_{ji} from a discrete distribution specified by the parameter $\phi_{z_{ji}}$ of activity z_{ji}. ϕ_k is sampled from a Dirichlet prior $\mathrm{Dir}(\cdot;\beta)$ with a flat hyperparameter β.

The first term of Eq 3 adds a smoothness constraint to θ_j through a Dirichlet distribution. Let $n_{j'k}$ be the number of observations assigned to activity k on trajectory j'. Then $(\sum_{j'\in\Omega_j}n_{j'1},\ldots,\sum_{j'\in\Omega_j}n_{j'K})$ is the histogram of observations assigned to K activity categories on the neighboring trajectories of j. It is used as the Dirichlet parameter for θ_j, after being weighted by a positive scalar γ and added to a flat prior α. Let $\rho_k=\alpha+\gamma\cdot\sum_{j'\in\Omega_j}n_{j'k}$. According to the properties of the Dirichlet distribution, if $\theta_j\sim\mathrm{Dir}(\rho_1,\ldots,\rho_K)$, the expectation of θ_j is $(\rho_1/\sum\rho_k,\ldots,\rho_K/\sum\rho_k)$ and its variation is small if $\sum\rho_k$ is large. Notice that z_{ji} is sampled from θ_j and θ_j has a constraint added by $z_{j'i'}$ on its neighboring trajectories. So trajectory j tends to have a similar distribution over activities as its neighboring trajectories, which means that they are smooth. A large γ puts a stronger constraint on the smoothness. If two trajectories are connected by an edge in the network, they are more likely to correspond to the same object. So trajectories of the same object tend to have similar distributions over activities.

5.1. Inference

We do inference by Gibbs sampling. It turns out that $\{\theta_j\}$ and $\{\phi_k\}$ can be integrated out during the Gibbs sampling procedure.

$$
\begin{aligned}
&p(\{z_{ji}\},\{w_{ji}\}|\alpha,\beta,\gamma)\\
=&\int_{\{\phi_k\},\{\phi_j\}}\int_{\{\theta_k\}}p(\{\theta_j\},\{z_{ji}\},\{w_{ji}\}|\alpha,\beta,\gamma)d\{\theta_k\}d\{\phi_k\},\{\phi_j\}\\
\propto&\int_{\{\phi_j\}}\int_{\{\theta_k\}}\prod_{kw}(\phi_{kw})^{\beta+m_{kw}-1}\,d\{\theta_k\}d\{\phi_j\}\\
&\prod_j\prod_k(\theta_{jk})^{\alpha+n_{jk}+\gamma\cdot\sum_{j'\in\Omega_j}n_{j'k}-1}\\
=&\prod_k\frac{\prod_w\Gamma(\beta+m_{kw})}{\Gamma(W\cdot\beta+m_{k\cdot})}\\
&\prod_j\frac{\prod_k\Gamma\left(\alpha+n_{jk}+\gamma\cdot\sum_{j'\in\Omega_j}n_{j'k}\right)}{\Gamma\left(K\cdot\alpha+n_{j\cdot}+\gamma\cdot\sum_{j'\in\Omega_j}n_{j'\cdot}\right)}, \qquad (4)
\end{aligned}
$$

where $\Gamma(\cdot)$ is the Gamma function, W is the size of the global codebook, m_{kw} is the number of observations assigned to activity k with value w, $m_{k\cdot}$ is the total number of observations assigned to activity k, n_{jk} is the number of observations assigned to activity k on trajectory j, and $n_{j\cdot}$ is the total number of observations on trajectory j. Then the conditional distribution of z_{ji} given all the other activity labels $\mathbf{z}^{-ji}$ is

$$
\begin{aligned}
&p(z_{ji}=k|\mathbf{z}^{-ji},\{w_{ji}\},\alpha,\beta,\gamma)\\
\propto&\frac{\beta+m^{-ji}_{k,w_{ji}}}{W\cdot\beta+m^{-ji}_{k,\cdot}}\cdot\frac{\alpha+n^{-ji}_{jk}+\gamma\sum_{j'\in\Omega_j}n_{j'k}}{K\cdot\alpha+n^{-ji}_{j\cdot}+\gamma\sum_{j'\in\Omega_j}n_{j'\cdot}}, \qquad (5)
\end{aligned}
$$

where $m^{-ji}_{kw_{ji}}$, $m^{-ji}_{k\cdot}$, n^{-ji}_{jk}, and $n^{-ji}_{j\cdot}$ are the same statistics as $m_{kw_{ji}}$, $m_{k\cdot}$, n_{jk}, and $n_{j\cdot}$ except that they have excluded observation i on trajectory j. To have a large posterior in

Eq 5, the first term requires that the value of observation i should fit the model of activity k, and the second term requires that its activity label is consistent with those of observations on the same trajectory and neighboring trajectories, with γ controlling the weight of neighboring trajectories. The models of activities are not explicitly learnt during the Gibbs sampling procedure, but they can be estimated from any single sample of $\{z_{ji}\}$,

$$\widehat{\phi}_{kw} = \frac{\beta + m_{kw}}{W \cdot \beta + m_{k\cdot}} \tag{6}$$

A trajectory is labeled as activity k, if most of its observations are assigned to k. The activity label of an observation can be obtained during the Gibbs sampling procedure based on Eq. 5. However, there may be an over smoothing effect, since in some cases most of the trajectories being the neighbors of trajectory j do not correspond the same object as j. In this work, we adopt an alternative labeling approach which actually achieves better performance in experiments. As shown by the experimental results in Section 6, the activity models learnt from Gibbs sampling are distinctive enough to label trajectories. After the activity models have been learnt and fixed at the of Gibbs sampling which uses Eq. 5 and 6, we ignore the smoothness constraint among trajectories and label the observation as $z_{ji} = \arg\max_k \hat{\phi}_{kw_{ji}}$.

6. Experimental Results

We evaluate our approach on two data sets, a parking lot scene and a street scene. There are tracking errors in both of the two data sets. For example, a track may break into fragments because of occlusions. As observed from experiments, our algorithm is robust to tracking errors.

6.1. Parking Lot Scene

The parking lot data set has $22,951$ trajectories, collected from 10 hours during the day time over 3 days. Inspection shows that it is a fairly busy scene. The topology of its four cameras is shown in Figure 1 (a). The view of camera 1 has no overlap with other camera views. However, the gap between views of cameras 1 and 2 is small. The views of cameras 2 and 3 have small overlap. The views of cameras 3 and 4 have large overlap. Our approach does not require knowledge of the topology of the cameras. Fourteen different activities are learnt from this data set. Because of space limitations, only six activities are shown in Figure 3. For each activity, we plot its distribution over space and moving directions in the four cameras and the trajectories clustered into this activity. When visualizing activity models, moving directions are represented by different colors, and the density of distributions over space and moving directions is proportion to the brightness of colors. When plotting trajectories, random colors are used to distinguish individual trajectories.

In Figure 3, activity 1 is vehicles and pedestrians entering the parking lot. It has a large extent in space and is observed by all of the four cameras. In activities 3 and 4, pedestrians are walking in the same direction but on different paths. From the distributions of their models, it is observed that the two paths are side by side but well separated in space. The path of activity 5 occupies almost the same region as that of activity 4. However, pedestrians are moving in opposite directions in these two activities, so the distributions of their models are plotted in different colors.

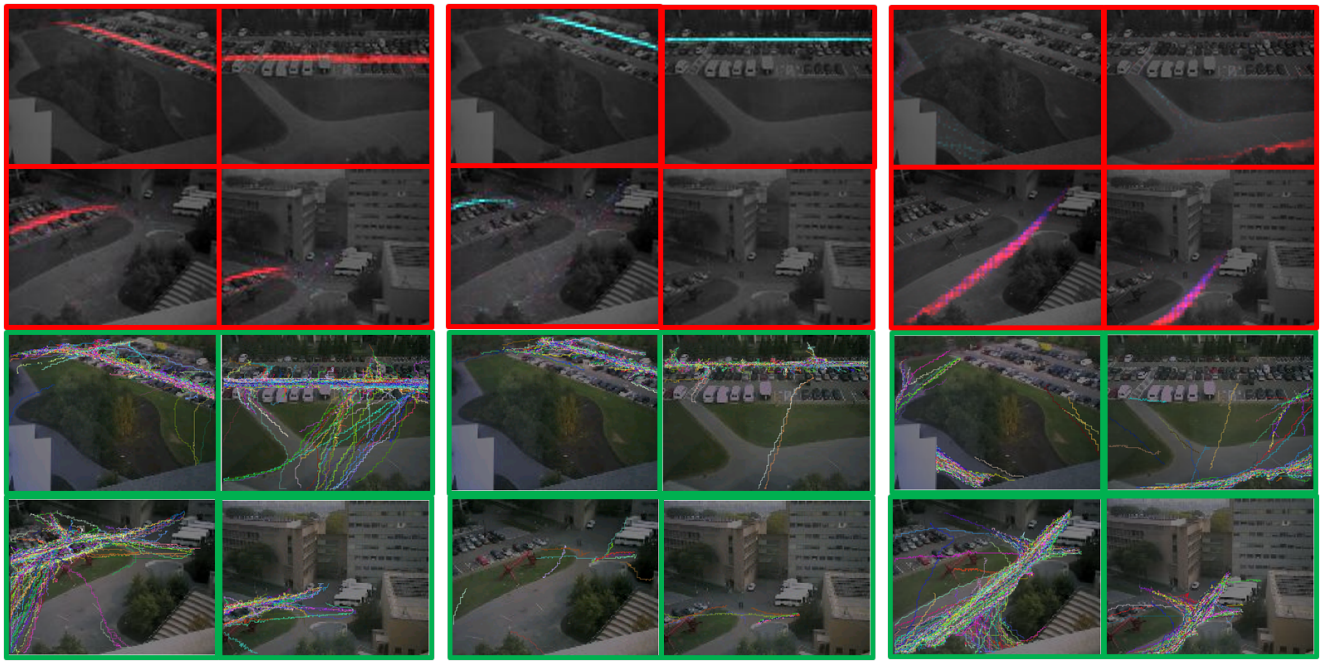
6.2. Street Scene

The topology of the four camera views of the street scene is shown in Figure 1 (b). Camera 1 has a distant view of the street. Camera 2 zooms in on the top-right part in the view of camera 1. The view of camera 3 has overlap with the views of cameras 1 and 2. It extends the top-right part of the view in camera 1 along the street. The view of camera 4 partially overlaps with the bottom region of the view in camera 1. There are $14,985$ trajectories in this data set, collected from 30 hours during day time in four days. Seventeen activities are learnt in this scene. Again, we only show the results of 6 activities in Figure 4. Activity 1 is vehicles moving on the road. It is observed by all four cameras. Vehicles first move from the top-right corner to the bottom-left corner of the view in camera 4. Then they enter the bottom region of the view in camera 1 and move upward. Some vehicles disappear at the exit points observed in the views of cameras 2 and 3, and some move further beyond the view of camera 3. In activities 2, 4 and 5, pedestrians first walk along the sidewalk in the view of camera 1, and then cross the street as observed by camera 4. The paths of activities 2 and 5 occupy similar regions in the view of camera 1, but their paths diverge in the view of camera 4.

As shown in Figure 3 and 4, the models of activities reveal some structures, such as paths commonly taken by objects, and entrance and exit points in the scene. Some paths are less related to the appearance of the scene. For example, some paths cross the street outside the crosswalk in the street scene. Usually paths have spatial extents in multiple cameras, which we call semantic regions. Semantic regions across cameras can be detected by simply thresholding the density of the distributions of activities (ϕ_k in Eq 3).

6.3. Perplexity

Perplexity is a measure commonly used to evaluate the performance of clustering algorithms. It is the number of bits required to encode the data and is proportional to the negative log likelihood of the data. It measures how unseen testing data fits the model learnt from training data. Two hundred randomly sampled trajectories from each camera



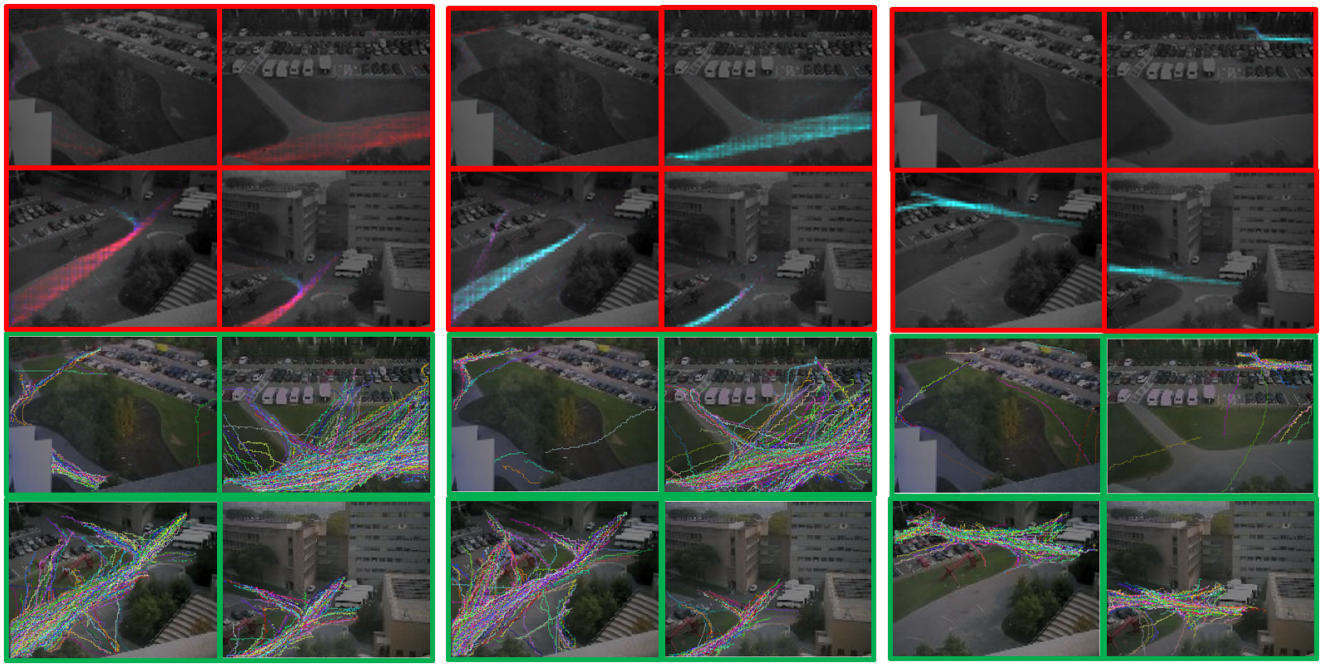
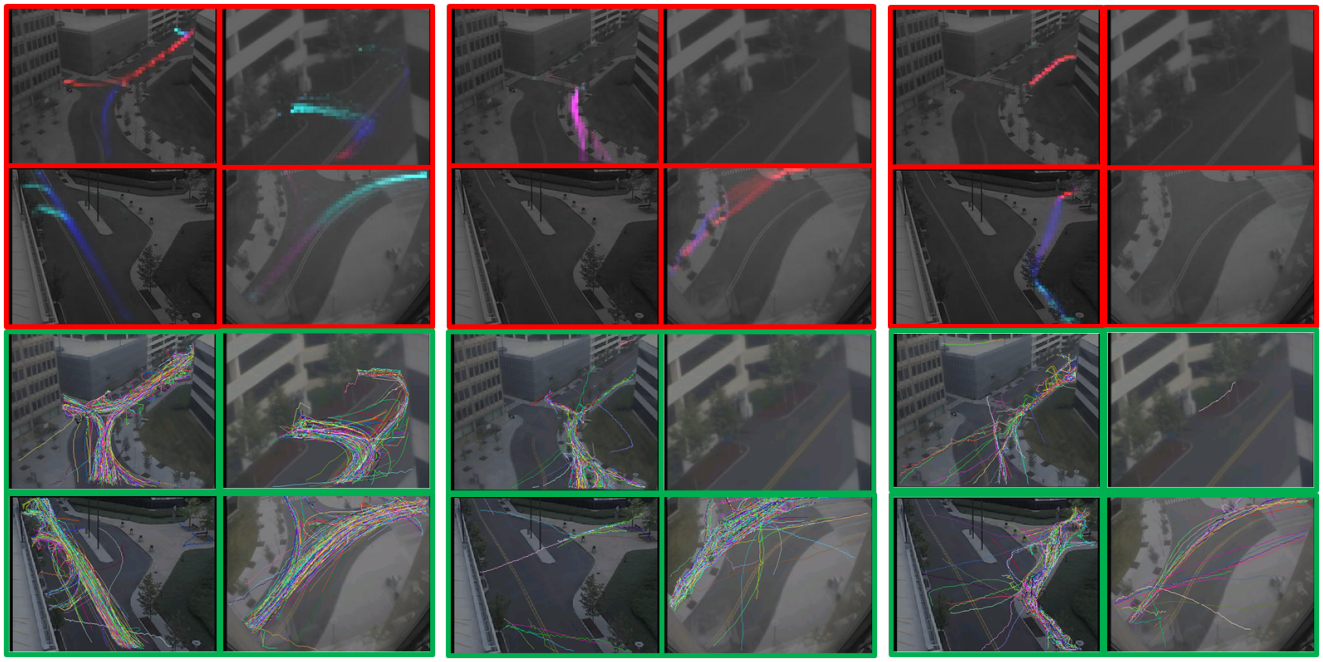


Figure 3. Distributions of activity models and clusters of trajectories of the parking lot scene. When plotting the distributions of activity models (in the four red windows on the top), different colors are used represent different moving directions: $\rightarrow$ (red), $\leftarrow$ (cyan), $\uparrow$ (blue), $\downarrow$ (magenta). When plotting trajectories clustered into different activities (in the four green windows at the bottom), random colors are used to distinguish individual trajectories.

serve as the test set; the remaining trajectories are used for training. To compare models with different trajectory networks, the activity models $\{\phi_k\}$ are learnt with the smoothness constraint added by the trajectory network. Once $\{\phi_k\}$ are learnt and fixed, the perplexity is computed on the test data ignoring the smoothness constraint.



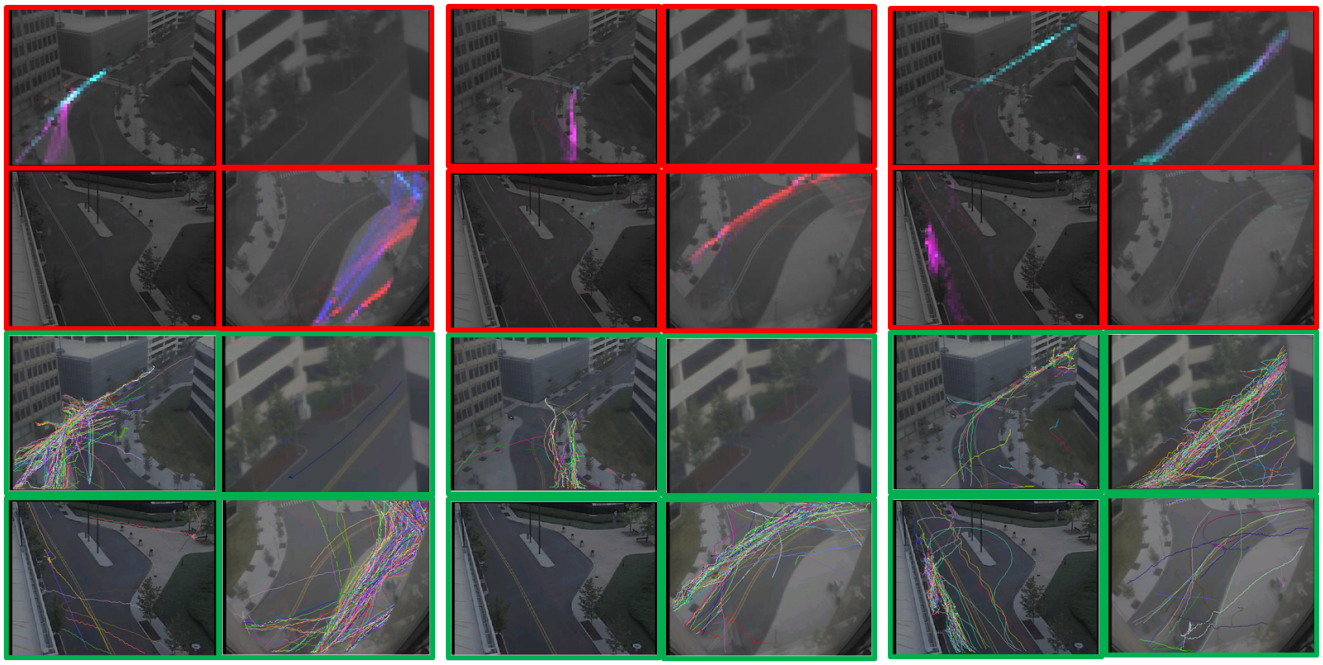


Figure 4. Distributions of activity models and clusters of trajectories of the street scene. The meaning of colors is the same as Figure 3. See text for details. Because of space limitations only six activities are shown.

First, we compare our approach with two alternatives: (1) unconnected network; (2) network with random correspondences[1]. The former completely abandons the smoothing constraint, so it cannot jointly model the distributions of a single activity in multiple camera views. The latter simulates the case when correspondence is poor. Both alternatives result in higher perplexity as shown in Table 1.

[1]First find correspondence candidates using Eq 1. Instead of fully connecting these candidates as in our model, a trajectory is randomly connected with only one of the candidates in a different camera view.

	Our approach	Unconnected	Random
Parking Lot	130.3	200.3	176.8
Street	85.7	228.8	135.2

Table 1. Perplexity under our approach and two alternative trajectory networks.

	1	2	3	4	Random
Parking Lot	120.9	121.3	122.8	123.3	425
Street	40.0	41.5	44.9	42.2	168

Table 2. Perplexity with models trained on a variable number of cameras. The test data is 200 trajectories from a single camera. The activity models in that camera are jointly learnt with different number of cameras (from 1 to 4). The last column is a baseline model trained on randomly assigned data.

We also compare against models learned with trajectories from a single to all of the cameras. Models learned from a subset of the cameras will necessarily have lower perplexity for trajectories within those cameras; however, they are limited to modeling joint activities only in a subset of the cameras. Our model captures joint activities in all cameras simultaneously, and only exhibits a small increase in perplexity as shown in Table 2.

6.4. Temporal Threshold

The temporal threshold T in Eq 1 determines the connectivity in the trajectory network. If a camera view A is disjoint from other views and it takes objects more than T seconds to cross the smallest gap between A and other views, then there is no way to extend the path in A to other views. If T is large and the scene is busy, there will be too many connected trajectories in the network even though they do not correspond to the same activities. Under-smoothing could lead to the same activity separated into different clusters, while over-smoothing could lead to different activities joined into the same cluster. Empirically, we achieved similar results with a wide range of values for T: for the street scene data set, good results are achieved when T varies between 0 and 30 seconds; for the parking lot data set, the range of good values of T is roughly from 3 to 15 seconds because the parking lot scene is busier and the view of camera 1 is disjoint from other camera views.

7. Conlusion

We propose a framework to model activities and cluster trajectories over a multi-camera network. It is unsupervised and does not require first solving the challenging multi-camera correspondence problem. Experiments on two data sets including a very large number of trajectories explain the effectiveness of this approach.

8. Acknowledgment

The authors wish to acknowledge DSO National Laboratories (Singapore) for partially supporting this research.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] J. Fernyhough, A. Cohn, and D. Hogg. Generation of semantic regions from image sequences. In *Proc. of ECCV*, 1996.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[4] N. Gheissari, T. B. Sebastian, J. Rittscher, and R. Hartley. Person reidentification using spatiotemporal appearance. In *Proc. of CVPR*, 2006.

[5] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameraswith disjoint views. In *Proc. of ICCV*, 2003.

[6] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *Proc. of CVPR*, 2005.

[7] I. Junejo and H. Foroosh. Trajectory rectification and path modeling for video surveillance. In *Proc. of ICCV*, 2007.

[8] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Trans. on PAMI*, 22:758–768, 2000.

[9] D. Makris and T. Ellis. Path detection in video surveillance. *Image Vision and Computation*, 20:859–903, 2002.

[10] D. Makris and T. Ellis. Automatic learning of an activity-based semantic scene model. In *Proc. of IEEE Conf. on Advanced Video and Signal Based Surveillance*, 2003.

[11] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Proc. of CVPR*, 2004.

[12] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Proc. of CVPR*, 2004.

[13] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. In *IEEE Trans. on PAMI*, 2000.

[14] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *Proc. of CVPR*, 2003.

[15] K. Tieu, G. Dalley, and E. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *Proc. of ICCV*, 2005.

[16] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. In *Proc. of ECCV*, 2006.

[17] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proc. of ICPR*, 2006.