Discovery of Shared Semantic Spaces for Multiscene Video Query and Summarization

Xun Xu, Timothy M. Hospedales, and Shaogang Gong

***Abstract*— The growing rate of public space closed-circuit television (CCTV) installations has generated a need for automated methods for exploiting video surveillance data, including scene understanding, query, behavior annotation, and summarization. For this reason, extensive research has been performed on surveillance scene understanding and analysis. However, most studies have considered single scenes or groups of adjacent scenes. The semantic similarity between different but related scenes (e.g., many different traffic scenes of a similar layout) is not generally exploited to improve any automated surveillance tasks and reduce manual effort. Exploiting commonality and sharing any supervised annotations between different scenes is, however, challenging due to the following reason: some scenes are totally unrelated and thus any information sharing between them would be detrimental, whereas others may share only a subset of common activities and thus information sharing is only useful if it is selective. Moreover, semantically similar activities that should be modeled together and shared across scenes may have quite different pixel-level appearances in each scene. To address these issues, we develop a new framework for distributed multiple-scene global understanding that clusters surveillance scenes by their ability to explain each other's behaviors and further discovers which subset of activities are shared versus scene specific within each cluster. We show how to use this structured representation of multiple scenes to improve common surveillance tasks, including scene activity understanding, cross-scene query-by-example, behavior classification with reduced supervised labeling requirements, and video summarization. In each case, we demonstrate how our multiscene model improves on a collection of standard single-scene models and a flat model of all scenes.**

***Index Terms*— Scene understanding, transfer learning, video summarization, visual surveillance.**

I. Introduction

THE widespread use of public space closed-circuit television (CCTV) camera systems has generated unprecedented amounts of data that can easily overwhelm human operators due to the sheer length of the surveillance videos and the large number of surveillance videos captured at different locations concurrently. This has motivated numerous studies into automated means to model, understand, and exploit these data. Some of the key tasks addressed by automated surveillance video understanding are as follows:

Manuscript received August 25, 2014; revised March 28, 2015 and May 26, 2015; accepted July 27, 2015. Date of publication February 29, 2016; date of current version June 5, 2017. This paper was recommended by Associate Editor V. Pavlovic.

The authors are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: xun.xu@qmul.ac.uk; t.hospedales@qmul.ac.uk; s.gong@qmul.ac.uk).

This paper has supplementary downloadable material provided by the author available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2016.2532719

Fig. 1. Example of a multicamera surveillance network with camera views distributed across different locations.

1) behavior profiling/scene understanding to reveal what are the typical activities and behaviors in the surveilled space [1]–[5];
2) behavior query-by-example, allowing the operator to search for similar occurrences to a specified example behavior [1];
3) supervised learning to classify/annotate activities or behaviors if events of interest are annotated in a training data set [2];
4) summarization to give an operator a semantic overview of a long video in a short period of time [6];
5) anomaly detection to highlight to an operator the most unusual events in a recording period [1]–[3].

So far, all of these tasks have generally been addressed within a single scene (single video captured by a static camera) or a group of adjacent scenes.

Compared with single-scene recordings, the multicamera surveillance network (cameras distributed over different locations) is a more realistic scenario in surveillance applications and thus of more interest to end users. An example of a multicamera surveillance network is given in Fig. 1, where surveillance videos mostly capture traffic scenes with various layouts and motion patterns. In such a multiscene context, new surveillance tasks arise. For behavior profiling/scene understanding, human operators would like to see which scenes within the network are semantically similar to each other (e.g., similar scene layout and motion patterns), which activities are in common, and which are unique across a group of scenes, and how activities group into behaviors. Here, activity refers to a spatiotemporally compact motion pattern due to the action of a single or small group of objects (e.g., vehicles making a turn) and behavior refers to the

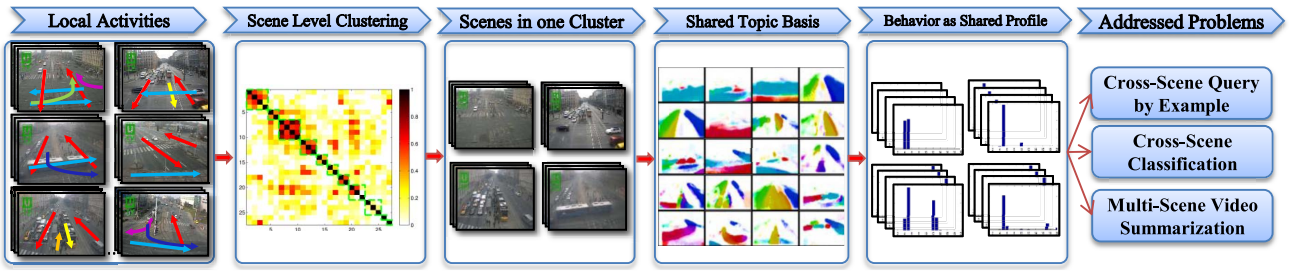


Fig. 2. Illustration of the proposed framework.

interaction between multiple activities within a short temporal segment (e.g., horizontal traffic flow with vehicles going east and west and making a turn). For query-by-example, searching for a specified example behavior should be carried out not only within a scene but also across multiple scenes. For behavior classification, annotating training examples in every scene exhaustively is not scalable. However, multiscene modeling potentially addresses this by allowing labels to be propagated from one scene to another. For summarization, generating a summary video for multiple scenes by exploiting cross-scene redundancy can provide the user who monitors a set of cameras with an overview of all the distinctive behaviors that have occurred in a set of scenes. Multiscene summarization can reduce the summary length and achieve higher compression than single-scene summarization. Combined with query-by-example (find more instances of a behavior in a summary), a flexible exploration of scenes at multiple scales is available.

Despite the clear potential benefits of exploiting multiscene surveillance, it cannot be achieved with existing single-scene models [1]–[5]. These approaches learn an independent model for each scene and do not discover corresponding activities or behaviors across scenes, even if they share the same semantic meaning. This makes any cross-scene reasoning about activities or behaviors impossible. In order to synergistically exploit multiple scenes in surveillance, a multiscene model with the following capabilities is required: 1) learning an activity representation that can be shared across scenes; 2) model behaviors with the shared representation so that they are comparable across scenes; and 3) generalizing surveillance tasks to the multiscene case, including behavior profiling/scene understanding, cross-scene query-by-example, cross-scene classification, and multiscene summarization. However, this is intrinsically challenging for three reasons.

1) *Computing Scene Relatedness:* Determining the relatedness of scenes is critical for multiscene modeling because naive information sharing between insufficiently related scenes can easily result in negative transfer [7], [8]. However, the relatedness of scenes is hard to estimate because the appearance of elements in a scene (e.g., buildings and road surface markings) is visually diverse and strongly affected by camera view, making appearance-based similarity measurement unreliable. Similarity measurement based on motion is less prone to visual noise in surveillance applications. However, most studies focus only on discovering the similarity in activity level [8], [9]. Thus, how to measure scene-level relatedness is still an open question.
2) *Selective Sharing of Information:* Large multicamera surveillance networks cover various types of scenes. Some scenes are totally unrelated, which means they convey different semantic meanings to a human. However, more subtly, even between similar scenes, there may be some activities in common and other activities that are unique to each. Learning a large universal model in this situation is prone to overfitting due to the high model complexity. Hence, a model that discovers (un)relatedness of scenes and selectively shares activities between them is necessary.
3) *Constructing a Shared Representation:* Within related scenes, a shared representation needs to be discovered in order to exploit their similarity for cross-scene query-by-example and multiscene summarization. Both common and unique activities should be preserved in this process to ensure the ability of discovering not only the commonality but also the distinctiveness between scenes.

To address these challenges, we develop a new framework illustrated in Fig. 2. We first learn local representations for each scene separately. Then related scenes are discovered by clustering. A shared semantic representation is constructed to represent activities and behaviors within each group of related scenes. Specifically, we first represent each scene with a low-dimensional semantic (rather than pixel-level) representation through learning a fast unsupervised topic model for each.[1] Using a topic-based representation allows us to reduce the impact of pixel noise in discovering activity and scene similarity. We next group *semantically* related scenes into a scene cluster by exploiting the correspondence of activities between different scenes. Finally, scenes within each cluster are projected to a shared representational space by not only computing a shared activity topic basis (STB), shared among all scenes, but also allowing each scene to have unique topics if supported by the data. Behaviors in each scene are represented with the learned STB.

In addition to profiling, for revealing the multiscene activity structure across all scenes, we use this structured representation to support cross-scene query, label propagation for classification, and multiscene summarization. Cross-scene query-by-example is enabled because within each cluster, the semantic representation is shared, so an example in one scene can retrieve related examples in every other scene in the cluster. Behavior classification/annotation in a

[1]Topics have previously been shown to robustly reveal semantic activities from cluttered scenes [1], [2].

new scene *without annotations* is supported because, once associated with a scene cluster, it can borrow the label space and classifier from that cluster. Finally, we define a novel multiscene approach to summarization that jointly exploits the shared representation to compress redundancy both within and across scenes of each cluster.

II. Related Work

A. Surveillance Scene Understanding

Scene understanding is a wide area that is too broad to review here. However, some relevant studies to this work include those based on object tracking [3], [10], [11], [12], which model behaviors, for example, by a hidden Markov model [3], [10], a Gaussian process [12], clustering [13], and stochastic context-free grammars [14], and those based on low-level feature statistics, such as optical flow [1], [2], [5], [15], that often model behaviors by a probabilistic topic model (PTM) [1], [2], [4]. The latter category of approaches are the most related to ours, as we also built on PTMs. However, all of these studies operate within scene rather than modeling globally distributed scenes and discovering shared activities.

B. Multiscene Understanding

We make an explicit distinction to another line of work that discovers connections and correlations among multiple overlapping or nonoverlapping scenes connected by a single camera network covering small areas [16], [17]. This is orthogonal to our area of interest, which is more similar to multitask learning [7]—how to share information among multiple scenes, some of which have semantic similarities but do not necessarily concurrently surveil topologically connected zones.

Fewer approaches have tried to exploit relatedness between scenes without a topological relationship [8], [9]. To recognize the same activity from another viewpoint, Khokhar *et al.* [9] proposed a geometric-transformation-based method to align two events, represented as Gaussian mixtures, before computing their similarity. Xu *et al.* [8] used a trajectory-based event description and learned motion models from trajectories observed in a source domain. This model was then used for cross-domain classification and anomaly detection.

In the context of static image (rather than dynamic scene) understanding [18], [19], studies have clustered images by appearance similarity. However, this does not apply directly to surveillance scenes because the background is no longer stationary nor uniform, e.g., building and road appearance are visually salient but can vary significantly between surveillance scenes at different locations. It is not reliable to relate surveillance scenes based on appearance—the important cue is activity instead.

C. Video Query and Annotation

Video query has always been an important issue in surveillance applications. A lot of work has been done on semantic retrieval [1], [20]. Hu *et al.* [20] used trajectories to learn an activity model and construct semantic indices for video databases. Wang *et al.* [1] represent video clips as topic profiles and measure the similarity between the query and candidate clips as relative entropy. Retrieved clips are sorted according to the distance to the query. However, none of these techniques take a multiscene scenario into consideration in which query examples are selected in one scene and candidate clips can be retrieved from other scenes at different locations.

Related to video query, video behavior annotation/classification has been addressed in the literature [1], also in terms of video segmentation [21]. However, these approaches are typically domain/scene specific, which means that *each scene* needs extensive annotation of training data whereby, ideally, labels should instead be borrowed from semantically related scenes. Although [9] recognized events across scenes at the activity level, scene-level behavior classification, dealing with a heterogeneous database of scenes is still an open problem.

D. Video Summarization

Video summarization has received much attention in the literature in recent years due to the need to digest large quantities of video for efficient review by users. A review can be found in [22]. There are a variety of approaches to summarization, varying both in how the summary is represented/composed, and how the task is formalized in terms of what type of redundancy should be compressed.

Summaries have been composed of *static keyframes*, which represent the summary as a collection of selected key frames [23], *dynamic skimming*, which composes a summary based on a collection of selected clips, and more recently *synopsis*. Synopsis [6], [24] temporally reorders (spatially nonoverlapping) activities from the original video into a temporally compact summary video by shifting activity tubes temporally so that they occur more densely. The objective of summarization can be formalized in various ways: to more abstractly achieve the highest rating in a user study [23], to show all foreground activities in the shortest time [24], or to minimize the reconstruction error between the summary and the original video, to show at least one example of every typical behavior.

As the number of scenes grows, multiview summarization becomes increasingly important to help operators monitor activities in numerous scenes. However, multiview summarization is much less studied compared with that of a single view. Lou *et al.* [25] adopted multiview video coding to deal with multiview video compression but did not tackle the more challenging compression of semantic redundancy. Fu *et al.* [26] addressed generating concise multiview video summaries by multiobjective optimization for generating representative summary clips. Recently, de Leo and Manjunath [27] proposed a multicamera video summarization framework that summarizes at the level of activity motif [28]. Due to the severe occlusion, far field of view, and high-density activities in surveillance videos, none of the existing techniques solve the problem of distributed multiscene surveillance video summarization.

In this paper, we pursue video summarization from the perspective of selecting the smallest set of representative video clips that still have good coverage of all the behaviors in the scene(s). Such *multiscene* summarization compresses redun-

dancy across as well as within scenes. This corresponds to an application scenario in which the user tasked with monitoring a set of cameras wants an overview of all the behaviors that occurred in a set of video streams during a recording period regardless the source of the video recordings, which typically come from different locations. This perspective on summarization is attractive because it makes sense of video content independent of location and local context. This offers a more holistic conceptual summarization in a global context compared with a summarization as visualization of a single scene in a local context such as video synopsis. Interestingly, combined with our query-by-example, we can take a behavior of interest shown in the summary as query to search for similar behaviors in other scenes. Thus, the framework presents both compact multiscene summarization and a finer scene-specific zoomed-in view, capable of compressing semantically equivalent examples no matter what scene they occur in.

E. Our Contributions

A system based on our framework can answer questions such as *show me which scenes are similar to this?* (scene clustering), *show me which activities are in common and which are distinct between these scenes?* (multiscene profiling) *show me all the distinct behaviors in this group of scenes?* (multiscene summarization), *show me other clips from any scene that are similar to this nominated example?* (cross-scene query), and *annotate this newly provided scene with no labels?* (cross-scene classification). Specifically, we make the following key contributions.

1) Introducing the novel and challenging problems of joint multiscene modeling and analysis.
2) Developing a framework to solve the proposed problem by discovering similarity between activities and scenes, clustering scenes based on semantic similarity, and learning a shared representation within scene clusters.
3) We show how to exploit this novel structured multiscene model for practical yet challenging tasks of cross-scene query-by-example and behavior annotation.
4) We further exploit this model to achieve multiscene video summarization, achieving compression beyond standard single-scene approaches.
5) We introduce a large multiscene surveillance data set containing 27 distinct views from distributed locations to encourage further investigation into realistic multiscene visual surveillance applications.

III. Learning Local Scene Activities

Given a set of surveillance scenes, we first learn local activities in each individual scene by using *latent Dirichlet allocation* (LDA) [29]. Although there are more sophisticated single-scene models [1], [2], [4], we use LDA because it is the simplest, most robust, most generally applicable to a wide variety of scene types, and the fastest for learning on large-scale multiscene data. However, it could easily be replaced by more elaborate topic models (e.g., HDP [1]). LDA generates a set of topics to explain each scene. Topics are usually spatially and temporally constrained subvolumes reflecting the activity of a single or small group of objects. Following [1] and [2], we use activities to refer to topics and behaviors to refer to the scene-level state defined by the coordinated activities of all scene participants.

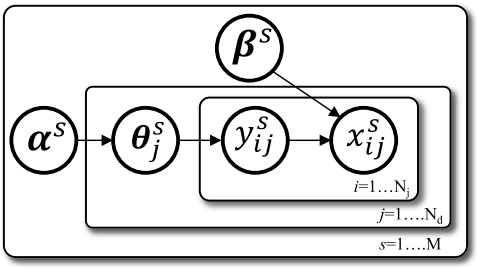


Fig. 3. Graphical model for LDA.

A. Video Clip Representation

We follow the general approach [1] to construct visual features for topic models. For each video out of an M scene data set, we first divide the video frame into $N_a \times N_b$ cells, with each cell covering $H \times H$ pixels. Within each cell, we compute optical flow [30], taking the mean flow as the motion vector in that cell. Then we quantize motion vector into N_m fixed directions. Note that stationary foreground objects can be readily added as another cell state, as described in [2] and [31]. Therefore, a codebook $\mathbf{V}$ of size $N_v = N_a \times N_b \times N_m$ is generated by mapping motion vectors to discrete visual words (from 1 to N_v). N_d visual documents $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^{N_d}$ are then constructed by segmenting the video into nonoverlapping clips of fixed length, where each clip $\mathbf{x}_j = \{x_{ij}\}_{i=1}^{N_j}$ has N_j visual words x_{ij}. The clip and the document are used interchangeably here, with both indicating visual words accumulated in a temporal segment.

B. Learning Local Activities With Topic Model

Learning LDA for scene s discovers the dynamic appearance of $k = 1 \ldots K$ typical topics/activities[2] (multinomial parameter $\boldsymbol{\beta}_k^s$) and explains each visual word x_{ij}^s in each clip $\mathbf{x}_j^s$ by a latent topic y_{ij}^s, specifying which activity generated it, as shown in Fig. 3. The topic selection y_{ij}^s is drawn from a multinomial mixture of topics parametrized by $\boldsymbol{\theta}_j^s$, which is further governed by a Dirchelet distribution with the parameter $\boldsymbol{\alpha}^s$. In scene s, the joint probability of N_d visual documents $\mathbf{X}^s = \{\mathbf{x}_j^s\}_{j=1}^{N_d}$, topic selection $\mathbf{Y}^s = \{\mathbf{y}_j^s\}_{j=1}^{N_d}$, and topic mixture $\boldsymbol{\theta}^s = \{\boldsymbol{\theta}_j^s\}_{j=1}^{N_d}$ for the given hyperparameters $\boldsymbol{\alpha}^s$ and $\boldsymbol{\beta}^s$ is

$$p(\boldsymbol{\theta}^s, \mathbf{Y}^s, \mathbf{X}^s \mid \boldsymbol{\alpha}^s, \boldsymbol{\beta}^s) = \prod_{j=1}^{N_d} p\left(\boldsymbol{\theta}_j^s \mid \boldsymbol{\alpha}^s\right) \cdot \prod_{i=1}^{N_j} p\left(y_{ij}^s \mid \boldsymbol{\theta}_j\right) p\left(x_{ij}^s \mid y_{ij}^s, \boldsymbol{\beta}^s\right). \quad (1)$$

Here we introduce an efficient way to infer the LDA model.

[2]In text analysis, a topic refers to a group of co-occurring words in a document. Activity refers to a motion pattern, which defines the group of co-occurring visual words in a video clip. They are used interchangeably in the following text.

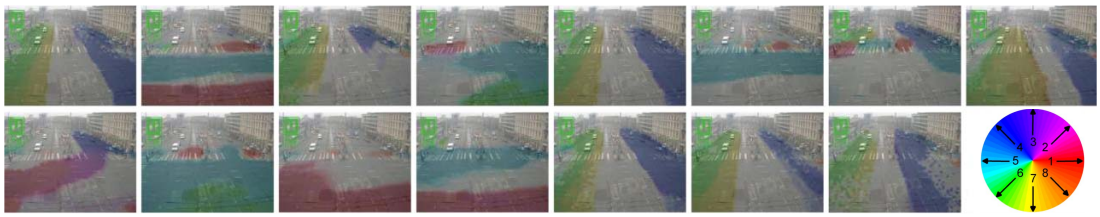


Fig. 4. Locally learned activities/topics in an example scene. The optical flow is quantized into $N_m = 8$ directions, as shown in the color wheel.

Algorithm 1 Topic Model Learning for a Single Scene

```
initialize α_k = 1
initialize β = random(N_v, K)
initialize φ_ijk = 1/K
repeat
  E-Step:
  for j = 1 → N_d do
    for k = 1 → K do
      γ_jk = α_k + Σ_{i=1}^{N_j} φ_ijk
      for i = 1 → N_j do
        φ_ijk = β_{x_ij k} exp(Ψ(γ_jk))
      end for
    end for
  end for
  M-Step:
  for v = 1 → N_v do
    for k = 1 → K do
      β_vk = Σ_{j=1}^{N_d} Σ_{i=1}^{N_j} φ_ijk 1(x_ij = v)
    end for
  end for
until Converge
```

1) Model Inference: Exact inference in LDA is intractable due to the coupling between $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ [29]. Variational inference approximates a lower bound of log likelihood by introducing variational parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$. The Dirichlet parameter $\boldsymbol{\gamma}_j$ is a clip-level topic profile and specifies the mixture ratio of each activity $\boldsymbol{\beta}_k$ in a clip $\mathbf{x}_j$. Thus, each video clip is represented as a mixture of activities ($\boldsymbol{\gamma}_j$). The variational Expectation-Maximization (EM) procedure for LDA is given in Algorithm 1 where $1(\cdot)$ is an indicator function and $\Psi(\cdot)$ is the first derivative of the $\log\Gamma$ function. For efficiency, we apply the sparse updates identified in [32] for an order of magnitude of speed increase.

After learning all $s = 1 \ldots M$ scenes, every clip $\mathbf{x}_j^s$ is now represented as a topic profile $\boldsymbol{\gamma}_j^s$ and each scene is now represented by its constituent activities $\boldsymbol{\beta}_k^s$ (Fig. 4).

IV. Multilayer Activity and Scene Clustering

We next address how to discover related scenes and learn shared topics/activities across scenes. This multilayer process is illustrated in Fig. 5 for two typical Clusters 3 and 7: at the scene level, we group related scenes according to activity correspondence (Section IV-A); within each scene cluster, we further compute an STB so that all activities within that cluster are expressed in terms of the same set of topics (Section IV-B).

A. Scene-Level Clustering

In order to group related scenes, we first need to define a relatedness metric. Related scenes should have more common activities so that the model learned from them is compact. Therefore, we assume that the scenes with semantically similar activities are more likely to be mutually related. We thus define the relatedness between two (aligned) scenes a and b by the correspondence of their semantic activities.

1) Alignment: Comparing scenes directly suffers from cross-scene variance due to the view angle. To reduce this cross-scene variance, we first align two scenes with a geometrical transformation, including scaling t_s and translation $[t_x, t_y]$. Although this is not a strong transform, it is valid in the typical case that a camera is installed upright, and with surveillance cameras, there are classic views that can be simply aligned by scaling and translation. To achieve this, we first denote the transform matrix for normalizing visual words in each scene a and b to the origin by $\mathbf{T}^a_{\text{norm}}$ and $\mathbf{T}^b_{\text{norm}}$, respectively, defined as (2). Scaling (t_s^a) and translation (t_x, t_y) parameters are estimated by (3)

$$\mathbf{T}^a_{\text{norm}} = \begin{bmatrix} t_s^a & 0 & t_x^a \\ 0 & t_s^a & t_y^a \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\text{center} = \frac{1}{N_d \cdot N_j} \sum_{j=1}^{N_d} \sum_{i=1}^{N_j} x_{ij}^a$$

$$t_s^a = \frac{N_d \cdot N_j}{\sum_{j=1}^{N_d} \sum_{i=1}^{N_j} \left\| x_{ij}^a - \text{center} \right\|_2}$$

$$\begin{bmatrix} t_x^a \\ t_y^a \end{bmatrix} = -t_s^a \cdot \text{center}. \tag{3}$$

Two scenes can thus be aligned by transforming data from a to b via $\mathbf{T}^{a2b} = \mathbf{T}^{b-1}_{\text{norm}} \cdot \mathbf{T}^a_{\text{norm}}$. We then denote the kth topic in scene a by $\boldsymbol{\beta}_k^a$. Therefore, any topic k in a can be aligned for comparison with those in b by $\mathbf{T}^{a2b}$.

We denote the topic transformation procedure by $\boldsymbol{\beta}' = \mathbb{H}(\boldsymbol{\beta}; \mathbf{T})$. This transformation is applied to topics in a way similar to the image transform. That is, given that $\boldsymbol{\beta}$ is a $N_a \times N_b \times N_m$ matrix and a transform matrix $\mathbf{T}$ is defined as (2), we first estimate the size $N'_a \times N'_b \times N_m$ of the transformed topic $\boldsymbol{\beta}'$ by $N'_a = N_a \times t_s$ and $N'_b = N_b \times t_s$. To obtain the value of each element/pixel of $\boldsymbol{\beta}'(x', y', d')$, we trace back to the position $[x, y, d]$ in the original topic $\boldsymbol{\beta}$. If we only consider scaling and translation, direction d is then unchanged throughout the procedure, i.e., $d' = d$. Therefore, x and y are determined by

$$[x \quad y \quad 1] = [x' \quad y' \quad 1] \cdot (\mathbf{T}^{-1})^T. \tag{4}$$

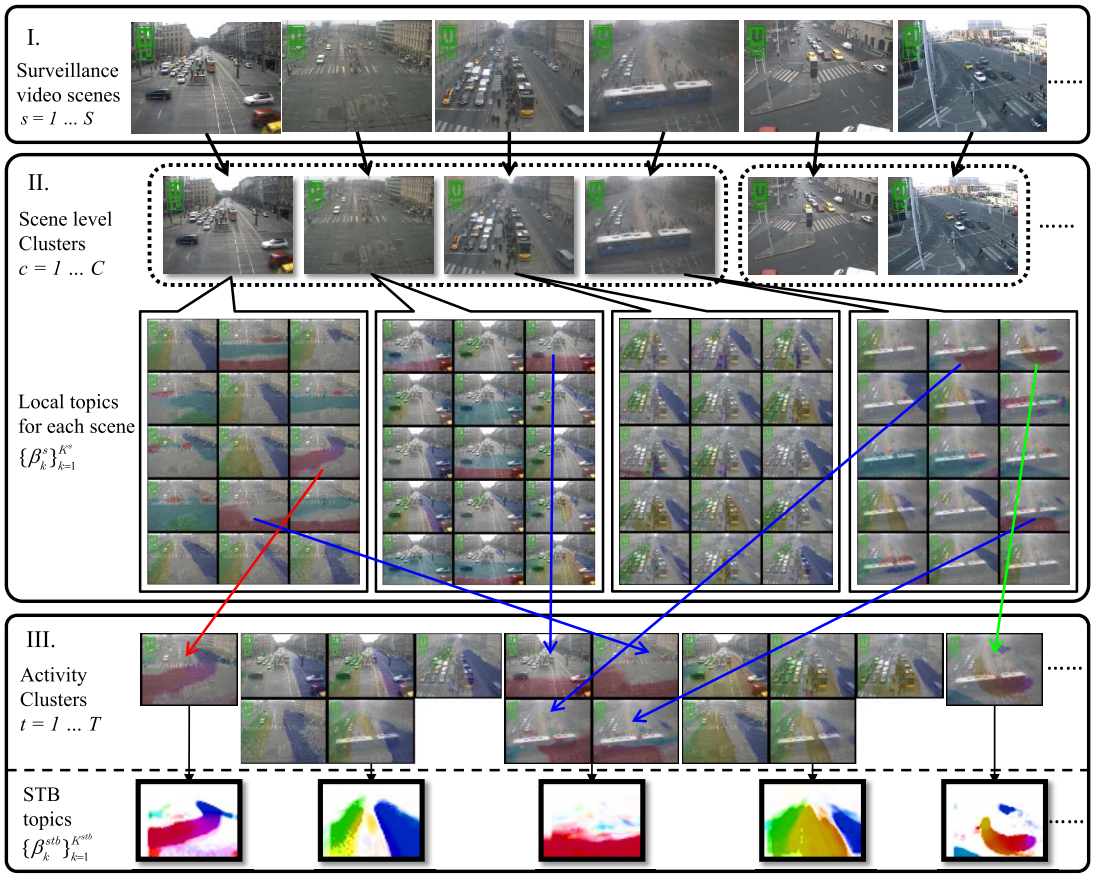


Fig. 5. Illustration of multilayer clustering of scenes and activities. Top: original surveillance video scenes. Middle: related scenes are grouped into clusters (green dashed boxes) and the local topics/activities are learned in each scene. Bottom: local topics are further grouped into activity clusters (color lines indicate some examples) and activity clusters are merged to construct an STB.

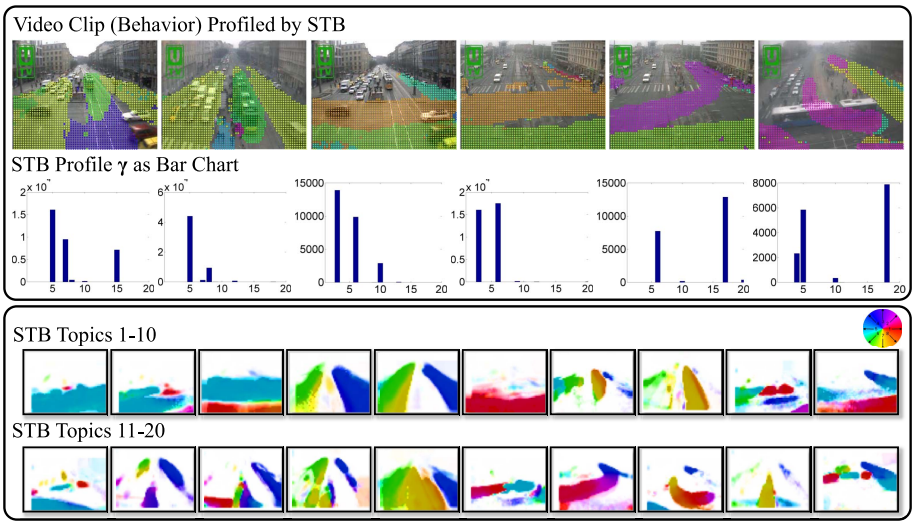


Fig. 6. Illustration of behavior profiling on STB. Left: visual words are profiled by STB and plotted as colored dots. Note that colors here indicate visual words belonging to individual activities in the STB instead of motion direction. Profiling γ is also given as bar chart in which the x-axis indexes STB activities. Right: STB activities where color patches indicate the distribution of motion vectors.

In most cases, x and y are not discrete values because of the matrix multiplication. In order to obtain the value of $\boldsymbol{\beta}(x', y', d')$, we perform interpolation, i.e., we use the values of adjacent pixels surrounding $[x, y, d]$ to determine the value of $\boldsymbol{\beta}(x', y', d')$. This interpolation is related only to spatial values in a single layer, i.e., d is fixed, and we use only the adjacent pixels by varying x and y. A number of standard interpolation techniques can be used for this task, including linear, bilinear, and bicubic interpolations, and we use bicubic interpolation here. After interpolation, we compute the exact value of each element/pixel $\boldsymbol{\beta}(x', y', d')$. Since this transformation involves translation, the transformed topic $\boldsymbol{\beta}'$ may extend out of the topic boundary, a $N_a \times N_b$ rectangle, defined by the original topic $\boldsymbol{\beta}$. To ensure that all topics are

comparable with the same codebook size, we keep only the part of $\boldsymbol{\beta}'$ that lies within the $N_a \times N_b$ rectangle defined by the original topic $\boldsymbol{\beta}$. After the above procedure, the transformed topic $\boldsymbol{\beta}'$ has the same size as the original $\boldsymbol{\beta}$, $N_a \times N_b \times N_m$. Finally, we normalize the transformed topic $\boldsymbol{\beta}'$ to obtain a multinomial distribution as follows:

$$\boldsymbol{\beta}' = \frac{\boldsymbol{\beta}'}{\sum_{x=1\ldots N_a}\sum_{y=1\ldots N_b}\sum_{d=1\ldots N_m}\boldsymbol{\beta}'(x,y,d)}. \tag{5}$$

2) Affinity and Clustering: Given the scene alignment above, we define the relatedness between scenes a and b by the percentage of corresponding topic pairs. More specifically, given K^a local topics $\{\boldsymbol{\beta}^a_{k^a}\}_{k_a=1}^{K^a}$ in scene a and K^b local topics $\{\boldsymbol{\beta}^b_{k^b}\}_{k_b=1}^{K^b}$ in scene b, the distance between topic $\boldsymbol{\beta}^a_{k^a}$ and topic $\boldsymbol{\beta}^b_{k^b}$ is defined as $\mathcal{D}_{\mathbf{KL}}$ in

$$\mathcal{D}_{\mathbf{KL}}\left(\boldsymbol{\beta}^a_{k^a}, \boldsymbol{\beta}^b_{k^b}\right) = \frac{1}{2}\left(\mathbf{KL}\left(\boldsymbol{\beta}^{a2b}_{k^a} \,||\, \boldsymbol{\beta}^b_{k^b}\right) + \mathbf{KL}\left(\boldsymbol{\beta}^{b2a}_{k^b} \,||\, \boldsymbol{\beta}^a_{k^a}\right)\right)$$

$$\mathbf{KL}\left(\boldsymbol{\beta}^a_{k^a} \,||\, \boldsymbol{\beta}^b_{k^b}\right) = \frac{1}{N_v}\sum_{v=1}^{N_v}\boldsymbol{\beta}^a_{k^a v}\cdot\log\left(\frac{\boldsymbol{\beta}^a_{k^a v}}{\boldsymbol{\beta}^b_{k^b v}}\right). \tag{6}$$

Given a threshold τ, the similarity between two topics can be binarized. Topic pairs with the distance less than a threshold are counted as inliers, defined by

$$\begin{aligned}\text{NumInlier} = &\sum_{k^a} 1\left(\min_{k^b}\left(\mathcal{D}_{\mathbf{KL}}\left(\boldsymbol{\beta}^a_{k^a}, \boldsymbol{\beta}^b_{k^b}\right)\right) < \tau\right)\\ &+ \sum_{k^b} 1\left(\min_{k^a}\left(\mathcal{D}_{\mathbf{KL}}\left(\boldsymbol{\beta}^b_{k^b}, \boldsymbol{\beta}^a_{k^a}\right)\right) < \tau\right)\end{aligned} \tag{7}$$

where $1(\cdot)$ is the indicator function. The final relatedness measure $\mathcal{D}(a,b)$ between scenes a and b is the percentage of inlier topic pairs

$$\mathcal{D}(a,b) = \frac{\text{NumInlier}}{K^a + K^b}. \tag{8}$$

Since (6) and (7) are symmetric, (8) is also symmetric. Given this relatedness measure, every scene pair is compared to generate an affinity matrix, and self-tuning spectral clustering [33] is used to group scenes into $c = 1\ldots C$ semantically similar scene-level clusters [see Fig. 5 (middle) for an example].

B. Learning a Shared Activity Topic Basis

Scenes clustered according to Section IV-A are semantically similar; however, the representation in each is still distinct. We next show how to establish a shared representation for every scene in a particular cluster. We denote the set of scenes in a cluster by $\mathcal{C}$. We first choose the scene with the lowest distance to all other scenes in the cluster as the reference scene/coordinate s_{ref}. Activities in all scenes $s \in \mathcal{C}$ can be projected to the reference coordinates via transform $\mathbf{T}^{s2s_{\text{ref}}}$ as stated in

$$\forall s \in \mathcal{C} \quad \forall k = 1\ldots K : \tilde{\boldsymbol{\beta}}^s_k = \mathbb{H}\left(\boldsymbol{\beta}^s_k; \mathbf{T}^{s2s_{\text{ref}}}\right). \tag{9}$$

Once every topic is in the same coordinate system, we create an affinity matrix for all the transformed topics $\{\tilde{\boldsymbol{\beta}}^s_k\}_{s\in\mathcal{C}}$ using the symmetrical Kullbeck–Leibler Divergence as a distance metric (6). Hierarchical clustering is then applied to group the projected activities into K^{stb} clusters $\{\mathcal{T}_k\}_{k=1}^{K^{\text{stb}}}$ ($\mathcal{T}_k$ denotes the set of activities in a cluster k). The result is that semantically corresponding activities across scenes are now grouped into the same cluster. We then take the mean of activities in each activity cluster $\mathcal{T}_k$ as one *shared activity topic* $\boldsymbol{\beta}^{\text{stb}}_k$ as in (10). An alternative to this approach is to relearn topics from the concatenation of visual words of all the scenes in a single cluster. However, this learning-from-scratch strategy prevents explicitly identifying shared and unique topics across scenes, because the trace of local topics from individual scenes to STB is lost. In contrast, our framework reveals how scenes are similar or different

$$\forall k = 1\ldots K^{\text{stb}} : \boldsymbol{\beta}^{\text{stb}}_k = \frac{1}{\mid \mathcal{T}_k \mid}\sum_{k',s'\in\mathcal{T}_k}\tilde{\boldsymbol{\beta}}^{s'}_{k'}. \tag{10}$$

We denote the set of *shared activity topics* $\{\boldsymbol{\beta}^{\text{stb}}_k\}_{k=1}^{K^{\text{stb}}}$ learned for the cluster as the STB. The resulting STB captures both common and unique activities in every scene member [see Fig. 5 (bottom) for an example]. We can now represent the behaviors in every scene as STB profiles: By projecting the STB back to each scene and recomputing the topic profile $\boldsymbol{\gamma}^{\text{stb}}_j$ now defined on $\{\boldsymbol{\beta}^{\text{stb}}_k\}_{k=1}^{K^{\text{stb}}}$, in contrast to the original scene-specific representation ($\boldsymbol{\gamma}^s_j$ defined in terms of $\{\boldsymbol{\beta}^s_k\}_{k=1}^{K}$), that is, rerunning Algorithm 1, but with $\boldsymbol{\beta}$ fixed to the STB values obtained from (10). An example of behavior profiling on STB is illustrated in Fig. 6. Visual words accumulated within a clip are profiled according to the STB. Thus, each behavior can be treated as a weighted mixture of multiple activities.

V. Cross-Scene Query-by-Example and Classification

Given the structured multiscene model introduced in the previous section, we can now describe how cross-scene query and classification can be achieved.

A. Cross-Scene Query

Activity-based query-by-example aims at retrieving semantically similar clips to a given query clip. In the cross-scene context, the pool of potential clips to be searched for retrieval includes clips from every camera in the network. Within a scene cluster $\mathcal{C}$, we segment each video s into $j = 1\ldots N_d$ short clips (Section III-A). We represent the jth video clip in scene s as topic profile $\boldsymbol{\gamma}^{\text{stb}}_{js}$ defined on STB $\boldsymbol{\beta}^{\text{stb}}_k$. A query clip q represented by STB profile $\boldsymbol{\gamma}^{\text{stb}}_{qs}$ can now be directly compared with all other clips in the cluster $\{\boldsymbol{\gamma}^{\text{stb}}_{js'}\}_{j,s'\in\mathcal{C}}$ using L2 distance. In this way, *cross-scene query-by-example* is achieved by sorting all clips in the cluster according to the distance to the query.

B. Cross-Scene Classification

Given an existing annotated database of scenes modeled with our multilayer framework, classification in a new scene s^* can now be achieved *without further annotation*. First s^* is associated with a cluster c^* (Section IV-A). Although s^* has no annotation, this reveals a set of semantically corresponding existing scenes from which annotation can meaningfully be

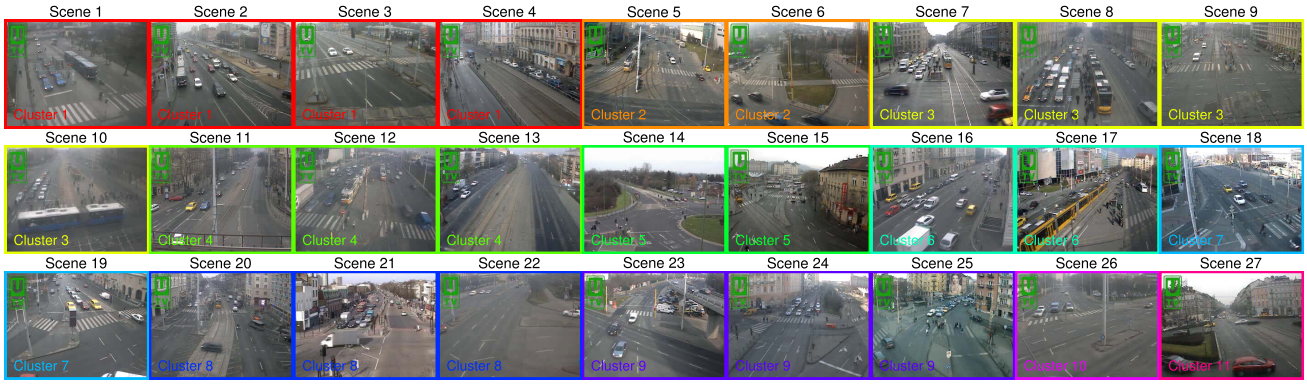


Fig. 7. Example frames for our multisurveillance video data set, with each scene assigned a reference number on top of the frame. The color of the bounding box and the text in the bottom left indicate the assigned cluster.

borrowed. Classification can thus be achieved by any classifier, using all other scenes/clips and labels from cluster c^* as the labeled training set.

It should be noted that our cross-scene classification differs from [34] and [35] in the following ways.

1) We train on a set of source scenes before testing on a held-out scene rather than one source to one test scene. The conventional 1-1 approach requires implicitly the source and target scenes to be *relevant*, which must be identified manually. Our model is able to group relevant scenes automatically, without requiring the user to know this as *a priori*.
2) Our model works in a transductive [7] manner, that is, it looks at target scene data during scene clustering but without looking at the target data label. This weak assumption is more desirable in practice because surveillance video data are often easy to collect without any labeling, whereas the effort required for labeling is the bottleneck.

VI. Multiscene Summarization

In this section, we present a multiscene video summarization algorithm that exploits the structure learned in Section IV to compress cross-scene redundancy. All clips are represented by their profile on STB. The general objective of multiscene summarization is to generate a *video skim* with at least one example of each distinct behavior in the shortest possible summary. We generate independent summaries for each scene cluster (since different scene clusters are semantically dissimilar) and multiscene summaries within each cluster (since scenes within a cluster are semantically similar).

k-Center Summaries: The multiscene summary video is of configurable length N_{sum}. Longer videos will show more distinct behaviors or more within-class variability of each behavior. We compose the summary Σ of N_{sum} clips $\{\boldsymbol{\gamma}_j^{\text{stb}}\}$, $j \in \Sigma$ drawn from all scenes in the cluster. The objective is that all clips in the cluster $\{\boldsymbol{\gamma}_{js}^{\text{stb}}\}_{j,s\in\mathcal{C}}$ should be near to at least one clip in the summary (i.e., the summary is representative). Formally, this objective is to find the summary set Σ that minimizes the cost J in

$$J = \max_{j,s\in\mathcal{C}} \left(\max_{j'\in\Sigma} \mathcal{D}_\gamma \left(\boldsymbol{\gamma}_{j'}^{\text{stb}}, \boldsymbol{\gamma}_{js}^{\text{stb}} \right) \right) \tag{11}$$

where $\mathcal{D}_\gamma$ is the L2 distance. This is essentially a k-center problem [36]. Since it is intractable to enumerate all combinations/potential summaries Σ, we adopt the two-approximation algorithm [37] to this optimization. The resulting $K = N_{\text{sum}}$ centers identify the summary clips.

VII. Experiments

Data Set: We collected 25 real traffic surveillance videos from publicly accessible online Web cameras in Budapest, Hungary. These videos are combined with two surveillance video data sets Junction and Roundabout [15] for a total of 27 videos. Sample frames for each scene are illustrated in Fig. 7(a). We trim each video to 18 000 frames in 10 frames/s, of which 9000 are used to learn the model, and the remaining 9000 frames are used for testing (query, classification, and summarization). For activity learning, we segment each training video into 25 frame clips, so 360 clips are generated for each scene. For both query and summarization applications, we segment test videos into clips with 80 frames, so 112 clips for query and summarization are generated from each scene. Thus, we have three types of video clips: 1) clips for unsupervised training of LDA; 2) clips for training cross-scene classification, retrieval, and multiscene summarization (semantic training clips); and 3) clips for testing on the same classification, retrieval, and summarization tasks (semantic testing clips). LDA clips are shorter (25 frames) to facilitate learning more cleanly segmented activities. Semantic clips are longer (80 frames) as a more human-scale user-friendly unit for visualization and annotation.

Learning Activities: We computed optical flow [30] for all videos by quantizing the scenes with 5×5 pixel cells and eight directions. Local activities are learned from each video independently by using LDA with $K = 15$ activities per scene.

Behavior Annotation: Behavior is a clip-level semantic tag defining the overall scene activity. Due to the semantic gap between behaviors in the video clip and (potentially task-dependent) human interpretation, it is difficult to give video a concise and consistent semantic label (in contrast to event [9] recognition and human action [34]). Instead of annotating each video clip explicitly, we give a set of binary activity tags (each representing the action of some objects within the scene) to each video clip, as shown in Table I. All the tags

TABLE I

ORIGINAL ANNOTATION ONTOLOGY AND TWO MERGING SCHEMES GIVE MULTIPLE GRANULARITIES OF ANNOTATION

No.	Original Annotation	Merge Scheme 1	Merge Scheme 2
1	Vehicle Left Sparse	Vehicle Left	Vehicle Horizontal
2	Vehicle Left Dense		
3	Vehicle Right Sparse	Vehicle Right	
4	Vehicle Right Dense		
5	Vehicle Up Sparse	Vehicle Up	Vehicle Vertical
6	Vehicle Up Dense		
7	Vehicle Down Sparse	Vehicle Down	
8	Vehicle Down Dense		
9	Vehicle Southeast Sparse	Vehicle Southeast	Vehicle SE& NW
10	Vehicle Southeast Dense		
11	Vehicle Northwest Sparse	Vehicle Northwest	
12	Vehicle Northwest Dense		
13	Vehicle Up2Right Turn	Vehicle Up2Right Turn	Vehicle Up2Right Turn
14	Vehicle Left2Up Turn	Vehicle Left2Up Turn	Vehicle Left2Up Turn
15	Vehicle Up2Left Turn	Vehicle Up2Left Turn	Vehicle Up2Left Turn
16	Tram Up	Tram Up	Tram Up
17	Tram Down	Tram Down	Tram Down
18	Pedestrian Horizontal	Pedestrian Horizontal	Pedestrian Horizontal
19	Pedestrian Vertical	Pedestrian Vertical	Pedestrian Vertical

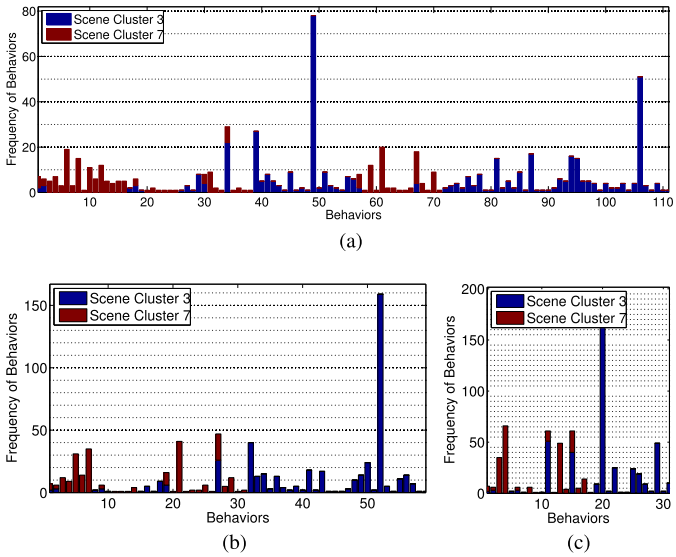


Fig. 8. Frequencies of behaviors of each category. (a) Original annotation. (b) Merge scheme 1. (c) Merge scheme 2.

associated with vehicles have a sparse or dense option. When there are fewer than three vehicles traveling in a clip, it is labeled as sparse; otherwise, it is labeled as dense. Each unique combination of activities that exists in the labeled clips then defines a unique scene-level behavior category. We explore this through multiple sets of annotations: an original annotation with 19 distinct tags and subsequent coarser label sets derived by Merge Scheme 1 with 13 distinct tags and Merge Scheme 2 with 10 distinct tags. The activity tags are given in Table I. We exhaustively annotate video clips in two example scene clusters (3 and 7, as shown in Fig. 7). Across the two clusters, there are six scenes with 112 clips per scene annotated (672 clips in total). In the original annotation case, there are 111 total behaviors identified. The distribution of behaviors is illustrated in Fig. 8(a). However, this number is more than necessary in terms of limited distinctiveness of the numerous entailed behaviors. By merging some activity annotations, we generate 59 or 31 (Merge Scheme 1 or 2 in Table I) unique behaviors. It should be noted that the frequency of behaviors is rather imbalanced, as indicated in Fig. 8(a)–(c). There is also a very limited overlap of behaviors between Scene Clusters 3 and 7. To assess annotation consistency and bias, we invited eight independent annotators to annotate all the video clips separately. We observe that the additional annotations are fairly consistent with the original annotation: with more than 80% agreement (Hamming distance) between the additional and the original annotations. A detailed analysis of these additional annotations is given in Supplementary Information.

A. Multilayer Scene Clustering

The multilayer scene clustering is conducted in two stages. We first group scenes into clusters and then within each scene cluster shared activity topics are learned.

1) Scene-Level Clustering: We first group the scenes into semantically similar clusters by spectral clustering. The similarity measurement between scenes is the number of corresponding activities, as defined in Section IV-A. The self-tuning spectral clustering automatically determines the appropriate number of clusters, which in the case of our 27-scene data set is 11 clusters. Fig. 7 shows the results in which semantically similar scenes are indeed grouped (e.g., camera toward one direction at road junctions in Cluster 3) and unique views are separated into their own cluster (e.g., Cluster 11).

2) Learning a Shared Activity Topic Representation: Within each scene cluster, we unify the representation by computing an STB. We automatically set the number of shared activities K^{stb} in each scene cluster with N_s scenes as $K^{\text{stb}} = \text{coeff} \times N_s$, where coeff is set to 5. The discovered basis from an example cluster (Scene Cluster 3 shown in Fig. 7) with four scene members is illustrated in Fig. 9. Fig. 9 reveals both activities unique to each scene (Topics 1–15) and activities common among multiple scenes (Topics 16–20). Thus, some shared activity topics are composed of single local/original topics and others of multiple local topics.

B. Cross-Scene Query-by-Example and Classification

In this section, we evaluate the ability of our framework to support two tasks: cross-scene query-by-example and cross-scene behavior classification. We compare our scene cluster model (SCM) with a baseline flat model (FM). Our SCM first groups scenes into scene clusters according to their relatedness and learns STB for every scene cluster. Video clips in each scene cluster are thus represented as topic profiles on the STB of the scene cluster. As with our model, an FM first learns a local topic model per scene; however, it then learns a single STB from all labeled scenes (six scenes from two clusters) without scene level clustering, instead of one STB per cluster. The only difference between the SCM and the FM is the absence of scene-level clustering in the FM. Note that the FM is a special case of our SCM with one scene-level cluster. Moreover, the individual scenes are also a special case of our SCM with one cluster per scene.

1) Query-by-Example Evaluation: To quantitatively evaluate query-by-example, we exhaustively take each scene and each clip in turn as the query, and all other scenes are considered as the pool. All clips in the pool are ranked

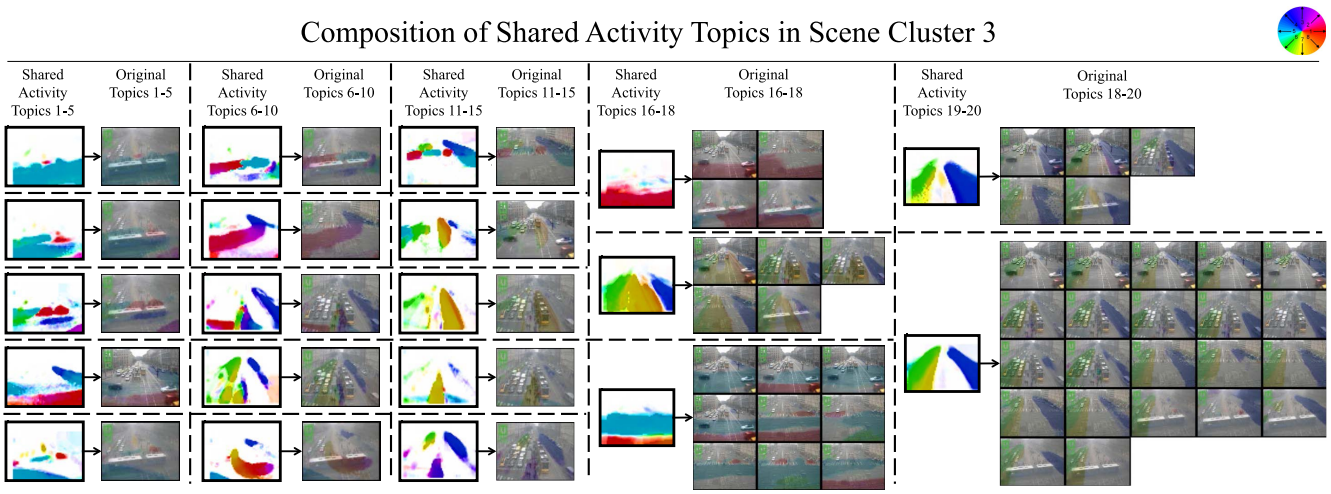


Fig. 9. Example STB learned from Scene Cluster 3. Shared activity topics may be composed of one or more local/original topics. Original topics are overlaid on the background frame. The color patches indicate the distribution of motion vectors for a single activity.

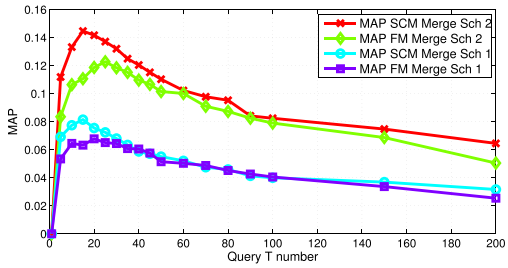


Fig. 10. Query-by-example MAP with different number of retrievals.

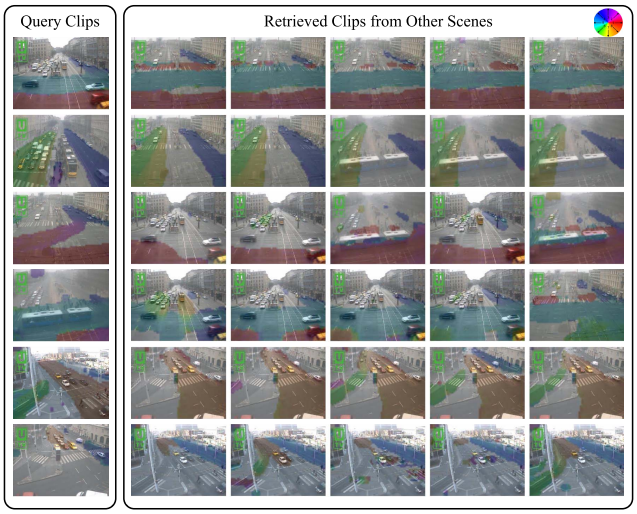


Fig. 11. Examples of cross-scene query-by-example. Left (Query Clips): six query clips randomly chosen from six scenes. Right (Retrieved Clips): image matrix illustrates the retrieved clips from the remaining five scenes, sorted by distance to query, from left to right in the matrix. Color patches overlaid on the background indicate the visual words accumulated within a video clip.

according to similarity (L2 distance on the STB profile) to the query. The performance is evaluated according to how many clips with the same behavior as the query clip are in the top T responses. We retrieve the best $T = 1 \ldots 200$ clips and calculate the Average Precision of each category for each T. Mean Average Precision (MAP) is computed by taking the mean value of Average Precision over all categories. The MAP curve by the top T responses to a query for both SCM and FM and merge schemes 1 and 2 are plotted in Fig. 10.

TABLE II

CROSS-SCENE CLASSIFICATION ACCURACY WITH 31 AND 59 CATEGORIES FOR BOTH SCM AND FM

Category	31		59	
	SCM	FM	SCM	FM
Scene 1	**55.36%**	50.89%	**42.86%**	40.18%
Scene 2	27.68%	**39.29%**	**18.75%**	16.96%
Scene 3	**49.11%**	41.96%	**39.29%**	37.50%
Scene 4	**54.46%**	46.43%	**37.50%**	36.61%
Scene 5	**30.36%**	26.79%	**17.86%**	**17.86%**
Scene 6	**38.39%**	25.00%	**20.54%**	12.50%
Average	**42.56%**	38.39%	**29.47%**	26.94%

It is evident that for Merge Schemes 1 and 2, the proposed SCM performs consistently better than the FM regardless of the number of top retrievals T. This is because in the SCM, the STB learned from this set of scenes is highly relevant to each scene in the cluster. In contrast, the FM learns a single STB for all scenes, making the STB less relevant to each individual scene and hence less informative as a representation for retrieval.

Qualitative results are also given in Fig. 11 by presenting six randomly chosen queries and their retrieved clips. Different types of behaviors are covered by query clips and most retrieved clips are semantically similar to query clips. The only exception is in the third row where the query clip indicates traffic going east and turning from left to up. This is because there is no corresponding behavior in the other scenes.

2) Classification Evaluation: In this experiment, we quantitatively evaluate classification performance where the test scene has *no labels*. Successful classification thus depends on correctly finding semantically related scenes and appropriately transferring labels from them (Section V). We perform leave-one-scene-out evaluation by holding out one scene as the unlabeled testing set and predicting the labels for the test set clips using the labels in the remaining scenes using the K Nearest Neighbor (KNN) classifier. The KNN K parameter is determined by cross validating among the remaining scenes. Classification performance is evaluated by the accuracy for each category of behavior, averaged over all held-out scenes.

From Table II, we observe that at either granularity of annotation (59 or 31 categories), our SCM outperforms the

TABLE III
SUMMARIZATION SCHEMES FOR CONDITION WC

Summarization Method	Description
Random	This lower-bound picks clips randomly from multiple scenes to compose the summary
Single-Scene Graph	The overall summary is a concatenation of independent summaries for each video by doing recursive Normalized cut [38] on a graph constructed by taking each video clip as vertices and L2 distance between topic profile γ of each clip as edges. Here each video clip is represented by scene-specific local topics. This corresponds to [39], but without temporal graph.
Single-Scene Kcenter	Similar to *Single-Scene Graph* method, but using Kcenter algorithm in Eq. (11) for summarization instead of Normalized Cut.
Multi-Scene Graph	This model learns a STB to represent video clips from all scenes with STB profile. Then Normalized Cut is applied to cluster clips and find multi-scene summaries.
Multi-Scene Kcenter	Our full model builds a STB from all scenes within a cluster, then uses the Kcenter algorithm to select summary clips from all scenes.

FM on average. This shows that, again, in order to borrow labels from other scenes for cross-scene classification, it is important to select relevant sources, which we achieve via scene clustering. The FM is easily confused by the wider variety of scenes to borrow labels from, whereas our SCM structures similar scenes and borrows labels from only semantically related scenes to avoid negative transfer [7], [8].

C. Multiscene Summarization

In the final experiment, we evaluate our multiscene summarization model against a variety of alternatives. We consider two conditions: 1) multiscene summarization within a scene cluster [Condition Within-Cluster Summarization (WC)] and 2) unconstrained multiscene summarization, including videos spanning multiple-scene clusters [Condition Across-Cluster Summarization (AC)].

Condition WC: In this experiment, we focus on the comparison between the multiscene model and the single-scene model for the given various summarization algorithms. The multiscene model represents all video clips from different scenes within a cluster with a single STB learned from the scene cluster, whereas the single-scene model represents each video with scene-specific activities, and the overall summary is the mere concatenation of summaries from each scene. Specifically, we compare the summarization methods listed in Table III.

Condition AC: In this experiment, analogous to query and classification, we focus on the comparison between the FM and the SCM for the given different summarization algorithms. The FM learns a single STB from all scenes available without discrimination, while the SCM learns an STB per scene cluster. Specifically, we compare the summarization schemes in Table IV.

Settings: To systematically evaluate summarization performance, we vary the length of the requested summary. In Condition WC, the summary varies from eight to 120 clips (64 s to 16 min) out of the overall 448 video clips (59.7 min) in Scene Cluster 3 [as shown in Fig. 7(a)] and 224 video clips (29.9 min) in Scene Cluster 7. In Condition AC, the summary varies from six to 120 clips (48 s to 16 min) out of 672 video clips (89.7 min), the total of which is a combination

TABLE IV
SUMMARIZATION SCHEMES FOR CONDITION AC

Summarization Method	Description
Random	This picks clips randomly from multiple scenes to compose the summary
Flat Multi-Scene User Attention	Leverages the magnitude, spatial and temporal phase of optical flow vectors to index videos. This is the visual attention measurement of ([40], Eq. (6)). We tested the model on a combined video by concatenating each individual video.
Flat Multi-Scene Graph	This model uses Normalized Cut [38] to cluster all video clips represented as single STB profiles. This is similar to [39].
Flat Multi-Scene Kcenter	Same as *Flat Multi-Scene Graph*, but using Kcenter to select summary clips.
Scene Cluster Multi-Scene Kcenter	Our full model clusters the scenes, learns STBs on each scene cluster, followed by Kcenter to summaries within each scene cluster

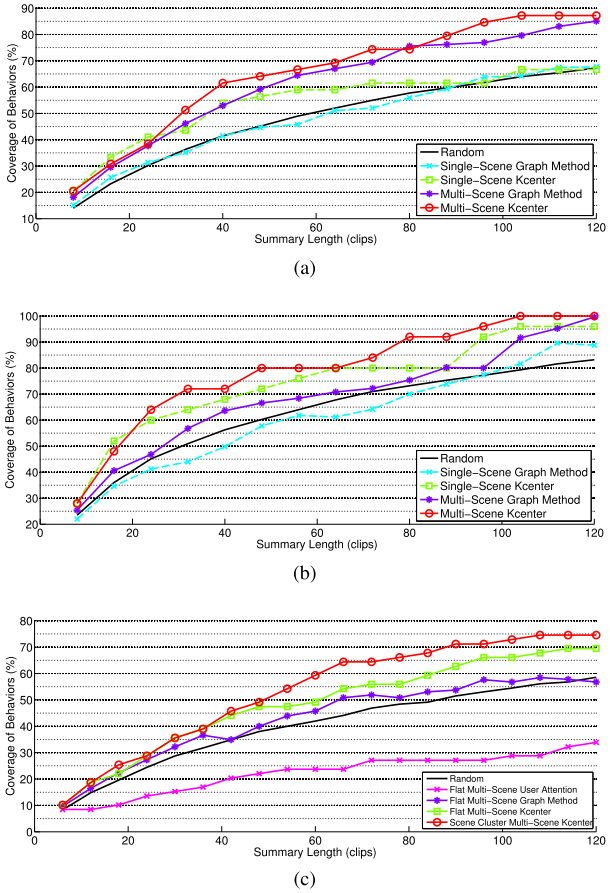


Fig. 12. Video summarization results: coverage of behaviors versus summary clip length. (a) Condition WC: Scene Cluster 3 (four scenes in total). (b) Condition WC: Scene Cluster 7 (two scenes in total). (c) Condition AC: all scenes (six scenes in total).

of Scene Clusters 3 and 7. All video clips for summarization are represented as topic profile γ. Recall that each local scene is learned with $K = 15$ topics and scene clusters with N_s scenes are learned with $K = \text{coeff} \times N_s$ topics where coeff is set to 5. For fair comparison, FM baselines are learned with the sum of the number of topics for each cluster.

Summarization Evaluation: The performance is evaluated by the coverage of identified behaviors in the summary, averaged over 50 independent runs. Fig. 12(a) and (b) shows the results for multiscene summarization within two example clusters (Condition WC). Clearly, our multiscene k-center

algorithm (red curve) outperforms the baselines: both graph method alternative (purple curve) and single-scene alternative (dashed line). The performance margin is greater between the multiscene and single-scene models for the first cluster because there are four scenes, so greater opportunity to exploit interscene redundancy. This validates the effectiveness of jointly exploiting multiple scenes for summarization. Fig. 12(c) shows the result for multiscene summarization across both clusters (Condition AC); our SCM builds one summary for each cluster to exploit the expected greater volume of within-cluster redundancy. In contrast, the FM builds one single summary but for a much more diverse group of data, and the single-scene models have no across-cluster redundancy to exploit. Even in the flat case, our k-center model (shown by the green curve) still outperforms all other alternatives (shown by the purple and magenta curves). It is also worth noting that the user attention model degenerates severely on our data set because it is unable to extract semantic meaning from videos in which pure motion strength is not informative enough to distinguish semantic behaviors. The qualitative results for the multiscene summarization are presented in Supplementary Information.

D. Further Analysis

In this section, we further analyze the robustness of our framework by varying key parameters and investigate their impact on the model performance.

1) Generalized Scene Alignment: We assume currently that cameras are installed upright and only scaling and translational transforms are applied to the scene alignment. However, under more generality, rotational transforms may also be considered. To that end, one can consider a generalized scene alignment that includes a rotational parameter ϕ in the transformation. Recall that in Section IV-A, we estimate the size of transformed topics. We can extend that to $N'_a = N_a \times t_s \times \cos(\phi)$ and $N'_b = N_b \times t_s \times \sin(\phi)$. The generalized transform matrix $\mathbf{T}$ is then defined as

$$\mathbf{T} = \begin{bmatrix} t_s \cdot \cos(\phi) & -t_s \cdot \sin(\phi) & t_x \\ t_s \cdot \sin(\phi) & t_s \cdot \cos(\phi) & t_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

The procedure to transform a topic under this generalized alignment differs from the original alignment only in the estimation of direction d. To determine d, given d', we represent the quantized optical flow as vector $\text{vec}' = [\cos(2\pi d'/N_m), \sin(2\pi d'/N_m)]^{\mathbf{T}}$. Then we estimate the original flow vector $\text{vec} = \mathbf{T}^{*-1}\text{vec}'$, where $\mathbf{T}^*$ is a 2×2 matrix from the first two dimensions of $\mathbf{T}$ because translation does not change motion direction. We determine d by nearest neighbor as follows:

$$\hat{d} = \underset{d=1\ldots N_m}{\arg\min} \left\| \text{vec} - \begin{bmatrix} \cos(2\pi d/N_m) \\ \sin(2\pi d/N_m) \end{bmatrix} \right\|. \tag{13}$$

To align scene A to scene B with this generalized alignment, we can estimate parameters by maximizing the marginal likelihood of target document $\mathbf{X_b}$ for the given source topics β_a. Specifically, we denote the transform operation with specified parameters as $\mathbb{H}(\beta|t_s, t_x, t_y, \phi)$.

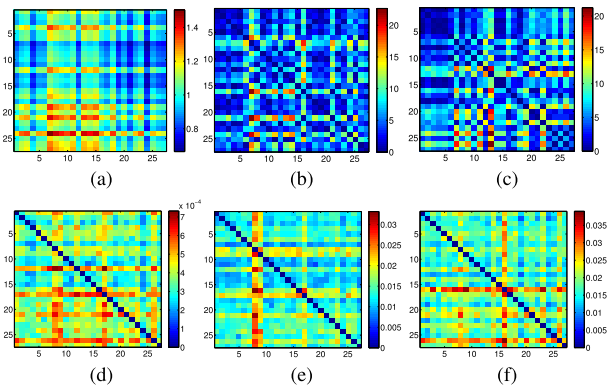


Fig. 13. Alignment and stability across all pairs of 27 scenes. (a) Absolute reference value of scaling. (b) Absolute reference value of x translation. (c) Absolute reference value of y translation. (d) RMSE of scaling. (e) RMSE of x translation. (f) RMSE of y translation.

Given target document $\mathbf{X_b}$, the marginal likelihood is $p(\mathbf{X_b}|\alpha_a, \mathbb{H}(\beta_a|t_s, t_x, t_y, \phi))$, where α_a is the Dirichlet prior in scene A. Because scaling and translational parameters are computed by a closed-form solution (3), we need only to search $\hat{\phi} = \text{argmax}_\phi\, p(\mathbf{X_b}|\alpha_a, H(\beta_a|s, dx, dy, \phi))$. However, in our experiments, by applying this generalized alignment process, we observed many local minima—suggesting that the rotational transform is underconstrained and not very repeatable.

2) Scene Alignment Stability: We first evaluate the stability of scene-level alignment. Recall that given two scenes a and b, we first normalize each scene with geometrical transformations $\mathbf{T}^a_{\text{norm}}$ and $\mathbf{T}^b_{\text{norm}}$. The scene a to b transform is thus defined by

$$\mathbf{T}^{a2b} = \mathbf{T}^{b-1}_{\text{norm}} \cdot \mathbf{T}^a_{\text{norm}} = \begin{bmatrix} \frac{t^a_s}{t^b_s} & 0 & \frac{t^a_x}{t^b_s} - \frac{t^b_x}{t^b_s} \\ 0 & \frac{t^a_s}{t^b_s} & \frac{t^a_y}{t^b_s} - \frac{t^b_y}{t^b_s} \\ 0 & 0 & 1 \end{bmatrix}. \tag{14}$$

We denote $s^{a2b} = (t^a_s/t^b_s)$, $dx^{a2b} = (t^a_x/t^b_s) - (t^b_x/t^b_s)$, and $dy^{a2b} = (t^a_y/t^b_s) - (t^b_y/t^b_s)$. The parameters estimated from the full data in each scene are denoted by s^{a2b}_{ref}, dx^{a2b}_{ref}, and dy^{a2b}_{ref}. To evaluate the stability of this alignment, we randomly sample 50% of the original data from each scene and estimate again the parameters as s^{a2b}_{50}, dx^{a2b}_{50}, and dy^{a2b}_{50}. We run this process 20 times and calculate the root-mean-square error (RMSE), defined in (15) for s^{a2b}. RMSEs for dx and dy are defined in the same way by replacing s^{a2b} with dx^{a2b} and dy^{a2b}, respectively

$$\text{RMSE}(s) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(s^{a2b}_{50i} - s^{a2b}_{\text{ref}}\right)^2}. \tag{15}$$

We show both the absolute value of reference parameters and the RMSE when aligning each pair of scenes in Fig. 13. It is evident that most scene pairs are scaled between 0.7 and 1.5 [Fig. 13(a)]. The worst RMSE(s) among all scene pairs are 0.0007 [Fig. 13(d)]. The same observations can be made on variability of x translation and y translation with the largest RMSE(dx) and RMSE(dy) being 0.035 pixels

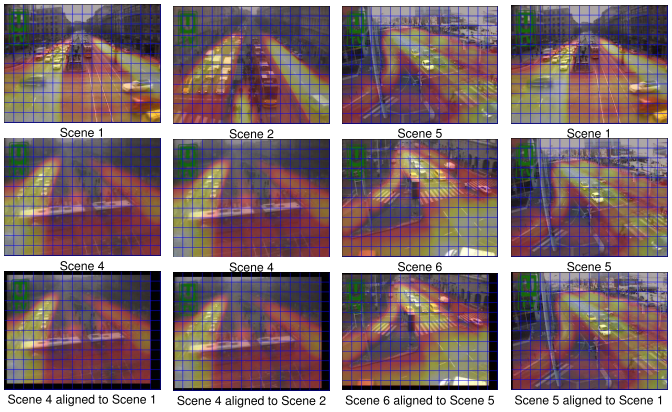


Fig. 14. Examples of scene alignment pairs. Each column indicates one example alignment. The first row is the target scene, the second row is the source scene to be aligned/transformed, and the last row is the source scene after alignment to the target. Both within-scene cluster (first three columns: Clusters 3 and 7, respectively) and across cluster (fourth column: Clusters 3 and 7) examples are presented. The overlaid heat map is the spatial frequency of visual words.

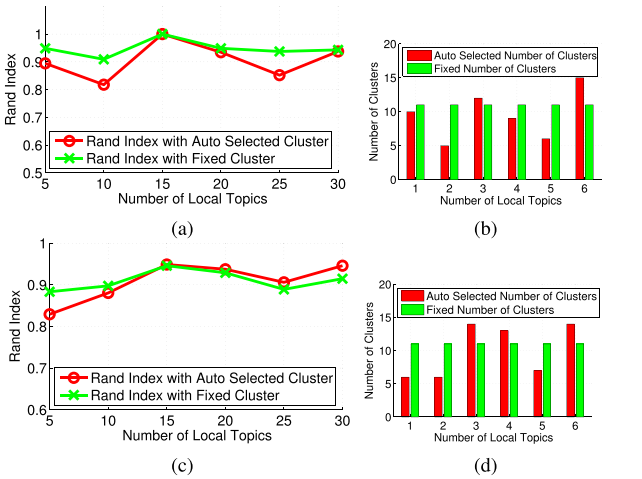


Fig. 15. Stability of scene-level clustering. (a) RI cell size = 5. (b) Number of cluster cell size = 5. (c) RI cell size = 10. (d) Number of cluster cell size = 10.

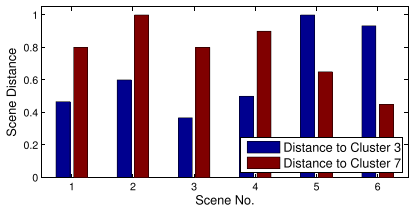


Fig. 16. Association of held-out scenes with clusters. Scenes 1–4 are held out from Cluster 3 and Scenes 5 and 6 are held-out scenes from Cluster 7. All held-out scenes are associated correctly.

or less, whereas the absolute values of reference x and y translations are between 0 and 20 pixels. The small values of these deviations verify that the scene alignment model is robust and repeatable. Some examples of scene alignment are shown in Fig. 14. Although the majority of activities are aligned well, some are less so because of the limitation of a global rigid transform over a whole scene. Further extension could exploit individual activity-centered alignment in addition to the holistic-scene alignment.

3) Scene Cluster Stability: We tested the stability of scene-level clustering by varying the cell size, number of local topics, and clustering strategy.

1) We compared visual word quantization with 5×5 and 10×10 cell sizes.
2) We evaluated from 5 to 30 local topics in each scene by step of 5. i.e., $5, 10, \ldots, 25, 30$.
3) We performed self-tuning spectral clustering with two alternative settings.

The first is that we allowed the model to automatically determine the number of clusters, and the second is that we fixed the number of clusters to the same as in the reference clustering, that is, 15 local topics and 5×5 cell size. We measured the discrepancy between the results from automatic clustering and the reference clustering using the rand index (RI) [41]. It describes the discrepancy between two set partitions and is frequently used as the evaluation metric for clustering. The RI is between 0 and 1, with the higher value indicating more similarity between two partitions. If two partitions are exactly the same, the RI is 1. We show the results on the stability test of scene-level clustering in Fig. 15.

For both cell sizes 5 and 10, automatic cluster selection generates consistent partitions (high RI). Therefore, the framework is robust to motion quantization cell size. However, it is also evident that automatic cluster number selection is less stable in determining the number of clusters, as indicated by the red bars in Fig. 15(b) and (d). On the other hand, by fixing the number of clusters, the partitioning is more stable (consistent high RI).

4) Associating New Scenes: Our model is able to group scenes according to the semantic relatedness if all the recorded data are available in advance. In addition, the model is capable of associating new scenes with existing clusters, e.g., given input from newly installed cameras at different locations, without the need to completely relearn the model. This is achieved by comparing the local topics of a new scene with those of the STB in each scene cluster and choosing the cluster with the highest relatedness. Only the updated cluster needs to be relearned to incorporate the new scene. We tested this approach in Scene Clusters 3 and 7 by: 1) holding out each scene in turn as the candidate scene to be associated and learning STB in each cluster with the other scenes; 2) computing the relatedness between the held-out scene and both clusters using (8); and 3) associating the candidate scene with the cluster with the highest relatedness. We illustrate the result of this via the distance (defined as $1-$ relatedness) between held-out candidate scenes and clusters in Fig. 16. It is evident that each held-out scene is closer to its corresponding cluster, so 100% of scenes are associated correctly. However, this approach is limited to associating new scenes with existing scene clusters (scenes). A full online learning multiscene model is desirable but also challenging and remains to be developed.

5) STB Stability: Finally, we investigate the stability of learning the STB with different number of shared topics. Recall that in Section VII-A1, the number of STB topics for the SCM and the FM is $K = \text{coeff} \times N_s$. Now let us change coeff from 3 to 10 and evaluate how this affects the cross-scene classification accuracy for both annotation schemes 1

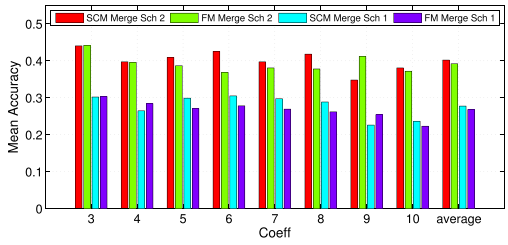


Fig. 17. Effect of varying number of topics used. Classification accuracy of the SCM and the FM.

(59 categories) and 2 (31 categories). The results are shown in Fig. 17. It is evident that for both 59 and 31 categories, our SCM is mostly better than the FM over a range of topic numbers.

VIII. Conclusion

In this paper, we introduced a framework for synergistically modeling multiple-scene data sets captured by multicamera surveillance networks. The paper deals with variable and piecewise interscene relatedness by semantically clustering scenes according to the correspondence of semantic activities and selectively shares activities across scenes within clusters. Besides revealing the commonality and uniqueness of each scene, multiscene profiling further enables typical surveillance tasks of query-by-example, behavior classification, and summarization to be generalized to multiple scenes. Importantly, by discovering related scenes and shared activities, it is possible to achieve cross-scene query-by-example (in contrast to a typical within-scene query) and to annotate behavior in a novel scene without any labels—which is important for making deployment of surveillance system scale in practice. Finally, we can provide video summarization capabilities that uniquely exploit redundancy both within and across scenes by leveraging our multiscene model.

There are still several limitations to our work, which can be addressed in the future.

1) In the current framework, scenes that can be grouped together are usually morphologically similar, which means that the underlying motion patterns and view angles are essentially similar. More advanced geometrical registration techniques could be applied, including similarity and affine transformations, to allow scenes with more dramatic viewpoint changed to be grouped.
2) In this work, motion information is mostly contributed by traffic. However, studying pedestrian/crowd behavior is becoming more interesting [42] due to its wide application in crime prevention and public security. However, compared with traffic, pedestrian crowd behaviors are less regulated and coherent. Thus, exacting suitable features and improving the model to deal with this are nontrivial tasks.
3) Finally, an improved multiscene framework that can fully and incrementally add new scenes in an online manner is of interest.

References

[1] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 539–555, Mar. 2009.

[2] T. Hospedales, S. Gong, and T. Xiang, "Video behaviour mining using a dynamic topic model," *Int. J. Comput. Vis.*, vol. 98, no. 3, pp. 303–323, 2012.

[3] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1450–1464, Sep. 2006.

[4] J. Varadarajan, R. Emonet, and J.-M. Odobez, "A sequential topic model for mining recurrent activities from long term video logs," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 100–126, 2013.

[5] D. Kuettel, M. D. Breitenstein, L. Van Gool, and V. Ferrari, "What's going on? Discovering spatio-temporal dependencies in dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1951–1958.

[6] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1971–1984, Nov. 2008.

[7] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[8] X. Xu, S. Gong, and T. Hospedales, "Cross-domain traffic scene understanding by motion model transfer," in *Proc. 4th ACM/IEEE Int. Workshop ARTEMIS*, Oct. 2013, pp. 77–86.

[9] S. Khokhar, I. Saleemi, and M. Shah, "Similarity invariant classification of events by KL divergence minimization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1903–1910.

[10] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2011.

[11] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using nonparametric hierarchical Bayesian models," *Int. J. Comput. Vis.*, vol. 95, no. 3, pp. 287–312, 2011.

[12] K. Kim, D. Lee, and I. Essa, "Gaussian process regression flow for analysis of motion trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1164–1171.

[13] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1544–1554, Nov. 2008.

[14] M. Fanaswala and V. Krishnamurthy, "Detection of anomalous trajectory patterns in target tracking via stochastic context-free grammars and reciprocal process models," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 1, pp. 76–90, Feb. 2013.

[15] J. Li, S. Gong, and T. Xiang, "Learning behavioural context," *Int. J. Comput. Vis.*, vol. 97, no. 3, pp. 276–304, 2012.

[16] C. C. Loy, T. Xiang, and S. Gong, "Incremental activity modeling in multiple disjoint cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1799–1813, Sep. 2012.

[17] X. Wang, K. Tieu, and W. E. L. Grimson, "Correspondence-free activity analysis and scene modeling in multiple camera views," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 56–71, Jan. 2010.

[18] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 524–531.

[19] L.-J. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2036–2043.

[20] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1168–1181, Apr. 2007.

[21] T. Xiang and S. Gong, "Activity based surveillance video content modelling," *Pattern Recognit.*, vol. 41, no. 7, pp. 2309–2326, 2008.

[22] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 3, no. 1, 2007, Art. no. 3.

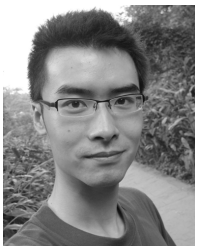
[23] S. E. F. de Avila, A. P. B. Lopes, A. da Luz, Jr., and A. de Albuquerque Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognit. Lett.*, vol. 32, no. 1, pp. 56–68, 2011.

[24] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg, "Webcam synopsis: Peeking around the world," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[25] J.-G. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 161–170.

[26] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z.-H. Zhou, "Multi-view video summarization," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 717–729, Nov. 2010.

[27] C. de Leo and B. S. Manjunath, "Multicamera video summarization and anomaly detection from activity motifs," *ACM Trans. Sensor Netw.*, vol. 10, no. 2, pp. 27:1–27:30, Jan. 2014.
[28] J. Varadarajan, R. Emonet, and J.-M. Odobez, "Probabilistic latent sequential motifs: Discovering temporal activity patterns in video scenes," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 1–11.
[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[30] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.
[31] J. Varadarajan and J.-M. Odobez, "Topic models for scene analysis and abnormality detection," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, Sep./Oct. 2009, pp. 1338–1345.
[32] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Learning multimodal latent attributes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 303–316, Feb. 2014.
[33] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1–8.
[34] J. Zheng and Z. Jiang, "Learning view-invariant sparse representations for cross-view action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3176–3183.
[35] J. Zheng, Z. Jiang, P. J. Phillips, and R. Chellappa, "Cross-view action recognition via a transferable dictionary pair," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.
[36] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theor. Comput. Sci.*, vol. 38, pp. 293–306, 1985, doi: http://dx.doi.org/10.1016/0304-3975(85)90224-5.
[37] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," *Math. Oper. Res.*, vol. 10, no. 2, pp. 180–184, 1985.
[38] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[39] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.
[40] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 907–919, Oct. 2005.
[41] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statist. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.
[42] J. Shao, C. C. Loy, and X. Wang, "Scene-independent group profiling in crowd," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2227–2234.



Xun Xu received the B.E. degree from the School of Electrical Engineering and Information, Sichuan University, Chengdu, China, in 2010. He is currently pursuing the Ph.D. degree with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K.

His research interests include surveillance video understanding, transfer learning, and event recognition.

Timothy M. Hospedales received the Ph.D. degree in neuroinformatics from The University of Edinburgh, Edinburgh, U.K., in 2008.

He is currently a Lecturer (Assistant Professor) of Computer Science with the Queen Mary University of London, London, U.K. He has authored over 30 papers in major international journals and conferences. His research interests include transfer and multitask machine learning applied to problems in computer vision and beyond.

Shaogang Gong received the D.Phil. degree in computer vision from Keble College, Oxford University, Oxford, U.K., in 1989.

He is currently a Professor of Visual Computation with the Queen Mary University of London, London, U.K. His research interests include computer vision, machine learning, and video semantic analysis.

Dr. Gong is a fellow of the Institution of Electrical Engineers and the British Computer Society.