Machine Vision and Applications
DOI 10.1007/s00138-013-0573-2

ORIGINAL PAPER

Summarizing high-level scene behavior

Kevin Streib · James W. Davis

Received: 8 August 2012 / Revised: 27 March 2013 / Accepted: 15 October 2013

Abstract We present several novel techniques to summarize the high-level behavior in surveillance video. Our proposed methods can employ either optical flow or trajectories as input, and incorporate spatial and temporal information together, which improve upon existing approaches for summarization. To begin, we extract common pathway regions by performing graph-based clustering on similarity matrices describing the relationships between location/orientation states. We then employ the activities along the pathway regions to extract the aggregate behavioral patterns throughout scenes. We show how our summarization methods can be applied to detect anomalies, retrieve video clips of interest, and generate adaptive-speed summary videos. We examine our approaches on multiple complex urban scenes and present experimental results.

Keywords Behavioral summarization · Activity analysis · Video surveillance applications

1 Introduction

Over the past few decades, the number of surveillance cameras being deployed has increased substantially. Unfortunately, the number of security operators responsible for monitoring those sensors has not grown at a proportional rate. As a result, security operators are often tasked with simultaneously monitoring videos from hundreds of surveillance cameras and are often overloaded by the amount of data being received.

Recently, several algorithms have been proposed to automatically summarize videos in some manner to help alleviate this workload. For instance, object trajectories have been be employed to detect common pathways throughout a scene. Videos have also been separated into short clips and clustered together to determine common scene activity patterns. Moreover, multiple techniques have been proposed to recognize anomalous activity, provide shortened representations of long videos, and retrieve segments of video that resemble a given query.

Throughout this paper, we explore various ways to summarize the behavior in surveillance video. In the surveillance domain, the term "behavior" is somewhat ambiguous. In this paper, we are more interested in the high-level (scene-based) behavior—the when, where, and how objects move throughout the scene.

To summarize the high-level behavior of a scene, we present novel behavior analysis techniques to extract common pathway regions and aggregate behavioral patterns from complex scenes. Instead of following the recent trend of employing complicated and computationally expensive clustering algorithms to summarize behavior, e.g., [32,30], we develop sufficient proximity measurements and employ simpler, more efficient, clustering algorithms to achieve strong performance. We then show how our approaches can be employed to aid operators through various applications.

We begin by extracting the behavioral superpixels of the scene, which we define as groups of adjacent pixels which are essentially uniform in the number and speed of tracks at various orientations throughout time. Next, we map the scene activity to superpixel/orientation states and then extract the common pathway regions by clustering a state-wise similarity matrix. By mapping activity to local states, our approach is

K. Streib (✉) · J. W. Davis
Department of Computer Science and Engineering,
Ohio State University, Columbus, OH 43210, USA
e-mail: streib.116@osu.edu

J. W. Davis
e-mail: jwdavis@cse.ohio-state.edu

Published online: 08 November 2013

applicable to both motion flow and trajectories. Furthermore, by incorporating spatial and temporal information into our proximity metric, our approach emulates the main advantages of utilizing trajectories for activity analysis without actually needing trajectories (which are difficult to obtain in complex scenes).

After extracting the pathway regions, we map the activities from short, non-overlapping, video clips to the regions, and then extract the temporal behavioral patterns from a clip-wise similarity matrix. In addition to employing the standard approach of assuming each video clip is an independent entity, we also present a more appropriate grouping method that incorporates temporal information from surrounding clips. Furthermore, we examine the results of multiple proximity metrics when computing the similarity between clips. Finally, we demonstrate further usefulness of our techniques through applications in detecting anomalies, retrieving video clips of interest, and creating speed-adaptive playback of videos.

2 Related work

Scene-based behavior analysis is typically performed by utilizing motion or appearance features [5,11,16,30,33], or by utilizing trajectories [14,22,26,28,32,29]. The benefit of the approaches using motion or appearance features is that they analyze activities without relying on tracking. The majority of these papers employ features based on optical flow [5,16,30,33], or combinations of optical flow and appearance metrics [11]. Furthermore, instead of working at a pixel level, many approaches [5,16,30,33] use features extracted from small cells or spatiotemporal volumes where behavior patterns are generally more consistent.

Many different approaches have been developed which perform scene activity analysis via trajectories. An envelope approach is used in [14] to determine if tracks should be assigned to existing or form new routes. In [29] spectral clustering is employed on pair-wise trajectory-based similarity matrices to extract trajectory clusters. Kernel Density Estimation is employed in [22] to learn a model for the joint probability of a transition between any two image points and the time taken to complete the transition. Vector quantization is used in [26] to reduce trajectories to a set of prototypes. In [32], observations are treated as words and trajectories as documents, which are clustered via language processing algorithms. Tracks are quantized into sets of location/orientation states and spectral clustering is used to extract pathlets from similarity matrices combining temporal and scene entry/exit information in [28].

In addition to the aforementioned algorithms, several techniques have been proposed to analyze videos temporally. In [5,30] videos are separated into short, non-overlapping clips, and document clustering approaches are employed to group clips together. In adaptive fast-forward techniques [15,4], the playback speed of the initial video is adapted based on a given criteria. Among the criteria employed thus far are the similarity of the video to a given query [15] and the amount of temporal information present in the video [4].

Unlike adaptive fast-forward algorithms, video summarization techniques attempt to provide a summary of a video by creating smaller videos containing descriptive sections of the original video. Typically, these techniques employ static representations such as key-frames [3,34], or motion video representations [1,7,8,17–19,21,25]. In [3] frames are clustered, key-frames are selected as the centroids of the clusters, and video shots containing the key-frames are concatenated to form a video summary. Key-frames are extracted and multiple clustering stages are employed to produce a summary in [34].

Several of the summarization algorithms utilize space–time volumes of actions. In [21] space–time "worms" are correlated with a user-specified query to find actions of interest, which are then condensed by optimizing their temporal shift, allowing simultaneous display of multiple instances of relevant activity. In [17,19] activities of objects are represented via space–time tubes, and an energy function is minimized to create a video synopsis containing a stroboscopic effect. In [18] a video summary is generated with minimal length and minimum collision between activities that are found by clustering "tubelets" via their appearance and motion features. Objects are detected and tracked in [8], resulting in "tunnels" that are shifted using a proposed direct shift collision detection algorithm, yielding a video containing multiple, originally temporally disjoint, tunnels appearing simultaneously.

Other summarization methods work without utilizing space–time volumes of actions. In [7] visually informative space–time layers are extracted and packed together so the total amount of information in the output video volume is maximized. The technique for extracting epitomes introduced in [6] was extended to videos in [1], resulting in a smaller video containing many of the spatial and temporal patterns present in the input video. In [25] a patch-based bidirectional similarity is employed to determine if a video summary is both complete and coherent with respect to the video it is summarizing. Finally, ribbons are carved out of videos by minimizing an activity-aware cost function in [12] using a model that tunes the compromise between temporal condensation and anachronism of events.

In this paper, we propose novel techniques to extract the common pathway regions and aggregate behavioral patterns that exist throughout a scene. Unlike the aforementioned approaches, we combine temporal information and spatial constraints with local motion information to emulate the benefits of trajectories (which may difficult to collect). Moreover,

our approaches enable employing computationally efficient clustering algorithms to summarize video of complicated scenes. After extracting the common pathway regions and aggregate behavioral patterns that exist throughout a scene, we demonstrate how our techniques can be exploited to detect anomalies, retrieve video clips of interest, and create adaptive fast-forward videos.

3 Behavioral superpixels

Our first step in summarizing scene behavior is to track moving objects. To track objects, we use a modified Kanade–Lucas–Tomasi (KLT) feature tracker [24], which is able to track hundreds of simultaneously moving feature points in real time. The base tracker employs the OpenCV implementation of the KLT feature tracker. Moreover, the tracker limits features/tracks to moving objects and limits any erroneous drift or feature matching during tracking by only accepting features that lie in a motion mask and move in a continuous direction. We refer to the KLT tracker as a "weak" tracker because multiple short/broken tracklets per target are typically produced. However, we show how our algorithms can accommodate such data. Figure 1 shows example of weak tracks.

When extracting long pathways traversed throughout a scene, it is common to divide scenes into small square cells, where activity patterns are assumed to be more consistent than at the pixel level [30,28]. However, square cells are susceptible to encompassing regions of the scene that exhibit varying activity patterns (e.g., a road and a sidewalk), and can cause a staircase effect on the borders of resulting pathway regions.

As opposed to simply gridding the scene, we partition the scene using a more adaptive approach. Our motivation comes from the approach presented in [20], which seeks to group adjacent pixels that are essentially uniform in color and texture into superpixels. Instead of grouping pixels based on their color and texture, we group adjacent pixels which are essentially uniform in the number and speed of tracks through the pixels at each orientation throughout time. We call the resulting regions (groups) the behavioral superpixels.

To extract the behavioral superpixels of a scene, we first map the tracks into location/orientation states, where track orientations are quantized into eight bins. We then create an activity mask by removing pixels containing negligible activity. Next, we separate the input video into N non-overlapping temporal clips and compute two $8N \times 1$ vectors for each pixel within the activity mask containing temporal traces of the number and average speed of tracks at each orientation, respectively. We then build count-based and speed-based similarity matrices using Gaussian kernels ($W = \exp(-d^2/\sigma^2)$), where the distance d between pixels is computed as the χ^2 statistic between traces, and $\sigma = 0.05 \cdot \max(d)$ for the respective measurements (count or speed). After the two similarity matrices are built, we combine them by computing their point-wise product, which ensures two pixels are only highly similar when both their count and speed traces are similar. Finally, we extract the behavioral superpixels from the combined similarity matrix using the normalized cuts algorithm [23].

4 Extracting common pathway regions

After mapping the observed activities to the behavioral superpixels, we extract the common pathway regions in the scene. In this paper, we define common pathway regions as areas where the temporal traces of the speed and number of tracks are locally similar. Thus, given this definition, entire paths may be composed of one or more pathway regions. For example, the two paths in the 'Y' junction shown in Fig. 2a contain the three pathway regions shown in Fig. 2b. We desire to extract pathway regions instead of extracting entire paths as pathway regions enable a more local, yet descriptive, mapping of tracks (or motion flow), and hence enable instantaneous behavioral analysis.

Even though we employ weak tracks throughout our experiments, we recognize that either motion flow or stronger tracks may be desirable in certain scenarios. Consequently,

Fig. 1 Example tracks for multiple moving objects in an urban environment

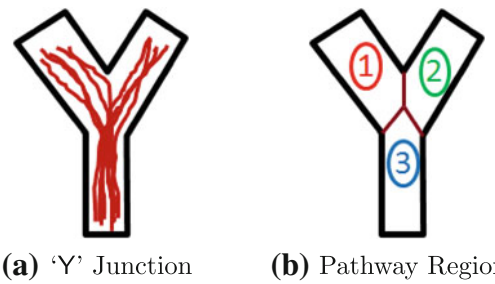

Fig. 2 An example of **a** a 'Y' junction and **b** the corresponding pathway regions

we designed our approach to be applicable when employing motion flow, weak tracks, or strong tracks.

To begin, we separate videos into short, non-overlapping, clips. We then map the track observations (or motion) to superpixel/orientation "states", where we continue to quantize track orientations into eight bins, yielding temporal traces of the count and average speed of tracks for each state.

Since we desire to extract the common pathway regions, we remove states that are not sufficiently traveled. More specifically, we remove states that do not contain at least 5 % of the maximum number of observations in a single state throughout a single clip in at least 2 % of the video clips. Next, we compute a similarity matrix to represent the behavioral similarity of the kept states. In this paper, we define two states as being behaviorally similar if (1) the speed of objects throughout the states are similar, (2) their track count traces are similar, and (3) they are spatially close and oriented in a similar direction. Once the similarity matrix is built, we employ a graph-based clustering algorithm to extract the common pathway regions. We will now explain our approach in creating the behavioral similarity matrix through a series of progressive steps.

4.1 Speed similarity

For a given clip duration T, we compute the proximity of the speed of two tracks using the Mahalanobis distance. More formally, we define the similarity of the speed of tracks between two states x and y as

$$W_s(x, y, T) = \exp\left(-\left(\frac{x_{s_\mu} - y_{s_\mu}}{\min\left(x_{s_\sigma}, y_{s_\sigma}\right)}\right)^2\right), \tag{1}$$

where x_s corresponds to the speed component of state x, and x_{s_μ} and x_{s_σ} are the mean and standard deviation of the average speed of the tracks through x for clips which contain activity, respectively.

4.2 Activity count similarity

For two states to have similar activity count traces, they should be "on" and "off" simultaneously, as well as have similar levels of activity when they are "on". Let I_x be an indicator vector where the ith element of I_x is 1 if there is activity on state x within clip i, and 0 otherwise. We compute the proximity of the activity count traces for x and y for a given T as

$$d_z\left(x\middle|y, T\right) = \frac{\sum_{i=1}^{N_T} I_y(i) \cdot (x_z(i) - y_z(i))^2}{\sum_{i=1}^{N_T} I_y(i)}, \tag{2}$$

where x_z corresponds to the track count component of state x and N_T is the number of video clips given T. Conceptually, $d_z\left(x\middle|y, T\right)$ is the average squared "intensity" difference between states x and y when there is activity on state y. We then compute the similarity of the activity count traces for x and y for a given T as

$$W_z(x, y, T) = \exp\left(-\max\left(\frac{d_z\left(x\middle|y, T\right)}{y_{z_\sigma}^2}, \frac{d_z\left(y\middle|x, T\right)}{x_{z_\sigma}^2}\right)\right), \tag{3}$$

where x_{z_σ} is the standard deviation of the count of tracks through x across the clips where activity exists on x.

4.3 Incorporating temporal shifts

Intuitively, W_z will be high between two states x and y if the track count within the two states is similar across time (i.e., in each video clip). However, since video clips are short, objects may not traverse the entirety of a pathway region within a single video clip. For example, consider the two-clip case where a single object moves across the scene. In this scenario, there will be no similarity in W_z between the states the object traverses in the first clip with those it traverses in the second clip.

To rectify this situation, we incorporate a temporal shift into the activity count proximity measurement shown in Eq. (2). More specifically, for a set of temporal shifts Υ_{xy}, we compute $d_z^{\Upsilon_{xy}}\left(x\middle|y, T\right)$ as

$$d_z^{\Upsilon_{xy}}\left(x\middle|y, T\right) = \frac{\min_{\tau\in\Upsilon_{xy}}\left(\sum_{i=1}^{N_T} I_y(i) \cdot (x_z(i-\tau) - y_z(i))^2\right)}{\sum_{i=1}^{N_T} I_y(i)}. \tag{4}$$

Thus, $d_z^{\Upsilon_{xy}}\left(x\middle|y, T\right)$ quantifies the activity count proximity between x and y using the temporally shifted trace from x that is most similar to the trace from y, where Υ_{xy} contains the set of feasible temporal shifts.

Intuitively, the set of temporal shifts Υ_{xy} for two states x and y should reflect the location and orientation of the two states. For example, if x and y are both orientated upward, and x is located directly below y, then x can be envisioned as flowing into y. Hence, the activity of x and y should either be similar in clip i, or the activity of x in a clip $i - \tau$ prior to clip i should be similar to the activity of y in clip i. More specifically, if a line through y is drawn perpendicular to the orientation of y, then states on the opposite side of the line as the orientation vector should have values of $\tau \geq 0$ (i.e., compare current and previous clips). Conversely, states on the same side of the line as the orientation vector should have values of $\tau \leq 0$ (i.e., compare current and future clips). In this paper, we only consider first order temporal shifts ($|\tau| \leq 1$), as they are sufficient to build proper similarities between local states. We then rely on the clustering algorithm to propagate local similarities throughout the pathway regions. Thus,

$\Upsilon_{xy} = \{0, 1\}$ if x flows into y, and $\Upsilon_{xy} = \{-1, 0\}$ if y flows into x.

The temporal shifts are then incorporated into the activity trace similarity by substituting $d_z^{\Upsilon_{xy}}\left(x\middle|y, T\right)$ for $d_z\left(x\middle|y, T\right)$ in Eq. (3), yielding $W_{z^{\Upsilon_{xy}}}(x, y, T)$.

4.4 Incorporating multiple temporal scales

In crowded scenes with widespread activity, it is conceivable that the activity traces will appear similar for states that do not belong to the same path for a given clip duration T. This problem could be alleviated by choosing an appropriate clip duration based on the scene, but this requires a priori knowledge of how objects move throughout the scene. However, since we are automatically analyzing how objects move throughout the scene, this knowledge does not exist, and it is difficult to specify a single correct clip duration for a scene.

Intuitively, if two states truly belong to the same pathway region, then their activity traces should be similar throughout multiple clip durations. Conversely, if two states do not belong to the same pathway region, we would expect their traces to be dissimilar throughout certain clip durations. Motivated by scale-space techniques [10,31] and pyramid match kernels [2,9], we propose a pyramid-based approach to compute the similarity between states at multiple temporal scales (clip durations).

For an L-level pyramid, we examine the similarity between the speed and count of tracks through two states using clips of duration $T = \left\{T_0, 2 \cdot T_0, \ldots, 2^{L-1} \cdot T_0\right\}$. Since the likelihood that activity on nearby states corresponds to objects traversing the same pathway region increases as clip duration decreases, we weight the similarity of activity traces for short clip durations more highly. More specifically, the jth level of the pyramid is weighted as

$$\omega_j = \begin{cases} \frac{1}{2^{j+1}} & j = 0, 1, \ldots, L-2 \\ \frac{1}{2^j} & j = L-1 \end{cases}, \tag{5}$$

where $j = 0$ corresponds to the finest layer of the pyramid (i.e., a clip duration of $T = T_0$). We then compute the speed and intensity-based similarity values for two states in a weighted fashion.

For an L-level pyramid, the weighted speed and activity count similarities are computed as

$$W_s^L(x, y, T_0) = \sum_{j=0}^{L-1} \omega_j \cdot W_s\left(x, y, 2^j \cdot T_0\right) \tag{6}$$

and

$$W_{z^{\Upsilon_{xy}}}^L(x, y, T_0) = \sum_{j=0}^{L-1} \omega_j \cdot W_{z^{\Upsilon_{xy}}}\left(x, y, 2^j \cdot T_0\right), \tag{7}$$

respectively, where T_0 is the duration of the shortest clip. Using the two similarity matrices, we define the behavioral proximity of two states as

$$d_b^L(x, y, T_0) = 1 - \left(W_s^L(x, y, T_0) \cdot W_{z^{\Upsilon_{xy}}}^L(x, y, T_0)\right). \tag{8}$$

We set $L = 4$ and $T_0 = 1$ s. throughout our experiments.

4.5 Pathway region extraction

Using the behavioral proximity defined in Eq. (8), we compute the proximity between states and form a state-wise behavioral proximity matrix. We then input the behavioral proximity matrix into the algorithm to build automatically tuned similarity matrices based on local point distributions presented in [27]. Next, we remove connections between states in the resulting similarity matrix that are not spatially close and oriented in a similar direction. Throughout our experiments, we define x and y to be spatially close if the minimum infinity norm between pixels from x and y is at most 20 pixels, and define x and y to be oriented in a similar direction if their orientations are within 45° (i.e., one quantization bin).

We cluster the final similarity matrix using the graph-based hierarchical clustering algorithm presented in [27], which merges clusters together in a manner that is analogous to solving a jigsaw puzzle piece-by-piece by utilizing local connection strengths. Motivated by the common Eigen-gap approaches [13], we automatically determine which layer of the hierarchy to keep using a hysteresis thresholding approach. Namely, we use the cost incurred to merge the clusters at a given level as a "score" for that level. We then compute the percent change between scores of successive levels, find the level $l_{\min}$ with the smallest number of clusters that results in at least a 20 % change, and search for the highest level l^* (containing more clusters than $l_{\min}$) such that there is at least a 20 % change between score values for all levels between $l_{\min}$ and l^*. Intuitively, a high percent change means it was costly to merge the clusters at a given level. This approach provides a method to automatically choose which hierarchy level to maintain.

5 Extracting behavioral patterns

In addition to extracting common pathway regions, another important task in visual surveillance is determining the behavioral patterns that appear throughout the day (e.g., once or periodic). In this section, we describe approaches to group video clips together based on their aggregate scene behavior. In each of the proposed methods, we focus solely on the location and count of tracks throughout the scene, and do not

incorporate the speed of the tracks. However, the speed of the tracks could be included in our approaches, if desired.

5.1 Instantaneous proximity

Letting P be the number of pathway regions, we represent each clip via a $P \times 1$ vector, whose elements contain the count of tracks along each pathway region within the clip. The video clips can then be grouped by clustering similar vectors together. Intuitively, multiple proximity measurements can be employed to group the video clips, where varying measurements utilize different properties to determine the similarity between two clips.

Throughout this paper, we will explore grouping clips together using two different proximity metrics. First, we will employ the χ^2 statistic

$$d_{\chi^2}(a,b) = \frac{1}{2}\sum_{i=1}^{n}\frac{(a_i - b_i)^2}{(a_i + b_i)}, \tag{9}$$

a popular statistical measurement which weights the individual dimension differences by the sum of the dimension values.

When employed to measure the proximity of video clips, the χ^2 statistic will focus more on the total amount of activity (aggregate) throughout the scene than it does on specifically where the activity occurs. The differences between dimensions with larger values are penalized so they do not dominate the proximity measurement (as is the case with Euclidean distance). For example, consider a scenario with three video clips of a scene with two pathway regions X and Y. Furthermore, let the count of the tracks on pathway regions X and Y be 10 and 0, 1 and 0, and 0 and 1, throughout the first, second, and third clips, respectively. Denoting the activity vector for the ith video clip as v_i, the proximity between the three pairwise combinations are $d_{\chi^2}(v_1, v_2) = 3.68$, $d_{\chi^2}(v_1, v_3) = 5.5$, and $d_{\chi^2}(v_2, v_3) = 1$. Thus, even though v_2 and v_3 do not contain activity on the same pathway regions, v_2 will be considered closer to v_3 than it is to v_1 (which has activity on the same pathway region as v_2) since the total activity between v_2 and v_3 is more similar than the total activity between v_2 and v_1.

In certain scenarios, it may be more desirable to focus more on where activity occurs throughout the scene than the total amount of activity within the scene. For example, in the aforementioned example with three video clips from a scene with two pathway regions, it may be desirable to group v_1 and v_2 together, since they contain activity on the same pathway region, and keep v_3 in a group by itself, since it contains activity on a different pathway region. To accomplish this task, we normalize the activity vector for each clip (by dividing by the total activity within the clip), and measure the proximity of two clips using the Jeffrey divergence

$$d_J(a,b) = \sum_i a_i \log\left(\frac{a_i}{(a_i+b_i)/2}\right) + b_i \log\left(\frac{b_i}{(a_i+b_i)/2}\right), \tag{10}$$

a symmetric metric to compare distributions that employs the KL divergence.

Clustering the clips together using the Jeffrey divergence results in groups whose clips contain similar distributions of activity throughout the clips. However, it is often desirable to separate periods of high traffic from those of low traffic. Thus, after grouping clips using the Jeffrey divergence, we further divide each cluster based on the total amount of activity within the clips. To accomplish this task, for each cluster, we model the count of tracks throughout the clips within the cluster via a Gaussian mixture model, where the number of Gaussians is selected as to minimize the Bayesian Information Criterion. We then assign each clip the label of the Gaussian with the highest likelihood.

5.2 Pyramidal proximity

While it is common to treat each clip as an independent entity when grouping video clips together [30], the activity throughout video clips is typically not independent. Instead, the activity throughout a video clip is often impacted by the activity within the clips directly before and/or impacts the activity within the clips directly after. Furthermore, many scenes are periodic in nature and employing temporal information may reduce spurious clip assignments caused by slight activity fluctuations or the temporal quantization. In this section, we present a method which computes the proximity between video clips using a pyramidal approach that incorporates information from clips temporally surrounding the two clips of interest.

For an L-level pyramid, the proximity between video clips a and b is computed as

$$d_v^L(v_a, v_b) = \sum_{j=0}^{L-1}\omega_j \cdot d\left(v_a^j, v_b^j\right), \tag{11}$$

where ω_j is computed as in Eq. (5), v_a^j is formed by concatenating v_{a-j} through v_{a+j} (the activity vectors for clips $a-j$ through $a+j$, and $d(\cdot)$ is either the χ^2 statistic or Jeffrey divergence.

5.3 Behavioral pattern extraction

Once the proximity values between video clips are computed, we build an automatically tuned similarity matrix to represent the similarity between clips and extract the behavioral patterns by clustering the similarity matrix using the approaches presented in [27] as described in Sect. 4.5.

6 Applications

The behavioral superpixels, common pathway regions, and various proximity measurements presented can be employed in several useful monitoring applications. In this section, we present approaches to detect video clips containing anomalous behavior, retrieve video clips similar to a given query, and generate adaptive fast-forward videos which adjust the playback speed of videos clips based on the behavior they contain.

6.1 Anomaly detection

A common task in visual surveillance is to detect the anomalous behavior that occurs within a video. Although widely used throughout the surveillance community, the term "anomalous" is rather vague, and can refer to multiple concepts. In this section, we present methods to detect video clips which are anomalous based on four different criteria. Namely, we propose methods to detect video clips containing rare activity, clips where the amount of activity or the speed of objects on a pathway region are abnormal, and clips where the aggregate scene behavior is abnormal. In all cases, our anomalies are purely statistical and do not necessarily correspond to an alarming event.

6.1.1 Rare activity

To detect video clips containing rare activity, we utilize the uncommon states that do not belong to any pathway region (i.e., those which do not contain at least 5 % of the maximum number of tracks observed in a single state throughout a single clip in at least 2 % of the video clips). For each clip, we compute the rarity of the activity throughout the clip as the sum of the likelihood of the activity along the uncommon states throughout the clip. For each state, we compute the likelihood as a function of (1) the count of tracks through the state; (2) the directionality of tracks through the behavioral superpixel to which the state belongs; and (3) how popular the state is throughout the video.

Since the behavioral superpixels can contain varying number of pixels, we normalize the count of tracks through each state by the number of pixels within the behavioral superpixel to which the state belongs. Thus, we represent the count of tracks through a state using the density of tracks throughout the state. We will represent the density through state x during video clip i as $x_{z_d}(i)$.

Since we are employing a KLT-based tracker, we expect the orientation of tracks to be somewhat noisy. Furthermore, since we quantize the orientations into a small number of angular bins, we expect the noise to cause the orientation of some tracks to be quantized into bins adjacent to the true underlying orientation of their corresponding objects. Moreover, we expect quantization errors to be more likely for states belonging to superpixels where the activity is more uniform across bins. To account for these expected errors, we weight each state based on the activity in all of the other states belonging to the same superpixel. Let state x belong to superpixel A, and x_{z_d} be the intensity of tracks throughout x across the entire video. We define the directionality weight for x to be

$$x_\theta = 1 - \frac{x_{z_d}}{\max_{y \in A} y_{z_d}}. \tag{12}$$

Finally, we weight how popular each state is throughout the video relative to the other uncommon states. Mathematically, we define the popularity weight as

$$x_\pi = 1 - \frac{x_{z_d}}{\max_{y \in \mathcal{X}} y_{z_d}}, \tag{13}$$

where $\mathcal{X}$ is the set of all uncommon states.

Using the aforementioned measurements, we compute the amount of rare activity in the ith video clip as

$$\zeta(i) = \frac{\sum_{x \in \mathcal{X}} x_{z_d}(i) \cdot x_\theta \cdot x_\pi}{\sum_{x \in \mathcal{X}} I(x_{z_d}(i))}, \tag{14}$$

where $I(x_{z_d}(i))$ is a binary value which is 1 if $x_{z_d}(i) > 0$, and 0 otherwise. Video clips with larger values of ζ contain activity that is more rare. It is important to note that, since our method employs solely the uncommon states, we are able to detect video clips containing rare activity even when they also contain large amounts of common activity (i.e., the rare activity throughout a clip is not masked by the common activity).

6.1.2 Abnormal activity on pathway regions

To detect abnormal behavior on the individual pathway regions, for each pathway region, we model the count and speed of tracks along the region using a Gaussian Mixture Model (GMM), where the number of Gaussians in the model is chosen as to minimize the Bayesian Information Criterion. We then define the speed and count likelihoods for each clip as the minimum of all of the individual pathway region likelihoods for the respective properties. Video clips which have the lowest likelihoods are then tagged as having the most anomalous activity along a pathway (a threshold could also be set).

6.1.3 Abnormal aggregate scene behavior

To detect if the aggregate scene behavior is abnormal, we employ the various proximity measurements used to group the video clips. More specifically, for a given proximity metric, we define those clips having the farthest nearest neighbors to be the most abnormal (again, a threshold could also be set).

6.2 Video clip retrieval

The next application we explore is retrieving video clips that are similar to a given query (e.g., traffic is flowing in a specific location/direction). Motivated by the approach in [30], we created a graphical user interface which enables users to mask out regions of interest within a scene, and define their corresponding orientations. We then create a query vector where all pathway regions which overlap the specified areas of interest are given a uniform weight which sums to 1, and pathway regions which do not overlap the specified areas of interest are given zero weight. Finally, we compute the similarity of all of the video clips to the query vector using the KL divergence, and return the clips yielding the lowest KL divergence values.

In addition to searching for single video clips which resemble a given query, it is conceivable that a user may wish to retrieve sections of the video where sequences of behavioral patterns occur. For example, if monitoring a structured traffic environment where traffic usually flows leftward, then rightward, and then vertically, a user may want to find sections of the video where traffic flows rightward and then leftward, as they deviate from the standard cycle.

To retrieve desired sequences of clips, we begin by making a query vector for each pattern using the same approach described above. To explain our approach at retrieving sequences of clips, we will assume the desired sequence is a bigram (i.e., contains two clips). In this scenario, we generate the vectors $q_1(i)$ and $q_2(i+1)$ containing the KL divergence values between the ith and $(i+1)$th video clips and the first and second query vectors, respectively. To ensure the two queries receive equal weight, we scale the values of q_1 and q_2 so they range from 0 to 1. Next, we define the final bigram score for the bigram beginning at clip i using the harmonic mean between the two scores. Thus, the score for the clip i is mathematically defined as

$$\xi(i) = \frac{2 \cdot q_1(i) \cdot q_2(i+1)}{q_1(i) + q_2(i+1)}. \tag{15}$$

The bigrams with the smallest ξ values are then returned. This approach could be extended to longer sequences of behavioral patterns if desired.

6.3 Creating adaptive fast-forward videos

The final application we explore is creating adaptive fast-forward videos. Watching long durations of surveillance video of an area can be a tedious and, given the number of cameras in existence today, an impossible task. In this section, we present methods to summarize videos using adaptive fast-forward techniques based on a given objective function, where the playback speeds of the videos are adjusted such that clips resembling target behaviors are played closer to real time, and clips that do not resemble the target behaviors are played faster than real time. While any given objective function could be employed, throughout our experiments we will focus on adapting the playback speed of the video based on (1) its similarity to a user query and (2) the amount of rare activity it contains.

To compute the playback speed for a clip, we employ a given objective function c, where a higher value of $c(i)$ for the ith clip denotes the clip should be played slower (i.e., closer to real time). First, we scale the values of c so the range of values is between 0 and 1. Then, we compute the playback frame rate for the ith clip as

$$f_i = f_{\min} + (1 - c(i)) \cdot (f_{\max} - f_{\min}), \tag{16}$$

where $f_{\min}$ and $f_{\max}$ are the minimum and maximum desired frame rates for the summary video, respectively. While we employ a linear function to compute the playback speed for each clip, a sigmoid or other functions could also be employed.

7 Experiments

We test our approaches on the three complex urban scenes shown in Fig. 3. The Roundabout and Junction scenes are from [11], while the Smith scene is from our campus area camera network. The video from the Roundabout scene is 62 min long and depicts a roundabout where traffic enters from either the left, bottom, or top of the image, and exits in the bottom-left, top, and right of the image. The video from the Junction scene is 50 min long and contains bidirectional traffic moving vertically and horizontally, along with several pedestrians crossing the roads and walking on the adjacent sidewalks. Both of these videos have resolutions of 360×288 pixels. Finally, the video from the Smith scene has a resolution of 704×480 pixels, is 45 min long, and contains a one-way road with two less traveled connecting roads, all surrounded by walkways containing large amounts of pedestrian traffic.

Using the KLT-based tracker described in Sect. 3, we extract behavioral superpixels that have an average area of ≈ 150 pixels and contain pixels which exhibit single behavioral patterns (as opposed to square cells that result from simply gridding the scene).

Fig. 3 Images from the Roundabout, Junction, and Smith scenes

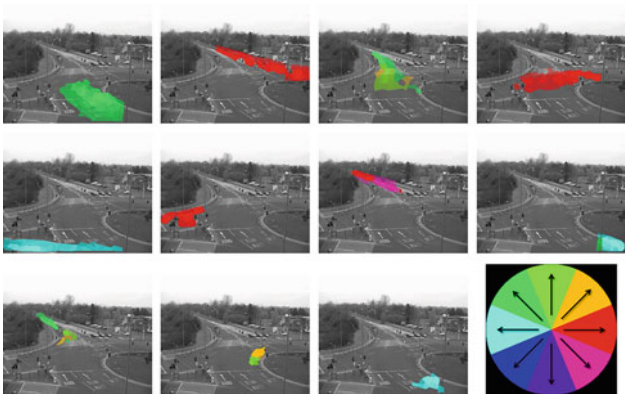

Fig. 4 Extracted pathway regions for the Roundabout scene

7.1 Common pathway regions

Figures 4, 5, 6 show the common pathway regions extracted for the Roundabout, Junction, and Smith scenes. In all three figures, the pathway regions are sorted in descending order based on the popularity of the region throughout time (left to right, top to bottom). Furthermore, the pathway regions are color-coded based on their orientations using the legend in Fig. 4, and the brightness for each state along the pathway region is based on the count of tracks throughout the state.

For the Roundabout scene (Fig. 4), we extract a pathway region for traffic that enters from the bottom of the scene and quickly exits at the bottom-left. Furthermore, both the horizontal and vertical roads are split into multiple pathway regions, where the breaks generally coincide with traffic lights. This result is desired, as vehicles often wait at the traffic lights for multiple clips.

In the Junction scene (Fig. 5), we correctly separate the horizontal pedestrian crosswalks from the pathway regions representing the vehicular motion. Furthermore, we correctly divide the vertical road with upward motion into two pathway regions, as there is an adjoining road towards the middle of the image whose traffic merges with the traffic already on the road. We also extract separate pathway regions for both turn lanes. Ideally, we would have separated the individual lanes of traffic. However, the combination of perspective effects and similar traffic flow in each lane resulted in our approach having difficulty separating the individual lanes (as would and any other clip-based approach).

For the Smith scene (Fig. 6), we extract long pathway regions for the main road and adjacent sidewalks. Furthermore, we are also able to extract pathway regions for the horizontal sidewalk on the middle-left, the crosswalks towards the middle of the image, and the horizontal motion in the far-field of the scene. In addition to these primary pathway regions, there are also a handful of secondary pathway

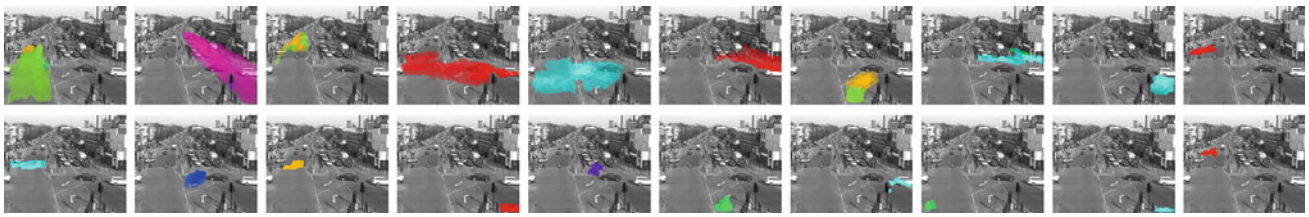

Fig. 5 Extracted pathway regions for the Junction scene

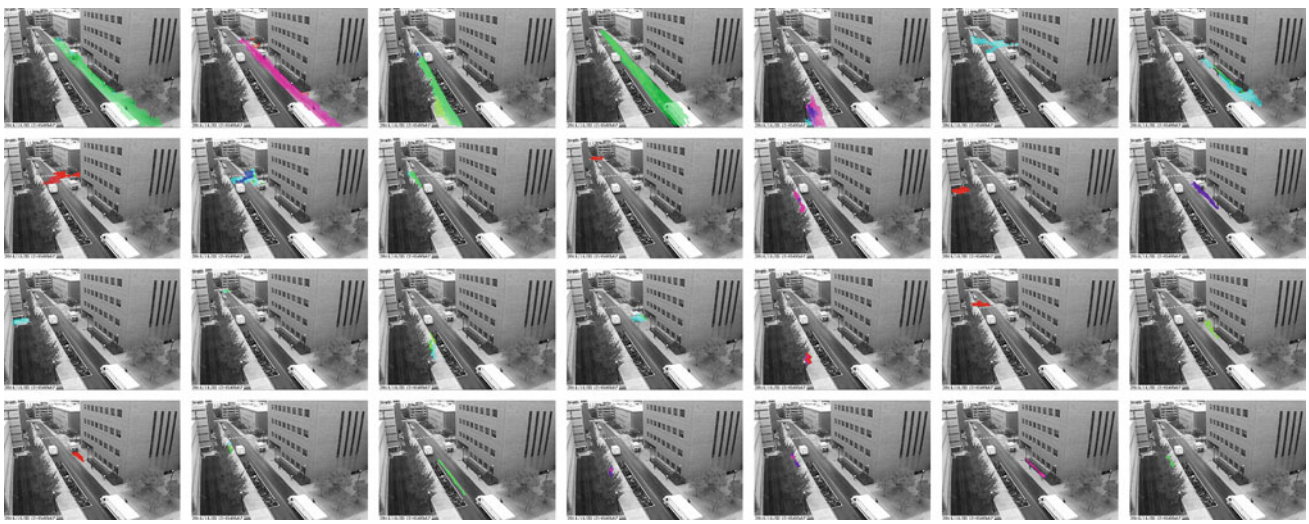

Fig. 6 Extracted pathway regions for the Smith scene

regions which are spatially close to the primary regions, but are composed of states that are less frequently traveled. These secondary regions are partly a result of quantizing the track orientations into a finite set of angular bins and employing imperfect tracks.

7.1.1 Temporal shift and scale comparison and discussion

The above experiments employed our proximity metric defined in Eq. (8), which compares the activity along states at different temporal shifts and scales. However, to our knowledge, existing flow-based algorithms (e.g., [30]) do not include this information. Instead, they simply compare simultaneously occurring activity at different states for a single temporal scale.

Figure 7a shows example states from the Junction scene and the 30 nearest states when using (b) our proximity metric and (c) a traditional proximity metric that does not account for temporal shifts or scales (i.e., our proximity metric with $\tau = 0$ and $L = 1$). The states remain color-coded based on the map in Fig. 4. Based on these results, it is clear that our proximity measurement is better suited for comparing the activity along states than the traditional approaches, which can define states in completely opposite directions as similar.

7.2 Behavioral patterns

We next separate the videos into clips of 5 s. duration and employ the approaches discussed in Sect. 5 to extract the behavioral patterns. Figure 8 shows the behavioral patterns extracted for the Roundabout and Junction scenes using the (top row) instantaneous proximity and (bottom row) pyramidal proximity with $L = 3$ based on the χ^2 statistic. Each scene/proximity measurement combination contains a behavioral pattern image (top-left), timeline image (bottom-left), and representative images from the primary clip within each pattern (right).

Fig. 7 Examples of **a** target states and the 30 nearest neighbors using **b** our proximity measurement and **c** a traditional proximity measurement

The top rows of each behavioral pattern image correspond to the pathway regions and the columns correspond to individual video clips, where the clips are sorted based on the behavioral pattern to which they were assigned. The intensity of an element (i, j) corresponds to the percentage of activity in clip j that occurred on pathway region i (using the same ordering for the pathway regions that they were displayed in throughout Figs. 4 and 5). The bottom row of the behavioral pattern image displays the color-coded label of the pattern to which the clips were assigned. The timeline images depict how the behavioral patterns vary throughout time, where the x and y axes correspond to time and the labels of the patterns to which the clips were assigned, respectively. Finally, representative images from the primary clip within each pattern, where we define the primary clip to be the clip which is most similar to the other clips within its pattern (i.e., has the highest within-class similarity), are shown.

Using the instantaneous proximity based on the χ^2 statistic, the video clips from the Roundabout scene are divided into five patterns, where each group corresponds to activity along different combinations of the pathway regions. Based on the timeline image, the scene appears to exhibit periodic tendencies at the temporal scale examined, which is expected, as the scene contains a road network governed by traffic lights. Like the Roundabout scene, the Junction scene is also highly periodic. The clips from the Junction scene are divided into six patterns. Three of the patterns correspond to clips containing different combinations of upward and downward traffic. There are also patterns which are primarily composed of leftward and rightward traffic, respectively. The remaining pattern is made up of clips which are dominated by vehicles moving from the turn lanes in the center of the scene.

The bottom row of Fig. 8 displays the behavioral patterns extracted from the scenes when employing the χ^2 statistic and a pyramid with $L = 3$ levels to compute the proximity between clips. As when employing instantaneous proximity, there are five patterns extracted from the Roundabout scene. By incorporating temporal information, the clips are now primarily divided into two patterns (traffic flowing upward and traffic flowing both rightward and downward). For the Junction scene, incorporating temporal information reduces the number of patterns extracted from six to three, where the three patterns primarily consist of leftward, vertical, and rightward traffic, respectively. Furthermore, the periodic nature of the Junction scene is even more evident in the timeline image when incorporating temporal information. Overall, when employing the χ^2 statistic, including temporal information into the proximity metric seems to generalize the behavioral patterns, enabling a more succinct summarization.

The top row of Fig. 9 displays the behavioral patterns extracted from the scenes when employing the instantaneous proximity between clips based on the Jeffrey divergence. The Roundabout scene is divided into four patterns. One pattern

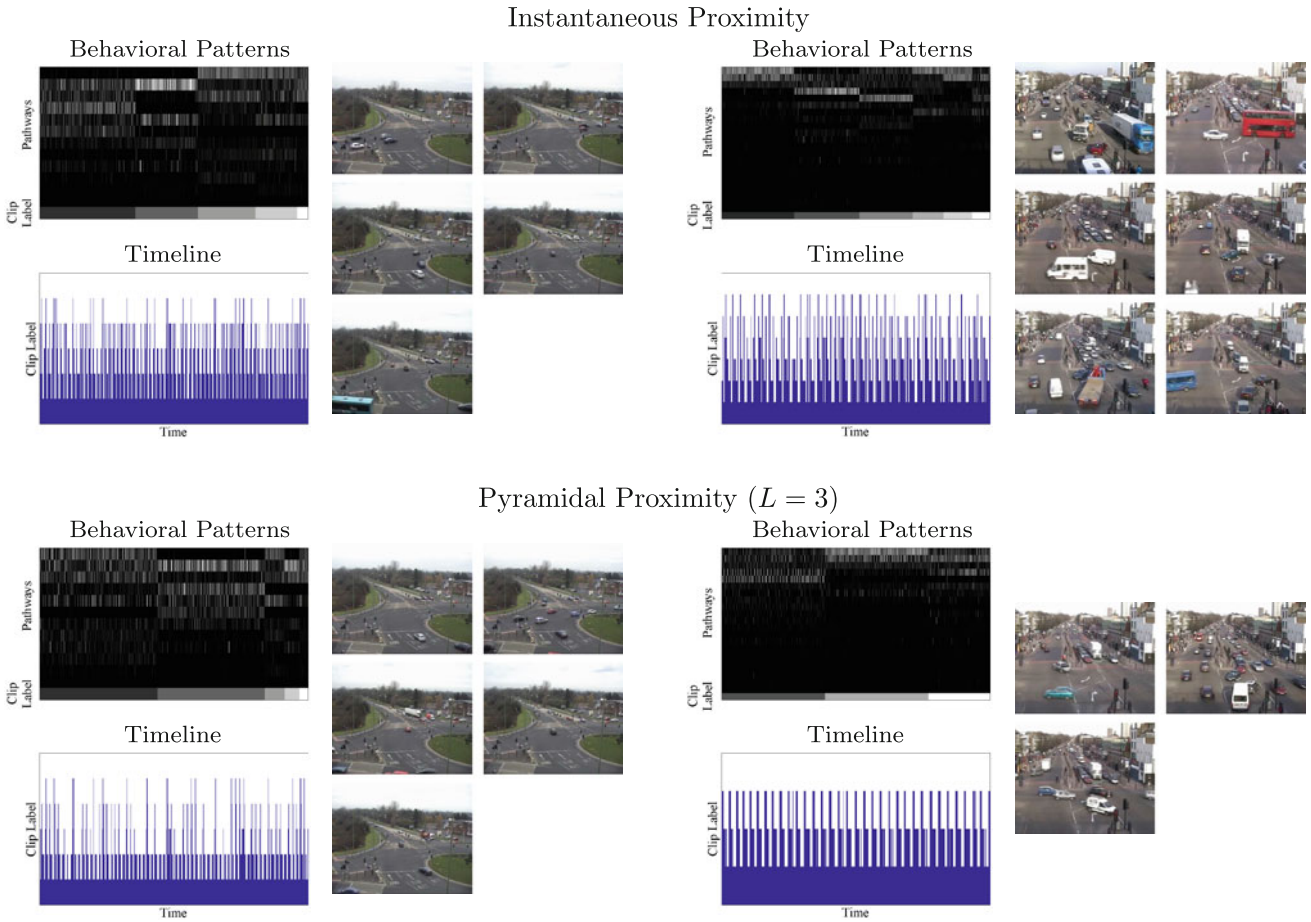

Fig. 8 Behavioral patterns (clips are sorted by the pattern to which they were assigned), corresponding timelines, and representative frames from each pattern for the (*left*) Roundabout and (*right*) Junction scenes using (*top*) instantaneous and (*bottom*) pyramidal ($L = 3$) proximity metrics based on the χ^2 statistic

corresponds to traffic traveling upward and a second pattern represents clips where traffic is traveling rightward across the scene. The remaining two patterns correspond to clips where traffic is traveling downward, with one pattern also containing traffic traveling leftward at the bottom of the screen. The Junction scene contains eight patterns. Five of the patterns correspond to clips which contain varying combinations of vertical traffic. Another pattern contains clips which are dominated by rightward traffic. The remaining two patterns are from clips which are composed primarily of leftward traffic.

The bottom row of Fig. 9 displays the behavioral patterns extracted from the scenes when employing the Jeffrey divergence and a pyramid with $L = 3$ levels to compute the proximity between clips. Nine patterns are extracted from the Roundabout scene. Three of the patterns correspond to clips which contain primarily downward traffic. Another three patterns represent clips that are dominated by upward motion. There is also a pattern for clips that are mainly composed of rightward motion. The remaining two patterns contain clips with leftward traffic at the bottom of the scene. The Junction scene is divided into ten patterns. Eight of the patterns contain clips which are composed of various combinations of activities that all contain vertical traffic. The remaining two patterns are from clips dominated by leftward and rightward traffic, respectively. Since the Jeffrey divergence is a distributional approach, incorporating temporal information tends to result in more patterns being extracted.

7.2.1 Discussion

In the previous experiments, we employed vectors where each element represented the number of tracks on a pathway region. However, since pathway regions are of varying sizes, this may cause certain pathway regions to dominate simply because they are much larger. If desired, the activity along each pathway region can be normalized by the number of pixels within the region. In this case, the feature vectors will correspond to the density of tracks along the pathway regions.

Based on the results throughout this section, it is clear that video clips can be grouped in several different ways. For example, the χ^2 statistic will focus more on the total amount

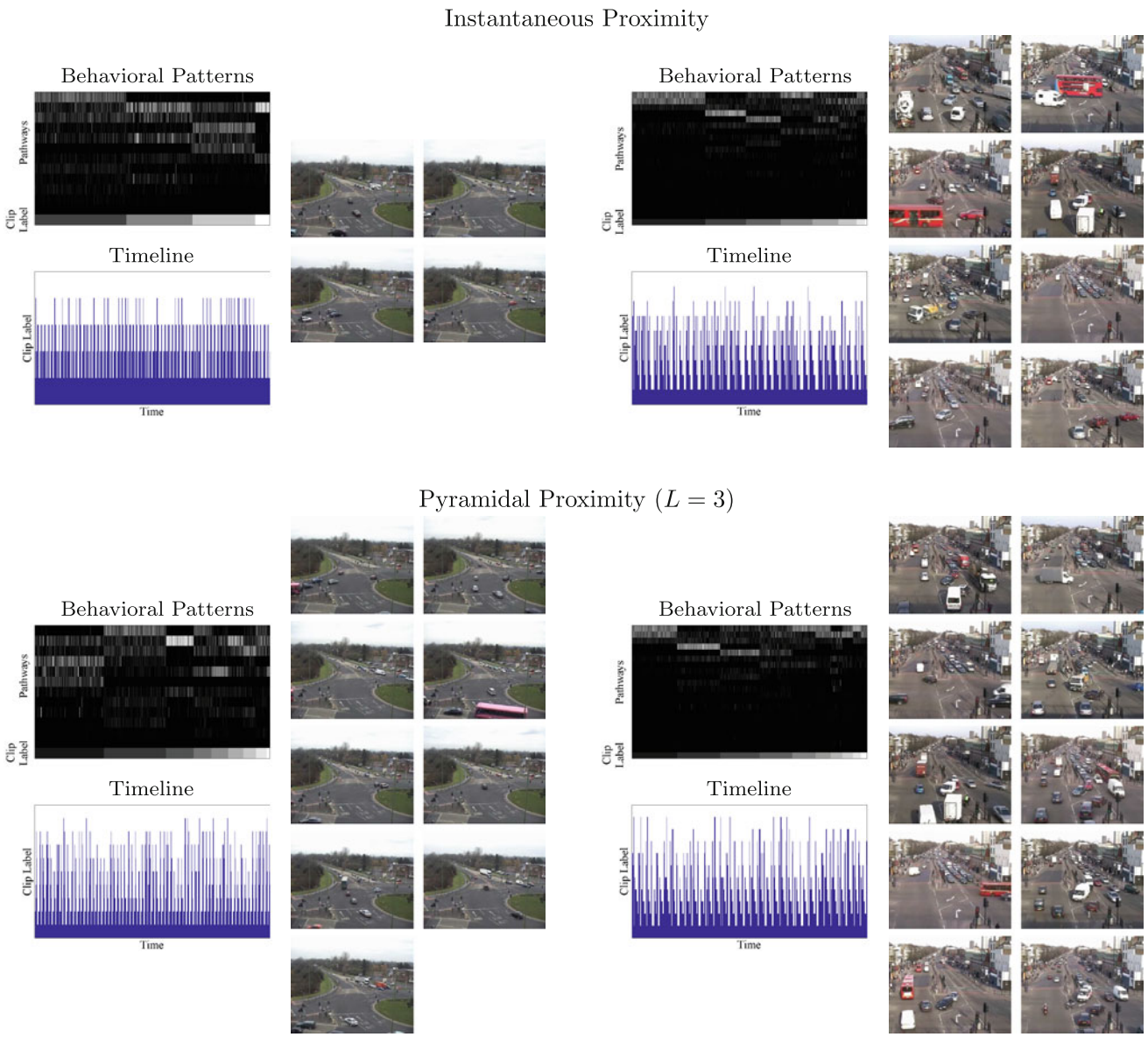

Fig. 9 Behavioral patterns (clips are sorted by the pattern to which they were assigned), corresponding timelines, and representative frames from each pattern for the (*left*) Roundabout and (*right*) Junction scenes using (*top*) instantaneous and (*bottom*) pyramidal ($L = 3$) proximity metrics based on the Jeffrey divergence

of activity throughout the scene than it does on where the activity occurs. Conversely, the Jeffrey divergence focuses more on where activity occurs throughout the scene than the total amount of activity within the scene, as it requires normalizing the activity along each pathway region by the total activity throughout the scene. Thus, it is important to ensure the proximity measurement employed matches the desired objective when grouping clips.

7.3 Anomaly detection

Once the pathway regions and various proximity measurements are computed, they can be employed in several surveillance monitoring applications. The first applications we examine are the anomaly detection techniques described in Sect. 6.1.

7.3.1 Rare anomalies

As described in Sect. 6.1.1, clips containing rare activity can be detected by examining the activity along states that do not belong to any of the pathway regions. The top of Fig. 10 shows images from clips of the Junction scene which contain rare activity, where the highlighted areas correspond to the locations where rare activity is occurring. The first and second images correspond to clips where a vehicle is traveling

Fig. 10 Images from video clips from the (*top*) Junction and (*bottom*) Smith scenes that have rare anomalies. The superpixels containing the rare activities are *highlighted*

the wrong way down the vertical road on the right. The third image is from a clip where a pedestrian crosses the street in the far-field. In the fourth clip there is a vehicle turning left higher in the scene than where most vehicles turn left. Finally, in the last image there is a vehicle which enters from a side street and directly crosses three lanes of traffic.

The bottom of Fig. 10 shows images from clips from the Smith scene which contain rare activity. In the second image there is a bicycle traveling the wrong way down a one-way street. In the remaining images the corresponding clips contain pedestrians crossing the street. Furthermore, the last image depicts a scenario where there are multiple rare activities within the same video clip.

7.3.2 Anomalies on individual pathway regions

As discussed in Sect. 6.1.2, models for the count and speed of tracks along individual pathway regions can be employed to detect anomalies on individual regions. Figure 11 displays images from clips from the Smith scene where the count of tracks along pathways is unlikely. In each image, the pathway region on which the unlikely activity exists is highlighted. The clips containing the images on the left and right are unlikely, as they contain a large number of pedestrians traversing the highlighted walkways. The clip containing the image in the middle is abnormal, as it contains a delivery van driving over a walkway as it is exiting out from under a building.

Fig. 11 Images from video clips from the Smith scene where the total count of tracks on a pathway region is anomalous. The pathway regions containing the anomalous activities are *highlighted*

Fig. 12 Images from video clips from the Smith scene where the speed of tracks on a pathway region is anomalous. The pathway regions containing the anomalous activities are *highlighted*

Fig. 13 Images from video clips from the Junction scene whose aggregate behavior is anomalous based on the χ^2 statistic when also considering the two clips before and after the target clip

Figure 12 contains images from clips from the Smith scene where the speed of tracks along a pathway region is unlikely. In all three images, bicycles are traversing pathway regions which are predominantly traveled by pedestrians walking.

7.3.3 Aggregate anomalies

As described in Sect. 6.1.3, the proximity measurements used for grouping video clips can be employed to detect clips containing aggregate anomalies. Figure 13 shows images from video clips from the Junction scene whose aggregate behavior is anomalous based on the pyramidal proximity using the χ^2 statistic. The image on the left corresponds to a clip where vertical traffic must stop to let a fire truck pass through the scene. The image in the middle is from a clip containing a vehicle traveling the wrong way down a street. Finally, in the

clip containing the image on the right, rightward traffic stops for a police car traveling leftward.

When employing the Jeffrey divergence to detect aggregate anomalies in the tested scenes, the most abnormal clips are often those containing a comparatively low amount of total activity, where that activity occurs on less frequently traveled pathway regions (e.g., pedestrians crossing the street in the Junction scene). This is a result of the Jeffrey divergence being a distributional approach.

7.4 Video clip retrieval

The second applications we examine are the video clip retrieval applications described in Sect. 6.2. Figure 14 shows images from the first five clips retrieved from the Junction scene with a query vector corresponding to traffic traveling upwards on the road on the left. Each of the five clips consist of predominantly upwards traffic on the road on the left, as desired.

Figure 15 shows images from the first five bigrams returned from the Junction scene when the user specified a bigram of leftward traffic followed by rightward traffic. Each column corresponds to a different bigram, with the top image being from the first clip in the bigram, and the bottom image being from the second clip. As shown in the figure, each of the returned bigrams correctly exhibit the desired patterns.

7.5 Adaptive fast-forward videos

The final application we explore is creating adaptive fast-forward videos using the technique described in Sect. 7.5. Figure 16 shows the playback speed of the adaptive fast-forward video of the Junction scene when the cost metric c corresponds to the negative KL divergence between the video clip patterns and the initial query described in the previous video retrieval section (upward motion on the left). The periodicity in the Junction scene is evident by the periodic nature of the adaptive frame rate curve.

Fig. 14 Images from the first five clips retrieved from the Junction scene when employing a query vector of traffic traveling upwards on the road on the left

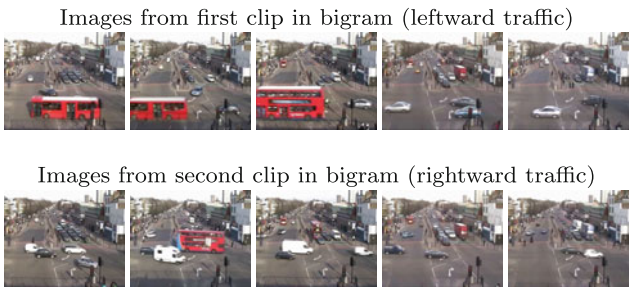

Fig. 15 Images from the first five bigrams retrieved from the Junction scene when employing a query of traffic going leftward and then rightward

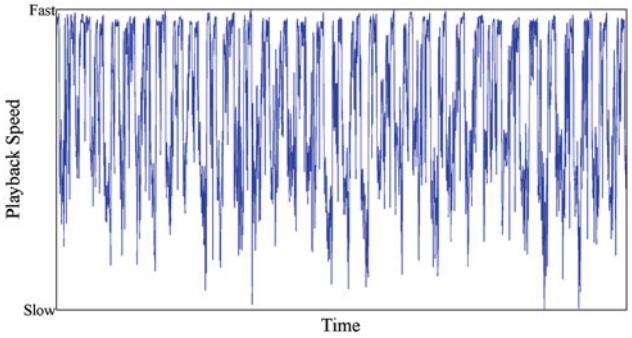

Fig. 16 Playback speed of video clips from the Junction scene when generating adaptive fast-forward videos based on the proximity to a query clip

Figure 17 shows the playback speeds of the adaptive fast-forward videos of the Junction and Smith scenes when the cost metric corresponds to the amount of rare activity in the video (i.e., $c(i) = \zeta(i)$). Based on the frame rate curve from the Junction scene, it is evident that the most rare clip from the scene is much more rare than the majority of the clips. Conversely, in the Smith scene there are several clips that contain similar amounts of rare activity as the most rare clip.

Overall, it is clear that the adaptive fast-forward videos will be much shorter in duration than the original videos and will enable security operators to focus on behavioral patterns of interest. Furthermore, qualitative analysis of the adaptive videos verified their playback was smooth and compelling to watch.

8 Summary

Throughout this paper, we proposed novel techniques to summarize the high-level behavior throughout scenes. First, we presented an approach to compute the proximity between location/orientation states that is applicable for motion flow and trajectories. Furthermore, by employing activity information from multiple temporal shifts and scales, we are able to compare and emulate the benefits of strong tracks for all input types.

Once the proximities between states were computed, we performed graph-based clustering to extract the common pathway regions. Next, we proposed methods to extract the aggregate behavioral patterns using feature vectors containing the activities along pathway regions throughout video clips. We then presented multiple useful mechanisms to fur-

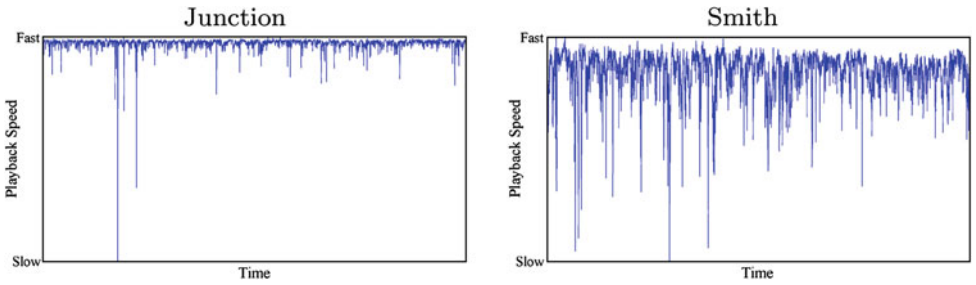

Fig. 17 Playback speed of video clips from the Junction and Smith scenes when generating adaptive fast-forward videos based on the amount of rare activity within the clips

ther exploit our techniques which help alleviate the workload of security operators. Namely, we presented approaches to detect anomalies, retrieve video clips containing behaviors of interest, and create adaptive fast-forward videos which adjust the playback speed of clips based on their behavior content. We tested our approaches on videos from multiple complicated urban scenes containing both pedestrian and vehicular traffic, and showed our methods were successful in summarizing their behavior.

Acknowledgments This research was supported in part by the Air Force Research Laboratories under contract No. FA8650-07-D-1220.

References

1. Cheung, V., Frey, B.J., Jojic, N.: Video epitomes. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2005)
2. Grauman, K., Darrell, T.: The pyramid match kernel: discriminative classification with sets of image features. In: Proceedings of IEEE International Conferenc on Computer Vision (2005)
3. Hanjalic, A., Zhang, H.: An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. IEEE TCSVT **9**(8), 1280–1289 (1999)
4. Hferlin, B., Hferlin, M., Weiskopf, D., Heidemann, G.: (2010) Information-based adaptive fast-forward for visual surveillance. Multimedia Tools Appl. **55**(1), 1–24
5. Hospedales, T., Gong, S., Xiang, T.: A markov clustering topic model for mining behaviour in video. In: Proceedings of the IEEE International Conference on Computer Vision (2009)
6. Jojic, N., Frey, B.J., Kannan, A.: Epitomic analysis of appearance and shape. In: Proceedings of IEEE International Conference on Computer Vision (2003)
7. Kang, H.W., Chen, X.Q., Matsushita, Y., Tang, X.: Space-time video montage. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2006)
8. Kasamwattanarote, S., Cooharojananone, N., Satoh, S., Lipikorn, R.: Real time tunnel based video summarization using direct shift collision detection. In: Advances in Multimedia Information Processing—PCM 2010, vol 6297, pp 136–147 (2010)
9. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2006)
10. Leung, Y., Zhang, J.S., Xu, Z.B.: Clustering by scale-space filtering. IEEE Trans. Pattern Anal. Mach. Intell. **22**(12), 1396–1410 (2000)
11. Li, J., Gong, S., Xiang, T.: Scene segmentation for behaviour correlation. In: Proceedings of the European Conference on Computer Vision (2008)
12. Li, Z., Ishwar, P., Konrad, J.: Video condensation by ribbon carving. IEEE Trans. Image Proc. **18**, 2572–2583 (2009)
13. Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**, 395–416 (2007)
14. Makris, D., Ellis, T.: Path detection in video surveillance. Image Vis. Comput. **20**, 895–903 (2002)
15. Petrovic, N., Jojic, N.: Adaptive video fast forward. Multimedia Tools Appl. **26**(2), 327–344 (2005)
16. Pop, I., Scuturici, M., Miguet, S.: Common motion map based on codebooks. In: 5th International Symposium, ISVC 2009, pp. 1181–1190. Las Vegas, NV, USA (2009)
17. Pritch, Y., Rav-Acha, A.: Nonchronological video synopsis and indexing. IEEE Trans. Pattern Anal. Mach. Intell. **30**, 1971–1984 (2008)
18. Pritch, Y., Ratovich, S., Hendel, A., Peleg, S.: Clustered synopsis of surveillance video. Advanced Video and Signal Based Surveillance (2009)
19. Rav-Acha, A., Pritch, Y., Peleg, S.: Making a long video short: Dynamic video synopsis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2006)
20. Ren, X., Malik, J.: Learning a classification model for segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2003)
21. Rodriguez, M.: CRAM: compact representation of actions in movies. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2010)
22. Saleemi, I., Shafique, K., Shah, M.: Probabilistic modeling of scene dynamics for applications in visual surveillance. IEEE Trans. Pattern Anal. Mach. Intell. **31**(8), 1472–1484 (2009)
23. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern. Anal. Mach. Intell. **22**(8), 888–905 (2000)
24. Shi, J., Tomasi, C.: Good features to track. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (1994)
25. Simakov, D., Caspi, Y., Shechtman, E., Irani, M.: Summarizing visual data using bidirectional similarity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2008)
26. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 747–767 (2000)
27. Streib, K., Davis, J.W.: Improving graph-based clustering via Ripley's K-function and local connection merging. In: Review in process, technical report pending (2012)
28. Streib, K., Davis, J.W.: Extracting pathlets from weak tracking data. Advanced Video and Signal Based Surveillance (2010)
29. Wang, X., Tieu, K., Grimson, W.E.L.: Learning semantic scene models by trajectory analysis. In: Proceedings of the European Conference on Computer Vision (2006)
30. Wang, X., Ma, X., Grimson, E.: Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models. IEEE Trans. Pattern. Anal. Mach. Intell. **31**(3), 539–555 (2009)

31. Wilson, R., Spann, M.: A new approach to clustering. Pattern Recognit. **23**(12), 1413–1425 (1990)
32. Wang, X., Ma, K.T., Ng, W.G., Grimson, W.E.L.: Trajectory analysis and semantic region modeling using a nonparametric Bayesian model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2008)
33. Yang, Y., Liu, J., Shah, M.: Video scene understanding using multiscale analysis. In: Proceedings of the IEEE International Conference on Computer Vision (2009)
34. Zhu, X., Wu, X., Fan, J.: Exploring video content structure for hierarchical summarization. Multimedia Syst. **10**(3), 98–115 (2004)

Author Biographies

Kevin Streib received his Ph.D. (2012), M.S. (2011), and B.S. (2006) degrees in Electrical and Computer Engineering from The Ohio State University. His research interests include computer vision and machine learning.

James W. Davis is a Full Professor of Computer Science and Engineering at Ohio State University. He received his Ph.D. from the Massachusetts Institute of Technology in 2000. His research specializes in computer vision approaches to video surveillance and human activity analysis.