A Data-Driven Approach for Event Prediction

Jenny Yuen and Antonio Torralba

CSAIL MIT
{jenny,torralba}@csail.mit.edu

Abstract. When given a single static picture, humans can not only interpret the instantaneous content captured by the image, but also they are able to infer the chain of dynamic events that are likely to happen in the near future. Similarly, when a human observes a short video, it is easy to decide if the event taking place in the video is normal or unexpected, even if the video depicts a an unfamiliar place for the viewer. This is in contrast with work in surveillance and outlier event detection, where the models rely on thousands of hours of video recorded at a single place in order to identify what constitutes an unusual event. In this work we present a simple method to identify videos with unusual events in a large collection of short video clips. The algorithm is inspired by recent approaches in computer vision that rely on large databases. In this work we show how, relying on large collections of videos, we can retrieve other videos similar to the query to build a simple model of the distribution of expected motions for the query. Consequently, the model can evaluate how unusual is the video as well as make event predictions. We show how a very simple retrieval model is able to provide reliable results.

1 Introduction

If we are told to visualize a street scene, we can imagine some composition with basic elements in it. Moreover, if we are asked to imagine what can happen in it, we might say there is a car moving through a road, being in contact to the ground and preserving some velocity and size relationships with respect to other elements in the scene (say a person or a building). Even when constrained by its

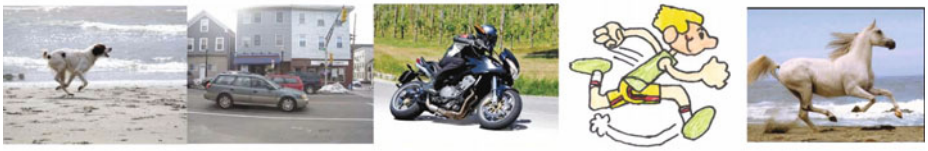

Fig. 1. What do these images have in common? They depict objects moving towards the right. These images do not contain motion cues such as temporal information or motion blur. The implied motion is known because we can recognize the image content and make reliable predictions what would occur if these were movies playing.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 707–720, 2010.

composition (e.g. when being shown a picture of it) we can predict things like an approximate speed of the car, and maybe even its direction (see fig. 1). Human capacity for mental imagery and story telling is driven by the years of prior knowledge we have about our surroundings. Moreover, it has been found that static images implying motion are also important in visual perception and are able to produce motion after-effects [1] and even activate motion sensitive areas in the human brain [2]. As a consequence, the human visual system is capable of accurately predicting plausible events in a static scene (or future events in a video sequence) as well as is finely tuned to flag unusual configurations or events.

Event and action detection are well-studied topics in computer vision. Several works have proposed models to study, characterize, and classify human actions ranging from constrained environments [3,4] to actions in the wild such as TV shows, sporting events, and cluttered backgrounds [5,6]. In this scenario, the objective is to identify the action class of a previously unknown query video given a training dataset of action exemplars (captured at different locations). A different line of work is that of event detection for video surveillance applications. In this case, the algorithm is given a large corpus of training video captured at a particular location as input, and the objective is to identify abnormal events taking place in the future in that same scene [7,8,9,10]. Consequently, deploying a surveillance system requires days of data acquisition from the target and hours of training for each new location.

In this paper we look into the problem of generic event prediction for scene instances different from the ones in some large training corpus. In other words, given an image (or a short video clip), we want to identify the possible events that may occur as well as the abnormal ones. We motivate our problem with a parallel to object recognition. Event prediction and anomaly detection technologies for surveillance are now analogous to object instance recognition. Many works in object recognition are moving towards the more generic problem of object category recognition [11,12]. We aim to push the envelope in the video aspect by introducing a framework that can easily adapt to new scene instances without the requirement of retraining a model for each new location. Moreover, other potential applications lie in the areas of video collection retrieval in online services such as YouTube, Vimeo, where video clips are captured in different locations and greatly differ with respect to controlled video sources such as surveillance feeds and tv programming as was pointed out by Zanetti *et al.* [13].

Given a query image, our purpose is to identify the events that are likely to take place in it. We have a rich video corpus with 2401 real world videos acting as our prior knowledge of the world. In an offline stage, we generate and cluster motion tracks for each video in the corpus. Using scene-matching, our system retrieves videos with similar image content. Track information from the retrieved videos is integrated to make a prediction of where in the image motion is likely to take place. Alternatively, if the input is a video, we track and cluster salient features in the query and compare each to the ones in the retrieved neighbor set. A track cluster can then be flagged as unusual if it does not match any in the retrieved set.

2 Related Work

Human action recognition is a popular problem in the video domain. The work by Efros *et al.* [14] learns optical flow correlations of human actions in low resolution video. Schechtman and Irani exploit self similarity correlations in space-time volumes to find similar actions given an exemplar query. Niebles *et al.* [5] characterize and detect human actions under complex video sequences by learning probability distributions of sparse space-time interest points. Laptev *et al.* densely extracts spatio-temporal features in a grid and uses a bag of features approach to detect actions in movies. Messing *et al.* models human activities as mixtures of bags of velocity trajectories, extracted from track data. None of these works study the task of event prediction and are constrained to human actions. Similar in concept to our vision is the work by Li *et al.* [15], where the objective is action classification given an object and a scene . Our work is geared towards localized prediction including trajectory generation, not classification.

Extensive work has also taken place in event and anomaly detection for surveillance applications. A family of works relies on detecting, tracking, and classifying objects of interest and learning features to distinguish events. Dalley *et al.* detect loitering and bag dropping events using a blob tracker to extract moving objects and detect humans and bags. The system idenfifies a loitering event if a person blob does not move for a period of time. Bag dropping events are detected by checking the distance between a bag and a person; if the distance becomes larger than some threshold, it is identified as a dropped bag. A second family of works clusters motion features and learning distributions on motion vectors across time. Wang *et al.* [7] uses a non-parametric Bayesian model for trajectory clustering and analysis. A marginal likelihood is computed for each video clip, and low likelihood events are flagged as abnormal. One common assumption of these methods is that training data for each scene instance where the system will be deployed is available. Therefore, the knowledge built is not transferrable to new locations, as the algorithm needs to be retrained with video feeds from each new location to be deployed.

Numerous works have demonstrated success using a rich databases for retrieving and/or transferring information to queries in both image [16,17,18,19] and video [20,21]. In video applications, Sivic *et al.* [21], proposed a video representation for exemplar-based retrieval within the same movie. Moving objects are tracked and their trajectories grouped. Upon selection of an image crop in some video frame, the system searches across video key frames for similar image regions and retrieves portions of the movie containing the queried object instance. The work proposed by Liu *et al.* [20] is the closest one to our system. It introduces a method for motion synthesis from static images by matching a query image to a database of video clip frames and transferring the moving regions from the nearest neighbor videos (identified as regions where the optical flow magnitude is nonzero) to the static query image. This work constructs independent interpretations per nearest neighbors. Instead, our work builds localized motion maps as probability distributions after merging votes from several nearest neighbors. Moreover, we aim to have a higher level representation where each moving object is modeled as a track blob while [20] generates hypotheses as one motion region per frame. In summary, these works demonstrate the strong potential of data-driven techniques, which to our knowledge no prior work has extended into anomaly detection.

3 Scene-Based Video Retrieval

The objective of this project is to use event knowledge from a training database of videos to construct an event prediction for a given a static query image. To achieve some semantic coherence, we want to transfer event information only from similar images. Therefore, we need a good retrieval system that will return matches with similar scene structures (*e.g.* a picture of an alley will be matched with another alley photo shot with a similar viewpoint) even if the scene instances are different. In this paper we will explore the usage of two scene matching techniques: GIST [22] and spatial pyramid dense SIFT [23] matching. The GIST descriptor encodes perceptual dimensions that characterize the dominant spatial structure of a scene. The spatial pyramid SIFT matching technique works by partitioning an image into subregions and computing histograms of local features at each sub-region. As a result, images with similar global geometric correspondence can be easily retrieved. The advantage of both the GIST and dense SIFT retrieval methods is their speed and efficiency at projecting images into a space where similar semantic scenes are close together. This idea has proven robust in many non-parametric data-driven techniques such as label transfer [17] and scene completion [18] amongst many others. To retrieve nearest videos from a database, we perform matching between the first frame of the video query and the first frame of each of the videos in the database.

4 Video Event Representation

We introduce a system that models a video as a set of trajectories of keypoints throughout time. Individual tracks are further clustered into groups with similar motion. These clusters will be used to represent events in the video.

4.1 Recovering Trajectories

For each video, we extract trajectories of points in the sequence using an implementation of the KLT tracker [24] by Birchfield [25]. The KLT tracking equation seeks the displacement $\mathbf{d} = [d_x, d_y]^T$ that minimizes the dissimilarity amongst two windows, given a point $\mathbf{p} = [x, y]^T$ and two consecutive frames I and J:

$$\varepsilon(w) = \int\int_W [J(\mathbf{p} + \frac{\mathbf{d}}{2}) - I(\mathbf{p} - \frac{\mathbf{d}}{2})]^2 w(\mathbf{p}) d\mathbf{p} \tag{1}$$

where W is the window neighborhood, and $w(\mathbf{d})$ is the weighing function (set to 1). Using a Taylor series expansion of J and I, the displacement that minimizes ε is:

$$\frac{\partial \varepsilon}{\partial \mathbf{d}} = \int\int_{W} [J(\mathbf{p}) - I(\mathbf{p}) + \mathbf{g}^T(\mathbf{p})\mathbf{d}]\mathbf{g}(\mathbf{p})w(\mathbf{p})d\mathbf{p} = 0 \tag{2}$$

where $\mathbf{g} = \left[\frac{\partial}{\partial x}\left(\frac{I+J}{2}\right) \frac{\partial}{\partial y}\left(\frac{I+J}{2}\right)\right]^T$

The tracker finds salient points by examining the minimum eigenvalue of each 2 by 2 gradient matrix. We initialize the tracker by extracting 2000 salient points at the first video frame. The tracker finds the correspondences of the points sequentially throughout the frames in the video. Whenever a track is broken (a point is lost due to high error or occlusions), new salient points are detected to maintain a consistent number of tracks throughout the video. As a result, the algorithm produces tracks, which are sequences of location tuples $\mathbf{T} = (x(t), y(t))_{t \in \mathbf{D}}$ within a duration $\mathbf{D}$ for each tracked point. For more details on the implementation, we refer to the the original KLT tracker paper.

4.2 Clustering Trajectories

Now that we have a set of trajectories for salient points in an image, we proceed to group them at a higher level. Ideally, tracks from the same object should be clustered together. We define the following distance function between two tracks

$$d_{track}(\mathbf{T}_i, \mathbf{T}_j) \equiv \frac{1}{|\mathbf{D}_i \cap \mathbf{D}_j|} \sum_{t \in \mathbf{D}_i \cap \mathbf{D}_j} \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} \tag{3}$$

We use the distance function to create an affinity matrix between tracks and use normalized cuts [26] to cluster them. Each entry of the affinity matrix is defined as $\mathbf{W}_{ij} = \exp(-d_{track}(\mathbf{T}_i, \mathbf{T}_j)/\sigma^2)$. The clustering output will thus be a group label assignment to each track. See fig. 3 for a visualization of the data. Since we do not know the number of clusters for each video in advance, we set a value of 10. In some cases this will cause an over segmentation of the tracks and will generate more than one cluster for some objects.

4.3 Comparing Track Clusters

For each track cluster $\mathbf{C} = \{\mathbf{T}_i\}$, we quantize the instantaneous velocity of each track point into 8 orientations To ensure rough spatial coherency between clusters, we superimpose a regular grid with a cell spacing of 10 pixels on top of the image frame to create a spatial histogram containing 8 sub-bins at each cell in the grid. Let H_1 and H_2 denote the histograms formed by the track clusters $\mathbf{C}_1$ and $\mathbf{C}_2$ such that $H_1(i, b)$ and $H_2(i, b)$ denote the number of velocity points from the first and second track clusters respectively that fall into the bth sub-bin of the ith bin of the histogram, where $i \in \mathbf{G}$ and $\mathbf{G}$ denotes the bins in the grid. We define the similarity between two track clusters as the intersection of their velocity histograms:

$$\mathbf{S}_{clust}(\mathbf{C}_1, \mathbf{C}_2) \equiv \mathbf{I}(H_1, H_2) = \sum_{i \in \mathbf{G}} \sum_{b=1}^{8} min\Big(H_1(i, b), H_2(i, b)\Big) \tag{4}$$

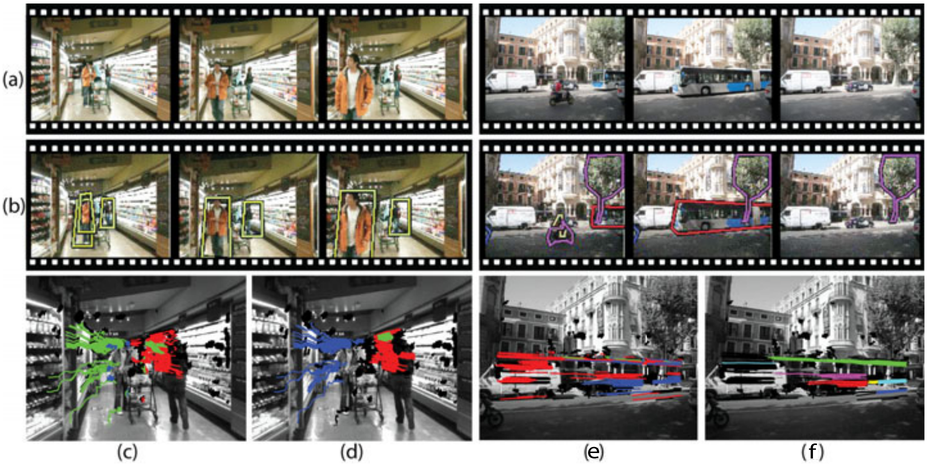

Fig. 2. Track clustering. Sample frames from the video sequence (a). The ground truth annotations denoted by polygons surrounding moving objects (b) can be used to generate ground truth labels for the tracked points in the video (c). Our track distance affinity function is used to to automatically cluster tracks into groups and generates fairly reasonable clusters where each roughly correspond to independent objects in the scene (d). The track clusters visualizations in (c) and (d) show the first frame of each video and the spatial location of all tracked points for the duration of the clip color-coded by the track cluster that each point corresponds to.

This metric was designed in the same spirit as the bottom level of the spatial pyramid matching method by Lazebnik *et al.* . We aim for matches that approximately preserve global spatial correspondences. Since our video neighbor knowledge-base is assumed to be spatially aligned to our query, a good match shall also preserve an approximate similar spatial coherence.

5 Video Database and Ground Truth

Our database consists of 2277 videos belonging to 100 scene categories. The categories with the most videos are: street (809), plaza (135), interior of a church (103), crosswalk (82), and aquarium (75). Additionally, 14 videos containing unusual events were downloaded from the web (see fig. 3 for some sample frames). 500 of the videos originate from the LabelMe video dataset [27]. As these videos were collected using consumer cameras without a tripod, there is slight camera shake. Using the LabelMe video system, the videos were stabilized. The object-level ground truth labeling in the LabelMe video database allows us to easily visualize the ground truth clustering of tracks and compare it with our automated results (see fig. 2). We split the database into 2301 training videos, selected 134 fully videos from outdoor urban scenes and the 14 unusual videos to create a test set with 148 videos.

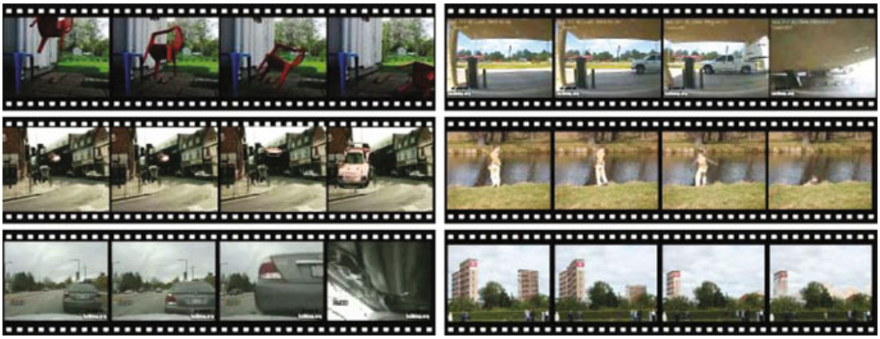

Fig. 3. Unusual videos. We define an unusual or anomalous event as one that is not likely to happen in our training data set. However, we ensured that they belong to scene classes present in our video corpus.

6 Experiments and Applications

We present two applications of our framework. Given the information from nearest neighbor videos, what can we say about the image if we were to see it in action? As an example, we can make good predictions of where motion is bound to happen in an image. We also present a method for determining the degree of anomaly of an event in a video clip using our training data.

6.1 Localized Motion Prediction

Given a static image, we can generate a probabilistic map determining the spatial extent of the motion. In order to estimate $p(motion|x, y, \text{scene})$ we use a parzen window estimator and the trajectories of the N=50 nearest neighbor videos retrieved with scene matching methods (GIST or dense SIFT-based).

$$p(motion|x, y, \text{scene}) = \frac{1}{N}\sum_{i}^{N}\frac{1}{M_i}\sum_{j}^{M_i}\sum_{t\in D} K(x - x_{i,j}(t), y - y_{i,j}(t); \sigma) \quad (5)$$

where N is the number of videos and M_i is the number of tracks in the ith video and $K(x, y; \sigma)$ is a gaussian kernel of width σ^2. Fig. 4 a shows the per-pixel prediction ROC curve compared using gist nearest neighbors, dense SIFT matching, and as a baseline, a random set of nearest neighbors. The evaluation set is composed of the first frame of each test video. We use the location of the tracked points in the test set as ground truth. Notice that scenes can have multiple plausible motions occurring in them but our current ground truth only provides one explanation. Despite our limited capacity of evaluation, notice the improvement when using SIFT and GIST matching to retrieve nearest neighbors. This graph suggests that (1) different sets of motions happen in different scenes, and (2) scene matching techniques do help filtering out distracting scenes to

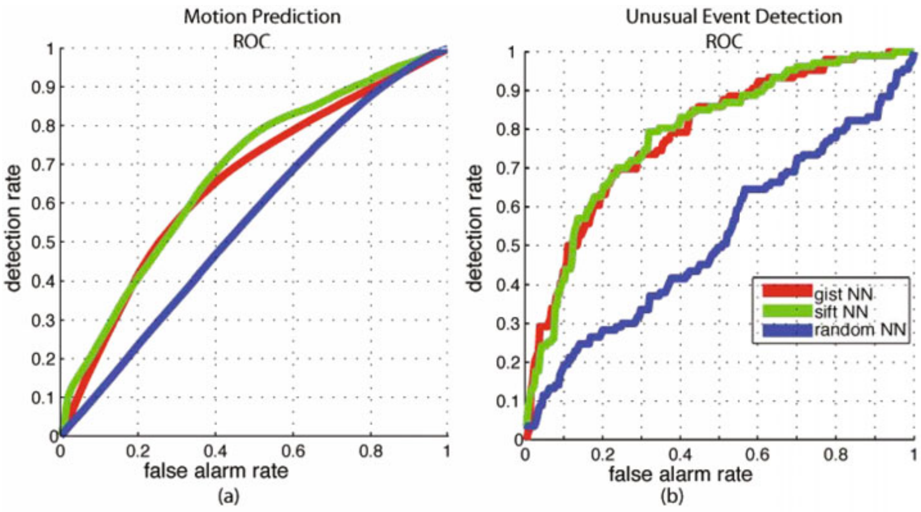

Fig. 4. Localized motion prediction (a) and unusual event detection (b). The algorithm was compared against two scene matching methods (GIST and dense SIFT) as well as a baseline supported by random nearest neighbors. Retrieving videos similar to the query image improves the classification rate.

make more reliable predictions (for example, a person climbing the wall of a building in a street scene would be considered unusual but a person climbing a wall in a rock climbing scene is normal). Fig. 6 c and 7 c contain the probability motion map constructed after integrating the track information from the nearest neighbors of each query video depicted in column (a). Notice that the location of high probability regions varies depending on the type of scenes. Moreover, the reliability of the motion maps depends on (1) how accurately the scene retrieval system returns nearest neighbors from the same scene category (2) whether the video corpus contains similar scenes. The reader can get an intuition of this by looking at column (e), which contains the average nearest neighbor image.

6.2 Event Prediction from a Single Image

Given a static image, we demonstrated that we can generate a probabilistic function per pixel. However, we are not only constrained to per-pixel information. We can use the track clusters of videos retrieved from the database and generate coherent track cluster predictions. One method is by directly transferring track clusters from nearest neighbors into the query image. However, this might generate too many similar predictions. Another way lies in clustering the retrieved track clusters. We use normalized cuts clustering for this step at the track cluster level using the distance function described in equation 4 to compare pairs of track clusters. Fig. 5 shows example track clusters overlaid on top of the static query image. A required input to the normalized cuts algorithm is the number of

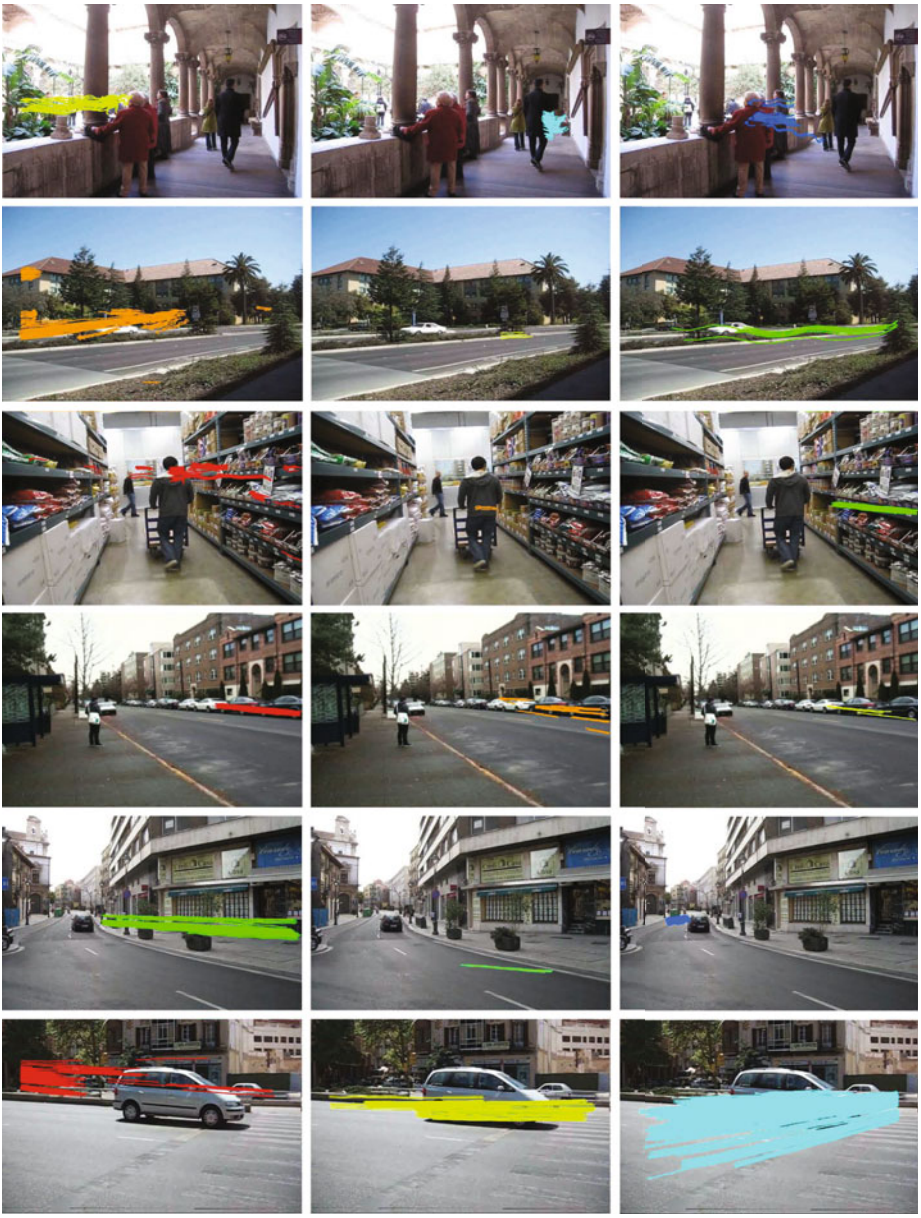

Fig. 5. Event prediction. Each row shows a static image with its corresponding event predictions. For each query image, we retrieve their nearest video clips using scene matching. The events belonging to the nearest neighbors are resized to match the dimensions of the query image and are further clustered to create different event predictions. For example, in a hallway scene, the system predicts motions of different people; in street scenes, it predicts cars moving along the road, etc.

clusters. We try a series of values from 1 to 10 and choose the clustering result that maximizes the distance between clusters. Notice how for different query scenes different predictions that take the image structure are generated.

6.3 Anomaly Detection

Given a video clip, we can also determine if an unusual event is taking place. First, we break down the video clip into query track clusters (which roughly represent object events) using the method described in section 4. We also retrieve the top 200 nearest videos using scene matching. We negatively correlate the degree of anomaly of a query track cluster with the maximum track cluster similarity between the query track cluster and each of the track clusters from the nearest neighbors:

$$anomaly(H_{query}) = -\underset{H_{neigh}}{\mathrm{argmax}}\Big(\mathbf{I}(H_{query}, H_{neigh})\Big) \tag{6}$$

where H_{query} is the spatial histogram of the velocity histories of the query track cluster and H_{neigh} denotes the histogram of a track cluster originated from a nearest neighbor. Intuitively, if we find a similar track cluster in a similar video clip, we consider it as normal. Conversely, a poor similarity score implies that such event (track cluster) does not usually happen in similar video clips. Fig. 6 shows examples of events that our system identified as common by finding a nearest neighbor that minimized its anomaly score. Notice how the nearest track clusters are fairly similar to the query ones and also how the spatial layout of the nearest neighbor scenes matches that of the query video. As a sanity check, notice the similarity of the nearest neighbors average image to the query scene suggesting that the scene retrieval system is picking the right scenes to make accurate predictions. Fig. 7 shows events with a higher anomaly score. Notice how the nearest neighbors differ from the queries. Also, the average images are indicators of noisy and random retrievals. By definition, unusual events will be less likely to appear in our database. However, if the database does not have enough examples of particular scenes, their events will be be flagged as unusual.

Fig. 4(b) shows a quantitative evaluation of this test. Our automatic clustering generates 685 normal and 106 unusual track clusters from our test set. Each of these clusters was scored achieving in similar classification rates when the system is powered by either SIFT or GIST matching methods reaching a 70% detection rate with a 22% false alarm rate. We use the scenario of a random set of nearest neighbors as a baseline. Due to our track cluster distance function, if a cluster similar to the query cluster appears in the random set, our algorithm will be able to identify it and classify the event as common. However, notice that the scene matching methods are demonstrating great utility cleaning up the retrieval set and narrowing videos to a fewer relevant ones. Fig. 8 shows some examples of our system in action.

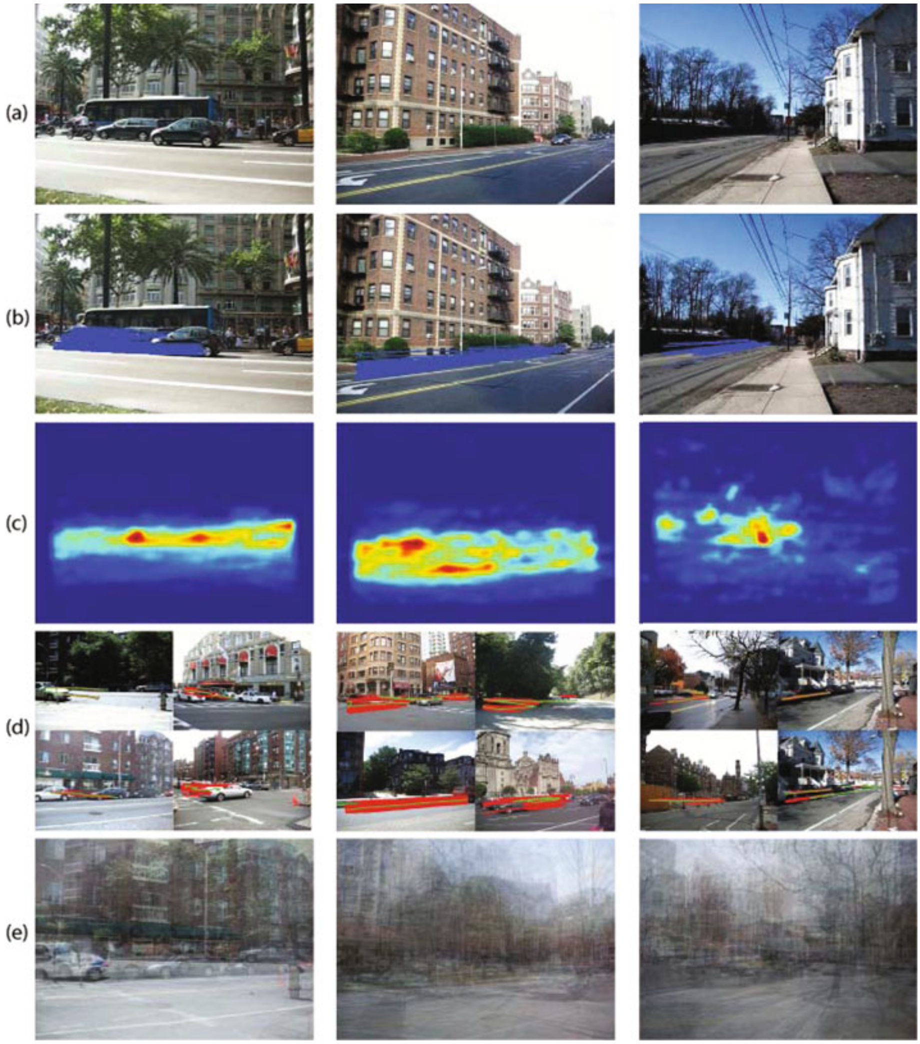

Fig. 6. Track cluster retrieval for common events. A frame from a query video (a), the tracks corresponding to one event in the video (b), the localized motion prediction map (c) generated after integrating the track information of the nearest neighbors (some examples shown in d), and the average image of the retrieved nearest neighbors (e). Notice the definition of high probability motion regions in (c) and how its shape roughly matches the scene geometry in (a). The maps in (c) were generated with no motion information originating from the query videos videos.

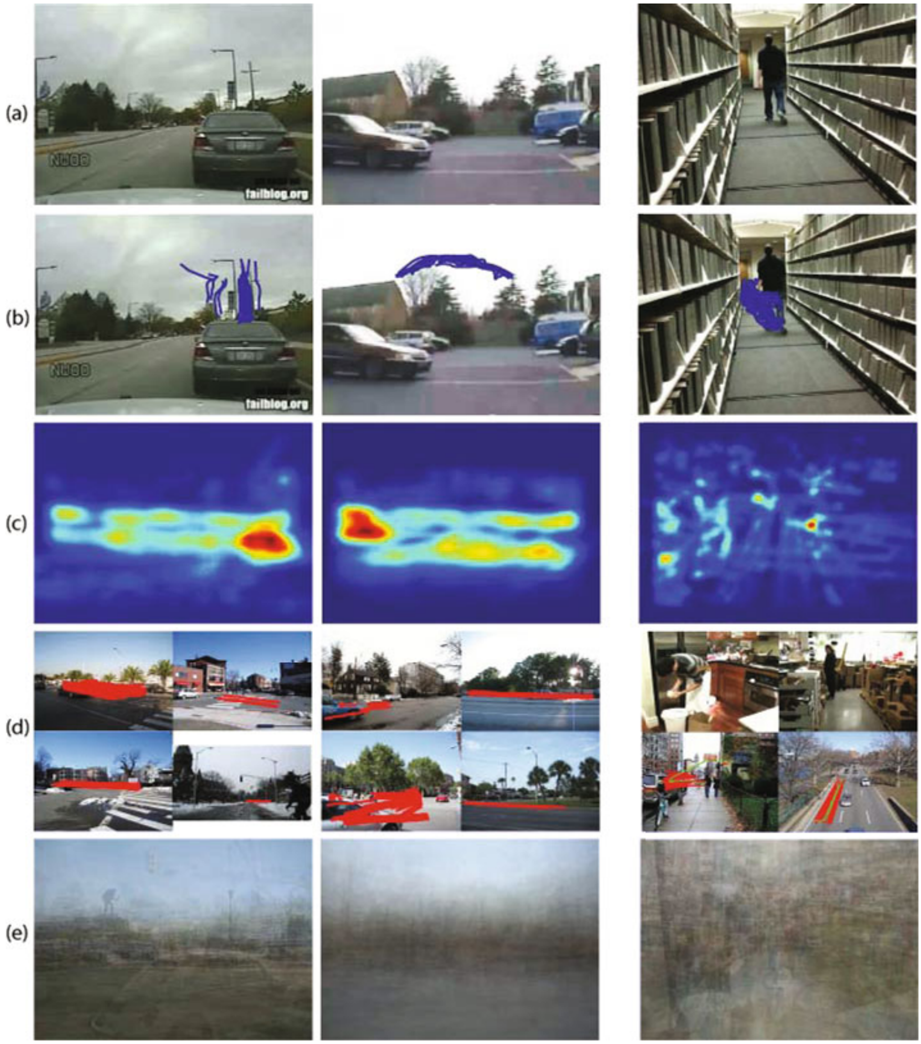

Fig. 7. Track cluster retrieval for unusual events (left) and scenes with less samples in our data set. When presented with unusual events such as a car crashing into the camera or a person jumping over a car while in motion (left and middle columns; key frames can be seen in fig. 8) our system is able to flag these as unusual events (b) due to their disparity with respect to the events taking place in the nearest neighbor videos. Notice the supporting neighbors belong to the same scene class as the query and the motion map predicts movements mostly in the car regions. However, our system fails when an image does not have enough representation in the database (right).

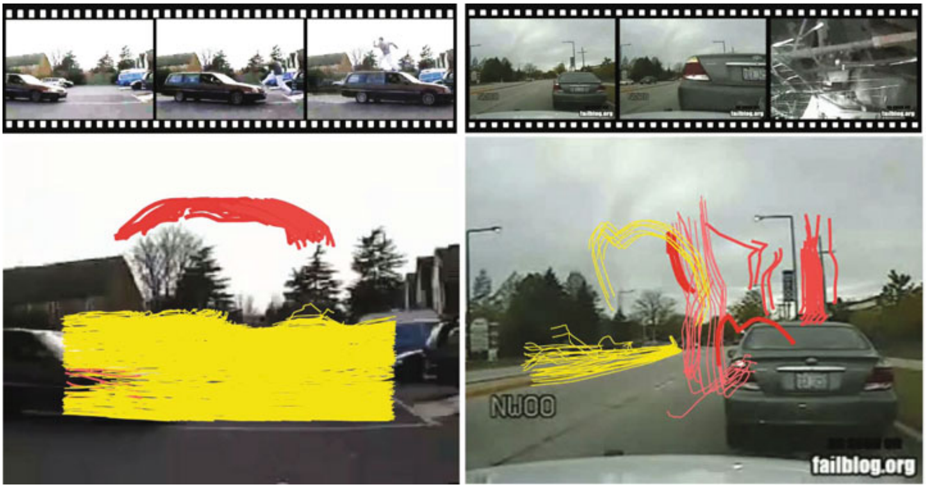

Fig. 8. Unusual event detection. Videos of a person jumping over a car and running across it (left) and a car crashing into the camera (right). Our system outputs anomaly scores for individual events. Common events shown in yellow and unusual ones in red. The thickness and saturation of the red tracks is proportional to the degree of anomaly.

7 Discussion and Concluding Remarks

We have presented a flexible and robust system for unsupervised localized motion prediction and anomaly detection powered by two phases: (1) scene matching to retrieve similar videos given a query video or image, and (2) motion matching via a scene-inspired and spatially aware histogram matching technique for velocity information. We emphasize that most of the work in the literature focuses on action recognition and detection and requires training models for each different action category. Our method has no training phase, is quick, and naturally extends into applications that are not available under other supervised learning scenarios. Experiments demonstrate the validity of our approach when given enough video samples of real world scenes. We envision its applicability in areas such as finding unique content in video sharing websites and future extensions in surveillance applications.

Acknowledgements

This work was funded by NSF Career Award ISI 0747120 and an NDSEG graduate fellowship.

References

1. Winawer, J., Huk, A.C., Boroditsky, L.: A motion aftereffect from still photographs depicting motion. Psychological Science 19, 276–283 (2008)
2. Krekelberg, B., Dannenberg, S., Hoffmann, K.P., Bremmer, F., Ross, J.: Neural correlates of implied motion. Nature 424, 674–677 (2003)

3. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: ICPR (2004)
4. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV. IEEE Computer Society, Washington (2009)
5. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. Int. J. Comput. Vision 79, 299–318 (2008)
6. Laptev, I., Perez, P.: Retrieving actions in movies. In: ICCV (2007)
7. Wang, X., Ma, K.T., Ng, G., Grimson, E.: Trajectory analysis and semantic region modeling using a nonparametric bayesian model. In: CVPR (2008)
8. Wang, X., Tieu, K., Grimson, E.: Learning semantic scene models by trajectory analysis. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 110–123. Springer, Heidelberg (2006)
9. Junejo, I.N., Javed, O., Shah, M.: Multi feature path modeling for video surveillance. In: International Conference on Pattern Recognition, vol. 2 (2004)
10. Zhong, H., Shi, J., Visontai, M.: Detecting unusual activity in video. In: CVPR (2004)
11. Dalal, N., Triggs, W.: Generalized SIFT based Human Detection. In: CVPR (2005)
12. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV (2009)
13. Zanetti, S., Zelnik-Manor, L., Perona, P.: A walk through the web's video clips. In: IEEE Workshop on Internet Vision, associated with CVPR (2008)
14. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV (2003)
15. Li, L.J., Fei-Fei, L.: What, where and who? classifying event by scene and object recognition. In: ICCV (2007)
16. Torralba, A., Fergus, R., Freeman, W.: Tiny images. Technical Report AIM-2005-025, MIT AI Lab Memo (September 2005)
17. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: Label transfer via dense scene alignment. In: CVPR (2009)
18. Hays, J., Efros, A.A.: Scene completion using millions of photographs. In: SIGGRAPH (2007)
19. Hays, J., Efros, A.A.: IM2GPS: estimating geographic information from a single image. In: CVPR (2008)
20. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: dense correspondence across different scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008)
21. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003)
22. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV 42, 145–175 (2001)
23. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
24. Tomasi, C., Kanade, T.: Detection and tracking of point features. In: IJCV (1991)
25. Birchfield, S.: Derivation of kanade-lucas-tomasi tracking equation. Technical report (1997)
26. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence (2000)
27. Yuen, J., Russell, B.C., Liu, C., Torralba, A.: Labelme video: Building a video database with human annotations. In: ICCV (2009)