Event Detection in Crowded Videos

Yan Ke[1], Rahul Sukthankar[2,1], Martial Hebert[1]
[1]School of Computer Science, Carnegie Mellon; [2]Intel Research Pittsburgh
{yke,rahuls,hebert}@cs.cmu.edu

Abstract

Real-world actions occur often in crowded, dynamic environments. This poses a difficult challenge for current approaches to video event detection because it is difficult to segment the actor from the background due to distracting motion from other objects in the scene. We propose a technique for event recognition in crowded videos that reliably identifies actions in the presence of partial occlusion and background clutter. Our approach is based on three key ideas: (1) we efficiently match the volumetric representation of an event against oversegmented spatio-temporal video volumes; (2) we augment our shape-based features using flow; (3) rather than treating an event template as an atomic entity, we separately match by parts (both in space and time), enabling robustness against occlusions and actor variability. Our experiments on human actions, such as picking up a dropped object or waving in a crowd show reliable detection with few false positives.

1. Introduction

The goal of event detection is to identify and localize specified spatio-temporal patterns in video, such as a person waving his or her hand. As observed by Ke *et al*. [14] and Shechtman & Irani [24], the task is similar to object detection in many respects since the pattern can be located anywhere in the scene (in both space and time) and requires reliable detection in the presence of significant background clutter. Event detection is thus distinct from the problem of human action recognition, where the primary goal is to classify a short video sequence of an actor performing an unknown action into one of several classes [2, 23, 32].

Our goal is to perform event detection in challenging real-world conditions where the action of interest is masked by the activity of a dynamic and crowded environment. Consider the examples shown in Figure 1. In Figure 1(a), the person waving his hand to flag down a bus is partially occluded, and his arm motion occurs near pedestrians that generate optical flow in the image. The scene also contains multiple moving objects and significant clutter that make it difficult to cleanly segment the actor from the background. In Figure 1(b), the goal is to detect the person picking up an object from the floor. In this case, the image flow is dominated by the motion of the crowd surrounding the actor, and the actor's clothing blends into the scene given the poor lighting conditions.



(a) (b)

Figure 1: Examples of successful event detection in crowded settings. (a) The hand wave is detected despite the partial occlusion and moving objects near the actor's hand; (b) The person picking up the dropped object is matched even though the scene is very cluttered and the dominant motion is that of the crowd in the background.

Earlier work has identified several promising strategies that could be employed for event detection. These can be broadly categorized into approaches based on tracking [21, 26], flow [8,14,24], spatio-temporal shapes [2,3,31,32], and interest points [7, 20, 23]. A more comprehensive review of historical work is presented by Aggarwal and Cai [1]. Methods based on tracking process the video frame-by-frame and segment an object of interest from background clutter, typically by matching the current frame against a model. By following the object's motion through time, a trace of model parameters is generated; this trace can be compared with that of the target spatio-temporal pattern to determine whether the observed event is of interest. Tracking-based approaches can incorporate existing domain knowledge about the target event in the model (e.g., joint angle limits in human kinematic models) and the system can support online queries since the video is processed a single frame at a time. However, initializing tracking

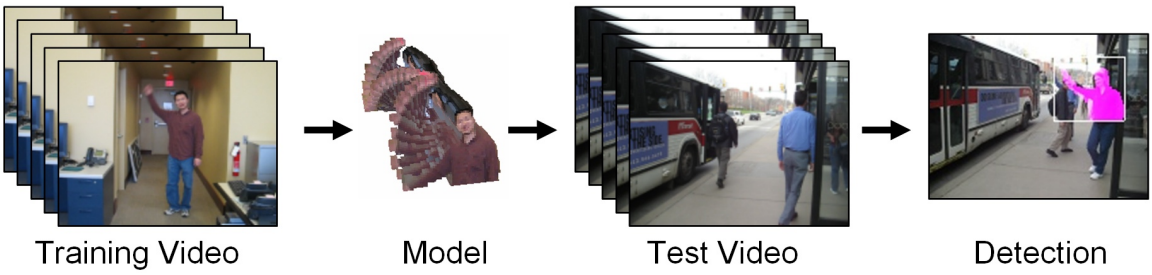


Figure 2: An overview of the proposed approach. An event model is constructed from a single training example and efficiently matched against oversegmented spatio-temporal volumes.

models can be difficult, particularly when the scene contains distracting objects. And while recent work has demonstrated significant progress in cluttered environments [22], tracking remains challenging in such environments, and the tracker output tends to be noisy. An alternate approach to tracking-based event detection focuses on multi-agent activities, where each actor is tracked as a blob and activities are classified based on observed locations and spatial interactions between blobs [12, 13]. These models are well-suited for expressing activities such as loitering, meeting, arrival and departure; the focus of our work is on finer-grained events where the body pose of the actor is critical to recognition.

Flow-based methods for event detection operate directly on the spatio-temporal sequence, attempting to recognize the specified pattern by brute-force correlation without segmentation. Efros *et al*. correlate flow templates with videos to recognize actions at a distance [8]. Ke *et al*. [14] train a cascade of boosted classifiers to process the vertical and horizontal components of flow in a video sequence using an algorithm that is similar in spirit to Viola and Jones' object detector for still images [27]. Shechtman and Irani propose an algorithm for correlating spatio-temporal event templates against videos without explicitly computing the optical flow, which can be noisy on object boundaries [24].

Shape-based methods treat the spatio-temporal volume of a video sequence as a 3D object. Different events in video generate distinctive shapes, and the goal of such methods is to recognize an event by recognizing its shape. Shape-based methods employ a variety of techniques to characterize the shape of an event, such as shape invariants [2, 32]. For computational efficiency and greater robustness to action variations, Bobick and Davis [3] project the spatio-temporal volume down to motion-history images, which Weinland *et al*. extend to motion-history volumes [31]. These techniques work best when the action of interest is performed in a setting that enables reliable segmentation. In particular, for static scenes, techniques such as background subtraction can generate high-quality spatio-temporal volumes that are amenable to this analysis. Unfortunately, these conditions do not hold in typical real-world videos due to the presence of multiple moving objects and scene clutter. Similarly, the extensive research on generalizing shape matching [11, 17] requires reliable figure/ground separation, which is infeasible in crowded scenes using current segmentation techniques. In this paper, we show how ideas from shape-based event detection can be extended to operate on oversegmented spatio-temporal volumes and perform well in challenging conditions.

Recently, space-time interest points [16] have become popular in the action recognition community [7, 20, 23], with many parallels to how traditional interest points [18] have been applied for object recognition. While the sparsity of interest points and their resulting computational efficiency is appealing, space-time interest points suffer the same drawbacks as their 2D analogues, such as failure to capture smooth motions and tendency to generate spurious detections at object boundaries.

We synthesize key ideas from each of the previous approaches and propose an algorithm to enable event detection in real-world crowded videos (Section 2). This paper focuses primarily on two topics: (1) effective representations of shape and motion for event detection, and (2) efficient matching of event models to over-segmented spatio-temporal volumes. The models that we match are derived from a single example and are manually constructed; automatic generation of event models from weakly-labeled observations is a related interesting problem and is not covered in the current work.

This paper is organized as follows. First, we show that spatio-temporal shapes are useful features for event detection. Where the previous work is typically limited to scenes with static backgrounds, we demonstrate shape matching in cluttered scenes with dynamic backgrounds. Second, we combine our shape descriptor with Shechtman and Irani's flow descriptor, which is a complementary feature that can be computed in cluttered environments without figure/ground separation (Section 3). Third, recognizing the value of a parts-based representation, which is explicitly modeled by the human tracking approaches, and implic-

itly modeled by the interest-point approaches, we break our action templates into parts and extend the pictorial structures algorithm [9, 10] to 3D parts for recognition (Section 4). Finally, we present an evaluation of event detection on crowded videos in Section 5. Figure 2 presents an overview of the approach.

2. Shape Matching

We briefly review the shape matching algorithm proposed in [15]. Space-time events are represented as spatio-temporal volumes in our system, as shown in Figure 2. The target events that we wish to recognize are typically one second long, and represent actions such as picking up an object from the ground, or a hand-wave. Denoting the template as T and the video volume as V, detecting the event involves sliding the template across all possible locations l in V and measuring the shape matching distance between T and V. An event is detected when the distance falls below a specified threshold. Similar to other sliding-window detection techniques, this is a rare-event detection task and therefore keeping the false-positive rate low is extremely important.

The first step is to extract spatio-temporal shape contours in the video using an unsupervised clustering technique. This enables us to ignore highly variable and potentially irrelevant features of the video such as color and texture, while preserving the object boundaries needed for shape classification. As a preprocessing step, the video is automatically segmented into regions in space-time using mean shift, with color and location as the input features [5, 6, 29]. This is the spatio-temporal equivalent of the concept of superpixels [19]. Figure 3 shows an example video sequence and the resulting segmentation. Note that there is no explicit figure/ground separation in the segmentation and that the objects are over-segmented. The degree to which the video is over-segmented can be adjusted by changing the kernel bandwidth. However, since finding an "optimal" bandwidth is difficult and not very meaningful, we use a single value of the bandwidth in all of our experiments, which errs on the side of over- rather than under-segmentation. Processing the video as a spatio-temporal volume (rather than frame-by-frame) results in better segmentations by preserving temporal continuity. We have found mean shift to work well in our task, but in general, any segmentation algorithm could be used as long as it produces an over-segmentation that tends to preserve object boundaries.

Our shape matching metric is based on the region intersection distance between the template volume and the set of over-segmented volumes in the video. Given two binary shapes, A and B, a natural distance metric between them is the set difference between the union and the intersection of the region, i.e., $|A \cup B \setminus A \cap B|$. Because of the over-segmentation, a video volume V at some location $l = (x, y, t)$ is composed of a set of k regions such



Figure 3: Input video and corresponding spatio-temporal segmentation using mean shift. The action is composed of a set of over-segmented regions.

that $V = \cup_{i=1}^{k} V_i$, as shown in Figure 4. A naive approach for computing the optimal distance between a template volume T and V is to enumerate through the 2^k subsets of V, compute the voxel distance between T and each subset, and choose the minimum. This would be prohibitively expensive even with a small number of regions. In [15], we showed that whether each region V_i should belong in the optimal set can be decided independently of all other regions, and therefore the distance computation is linear in k, the number of over-segmented regions. The distance between the template T and the volume V at location l is defined as

$$d(T, V; l) \quad = \quad \sum_{i=1}^{k} d(T, V_i; l), \tag{1}$$

where

$$d(T, V_i; l) = \begin{cases} |T \cap V_i| & \text{if } |T \cap V_i| < |V_i|/2 \\ |V_i - T \cap V_i| & \text{otherwise.} \end{cases} \tag{2}$$

This distance metric is equivalent to choosing the optimal set of over-segmented regions and computing the region intersection distance.

A consequence of employing automatic segmentation (as opposed to figure/ground separation) is that some objects will be over-segmented. Regions that are highly textured could be finely over-segmented, and therefore would result in a low matching distance to *any* template. To reduce this sensitivity to the segmentation granularity, we introduce a normalizing term that accounts for the flexibility of the candidate regions in matching arbitrary templates. The normalized distance is

$$d_N(T, V; l) = \frac{d(T, V; l)}{E_{\mathcal{T}}[d(\cdot, V; l)]}, \tag{3}$$

where the denominator is the expected distance of a template to the volume V, averaged over $\mathcal{T}$, the set of all pos-

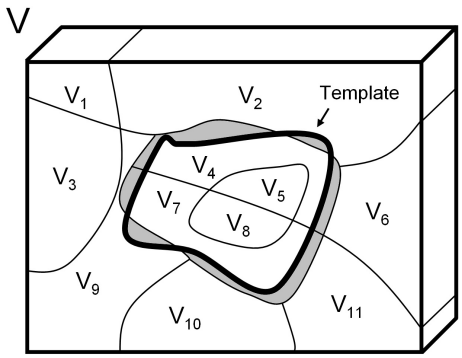


Figure 4: Example showing how a template is matched to an over-segmented volume using our shape matching method. The template is drawn in bold, and the distance (mismatch) is the area of the shaded region.

sible templates that fit within V. By enumerating through all possible templates that fit in V, we derive the expected distance to be

$$\sum_{i=1}^{k} \frac{1}{2^{|V_i|}} \sum_{j=1}^{|V_i|-1} \binom{|V_i|}{j} \min(j, |V_i| - j). \quad (4)$$

Note that the above term is a function of only $|V_i|$. Therefore, we can pre-compute this term so that the run-time computation reduces to table look-ups.

3. Flow Matching

Optical flow has been shown to be a useful feature for event detection in video [8, 14, 24]. Similar to our shape descriptor, it is invariant to appearance and lighting changes and does not require figure/ground separation. Shechtman and Irani [24] introduced a flow-based correlation technique that has been shown to complement our shape descriptor for event detection [15]. Given two spatial-temporal patches (of size $7 \times 7 \times 3$) centered at $P_1 = T(x_1, y_1, t_1)$ and $P_2 = V(x_2, y_2, t_2)$, traditional matching algorithms first compute the flow vectors, and then compute the distance between the flows. The results are often noisy or even inaccurate at object boundaries. Instead, Shechtman and Irani's algorithm computes whether the same flow could have generated the patches observed in the template and the video. The local inconsistency in motion between two patches P_1 and P_2 is given by

$$m_{12} = \frac{\Delta r_{12}}{\min(\Delta r_1, \Delta r_2) + \epsilon}, \quad (5)$$

where Δr is the rank increase between the three dimensional and the two dimensional Harris matrix of the patch from P_1, P_2, or the concatenation of the two patches in the case of Δr_{12}. The flow correlation distance is therefore

$$d_F(T, V; l) = \frac{\sum_{i \in T, j \in (T \cap V)} m_{ij}}{|T|}. \quad (6)$$

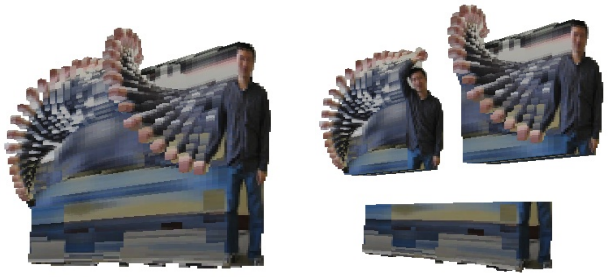
Our implementation of this algorithm generates an unusually-high number of false positives in regions with little texture. From the equations, it is obvious that uniform regions have *indeterminate* flow, and therefore could match *any* possible template. To eliminate such cases, we add a pre-filtering step to Shechtman and Irani's algorithm that discards uniform regions by thresholding on the Harris score of the region. We discuss how we combine the shape and flow distance metrics in the next section.

4. Recognition

The previous section describes a method for matching volumetric shape features on automatically-segmented video. The main strength of the algorithm is that it can perform shape matching without precise object masks in the input video [2, 3, 32]. Further, using template-based matching enables search with only one training example. However, like all template-based matching techniques [3, 24], it suffers from limited generalization power due to the variability in how different people perform the same action. A standard approach to improve generalization is to break the model into parts, allowing the parts to move independently, and to measure the joint appearance and geometric matching score of the parts. Allowing the parts to move makes the template more robust to the spatial and temporal variability of actions. This idea has been studied extensively in recognition in both images [30] and video [4, 25]. Therefore, we extend our baseline matching algorithm by introducing a parts-based volumetric shape-matching model. Specifically, we extend the pictorial structures framework [9, 10] to video volumes to model the geometric configuration of the parts and to find the optimal match in both appearance and configuration in the video.

4.1. Matching Parts

A key feature of our baseline algorithm is that it can perform shape matching with over-segmented regions. However, it assumes that the template consists of a single region, and that only the video is over-segmented. Given a single template, one must use prior knowledge to break the template into parts. For events that consist of human actions, these parts typically correspond to the rigid sections of the human body, and therefore the process is straightforward. We illustrate how one might manually break the handwave template into parts, as shown in Figure 5. We note that, for this action, only the upper body moves while the legs remain stationary. Therefore, a natural break should



(a) Whole (b) Parts

Figure 5: To generalize the model and allow for more variability in the action, we break the action template (a) into parts (b). The model can be split in both space or time to generate the parts.

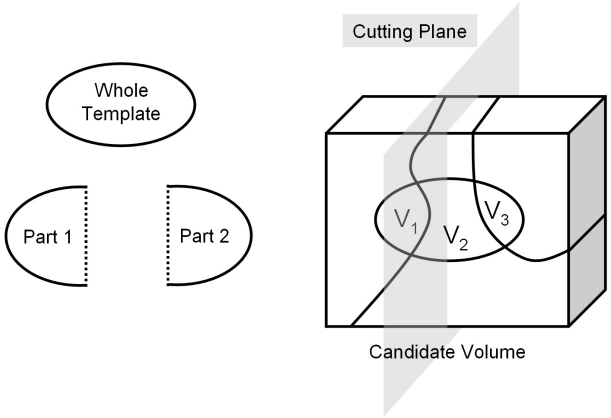


Figure 6: Illustration of how we artificially cut the candidate volume to match how the whole template is split into its constituent parts. The candidate volume is dynamically cut as we slide the template parts along it. The cutting process is very efficient.

be at the actor's waist. Such a break would allow the template parts to match people with different heights. Another natural break would be to split the top half of the action temporally, thus producing two parts that correspond to the upward and downward swing of the handwave action. This allows for some variation in the speed with which people swing their arms. It is important to note that, just like the whole template, the parts are also spatio-temporal volumes and could represent a body part in motion.

We now generalize our baseline algorithm (Section 2) and describe how we match template parts to over-segmented regions. Consider the oval template that has been split into two parts in the toy example illustrated in Figure 6. Although the whole template matches the oval ($V_1 \cup V_2 \cup V_3$) in the candidate volume, the parts would match poorly because the over-segmentation is inconsistent with the boundaries between the two parts. For example, our baseline algorithm would not match Part 1 to V_1, nor Part 2 to V_3. In general, there is no reason to believe that they should match because some of the part boundaries are artificially created (as shown by the dashed lines) and do not necessarily correspond to any real object boundaries. Our solution is to introduce additional cuts using a virtual plane that is aligned to and moves with the template part. For example, as we slide Part 1 across the video, we subdivide all the regions that intersect with the cutting plane placed on the right edge of the Part 1. V_2 is divided correctly, and Part 1 now matches the union of V_1 and the shaded region of V_2. For convenience, we only use cutting planes that are aligned with the principal axes, but in general the plane can be oriented in any direction. By pre-computing the cuts and with judicious bookkeeping, the parts-based matching can be performed with the same computational efficiency as our baseline shape-based matching algorithm.

4.2. Matching Part Configuration

We now describe how the framework of pictorial structures [9, 10] can be extended to parts-based event detection in video. Intuitively, each part in the template should match the video well, and the relative locations of parts should be in a valid geometric configuration. More formally, consider a set of n parts that form a tree in a graph. Adopting a notation based on Felzenszwalb and Huttenlocher [9], let the part model be specified by a graph $G = (P, E)$. Template part T_i is represented as a vertex $p_i \in P$ and the connection between parts p_i and p_j is represented as an edge $(p_i, p_j) \in E$. The configuration of the parts is specified by $L = (l_1, \ldots, l_n)$, where $l_i = (x_i, y_i, t_i)$ is the location of part T_i in the candidate volume V. Let $a_i(l_i)$ be the distance in appearance between the template part T_i and the video at location l_i. Let $d_{ij}(l_i, l_j)$ be the distance in configuration between parts T_i and T_j when they are placed at locations l_i and l_j, respectively. The general energy function that we want to minimize for an optimal match is:

$$L^* = \underset{L}{\operatorname{argmin}} \left(\sum_{i=1}^{n} a_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right). \quad (7)$$

The appearance distance $a()$ is a linear combination of our normalized distance metric (Equation 3) and Irani & Shechtman's flow-based correlation distance:

$$a_i(l_i) = d_N(T_i, V; l_i) + \alpha d_F(T_i, V; l_i), \quad (8)$$

where $\alpha = 0.2$ (we use the same weight for all experiments). For matching efficiency, our parts model is organized in a tree structure and we model the relative position of each part as a Gaussian with a diagonal covariance matrix. Therefore,

$$d_{ij}(l_i, l_j) = \beta \mathcal{N}(l_i - l_j, s_{ij}, \Sigma_{ij}), \quad (9)$$

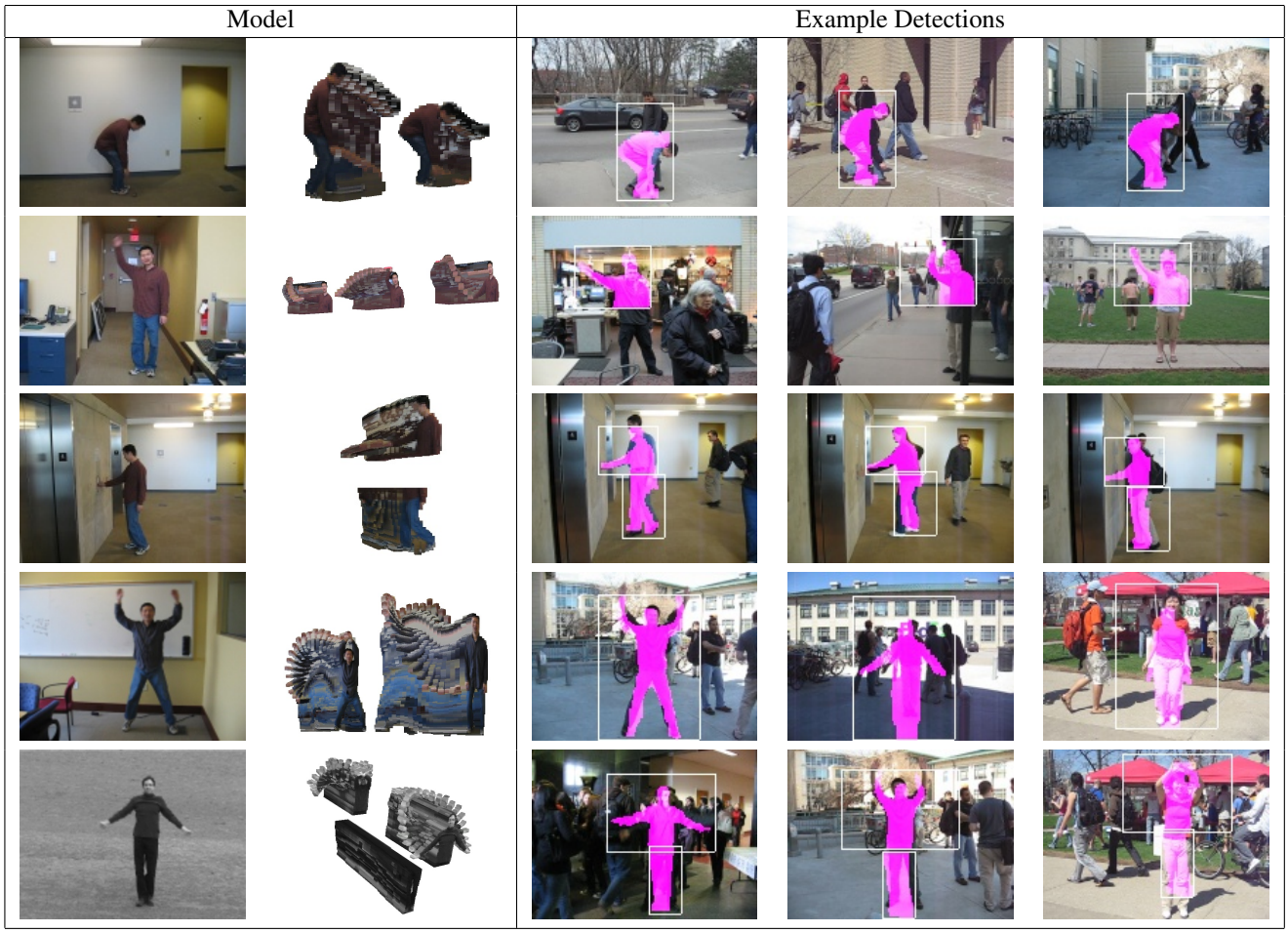


Figure 7: Examples of event detection in crowded video. Training sequences and event models are shown on the left. Detections in several challenging test sequences are shown on the right. The action mask from the appropriate time in the event model is overlaid on the test sequence frame, and a bounding box marks the matched location of each part.

where s_{ij} is the mean offset and Σ_{ij} is the diagonal covariance. β adjusts the relative weight of the configuration vs. appearance terms and for all of our experiments we use $\beta = 0.02$. The mean offset is taken from the location where we cut the parts, and the covariance is set manually, typically around 10% of the template size. Learning this matrix from multiple templates is part of our future work. As described by Felzenszwalb & Huttenlocher [9], the minimization can be efficiently solved using distance transforms and dynamic programming. Because we employ a sliding window approach to event detection, we also record the actual distance solved in the minimization and threshold on that distance. Only those locations with a distance below a specified threshold are considered as detections. As discussed earlier, a key feature of our algorithm is that although a segmented instance of the event template is needed, we do not assume that the input video can be reliably segmented. This makes event detection possible in challenging cases, such as crowded scenes, where reliable segmentation is difficult.

5. Results

To evaluate the effectiveness of our algorithms, we selected events that represent real world actions such as picking up an object from the ground, waving for a bus, or pushing an elevator button (Figure 7). In our previous work [15], we evaluated our initial matching algorithms on standard datasets (e.g., the KTH datasets [23]). This data was appropriate to evaluate the basic matching capabilities, but it is too "clean" to evaluate the effectiveness of the techniques described in this paper. Therefore, we acquired new videos by using a hand-held camera in environments with moving people or cars in the background. This data is designed to evaluate the performance of the algorithm in crowded scenes. We study the effects of using different combinations of shape and flow descriptors, and parts-based versus whole shape models. One subject performed one instance

of each action for training[1]. Between three to six other subjects performed multiple instances of the actions for testing. We collected approximately twenty minutes of video containing 110 events of interest. The videos were downscaled to 160x120 in resolution. There is high variability in both how the subjects performed the actions and in the background clutter. There are also significant spatial and temporal scale differences in the actions as well.

For each event, we create the model from a single instance by interactively segmenting the spatio-temporal volume using a graph-cut tool similar to [28]. The templates are typically $60 \times 80 \times 30$ in size and range from 20,000–80,000 voxels. The whole template is then manually broken into parts, as shown in Figure 7. The video is automatically segmented using mean shift; the average segment size is approximately 100 voxels. We scan the event template over the videos using the shape and flow distance metrics described earlier, and combine them using pictorial structures. There are approximately 120,000 possible locations to be scanned per second of video for a typical template. In these experiments, to evaluate the robustness of our matching algorithm to variations in observed scale, we match only at a single fixed scale; in practice, one could match over multiple scales. The algorithm returns a three-dimensional distance map representing the matching distance between the model and the video at every location in the video. For efficiency, we project the map to a one-dimensional vector of scores, keeping only the best detection for each frame, as shown in Figure 8(a). Since it is rare for two instances of an action to start and end at exactly the same time instant, this is a reasonable simplification. An event is detected when the matching distance falls below a specified threshold. We vary this threshold and count the number of true positives and false positives to generate the Precision-Recall graphs. A detected event is considered a true positive if it has greater than 50% overlap (in space-time) with the labeled event.

We now analyze the performance of our algorithm and compare it to baseline methods. Figure 7 shows example detections using our algorithm with the parts-based shape and flow descriptor in crowded scenes. Note the amount of clutter and movement from other people near the event. The precision-recall graphs for all of the actions are shown in Figures 8(b)–(f). We compare our results to Shechtman and Irani's flow consistency method [24] as a baseline, labeled as Flow (Whole) in our graphs. This state-of-the-art baseline method achieves low precision and recall in nearly all actions, demonstrating the difficulty of our dataset. Our combined parts-based shape and flow descriptor is significantly better and outperforms either descriptor alone, which confirms our previous findings [15]. The parts-based shape descriptor is better than the whole shape descriptor in the hand-wave, push button, and two-handed wave actions, while there is little benefit to adding the parts model for the jumping-jacks and pick-up actions.

[1]The two-handed wave template was taken from the KTH videos.

6. Conclusion

We present a method for detecting events in crowded videos. The video is treated as a spatio-temporal volume and events are detected using our volumetric shape descriptor in combination with Shechtman and Irani's flow descriptor. Unlike existing shape-based methods, our system does not require figure/ground separation, and is thus more applicable to real-world settings. We extend our baseline shape matching algorithm to detect event parts (sliced in both space or time), and generalize the model to recognize actions with higher actor variability. The parts are combined using pictorial structures to find the optimal configuration. Our approach detects events in difficult situations containing highly-cluttered dynamic backgrounds, and signficantly out-performs the baseline method [24]. This paper emphasizes the *matching* aspects of event detection and demonstrates robust performance on real-world videos. The biggest limitation of the current work is that the model is derived from a single exemplar of the event, thus limiting our ability to generalize across observed event variations. Future work will focus on the *modeling* aspects of the task, including the automatic selection of event parts and the aggregation of several training videos into a single model. Initial results show that greater generalization performance can be achieved by combining the matching scores from multiple event models.

7. Acknowledgements

This work was supported by NSF Grant IIS-0534962. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999.

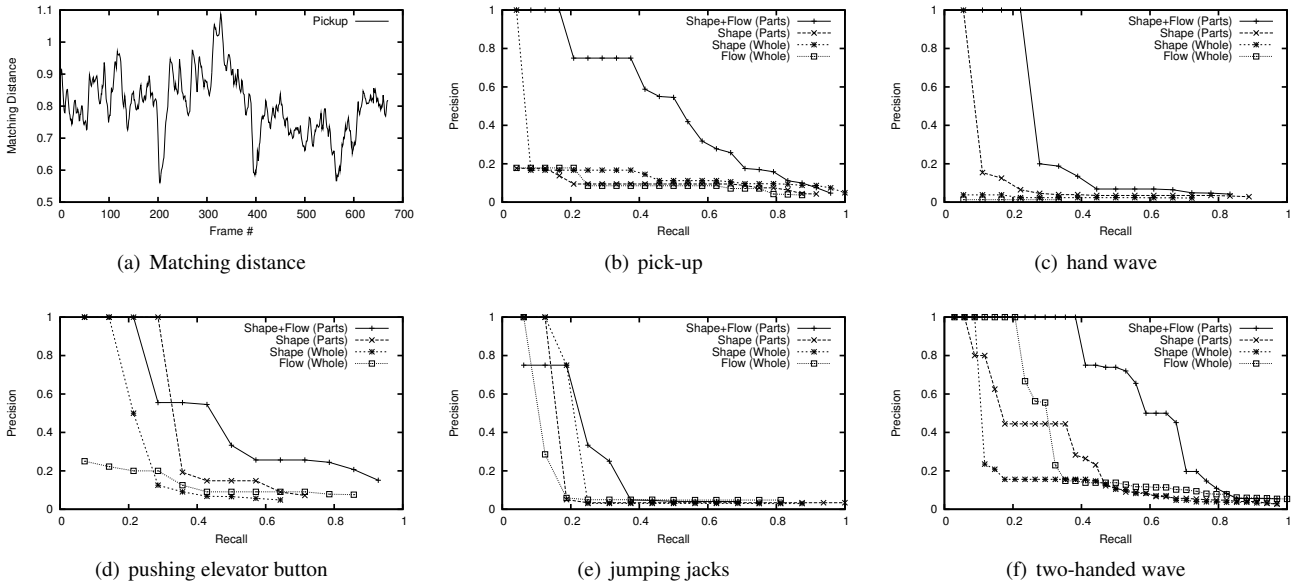
[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. ICCV*, 2005.

[3] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *PAMI*, 23(3), 2001.

[4] O. Boiman and M. Irani. Similarity by composition. In *NIPS*, 2006.

[5] Y. Cheng. Mean shift, mode seeking, and clustering. *PAMI*, 17(8), 1995.

[6] D. DeMenthon. Spatio-temporal segmentation of video by hierarchical mean shift analysis. In *Statistical Methods in Video Processing Workshop*, 2002.



(a) Matching distance (b) pick-up (c) hand wave

(d) pushing elevator button (e) jumping jacks (f) two-handed wave

Figure 8: (a) Projected matching distance on video with three pick-up events. A threshold of 0.6 successfully detects all of them. (b)–(f) Precision/recall curves for a variety of events. Our parts-based shape and flow descriptor significantly outperforms all other descriptors. The baseline method [24], labeled as "Flow (Whole)", achieves low precision and recall in most actions, demonstrating the difficulty of our dataset.

[7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE VS-PETS Workshop*, 2005.

[8] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. ICCV*, 2003.

[9] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005.

[10] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. on Computers*, 22(1), Jan. 1973.

[11] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt. Shape representation and classification using the poisson equation. *PAMI*, 28(12), 2006.

[12] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, and C. Isbell. Discovery and characterization of activities from event-streams. In *Proc. UAI*, 2005.

[13] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *Proc. ICCV*, 2001.

[14] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proc. ICCV*, 2005.

[15] Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal shape and flow correlation for action recognition. In *Workshop on Visual Surveillance*, 2007.

[16] I. Laptev and T. Lindeberg. Space-time interest points. In *Proc. ICCV*, 2003.

[17] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *PAMI*, 29(2), 2007.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.

[19] G. Mori. Guiding model search using segmentation. In *Proc. ICCV*, 2005.

[20] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *Proc. BMVC*, 2006.

[21] D. Ramanan and D. A. Forsyth. Automatic annotation of everyday movements. In *NIPS*, 2003.

[22] D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *PAMI*, 29(1), 2007.

[23] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Proc. ICPR*, 2004.

[24] E. Shechtman and M. Irani. Space-time behavior based correlation. In *Proc. CVPR*, 2005.

[25] E. Shechtman and M. Irani. Matching local self-similarities across images and video. In *Proc. CVPR*, 2007.

[26] Y. Sheikh, M. Sheikh, and M. Shah. Exploring the space of a human action. In *Proc. ICCV*, 2005.

[27] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2), 2004.

[28] J. Wang, P. Bhat, A. Colburn, M. Agrawala, and M. Cohen. Interactive video cutout. In *ACM SIGGRAPH*, 2005.

[29] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *Proc. ECCV*, 2004.

[30] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. ECCV*, 2000.

[31] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2), 2006.

[32] A. Yilmaz and M. Shah. Actions as objects: A novel action representation. In *Proc. CVPR*, 2005.