Spatio-temporal Shape and Flow Correlation for Action Recognition

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

This paper explores the use of volumetric features for action recognition. First, we propose a novel method to correlate spatio-temporal shapes to video clips that have been automatically segmented. Our method works on over-segmented videos, which means that we do not require background subtraction for reliable object segmentation. Next, we discuss and demonstrate the complementary nature of shape- and flow-based features for action recognition. Our method, when combined with a recent flow-based correlation technique, can detect a wide range of actions in video, as demonstrated by results on a long tennis video. Although not specifically designed for whole-video classification, we also show that our method's performance is competitive with current action classification techniques on a standard video classification dataset.

1. Introduction

The goal of action recognition is to localize a particular event of interest in video, such as a tennis serve, both in space and in time. Just as object recognition is a key problem in image understanding, action recognition is a fundamental challenge for interpreting video. A recent trend in action recognition has been the emergence of techniques based on the *volumetric analysis* of video, where a sequence of images is treated as a three-dimensional space-time volume. Eschewing the building of explicit models of the actor or environment (*e.g.*, kinematic models of humans), these approaches attempt to perform recognition directly on the raw video. An obvious benefit is that recognition need not be limited to a specific set of actors or actions but can, in principle, extend to a variety of events — given appropriate training data. The drawback is that volumetric representations do not easily generalize across appearance changes due to different actors, varying environmental conditions and camera viewpoint. This observation has motivated the employment of video features that are robust to appearance; these can be broadly categorized as *shape-based* (*e.g.*, background subtracted human silhouettes) and *flow-based* (*e.g.*, motion fields generated using optical flow). However, as discussed below, both of these types of methods have significant limitations.

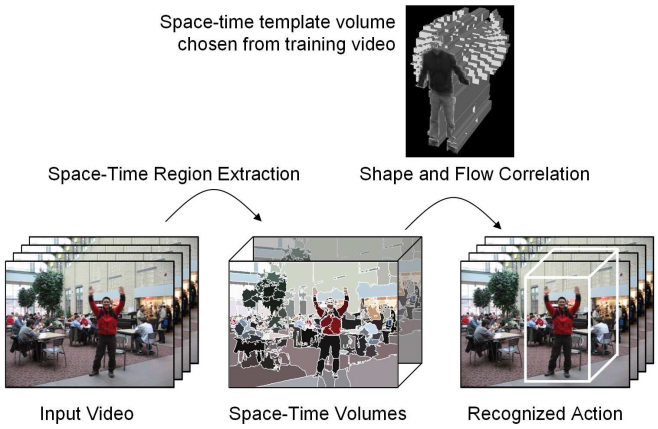
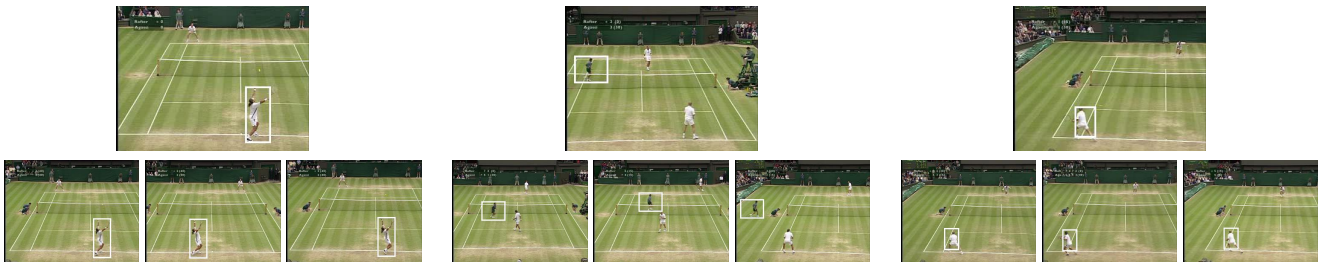


Figure 1. Our goal is to detect specific actions in realisitic videos with cluttered environments. First, we segment input video into space-time volumes. Then, we correlate action templates with the volumes using shape and flow features. We are able to localize events in space-time without the need for background-subtracted videos.

Silhouette-based approaches attempt to recognize actions by characterizing the shape of the actor's silhouette through space-time, and thus are robust to variations in clothing and lighting [2, 3, 21]. There are two major limitations with such approaches. First, they assume that the silhouettes can be accurately delineated from the background. Second, they assume that the entire person is represented as one region. Therefore, such techniques typically require static cameras and a good background model. Unfortunately, even state-of-the-art background subtraction techniques generate holes when parts of the actor blend in with the background, or create protrusions on the silhouette when strong shadows are present. These artifacts consequently reduce the accuracy of shape-based action recognition techniques. A more subtle limitation of silhouette-based techniques is that they ignore features inside the boundary, such as internal motion of the object.

Flow-based techniques estimate the optical field between adjacent frames and use that as the basis for action recogni-



a) serve action b) run right action c) return serve action

Figure 2. Illustration of actions detected in a tennis sequence. Top row: templates; Bottom row: example detections.

tion. Ke *et al*. learn a discriminative cascade of 3D box features on the flow [12]. Shechtman and Irani use a template matching approach to correlate the flow consistency between the template and the video [19]. In addition to being invariant to appearance variations, an important advantage of flow-based approaches is that they require no background subtraction and thus these methods can process videos with limited camera motion. However, optical flow is a very coarse feature and therefore many scenes are likely to exhibit similar flows over short periods of time. For example, Ke *et al*. observed that in the KTH actions dataset [18], their boxing detector was triggered near a hand-clap action because those regions contained the same flow [12].

Common to all appearance-based approaches are limitations due to changes in camera view and variability in the speed of actions. Very few representations are robust to these variations, and the standard approach is to span the space of variations using multiple training examples. Others have attempted to use space-time interest point features for added robustness [10, 16, 18]. While the sparsity of the interest points is certainly appealing from an efficiency standpoint, it is unclear how these methods compare against volumetric approaches. This paper evaluates shape- and flow-based volumetric features against interest point techniques on a standard dataset.

Our paper makes two major contributions. First, we propose a simple yet effective shape-based representation for matching videos that does not require background subtraction, nor explicit background models. Second, we combine our shape-based method with recent flow-based techniques and demonstrate improved recognition performance. Our shape-based matching consists of spatio-temporal region extraction and region matching. For region extraction, we employ an unsupervised clustering technique to segment the video into three-dimensional volumes that are internally consistent in appearance; we term these "supervoxels" since they are conceptually analogous to superpixels [17]. We observe that real object boundaries in spatio-temporal volumes typically fall on supervoxel borders, just as superpixel borders correspond to useful segmentation boundaries [15]. As with all bottom-up segmentation techniques, we do not expect the region extractor to segment the entire object as a single region, and thus we err on the side of over-segmentation. We propose a shape matching technique that works despite over-segmented videos. This is similar in spirit to recent work in shape-guided figure-ground segmentation [4]. We then discuss the limitations of shape and flow-based techniques for action recognition and argue that their complementary nature allows them to mitigate each other's limitations. To show the benefits of the combined features, we incorporate Shechtman and Irani's flow-based features [19] into our classifier and demonstrate improved performance on a challenging event detection task (see Figure 2) and a standard video classification task.

2. Shape-Based Matching

2.1. Spatio-Temporal Region Extraction

Region extraction is a process for automatically segmenting the video into 3D spatio-temporal volumes. An ideal region extractor would not only automatically segment individual objects in space, but it would also track their motion through time. Stable object segmentation is currently difficult for images [14] and video [20]. Because we want the region extraction to be general for many types of applications, we use mean shift [6, 8] to cluster the video into regions. Instead of indivually segmenting video frames and then linking the regions temporally (which causes unstable regions), we segment the three-dimensional spatio-temporal volume of pixels created by stacking a sequence of frames. The smallest processing unit is a voxel, taken from a 1×1 pixel from one 1 frame. The voxel location and color are used as features for mean shift. Our method works well despite having used simple features because it is not dependent on the precise segmentation of the object from the background. In general, any segmentation algorithm could have been used and we plan to explore more sophisticated algorithms and features in the future.

A critical parameter that must be chosen for mean shift, and nearly all clustering algorithms, is the kernel bandwidth size. Intuitively, the bandwidth size encodes the prior on the size of the objects that should be segmented. A small bandwidth will correctly segment small objects, but will over-segment large objects into multiple parts. Con-

versely, a large bandwidth will correctly segment large objects, but will incorrectly group small objects together. While there are proposed methods for adapting the kernel bandwidth [7] or automatically choosing a stable bandwidth based on scale-space theory [9,13], it is inherently impossible to choose the correct bandwidth for segmentation without higher-level semantic knowledge. Therefore, we perform hierarchical clustering using mean shift that segments the image into a pyramid of region sizes [9]. Because of the small scaling factor of the region sizes, this only increases the number of regions by a small constant factor, while enabling us to deal with arbitrarily-sized objects. Figure 3 shows hierarchical mean shift applied to a video sequence. As expected, larger regions are extracted with larger bandwidths. At run-time, we search over the hierarchy using the matching algorithm described below.

2.2. Volumetric Region Matching

We now present a novel method for matching action templates to over-segmented video that accomplishes three goals. First, the algorithm matches on the shape of the spatio-temporal volume, rather than the pixels in the volume. This is motivated by the fact that the spatio-temporal "shape" of an action is robust to variations in an object's appearance (*e.g.*, an actor's clothing). Second, the algorithm robustly matches over-segmented spatio-temporal volumes. In other words, it identifies the set of supervoxel regions that, when aggregated, best match the given template. Finally, the method must be computationally-efficient because video data is extremely large. Because our action representation is composed of three-dimensional shapes, it would seem straightforward to directly apply algorithms from the 3D shape matching literature to this task. Unfortunately, most of the existing algorithms cannot efficiently cope with over-segmented regions.

2.2.1 Proposed Algorithm

Our algorithm is based on the region intersection of binary volumes. One natural distance metric between two binary shapes (A, B) is the volume of the set difference between the union and the intersection of the regions, *i.e.*, $|A \cup B \setminus A \cap B|$. We adapt the algorithm to work with over-segmented regions as follows. First, we limit the search to a single level of the segmentation hierarchy, and then extend the search to multiple levels as described in Section 2.2.2. Given a template T of volume $|T|$, we slide the template along the x, y, and t dimensions of the video. Consider a candidate volume V with the template at some location (x, y, t). Because the video is over segmented, V could be composed of k regions V_i such that $V = \cup_{i=1}^{k} V_i$. Consider how one might calculate the voxel intersection distance between the template T and a subset of regions of V. Since every region V_i is either selected or not selected, a naive approach would enumerate all possible 2^k subsets of V, calculate voxel intersection between the template T and each subset, and choose the minimum. We propose a fast method for both identifying the subset of V that minimizes the distance and for calculating this distance.

There are four cases that we must consider when deciding whether a region V_i belongs in the minimum set. In Figure 4, we have drawn the template T in bold and overlaid onto the candidate volume V, which is segmented into 11 regions $V_1 \ldots V_{11}$. The set of regions that minimizes the distance to the template is $\{V_4, V_5, V_7, V_8\}$, and the actual distance is the area occupied by the shaded regions. By inspection, it is obvious that removing any region from the minimal set or adding any region not already in the minimal set, will increase the distance. The four cases of region intersections that we must consider are as follows. If a region V_i is completely enclosed by the template, such as V_5, then it is always contained in the minimal set. Similarly, if a region V_i does not intersect with the template, such as V_{11}, then it is never contained in the minimal set. The two interesting cases are when V_i intersects the template, such as V_2 and V_4. Let us consider V_2; it is obvious that excluding V_2 minimizes the distance between the template and the minimal set. Similarly, including V_4 in the minimal set minimizes the distance. Intuitively, we should include a region if there is a large overlap between the region and the template. More formally, we include region V_i if $|V_i \cap T| > |V_i|/2$. The distance is therefore

$$d(T, V_i) = \begin{cases} |T \cap V_i| & \text{if } |T \cap V_i| < |V_i|/2 \\ |V_i - T \cap V_i| & \text{otherwise.} \end{cases} \quad (1)$$

It is important to note that once the relative positions of the template T and the candidate volume V are specified, each of the regions V_i can be considered independently. In other words, whether V_i is in the minimal set is independent of any of the other regions $V_{\{1\ldots k\}\setminus i}$. Therefore, the distance between the template T and the candidate volume V is

$$d(T, V) \quad = \quad \sum_{i=1}^{k} d(T, V_i). \quad (2)$$

It can easily be shown that this function correctly computes the distance between the template and the minimal set. As we slide the window across the video, we mark all locations with a distance less than some threshold θ as a match. It can be shown that once the distance is computed at one location, distance computations at adjacent locations can be updated with only a small cost. The cost is proportional to the surface area of template, and in practice we reduce from 30 minutes to 3 minutes to match 10 seconds of video.

Frame Level 1 Level 2 Level 3 Level 4

Figure 3. Hierarchical mean shift automatically finds differently-sized regions at various levels of the hierarchy. The bottom level, with the smallest bandwidth, finds the smallest regions. Larger regions are found at higher levels of the hierarchy.

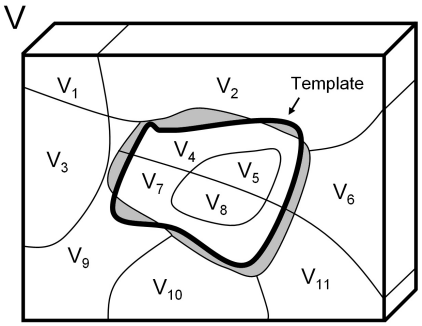


Figure 4. Example showing how a template is matched to an over-segmented volume using the Region Intersection method. The template is drawn in bold, and the distance (mismatch) is the area of the shaded region.

2.2.2 Modeling Segmentation Granularity

A potential problem with our method is that highly-textured regions of the video can generate many false positives. This is because such volumes consist of many tiny supervoxels that can be appropriately aggregated to match the given template. More formally, recall that the maximum error that a region V_i can contribute to the distance between the template and the volume is $|V_i|/2$. Therefore, as V is segmented into more regions, the smaller the size of each region, and therefore the more likely that some portion of V will match *any* template. In the limiting case, when V is segmented into $|V|$ unit-sized supervoxels, then the distance between V and any template is 0, since any volume can be trivially constructed from $1 \times 1 \times 1$ voxels. This motivates the need for a regularization term that balances the template match by the target volume's inherent flexibility. Therefore, we propose a normalization model as follows.

Instead of setting the decision boundary to $d(T, V) < \theta$, we set the decision boundary on the normalized distance,

$$\frac{d(T,V)}{E_{\mathcal{T}}[d(\cdot,V)]} < \theta, \tag{3}$$

where the denominator is the expected distance of a template to volume V, averaged over $\mathcal{T}$, the set of all possible templates that fit within V. Essentially, this is an estimate of the match confidence. Enumerating through all possible templates to compute the expected value may seem intractable at first, but we show that it is possible to compute this efficiently. Writing out the definition of the expectation, we have

$$\begin{aligned} E_{\mathcal{T}}[d(\cdot,V)] &= \frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} d(\tau,V) \\ &= \frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}}\sum_{i=1}^{k} d(\tau,V_i), \text{ by Eqn. 2} \\ &= \frac{1}{|\mathcal{T}|}\sum_{i=1}^{k}\sum_{\tau\in\mathcal{T}} d(\tau,V_i), \text{ by indep.} \end{aligned} \tag{4}$$

For each region V_i, we enumerate all possible templates that have j pixels intersecting the region, which is $2^{|V|-|V_i|}\binom{|V_i|}{j}$. Then, we calculate the distance between the region and the template which is either the area of the intersecting region or the non-intersecting region, whichever is smaller. Therefore, the expected distance is equal to

$$\begin{aligned} &= \frac{1}{2^{|V|}}\sum_{i=1}^{k}\sum_{j=1}^{|V_i|-1} 2^{|V|-|V_i|}\binom{|V_i|}{j}\min(j,|V_i|-j) \\ &= \sum_{i=1}^{k}\frac{1}{2^{|V_i|}}\sum_{j=1}^{|V_i|-1}\binom{|V_i|}{j}\min(j,|V_i|-j). \end{aligned} \tag{5}$$

This can be simplified to:

$$= \sum_{i=1}^{k} f(|V_i|), \text{ where} \tag{6}$$

$$f(n) = \begin{cases} \frac{n}{2} - \frac{1}{2^n}\binom{n}{n/2}(n/2), & n \text{ even}, \\ \frac{n}{2} - \frac{1}{2^n}\binom{n-1}{(n-1)/2}n, & n \text{ odd}. \end{cases} \tag{7}$$

There exists a simple recurrence for computing $f(n)$ exactly. Note that the term within the sum depends only on the size of the regions V_i and therefore can be pre-computed. At run-time, we only need to perform one table look-up for each supervoxel in the volume.

Not only does this algorithm automatically filter out cluttered backgrounds, it also chooses the best level in the segmentation hierarchy against which to match. Objects that are under-segmented will not match the template at all. And although over-segmented objects will match the template, they will be penalized for having too many regions. The "correct" segmentation level, the one that least over-segments without under-segmenting, will get the highest score.

3. Complementary Nature of Shape and Flow

We now highlight some fundamental limitations of shape- and flow-based features and how these can be overcome when the two feature types are combined. Previous work that employs shape features, whether in images or video, typically extracts the outline or silhouette of the object. This raw shape is then frequently represented as a binary image. Since silhouettes are robust to appearance variations due to internal texture and illumination, they are unable to represent the internal motion of an object. For example, a textured rolling ball is indistinguishable from a static ball based on shape alone — yet could easily be recognized based on flow. Figure 5 shows a portion of a hand-clap action sequence. When viewed from the front, the silhouette changes very little, although there is a distinctive change of flow at the hands. Therefore, one would expect the addition of flow features to help particularly in cases where an action cannot be distinguished from its silhouette alone.

Conversely, some actions cannot be distinguished using flow-based features alone. While such features explicitly model the motion of an object, they only implicitly model the object shape; more importantly, the shape of stationary parts of the object are ignored. For example, as observed by Ke *et al*. [12], in the KTH action recognition database, the flow of the boxing action looks very similar to that of the hand-clap (see Figure 6). This is because the horizontal trajectories of the arms is similar and the (stationary) body of the actor is invisible; thus the outward motion of the punch matches the inward motion of the clap. However, a shape-based feature could trivially distinguish between the person and the grassy background and disambiguate these actions. Therefore, we argue that shape- and flow-based features are complementary and should be used in conjunction for action recognition. We believe that we are the first to propose a volumetric approach that combines these two feature types and show their effectiveness on non-background subtracted videos.

Despite the normalization, our shape-based correlation algorithm can sometimes generate false positives on highly-textured regions, which are finely segmented (Figure 7a). However, we can obtain accurate flow measurements on these regions and a flow-based algorithm such as Shechtman and Irani's flow consistency [19] can filter out these false positives. Similarly, uniform regions pose an analogous problem for flow-based algorithms because these regions have *indeterminate* flow, and therefore can match *all* possible templates. Consequently, we add a pre-filtering step to Shechtman and Irani's technique to discard uniform regions by thresholding on the Harris score of the region. Even with this filtering, we observe that the majority of false-positives occur in low-textured regions (Figure 7b). Fortunately, our shape-based correlation works well on those regions and can be used to filter out the false positives. We quantify the benefits of combining shape and flow in Section 5.

4. Classification

This section describes how our spatio-temporal shape correlation technique can be applied to detect events in video and to classify video sequences. We also describe how Shechtman and Irani's flow-based correlation is incorporated into our framework. Suppose first (for now) that we have a template of a single instance of an action of interest. To find other instances of this action in a video clip, we can slide the template over the entire video and measure the correlation distance at all locations in space and time. Thresholding the correlation distance and finding the peaks would give us locations of potential matches. Figure 8 shows the minimum correlation distance of a hand-wave action projected on a time axis. Note that the cyclic nature of the action and the distance is minimized when the phases of the template and the action match. Although the action in the video is periodic, our algorithm does not assume periodic motion and thus we can can detect all instances of the event and localize them in both space and time. The advantage of using single templates for matching is that minimal human effort is required to bootstrap the system. This works well in scenarios where we have not trained the system on a large collection of template actions or where a human operator is interactively searching for novel events in large video databases. In such scenarios, the user can manually adjust the threshold to balance the detection and the false positive rates. The flow-based correlation technique by Shechtman and Irani [19] assumes that the action of interest always appears somewhere in the video database; thus, simply thresholding on 80% of the highest correlation score is sufficient to suppress most false positives. Similar thresholding techniques could also be employed here.

A limitation of using single templates for matching is that they typically generalize poorly. Some of the variations, such as scale and changes in action speed could be solved by scaling the template and searching over the scales. However, it would be difficult to generalize to different styles of the same action, or the same action seen from different viewpoints. We now describe how scores from multiple training templates can be combined to build a classifier. Employing multiple templates enables us to generalize over the variability in observed actions. A straightforward approach could attempt to match all templates to the video and use a k-nearest neighbor classifier. The challenge with this approach is that the distances to each template are not directly comparable. Furthermore, different features such as our region intersection and flow consistency lie in completely different spaces. While there are a number of classifiers that one could use, we chose to use SVM because of its reported success in a wide range of applications. We train

Figure 5. Notice how the silhouette stays constant during this part of the hand-clapping event. More generally, a fundamental limitation of such shape features is that they cannot represent motion inside the silhouette.

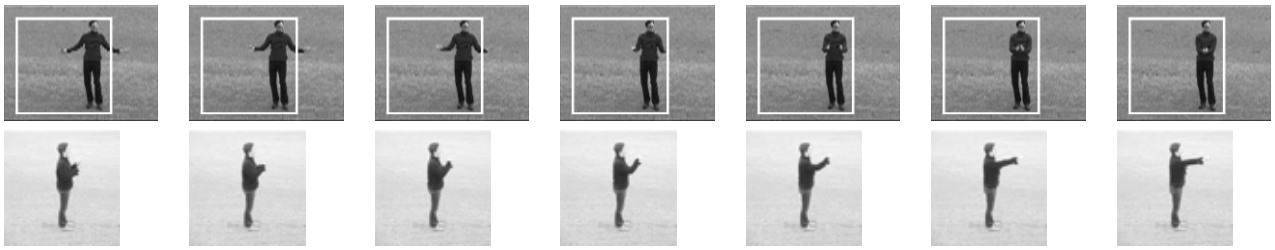


Figure 6. These two different actions (clapping and boxing) have very similar flow and are easily confused using flow-based features. The addition of a shape feature could easily tell that the grassy area does not contain a person and eliminate this false positive.

(a) shape fp (b) flow fp

Figure 7. False positives on found using a) shape correlation and b) flow consistency correlation. The false positives using shape features occur on highly textured regions, whereas the false positives using flow features occur on uniform regions. Using both features filters out each other's false positives.

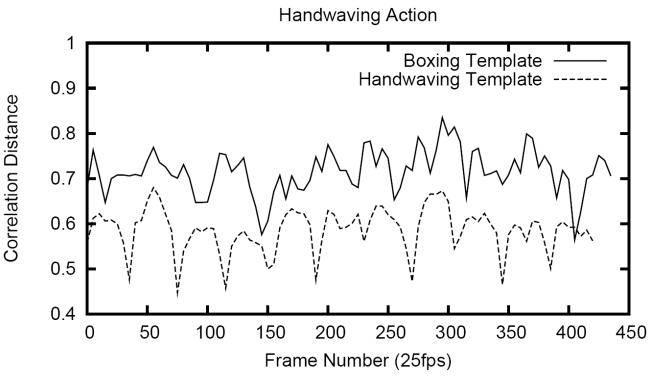
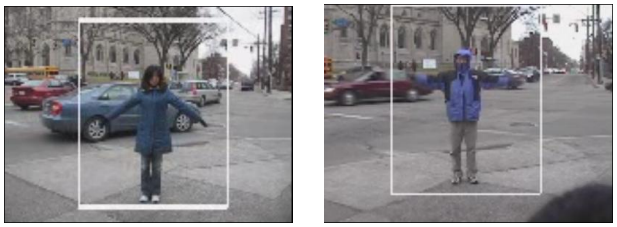


Figure 8. The minimum correlation distance of two templates on a hand-waving action. Notice the cycles in the action, where the distance is minimized when the phase of the template matches that of the action.

the SVM (using LIBSVM [5] and an RBF kernel) as follows. Given a candidate video at some space-time location, we correlate it with a database of n template actions. This gives us a feature vector of size n if use our region intersection algorithm, or $2n$ if we also include flow features. Each dimension of the feature vector corresponds to a distance between the candidate video location and a template action. We then train the SVM on both positively- and negatively-labeled regions.

5. Evaluation

We first illustrate how our algorithm performs on an event detection task, where we try to detect and localize an event in space-time. Only one template is used to search the video. This is useful in scenarios where we need to search for novel events using only one or two examples. For this experiment, we used a real life video — a Wimbledon 2000 match between Agassi and Rafter [1]. This experiment is difficult because the video contains a lot of clutter (*e.g.*, Figure 7a) and only a few instances of the actions are present in the video. We manually selected an example of Rafter serving (Figure 2a) and used it as a template to find all other instances of him serving in the first 30 minutes of the video. The template was scanned over all spatio-temporal locations (with 5 frame offsets for efficiency) and we kept the best match for each frame, assuming the action only occurs once per frame. There were 28 instances of the serve and we considered a detection to be a positive match if there was at least 75% overlap between the detection bounding volume and the manually-labeled event volume. Figure 10 shows the results of using various matching methods, where we varied the matching distance threshold to generate the precision-recall curve. "Shape Baseline" is the per-



(a) (b)

Figure 9. Handwave detections in a cluttered scene and with a moving background. Notice the difference in scale between the template (Figure 3) and the actors.

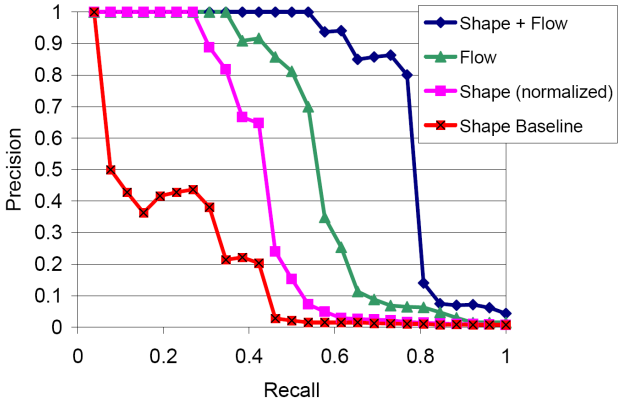


Figure 10. Comparison of various features on 30 minutes of tennis video in an event detection task.

formance of our shape-based region intersection algorithm without normalizing for segmentation granularity. "Shape (normalized)" normalizes for the segmentation granularity and performs markedly better. In this experiment, flow-based correlation performs better than shape-based correlation. This is partly due to false positives matching on finely-segmented crowd scenes, despite the normalization. However, combining both methods performs the best, achieving 80% recall at 80% precision. The two methods remove the false positives from each other and therefore results in a much higher precision. Qualitative examples of other actions we can detect are illustrated in Figures 2 and 9.

Although our goal is to detect and locate events, we adapted our algorithm to perform video classification on the KTH action database to compare against other algorithms. The KTH actions database contains 25 people performing six actions in four different scenarios [18]. Each video clip contains one person performing an action multiple times. This dataset is difficult because it contains drastic lighting, clothing, and scale changes (Figure 11). Different people also perform the actions at different speeds and orientations. The videos were recorded using a handheld camera which prevent simple background subtraction techniques from reliably extracting the person. The goal of the experiment is to classify the video clips into one of the six actions — walking, jogging, running, boxing, clapping, and waving. Classifying the entire video simplifies the training and recognition process because we do not have to label each instance of the action; we only need to label the sequence as a whole. Following the methodology of Niebles *et al*. [16], we use leave-one-out cross-validation grouped by person to measure the classification accuracy. We train the SVM classifier as follows. First, we manually label 4 templates for each action. Each template contains one cycle of the action, typically 15 to 30 frames long. The videos used to extract the templates are removed from the cross-validation set. For each template t_i, we scan over all space-time locations in a video clip. For each frame of the video, we extract the best correlation score for each template. There is one feature f per frame, where f_i is the best correlation score to template t_i. During classification, each frame in a video clip is classified as one of the six actions and votes for the label of the entire video clip.

(a) (b)

Figure 11. Notice the difference in scale between some videos in the KTH dataset. The contrast is also low making segmentation difficult.

Table 1 shows the confusion matrix on the KTH dataset. The result is generated using both shape and flow features and correlated against two templates per action. We achieve an accuracy of 80.9%, which is comparable to the most recent studies on the same dataset (Table 2). Unfortunately, we can only loosely compare the results in Table 2 because different groups employed different experimental methodologies. Like the other studies, we find that there is confusion mainly between walk-jog-run and box-clap-wave. As expected, running is more easily confused with jogging than with walking. Boxing is also more easily confused with clapping (horizontal motion) than waving (vertical motion). Figure 12 shows the effect of using different features and training on different number of templates. We are able to generalize the actions and increase the classification performance by training on more templates, but with diminishing returns. On this dataset, shape-based correlation performs better than the flow-based correlation, and performance improves slightly when we combine the two features.

6. Conclusion

We propose a new spatio-temporal shape-based correlation algorithm for action recognition that does not require background subtraction for silhouette extraction. The video

Table 1. Confusion matrix using our method (combined shape and flow) on the KTH actions database. Accuracy = 80.9%.

	walk	jog	run	box	clap	wave
walk	0.88	0.08	0.04	0.00	0.00	0.00
jog	0.14	0.66	0.19	0.00	0.00	0.00
run	0.04	0.15	0.81	0.00	0.00	0.00
box	0.03	0.00	0.00	0.84	0.10	0.03
clap	0.00	0.01	0.00	0.12	0.79	0.07
wave	0.00	0.00	0.00	0.06	0.06	0.88

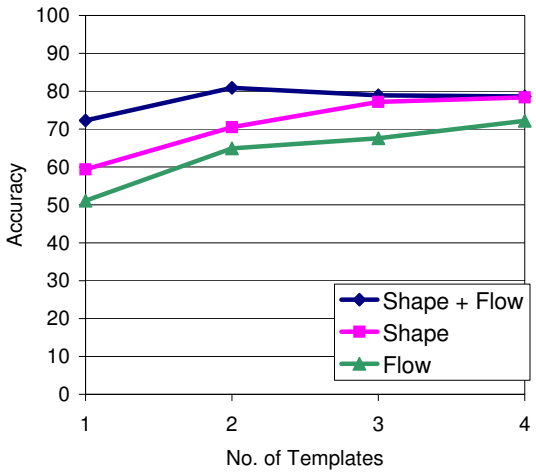


Figure 12. Results on the KTH actions database. Training on more templates improves results with diminishing returns. The combined shape and flow features perform better than either alone, especially with few training examples.

Table 2. Although our method is not specifically designed for whole-video classification, our results on the KTH actions dataset [18] are competitive with recent studies.

Related work	Accuracy
Our Method (shape + flow)	**80.9%**
Ke *et al*. [12]	63.0%
Schuldt *et al*. [18]	71.7%
Dollar *et al*. [10]	81.2%
Niebles *et al*. [16]	81.5%
Jiang *et al*. [11]	84.4%

is segmented in space-time using mean shift, which gives us a hierarchy of segmented regions. Our matching algorithm can efficiently calculate the distance between a template and the over-segmented video. The results are competitive with the state-of-the-art on a standard dataset and we show that combined shape and flow-based features perform better than either alone. For future work, we will explore the effects of other region segmentation and classification algorithms. Further, we plan to explore semi-supervised learning of template actions to minimize the labor required to label them.

7. Acknowledgements

Yan Ke is supported by NSF Grant IIS-0534962.

References

[1] Wimbledon 2000 Semi-Final - Agassi vs. Rafter. SRO Sports Entertainment. ISBN: 0-7697-7886-0.

[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*, 2001.

[4] E. Borenstein and J. Malik. Shape guided object segmentation. In *Proc. CVPR*, 2006.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] Y. Cheng. Mean shift, mode seeking, and clustering. *PAMI*, 1995.

[7] D. Comaniciu. An algorithm for data-driven bandwidth selection. *PAMI*, 2003.

[8] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.

[9] D. DeMenthon and D. Doermann. Video retrieval of near-duplicates using k-nearest neighbor retrieval of spatio-temporal descriptors. *MTAP*, 2005.

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

[11] H. Jiang, M. S. Drew, and Z.-N. Li. Successive convex matching for action detection. In *Proc. CVPR*, 2006.

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

[13] Y. Leung, J.-S. Zhang, and Z.-B. Xu. Clustering by scale-space filtering. *PAMI*, 2000.

[14] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, 2001.

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

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

[17] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. ICCV*, 2003.

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

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

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

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