Efficient Visual Event Detection using Volumetric Features

Y. Ke, R. Sukthankar, M. Hebert

Proceedings of International Conference on Computer Vision, 2005

IRP-TR-05-45

Research at Intel

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining applications.
Intel may make changes to specifications and product descriptions at any time, without notice.

Efficient Visual Event Detection using Volumetric Features

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
http://www.cs.cmu.edu/~yke/volumetric/

Abstract

This paper studies the use of volumetric features as an alternative to popular local descriptor approaches for event detection in video sequences. Motivated by the recent success of similar ideas in object detection on static images, we generalize the notion of 2D box features to 3D spatio-temporal volumetric features. This general framework enables us to do real-time video analysis. We construct a real-time event detector for each action of interest by learning a cascade of filters based on volumetric features that efficiently scans video sequences in space and time. This event detector recognizes actions that are traditionally problematic for interest point methods — such as smooth motions where insufficient space-time interest points are available. Our experiments demonstrate that the technique accurately detects actions on real-world sequences and is robust to changes in viewpoint, scale and action speed. We also adapt our technique to the related task of human action classification and confirm that it achieves performance comparable to a current interest point based human activity recognizer on a standard database of human activities.

1. Introduction

Event detection in video (see Figure 1) is becoming an increasingly important application for computer vision, particularly in the context of activity recognition [1]. Approaches to the problem are typically categorized as either model-based or appearance-based. The former [3, 12] attempt to estimate a set of model parameters, such as pose, from the video data and use them to recognize the activity. By contrast, the latter [2, 4, 10] perform inference directly on the observed pixels. Broadly speaking, model-based techniques are preferable when a good model for the actor is available *a priori*, whereas appearance-based techniques extend more easily to different activities.

Recently, there has been significant interest in approaches that exploit local descriptors on interest points in static images [7] and video [6]. These rely on expressing the local region around an area of interest using representations that are robust to geometric perturbations and noise, yet distinctive enough to reliably identify the local region. However, these techniques rely on the assumption that one can reliably detect a sufficient number of stable interest points in the video sequence. For space-time interest points this means that the video sequence must contain several instances of motion critical events — regions where an object rapidly changes its direction of motion — such as the reciprocating path traced by a walking person's shoe. Unfortunately, these techniques fail to detect useful interest points in many common situations where the motions contain no sharp extrema, such as those illustrated in Figure 2. Space-time interest points are also frequently triggered by the appearance of shadows and highlights in the video sequence, as shown in Figure 3. These unstable "events" are sensitive to lighting conditions and can reduce recognition accuracy for the action of interest.

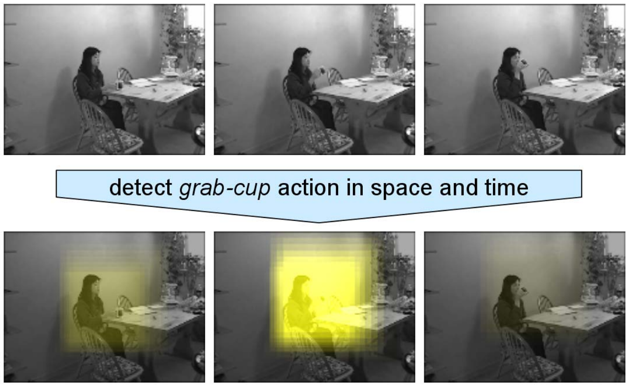
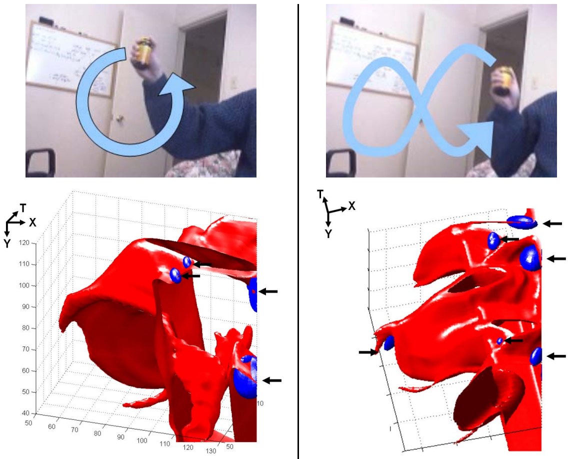


Figure 1: An example of our detector recognizing the *grab-cup* action. Note that the detection volume (shown highlighted) is localized both in space and time.

We propose a novel appearance-based framework that employs volumetric features for efficiently analyzing video. Applying this framework on the video's optical flow, we learn activity detectors that perform event detection in real time. The framework extends the rectangle features used by Viola and Jones [18] into the spatio-temporal domain for video analysis. Unlike their follow-up work in pedestrian detection in video [19], which uses only two adjacent frames, our *volumetric features* span longer time frames. To



circular motion

figure eight motion

Figure 2: Two examples of smooth motions where no stable space-time interest points are detected. The 3D plots of motion through time were generated using software from [6]. The highlighted ellipsoids show the detected interest points. All of these detections are non-informative, caused by boundary interactions between the arm and the edge of the frame. By contrast, our volumetric features are scanned over the video sequence through space and time, and can accurately recognize such motions.

maintain computational efficiency and build a real-time detector, we generalize the notion of integral images [18] to an efficient space-time representation that we term *integral videos*. These allow us to perform event detection on video sequences in real time, as shown in Figure 1.

Although our system is primarily designed to detect motion events, we have also extended it to the action classification task, where each video sequence contains several repetitions of a single action. Schuldt *et al.* [15] propose a technique that uses local space-time features to classify six human actions (walk, jog, run, wave, clap, and box) in challenging real-world video sequences. They argue that global image measurements based on optical flow are unstable when there are multiple moving objects in the scene, or when the camera is not stationary. We show on the same dataset used in their work that our technique achieves comparable performance in the presence of camera motion, scale variation, and viewpoint changes. We argue that the same problems that hinder the use of 2D local descriptors for object detection in static images also impact spatio-temporal local descriptors. For example, changes in object or background appearance will affect the location of the peaks found in spatio-temporal gradients. The trade-off between the stability of the interest points and the number of points found also exists in this domain. Because our technique uses dense optical flow measurements, our classifier is not limited to the sparse information found at peaks in spatio-temporal gradients nor affected by the instability of those peaks.

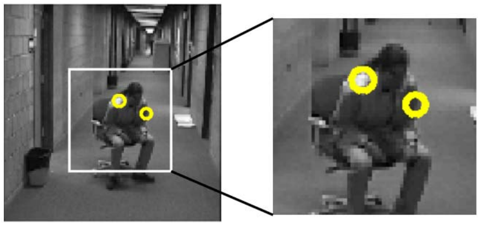


Figure 3: Space-time interest points are often found on highlights and shadows. These points are sensitive to lighting conditions and reduce recognition accuracy. This observation motivates our decision to apply volumetric features to the motion vectors rather than to the raw pixels.

The remainder of the paper is organized as follows. Section 2 summarizes the related work. Section 3 details our volumetric features. Section 4 describes classifier training. Section 5 explains how this classifier is applied to video sequences. Section 6 presents experimental results on several real-world video sequences. Section 7 concludes the paper.

2. Related Work

Aggarwal and Cai present an excellent overview of human motion analysis [1]. Of the appearance based methods, many approaches are domain specific, or have many constraints on the environment as well as the type of motion that can be detected. For example, Polana and Nelson [11] only detect periodic motion. Moore *et al.* [9] use Hidden Markov models and Bayesian relations to detect actions and classify objects, but are limited to an overhead view and only track hands. Bobick and Davis [2] use motion-energy images (MEI) and motion-history images (MHI) to recognize many types of aerobics exercises. While their method is efficient, their work and others [4] assume that the actor is well segmented from the background and centered for the detector. Our method does not need to pre-segment the motion and is robust to slight motions like those of a hand-held camera. Zelnik-Manor and Irani [22] cluster long sequences of events to detect classes of activities in those sequences.

More recently, Rea *et al.* [13] and Wang *et al.* [20] attempt to detect events in specific domains such as snooker and tennis. Peursum *et al.* [10] try to label static objects in the scene by tracking a person's height and speed and training an HMM to detect actions such as walking and sitting. Such a simple method may be suitable for these activities, but it does not seem likely that it would generalize to more complicated motions. Many of the approaches to date rely on the extraction of primitives, such as interest points and contours. In contrast to such "structural" approaches, we perform the detection directly on the pixel data. Recent work by Shechtman and Irani [16] demonstrates activity recognition using space-time correlation of the video

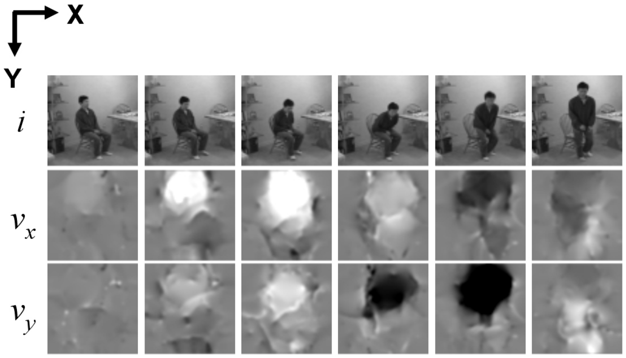


Figure 4: An example of the *stand-up* action and its optical flow. We separate the optical flow into the horizontal (v_x) and vertical (v_y) components. Lighter areas represent flow in the positive direction, while darker areas represent flow in the other direction.

with an action template. While their system can be trained on a single video sequence, the correlation is significantly less efficient than our cascade of detectors.

3. Features

Given a video sequence, our goal is to classify whether an action event occurs in any of the space-time volumes in the video sequence. This is similar to running a classifier on all of the sub-windows in a 2D image for object detection. Similarly, as detailed in Section 5, we run the classifier on all sub-volumes for event detection.

The framework is general enough that volumetric features can be computed over many types of low level features of the video, for example raw pixel intensities, spatio-temporal gradients, or optical flow. We believe that most of the information salient to recognizing actions can be captured from the optical flow, and that the appearance of the object is less relevant.

Initial experiments using pixel intensities performed poorly, mainly due to changes in appearance of the actor, the background, and lighting conditions. This motivates our decision to compute our volumetric features on the video's optical flow. We separate the optical flow into its horizontal and vertical components and compute volumetric features on each component. Let $v_x(x, y, t)$ and $v_y(x, y, t)$ denote the horizontal and vertical optical flow components, respectively, at pixel location (x, y) and time t. This is illustrated in Figure 4.

As shown in the first row of Figure 5, our volumetric features consist of one-box or two-box volumes. The value of the one-box feature is simply the sum of the pixels within the volume (v_x or v_y). Correspondingly, the value of a two-box feature is the difference of their individual sums. Just as in 2D object detection [18], where one computes multiple rectangle features in a sub-window of the image, we compute multiple one-box and two-box features in a detection sub-volume, as illustrated in the second row of Figure 5. The horizontal and vertical size of the volume must be varied to match the size of the object being detected. The temporal span of the detection volume is both application and motion dependent. It is not necessary for the time duration of the volume to exactly match that of the motion to be recognized. Provided that the detection volume encompasses the motion, we can achieve satisfactory recognition even if the action is performed at moderately different speeds (see Figure 10). If the training sequences are all aligned correctly at the start of the motion, the training algorithm described in Section 4 will learn that the tail ends of the sequences are noisy and thus less discriminative.

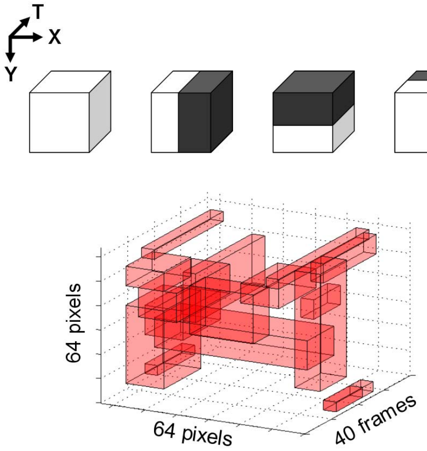


Figure 5: The top row illustrates the 3D volumetric features used in our classifiers. The first feature calculates the volume. The other three features calculate volumetric differences in X, Y, and time. The bottom row shows multiple features learned by the classifier to recognize the handwave action in a detection volume.

In our experiments, the features are computed over a volume of 64×64 pixels by 40 frames in time. An exhaustive enumeration of all possible one-box and two-box features over this volume would number in the billions. Therefore one must sub-sample the feature space to reduce the set to a reasonable size for learning. We discretize the space to sample approximately one million features in our experiments, which is a large, but manageable set. The smallest feature occupies a 4×4 pixel by 4 frame spatio-temporal volume.

We use an "integral video" data structure to efficiently calculate the box features described above. This is a direct spatio-temporal generalization of the integral image described by Viola and Jones [18]. An integral video at pixel location (x, y) and time t is defined as the sum of all pixels at locations less than or equal to (x, y, t). More formally,

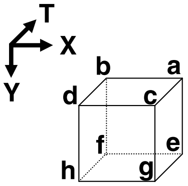


Figure 6: The volume of this box can be computed from the integral video with eight array references: $e - a - f - g + b + c + h - d$.

the integral video iv, is defined as

$$iv(x, y, t) = \sum_{x' \leq x} \sum_{y' \leq y} \sum_{t' \leq t} i(x', y', t'),$$

where $i(x, y, t)$ is the pixel value at the original image. In our framework, we compute two integral volumes for the two optical flow components, and use $v_x(x, y, t)$ and $v_y(x, y, t)$ in place of $i(x, y, t)$. iv can be easily computed using the following recurrences:

$$\begin{aligned} s_1(x, y, t) &= s_1(x, y-1, t) + i(x, y, t) \\ s_2(x, y, t) &= s_2(x-1, y, t) + s_1(x, y, t) \\ iv(x, y, t) &= iv(x, y, t-1) + s_2(x, y, t), \end{aligned}$$

where $s_1(x, -1, t) = s_2(-1, y, t) = iv(x, y, -1) \equiv 0$. Clearly, computing the integral video structure is only marginally more expensive than computing a series of integral images in a video sequence. To compute the one-box feature shown in Figure 6 at any scale, location, or time, only 8 array references to the integral video structure are needed. Care must be taken in real implementations of integral video to maintain numerical precision over long sequences.

4. Learning the Classifier

Since our volumetric features are efficient to compute, we employ a sliding-window approach over the video to detect actions. This is an example of a rare-event detection problem, and fits well in the framework of cascaded classifiers. We use the direct forward feature selection method of Wu *et al.* [21] to select a small subset and arrange them in a cascade for efficient detection. The algorithm selects features and their associated thresholds (termed filters), to classify whether an event occurs in a particular detection volume. The filters are binary classifiers that vote on the classification of the volume. While we could have used Adaboost as did Viola and Jones, Wu *et al.*'s method enables us to train the classifier in a reasonable amount of time despite the much larger set of initial candidate features. We summarize our implementation below.

Given a set of positive examples P, negative examples N, and features F, we construct a cascaded classifier that achieves high detection rate on the positive examples and low detection rate on the negative examples. For each node in the cascade (to be learned), we randomly choose a set of negative examples $N' \subseteq N$, that have been misclassified by the previous stages, where $|N'| = |P|$. Based on P and N', we select the optimal threshold θ_i for each feature $f_i \in F$ that minimizes the classification error rate independent of the other features. The optimal threshold for each feature can be found in $O(|P| \log(|P|))$ time by sorting the feature value of the positive and negative examples. We then iteratively add filters to the ensemble to maximize its detection rate or minimize its false positive rate. The decision output of the ensemble is simply the majority vote of the individual filters. The stopping criteria for a node in the cascade is when its target detection rate (100%) and false positive rate (less than 20%) are reached. Once the stopping criteria is reached, we eliminate the negative examples that were correctly classified and train the next node in the cascade using the remaining examples.

While training the cascade, it is critical to have a representative sampling of the space of negative samples. First, it is very difficult to pre-compute all of the negative examples. Because the cascade must have a very low false positive rate, we quickly exhaust all of the pre-computed negative examples after training only a few nodes. Second, this problem is compounded by our use of three dimensional features, where the feature space is much larger than the ones used for object detection on images. Sung and Poggio [17] propose a bootstrapping method which we use to generate more negative examples after training each node. As we train the new nodes in the cascade, we first run the already-trained cascade on video sequences and extract all of the false detections. We use these detections as negative examples to train the new node, guaranteeing a sufficient supply of negative examples.

5. Detection

To detect events in a video sequence, we first compute the integral video of the sequence. We then scan a detection volume over all locations in space and time. To make the detector work at different scales, we vary the width and height of the detection volume and the associated filters. This is analagous to varying the size of the detection window in 2D object detection on images. Our smallest window size is 64×64 pixels. We slide the position of the window in increments of 1/16 of the window size, and we increase the window size in increments of 1.25 in scale. A complication is that different people can perform the same actions at different speeds. In principle, this would indicate that we should also vary the time scale of the detection window. However, we chose instead to train the detector on actions performed at varying speeds and to detect actions using a fixed time scale window. Section 6.2 confirms that this approach works well in practice.

As with all similar scanning-window type detection systems, there will be multiple detections at adjacent locations, scales, and time around each "true" event. One might argue that this is a drawback of such systems, but we use it to our advantage to increase detection precision. Since we employ a purely discriminative technique (binary classifier), there is no explicit notion of detection "strength". However, the number of detections found in a small area in space-time *is* indicative of the quality of the detections. In other words, a dense cluster of detections is likely to indicate a true detection, while isolated detections are more likely to be false positives. We model this formally as follows, similar to Rosenberg's work in object detection [14].

Suppose that our goal is to detect the event E at space-time location L. Let D_i indicate that the detector also detects this event at some nearby space-time location L_i relative to L. Intuitively, the more D_i's that are true, the more likely that the event E occurred at L. Given the set of D_i's, we wish to find the likelihood ratio between the event occurring and the event not occurring at L, and threshold at some value:

$$\frac{P(E_L|\{D_i \dots D_n\})}{P(\neg E_L|\{D_i \dots D_n\})} > T_1.$$

Using the naive Bayes assumption, this is simply

$$\frac{P(D_1|E_L)\dots P(D_n|E_L)}{P(D_1|\neg E_L)\dots P(D_n|\neg E_L)} > T_2.$$

Assuming that $P(D_n|\neg E_L)$ has identical uniform distribution, this becomes

$$P(D_1|E_L)\dots P(D_n|E_L) > T_3$$

$$\sum_{i=1\dots n} \log P(D_i|E_L) > T_4,$$

which is equivalent to convolving the detections with a kernel that is dependent on the distribution of D_i's, and thresholding the output. We model this distribution as a 3D Gaussian kernel with diagonal covariance. We find the peaks of all clusters that are greater than the threshold and report them as events.

6. Evaluation

The first set of experiments examine our system's ability to detect and recognize non-periodic events in a long video sequence. Our detector is trained and tested on real videos with the *sit-down*, *stand-up*, *close-laptop*, and *grab-cup* actions.[1] We discuss our system's robustness to different camera views, scale variations, and changing speeds at which actions are performed. We also modify our action detector to do action classification in to evaluate our system on existing public data sets. We compute optical flow vectors using two off-the-shelf algorithms (Gautama *et al.* [5] and standard Lucas-Kanade [8]) and achieve similar results.

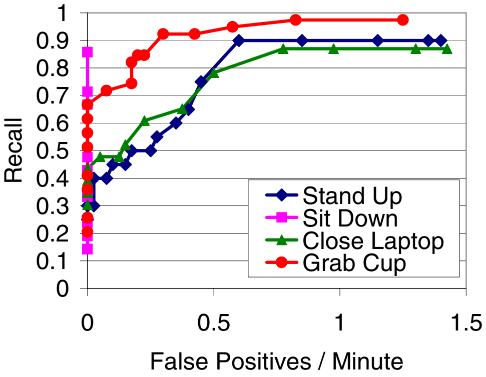


Figure 7: ROC curves for our activity detectors. Note that the sit detector achieved 86% detection rate with no false positives.

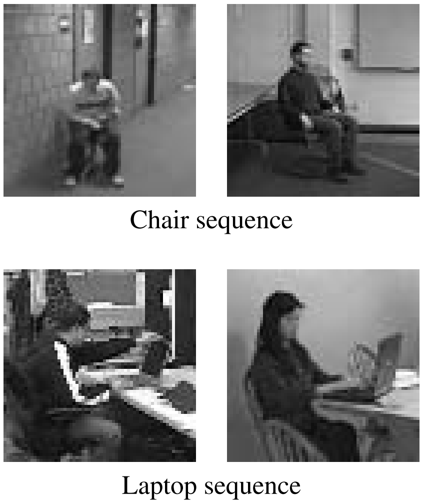
6.1 Action Detection

We train our system to detect the following four non-periodic actions: *sit-down*, *stand-up*, *close-laptop*, and *grab-cup*. The training set consists of four people performing a total of 60 repetitions of these actions. The test set consists of a different group of five people performing 20 repetitions. The ROC curve (see Figure 7) is generated by varying the threshold in the likelihood ratio test. For each action detector, we measure its false positive rate by running it on 40 minutes of video with moderate amounts of movement where none of these actions occur. The *sit-down* detector performs extremely well; it detects 86% of the events with no false positives. The *stand-up* detector achieves 90% recall with 0.6 false positives per minute. The *close-laptop* detector achieves 78% recall at 0.5 false positives per minute. Finally, the *grab-cup* detector achieves 92% recall at only 0.3 false positives per minute. Using Lucas-Kanade for computing the optical flow, our system runs at frame rate on 160×120 pixel size videos.

6.2 Analysis

Our detector is trained on the optical flow of the video, and as with all methods that operate on low level features (as opposed to recovered 3D pose), its performance is dependent on factors such as variations in camera view, the objects' scale, and the speed at which the actions are performed. First, we investigate how changes in camera view affect our system. The effects are motion dependent, where horizontal moving actions are more likely to be distorted than ver-

[1] This dataset is publicly available at our website.


Chair sequence

Laptop sequence

Figure 8: Our system is trained and tested in different environments and camera views.

tical motions as we pan the camera around the room. We give anecdotal evidence that our system is robust to modest changes in camera view, roughly within 45 degrees. Our data is gathered in uncontrolled office environments with no precise placement of cameras. Figure 8 shows two typical frames in the *sit* and *stand* sequences, where the two people face slightly different directions and sit on different chairs.

As described in Section 5, we make our detector scale invariant by adjusting the size of the detection volume. Figure 9 shows the amount of scale variations in the data, where the variations between training and testing objects are 1-2× for our data and 1-3× for Schuldt *et al.*'s data. Part of this scale variation is caused by the fact that we train the detector on 64×64 pixel size videos and test on 160×120 pixel size videos, and part of the variation is caused by the differences in distance of the object to the camera as well as camera zoom. The results show that our system is robust to the scale changes.

Because people perform actions at different speeds, our detector must also be robust to such variations. Figure 10 shows how three people perform the handclapping action at different speeds over 40 frame sequences. The sequences are shown to be aligned at the start of the motion, which we define to be when the hands are closed. Note that the actions diverge at the end of the sequences, where one person's arms are closed while another's arms are open. Our classifier automatically learns to ignore the noisy tail ends of the sequences. This is shown at the bottom of Figure 10, where the density of filters is greater near the beginning of the sequence.

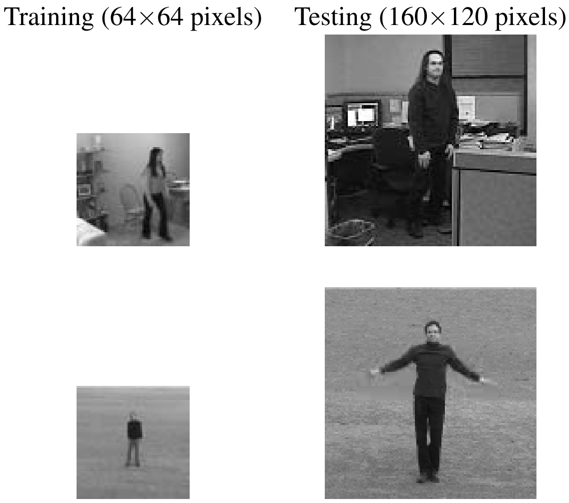


Figure 9: Examples of the amount of scale changes in our data (2×) and Schuldt *et al.*'s data (3×). We trained the system using 64×64 pixel size videos (left) and tested on 160×120 pixel size videos (right).

6.3 Action Classification

The second set of experiments compares our method against Schuldt *et al.*'s [15] in classifying periodic actions performed in short video sequences. We use the same training and test sequences as in their paper, which contains eight people in the training set and nine in the testing. Each person repeats six actions (walk, jog, run, box, clap, and wave) in each of four scenarios (outdoor, outdoor + camera zoom, outdoor + change of clothes, indoor) for a total of about twenty seconds. Since we designed our detector to recognize aperiodic motion, for training we manually segment several instances of each action, all starting at the same phase to be positive examples. For example, we define a simple hand wave action to be the one shown in Figure 11. For each action, we choose random parts of the other action sequences in addition to the pre-recorded videos to generate the negative examples. Figure 12 shows examples of the first few filters chosen to classify the handwave and boxing motions. It is easy to see that the handwave classifier is quite symmetric, capturing motion on both sides of the body. On the other hand, the boxing filters are on one side and high, representing the punch thrown by the boxer. We classify the entire sequence by summing likelihood ratios that pass the threshold and select the action whose detector reported the highest sum.

Tables 1 and 2 show the confusion matrix for these six actions for each technique. The trace of the matrix is a measure the accuracy of the classifier, ranging from 100 (expected value for random classification) to 600 (for perfect classification). The trace of our matrix is 377.8 while the trace of Schuldt *et al.*'s matrix is 430.3. Our results are only

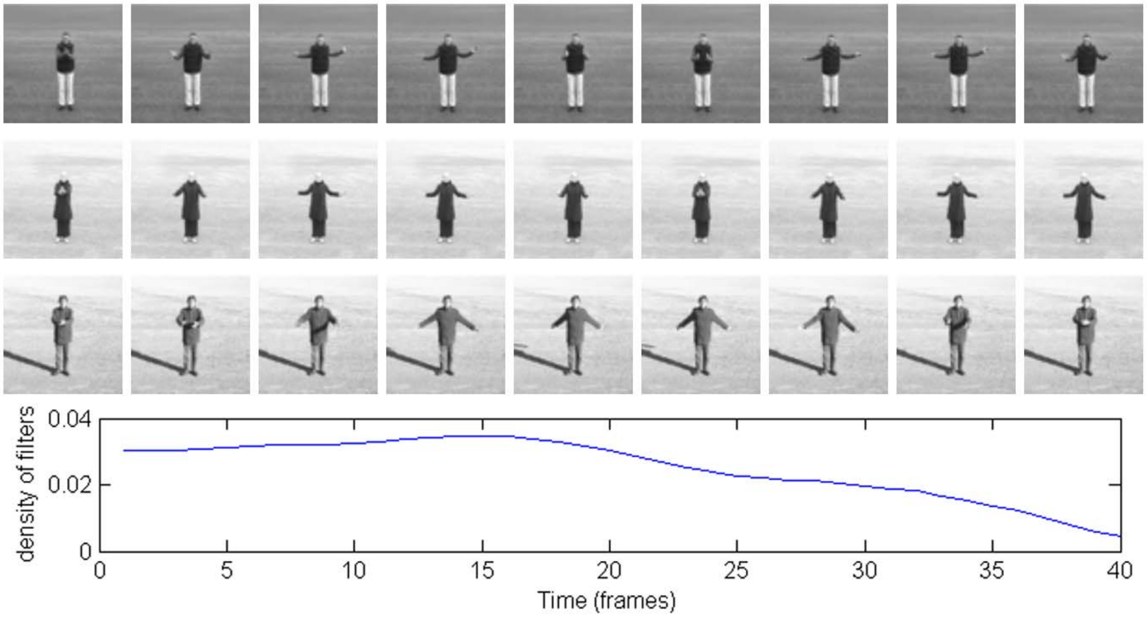


Figure 10: This figure shows three people performing the handclapping action at different speeds. Since the beginning of the sequences are aligned, our classifier learns that this region is more discriminative. The classifier selects more filters to classify the start of the sequences and fewer filters to classify the end of the sequences, where they diverge.

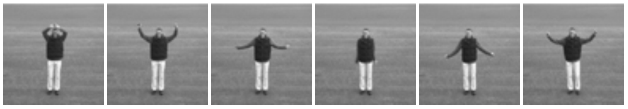
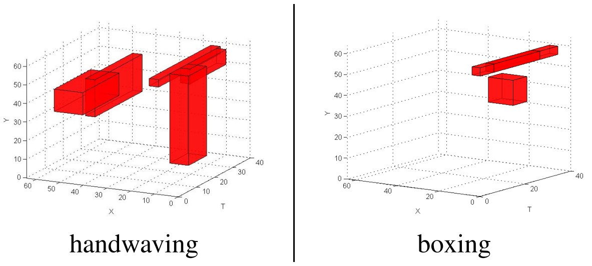


Figure 11: One cycle of the handwave motion, from the Schuldt *et al.* database.



handwaving

boxing

Figure 12: Examples of first few filters learned for the handwaving and boxing actions.

slightly worse, which is encouraging since our system was trained to detect a single instance of each action within arbitrary sequences while Schuldt *et al.*'s system has the easier task of classifying each complete sequence (containing several repetitions of the same action) into one of six classes. These experiments also highlight some of the limitations of relying exclusively on motion information. Figure 13 shows an example where a "clapping" motion incorrectly triggers the "boxing" detector because the left-to-right motion in the shaded area is the same for both motions, even though they vary considerably in appearance.

Table 1: Confusion matrix using our method. Trace = 377.8.

	walk	jog	run	box	clap	wave
walk	80.6	11.1	8.3	0.0	0.0	0.0
jog	30.6	36.1	33.3	0.0	0.0	0.0
run	2.8	25.0	44.4	0.0	27.8	0.0
box	0.0	2.8	11.1	69.4	11.1	5.6
clap	0.0	0.0	5.6	36.1	55.6	2.8
wave	0.0	5.6	0.0	2.8	0.0	91.7

Table 2: Schuldt's LF + SVM confusion matrix. Trace = 430.3.

	walk	jog	run	box	clap	wave
walk	83.8	16.2	0.0	0.0	0.0	0.0
jog	22.9	60.4	16.7	0.0	0.0	0.0
run	6.3	38.9	54.9	0.0	0.0	0.0
box	0.7	0.0	0.0	97.9	0.7	0.7
clap	1.4	0.0	0.0	35.4	59.7	3.5
wave	0.7	0.0	0.0	20.8	4.9	73.6

7. Conclusion

We introduced a novel volumetric feature framework for analyzing video by extending Viola and Jones' work for static-scene object detection to the spatio-temporal domain. Applying this framework on the video's optical flow, we learn activity detectors that can recognize events in video. We have shown that computing integral videos and box features

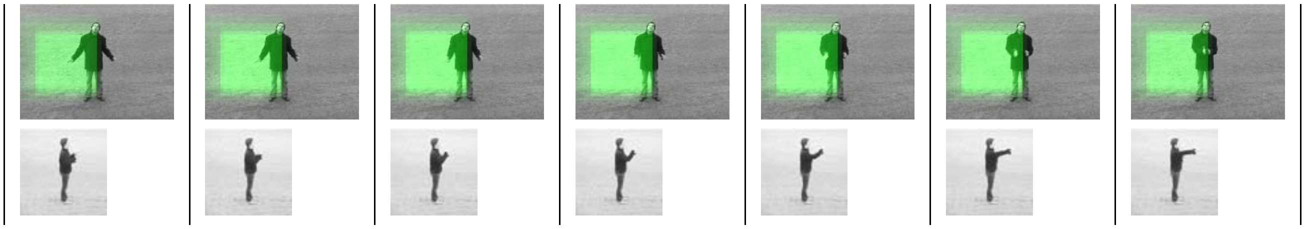


Figure 13: An example of an incorrect detection: the "clapping" motion shown in top row triggers the "boxing" detector. This is because the left-to-right motion in the shaded area (caused by the clap) is very similar to the motion generated by the extension of the arm in the boxing action shown in the bottom row. This illustrates the limitation of relying on motion features alone, as a model of appearance would easily be able to distinguish between the two actions.

is only a small constant factor slower than that of a series of integral images, and therefore we achieve real time recognition performance. We have demonstrated good performance on detecting non-periodic motions with low false positive rates on long sequences of video. Further, despite camera movements and scale changes, our results on classifying the six actions are comparable to the system by Schuldt *et al.*

We have successfully shown that object detection in static scenes and activity recognition in video can be integrated into a common framework. One possible future direction is to add an appearance model to increase accuracy. At the very least, this will enable us to differentiate between the clapping and boxing actions shown in Figure 13 by discriminating between pixels that correspond to a person rather than the background.

8. Acknowledgments

We would like to thank D. Hoiem and A. Efros for their useful feedback on this research. Yan Ke is supported by the Intel Research Scholar program.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 1999.

[2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Trans. PAMI*, 2001.

[3] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. CVPR*, 1997.

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

[5] T. Gautama and M. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Trans. Neural Networks*, 13(5), 2002.

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

[7] D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.

[8] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Image Understanding Workshop*, 1981.

[9] D. J. Moore, I. A. Essa, and M. H. Hayes III. Exploiting human actions and object context for recognition tasks. In *Proc. ICCV*, 1999.

[10] P. Peursum, S. Venkatesh, G. A. W. West, and H. H. Bui. Object labelling from human action recognition. In *IEEE Intl. Conf. Pervasive Computing and Communications*, 2003.

[11] R. Polana and R. Nelson. Detection and recognition of periodic, nonrigid motion. *IJCV*, 23(3), 1997.

[12] D. Ramanan and D. A. Forsyth. Using temporal coherence to build models of animals. In *Proc. ICCV*, 2003.

[13] N. Rea, R. Dahyot, and A. Kokaram. Semantic event detection in sports through motion understanding. In *Proc. Intl. Conf. Image and Video Retrieval*, 2004.

[14] C. Rosenberg. *Semi-Supervised Training of Models for Appearance-Based Statistical Object Detection Methods*. Ph.D. Thesis, Carnegie Mellon University, 2004.

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

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

[17] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. PAMI*, 1998.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, 2001.

[19] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. ICCV*, 2003.

[20] P. Wang, R. Cai, and S.-Q. Yang. Tennis video analysis based on transformed motion vectors. In *Proc. Intl. Conf. Image and Video Retrieval*, 2004.

[21] J. Wu, J. M. Rehg, and M. D. Mullin. Learning a rare event detection cascade by direct feature selection. In *Proc. NIPS*, 2002.

[22] L. Zelnik-Manor and M. Irani. Event-based video analysis. In *Proc. CVPR*, 2001.