Behavior Recognition via Sparse Spatio-Temporal Features

Piotr Dollár Vincent Rabaud Garrison Cottrell Serge Belongie

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093 USA
http://vision.ucsd.edu

Abstract

A common trend in object recognition is to detect and leverage the use of sparse, informative feature points. The use of such features makes the problem more manageable while providing increased robustness to noise and pose variation. In this work we develop an extension of these ideas to the spatio-temporal case. For this purpose, we show that the direct 3D counterparts to commonly used 2D interest point detectors are inadequate, and we propose an alternative. Anchoring off of these interest points, we devise a recognition algorithm based on spatio-temporally windowed data. We present recognition results on a variety of datasets including both human and rodent behavior.

1. Introduction

In this work we develop a general framework for detecting and characterizing behavior from video sequences, making few underlying assumptions about the domain and subjects under observation. Consider some of the well known difficulties faced in behavior recognition. Subjects under observation can vary in posture, appearance and size. Occlusions and complex backgrounds can impede observation, and variations in the environment, such as in illumination, can further make observations difficult. Moreover, there are variations in the behaviors themselves.

Many of the problems described above have counterparts in object recognition. The inspiration for our approach comes from approaches to object recognition that rely on sparsely detected features in a particular arrangement to characterize an object, e.g. [6, 1, 18]. Such approaches tend to be robust to pose, image clutter, occlusion, object variation, and the imprecise nature of the feature detectors. In short they can provide a robust descriptor for objects without relying on too many assumptions.

We propose to characterize behavior through the use of spatio-temporal feature points (see figure 1). A spatio-temporal feature is a short, local video sequence such as an eye opening or a knee bending, or for a mouse a paw rapidly moving back and forth. A behavior is then fully described in terms of the types and locations of feature points present. The motivation is that an eye opening can be characterized as such regardless of global appearance, posture, nearby motion or occlusion and so forth, for example, see figure 2. The complexity of discerning whether two behaviors are similar is shifted to the detection and description of a rich set of features.

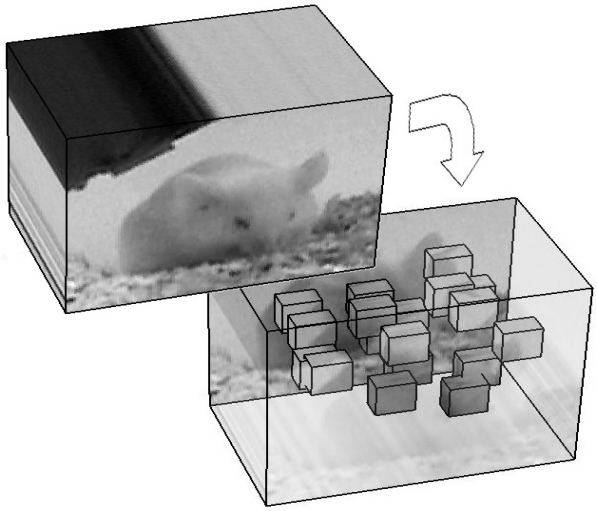

Figure 1: Visualization of cuboid based behavior recognition. Spatio-temporal volume of mouse footage shown at top. We apply a spatio-temporal interest point detector to find local regions of interest in space and time (cuboids) which serve as the substrate for behavior recognition.

Although the method is inspired by approaches to object recognition that rely on spatial features, video and images have distinct properties. The third dimension is temporal, not spatial, and must be treated accordingly. Detection of objects in 3D spatial volumes is a distinct problem, see for example [8].

In this work we show that direct 3D counterparts to commonly used 2D interest point detectors are inadequate for detection of spatio-temporal feature points and propose an alternative. We also develop and test a number of descrip-

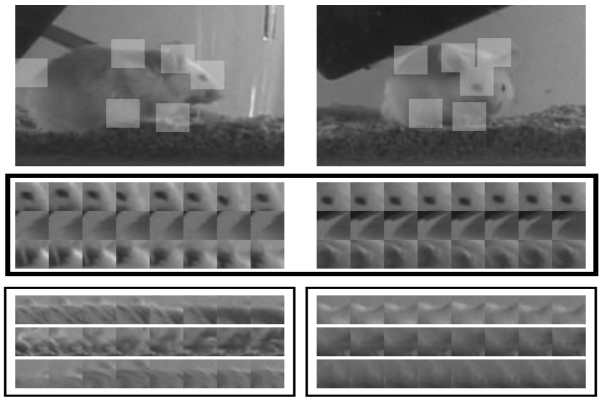

Figure 2: Example of six cuboids extracted from each of two different sequences of grooming, a single frame shown from each original sequence. Each cuboid is shown flattened with respect to time. Note that although the posture of the mouse is quite different in the two cases, three of the six cuboids (shown in the top three rows) for each mouse are quite similar. The other three have no obvious correspondences although its very hard to perceive what these are without motion.

tors to characterize the cuboids of spatio-temporally windowed data surrounding a feature point. Cuboids extracted from a number of sample behaviors from a given domain are clustered to form a dictionary of cuboid prototypes. The only information kept from all subsequent video data is the location and type of the cuboid prototypes present. We argue that such a representation is sufficient for recognition and robust with respect to variations in the data. We show applications of this framework, utilizing a simple behavior descriptor, to three datasets containing human and mouse behaviors, and show superior results over a number of existing algorithms.

The structure of the paper is as follows. In Section 2 we discuss related work. We describe our algorithm in Section 3. In Section 4 we present a detailed comparison of the performance of our algorithm versus existing methods on various datasets. We conclude in Section 5.

2. Related Work

Tracking and behavior recognition are closely related problems, and in fact many traditional approaches to behavior recognition are based on tracking models of varying sophistication, from paradigms that use explicit shape models in either 2D or 3D to those that rely on tracked features; for a broad overview see [9]. The basic idea is that given a tracked feature or object, its time series provides a descriptor that can be used in a general recognition framework.

In the domain of human behavior recognition for example, an entire class of approaches for recognition is based on first recovering the location and pose of body parts, see for example [29, 3]. However, it is unclear how to extend paradigms that rely on articulated models in either 2D or 3D to domains where behavior is not based on changes in configurations of rigid parts, as is the case for recognition of rodent behavior. Perhaps more fundamental, however, is that even in domains where explicit shape models are applicable, it is often very difficult to fit the models to the data accurately.

Another class of approaches performs recognition by first tracking a number of spatial features. [27] use spatial arrangements of tracked points to distinguish between walking and biking, using the intuition that people can identify such behaviors from Johansson displays. [24] use view invariant aspects of the trajectory of a tracked hand to differentiate between actions such as opening a cabinet or picking up an object. Recognition can also proceed from tracked contours, such as in [13].

In response to the practical difficulties of feature and contour tracking, [23] and [28] introduced the framework of 'tracking as repeated recognition,' in which the recovery of pose and body configuration emerges as a byproduct of frame-by-frame recognition using a hand labeled dataset of canonical poses. These approaches are based on the comparison of Canny edges. While the assumptions of edge detection are less restrictive than those of feature or contour tracking, it is still unreliable in domains with cluttered or textured backgrounds or in which the object of interest has poor contrast.

The work of Efros et al. [5] focuses on the case of low resolution video of human behaviors, targeting what they refer to as 'the 30 pixel man.' In this setting they propose a spatio-temporal descriptor based on optical flow measurements, and apply it to recognize actions on ballet, tennis and football datasets. Our proposed method bears some similarity to this approach, but is categorically different in that it uses local features rather than a global measurement. Earlier approaches in this vein are those of [30] and [4]. In [30] for example, Zelnik-Manor and Irani use descriptors based on global histograms of image gradients at multiple temporal scales. The approach shows promise for coarse video indexing of highly visually distinct actions. We examine the approaches of [30] and [5] in more detail in section 4.3.

Most closely related to our work is that of [26], who also use sparsely detected spatio-temporal features for recognition, building on the work on spatio-temporal feature detectors by [17]. They show promising results in human behavior recognition, demonstrating the potential of a method based on spatio-temporal features in a domain where explicit shape models have traditionally been used. The spatio-temporal detector, feature descriptor and behavior descriptor employed in their approach differ from ours. Their method assumes fixed length behaviors, and the similarity between a pair of behaviors is found using a greedy match of the features where multiple features can map to the same corresponding feature.

3. Proposed Algorithm

In the following sections we describe our algorithm in detail. In Section 3.1 we talk about detection of spatial interest points and extensions to the spatio-temporal domain. We describe cuboids in more detail in Section 3.2, and in Section 3.3 we describe the use and importance of cuboid prototypes. We describe the very simple behavior descriptor used in all of our experiments in Section 3.4.

3.1. Feature Detection

A variety of methods exist to detect interest points in the spatial domain, for an extensive review and comparison of methods see [25]. Typically, a response function is calculated at every location in the image and feature points correspond to local maxima.

One of the most popular approaches to interest point detection in the spatial domain is based on the detection of corners, such as [11, 7]. Corners are defined as regions where the local gradient vectors point in orthogonal directions. The gradient vectors are obtained by taking the first order derivatives of a smoothed image $L(x, y, \sigma) = I(x, y) * g(x, y, \sigma)$, where g is the Gaussian smoothing kernel. σ controls the spatial scale at which corners are detected. The response strength at each point is then based on the rank of the covariance matrix of the gradient calculated in a local window. Different measures of the rank lead to slightly different algorithms.

Another common approach is to use the Laplacian of Gaussian (LoG) for the response function. For example, Lowe [19] proposes to use an approximation of the LoG based on the difference of the image smoothed at different scales. Specifically, his response function is $D = (g(\cdot; k\sigma) - g(\cdot; \sigma)) * I = L(\cdot; k\sigma) - L(\cdot; \sigma)$ where k is a parameter that controls the accuracy of the approximation; D tends to the scale normalized LoG as k goes to 1. Under varying conditions either the LoG or Harris detector may have better performance; [21] proposes a technique that incorporates both approaches.

Kadir and Brady [14] approach feature detection with the specific goal of detecting features for object recognition. They define a local measure of patch complexity and look for points the maximize this measure spatially and across scales. Motivation for this type of approach is that salient point are precisely those which maximize discriminability between the objects. This feature detector was used by [6] in their object recognition framework.

3.1.1. Extensions to the Spatio-Temporal Case

The general idea of interest point detection in the spatio-temporal case is similar to the spatial case. Instead of an image $I(x, y)$, interest point detection must operate on a stack of images denoted by $I(x, y, t)$. Localization must proceed not only along the spatial dimensions x and y but also the temporal dimension t. Likewise, detected features also have temporal extent.

The only spatio-temporal interest point operator that we know of is an extension of the Harris corner detect to the 3D case, which has been studied quite extensively by Laptev and Lindeberg (for a recent work see [17]). The basic idea is simple and elegant. Gradients can be found not only along x and y, but also along t, and spatio-temporal corners are defined as regions where the local gradient vectors point in orthogonal directions spanning x, y and t. Intuitively, a spatio-temporal corner is an image region containing a spatial corner whose velocity vector is reversing direction. The second moment matrix is now a 3×3 matrix, and the response function is again based on the rank of this matrix.

The generalized Harris detector described above has many interesting mathematical properties, and in practice it is quite effective at detecting spatio-temporal corners. As mentioned [26] used spatio-temporal features detected by the generalized Harris detector to build a system that distinguishes between certain human behaviors. The behaviors their method can discriminate amongst, including walking, jogging, running, boxing, clapping and waving, are in fact well characterized by the reversal in the direction of motion of arms and legs. Hence these behaviors give rise to spatio-temporal corners, so the technique is well suited for dealing with their dataset.

In certain problem domains, e.g., rodent behavior recognition or facial expressions, we have observed that true spatio-temporal corners are quite rare, even when seemingly interesting motion is occurring. Sparseness is desirable to an extent, but features that are too rare can prove troubling in a recognition framework, as observed by Lowe [19].

In addition to the rarity of spatio-temporal corners, a more general question that remains unanswered is whether spatio-temporal corners are in fact the features one needs for general behavior recognition. Analogous to useful features for object recognition, we are interested in precisely those features that maximize discrimination between behaviors. Consider two examples, the jaw of a horse chewing on hay and the spinning wheel of a bicycle. Neither example gives rise to a spatio-temporal corner as the motions are subtle and gradually changing, yet both seem like particularly relevant features for behavior recognition.

We propose an alternative spatio-temporal feature detector for our behavior recognition framework. We have explicitly designed the detector to err on the side of detecting too many features rather than too few, noting that object recognition schemes based on spatial interest points deal well with irrelevant and possibly misleading features generated by scene clutter and imperfect detectors [19]. The resulting representation is still orders of magnitude sparser than a direct pixel representation.

Like much of the work on interest point detectors, our response function is calculated by application of separable linear filters. We assume a stationary camera or a process that can account for camera motion. The response function has the form $R = (I * g * h_{ev})^2 + (I * g * h_{od})^2$ where $g(x, y; \sigma)$ is the 2D Gaussian smoothing kernel, applied only along the spatial dimensions, and h_{ev} and h_{od} are a quadrature pair [10] of 1D Gabor filters applied temporally. These are defined as $h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{od}(t; \tau, \omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$. In all cases we use $\omega = 4/\tau$, effectively giving the response function R two parameters σ and τ, corresponding roughly to the spatial and temporal scale of the detector.

The detector is tuned to fire whenever variations in local image intensities contain periodic frequency components. In general there is no reason to believe that only periodic motions are interesting. Periodic motions, such as a bird flapping its wings, will indeed evoke the strongest responses, however, the detector responds strongly to a range of other motions, including at spatio-temporal corners. In general, any region with spatially distinguishing characteristics undergoing a complex motion can induce a strong response. Areas undergoing pure translational motion will in general not induce a response, as a moving, smoothed edge will cause only a gradual change in intensity at a given spatial location. Areas without spatially distinguishing features cannot induce a response.

3.2. Cuboids

At each interest point (local maxima of the response function defined above), a cuboid is extracted which contains the spatio-temporally windowed pixel values. The size of the cuboid is set to contain most of the volume of data that contributed to the response function at that interest point; specifically, cuboids have a side length of approximately six times the scale at which they were detected.

To compare two cuboids, a notion of similarity needs to be defined. Given the large number of cuboids we deal with in some of the datasets (on the order of 10^5), we opted to use a descriptor that could be computed once for each cuboid and compare using Euclidean distance.

The simplest cuboid descriptor is a vector of flattened cuboid values. More generally, a transformation can be applied to the cuboid, such as normalization of the pixel values, and given the transformed cuboid, various methods can be employed to create a feature vector, such as histogramming. The goal of both phases is to create a descriptor with invariance to small translations, slight variation in appearance or motion, changes in lighting, and so on, while retaining the descriptor's discriminative power. Instead of trying to predict the right balance between invariance and discriminative power, we design a number of descriptors and test each in our recognition framework.

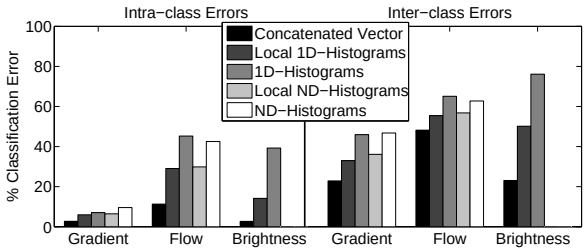

Figure 3: Shown is the intra and inter class performance of our recognition method on the face dataset using different cuboid descriptors. The full algorithm, dataset and methodology are discussed later, the sole purpose of this figure is to give a sense of the relative performance of the various cuboid descriptors. Recall that the descriptors we use involve first transforming the cuboid into: (1) normalized brightness, (2) gradient, or (3) windowed optical flow, followed by a conversion into a vector by (1) flattening, (2) global histogramming, or (3) local histogramming, for a total of nine methods, along with multi-dimensional histograms when they apply. Using the gradient in any form gave very reliable results, as did using the flattened vector of normalized brightness values.

The transformations we apply to each cuboid include: (1) normalized pixel values, (2) the brightness gradient, and (3) windowed optical flow. The brightness gradient is calculated at each spatio-temporal location (x, y, t), giving rise to three channels (G_x, G_y, G_t) each the same size as the cuboid. To extract motion information we calculate Lucas-Kanade optical flow [20] between each pair of consecutive frames, creating two channels (V_x, V_y). Each channel is the same size as the cuboid, minus one frame.

We use one of three methods to create a feature vector given the transformed cuboid (or multiple resulting cuboids when using the gradient or optical flow). The simplest method involves flattening the cuboid into a vector, although the resulting vector is potentially sensitive to small cuboid perturbations. The second method involves histogramming the values in the cuboid. Such a representation is robust to perturbations but also discards all positional information (spatial and temporal). Local histograms, used as part of Lowe's 2D SIFT descriptor [19], provide a compromise solution. The cuboid is divided into a number of regions and a local histogram is created for each region. The goal is to introduce robustness to small perturbations while retaining some positional information. For all the methods, to reduce the dimensionality of the final descriptors we use PCA [12].

Many of the above choices were motivated by research in descriptors for 2D features (image patches). For a detailed review of 2D descriptors see [22]. Other spatio-temporal descriptors are possible. For example, al. [26] used differential descriptors [16] for their spatio-temporal interest points, however, among the descriptors examined for 2D features, differential descriptors are not particularly robust.

We tested the performance of our overall algorithm changing only the cuboid descriptor on a dataset described later in this paper. Results are shown in figure 3. His-

tograms, both local and global did not provide improved performance; apparently the added benefit of increased robustness was offset by the loss of positional information. In all experiments reported later in the paper we used the flattened gradient as the descriptor, which is essentially a generalization of the PCA-SIFT descriptor [15].

3.3. Cuboid Prototypes

Our approach is based on the idea that although two instances of the same behavior may vary significantly in terms of their overall appearance and motion, many of the interest points they give rise to are similar. Under this assumption, even though the number of possible cuboids is virtually unlimited, the number of different *types* of cuboids is relatively small. In terms of recognition the exact form of a cuboid becomes unimportant, only its type matters.

We create a library of cuboid prototypes by clustering a large number of cuboids extracted from the training data. We cluster using the k-means algorithm. The library of cuboid prototypes is generated separately for each dataset since the cuboids types are very different in each (mouse cuboids are quite distinct from face cuboids). Clusters of cuboids tend to be perceptually meaningful.

Using cluster prototypes is a very simple yet powerful method for reducing variability of the data while maintaining its richness. After the training phase, each cuboid detected is either assumed to be one of the known types or rejected as an outlier.

Intuitively the prototypes serve a similar function as parts do in object recognition. The definition of parts varies widely in the literature on object recognition, the analogy here is most applicable to the work of [6] and especially [1], who refer to the local neighborhoods of spatially detected interest points as parts. In the case of static face detection, these might include the eyes or hairline features.

3.4. Behavior Descriptor

After extraction of the cuboids the original clip is discarded. The rationale for this is that once the interest points have been detected, together their local neighborhoods contain all the information necessary to characterize a behavior. Each cuboid is assigned a type by mapping it to the closest prototype vector, at which point the cuboids themselves are discarded and only their type is kept.

We use a histogram of the cuboid types as the behavior descriptor. Distance between the behavior descriptors (histograms) can be calculated by using the Euclidean or χ^2 distance. When more training data is available, we use the behavior descriptor and class labels in a classification framework.

The relative positions of the cuboids are currently not used. Previously mentioned algorithms for object recognition, such as [6] or [1] could be used as models for how to incorporate positional information.

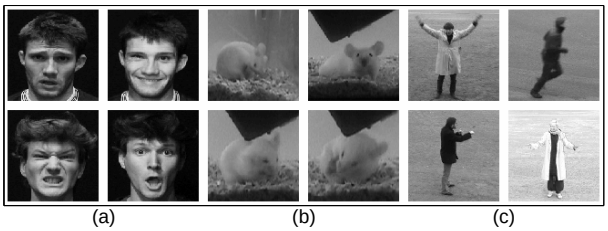

Figure 4: Representative frames from clips in each domain: (a) facial expressions, (b) mouse behavior, and (c) human activity.

4. Experiments

We explore results in three representative domains: facial expressions, mouse behavior and human activity. Representative frames are shown in figure 4. To judge the performance of our algorithm, we compare to results obtained using three other general activity recognition algorithms on these datasets. Each domain presents its own challenges and demonstrates various strengths and weaknesses of each algorithm tested.

We describe each dataset in detail in the following section, training and testing methodology in Section 4.2, the algorithms used for comparison in Section 4.3, and finally detailed results in Section 4.4.

4.1. Datasets

We compiled the facial expressions and mouse behavior datasets ourselves, they are available for download at `http://vision.ucsd.edu`. The human activity dataset was collected by [26] and is available online.

The face data involves 2 individuals, each expressing 6 different emotions under 2 lighting setups. The expressions are anger, disgust, fear, joy, sadness and surprise. Certain expressions are quite distinct, such as sadness and joy, others are fairly similar, such as fear and surprise. Under each lighting setup, each individual was asked to repeat each of the 6 expressions 8 times. The subject always starts with a neutral expression, expresses an emotion, and returns to neutral, all in about 2 seconds.

The mouse data includes clips taken from seven fifteen minute videos of the same mouse filmed at different points in the day. The set of behaviors includes drinking, eating, exploring, grooming and sleeping. The number of occurrences and characteristics of each behavior vary substantially for each of the seven videos; clips extracted from each video are kept separate. A total of 406 clips were extracted ranging from 14 occurrences of drinking to 159 occurrences of exploring, each lasting between 1 and 10 seconds. Typical mouse diameter is approximately 120 pixels although the mouse can stretch or compress substantially. All filming was done in the vivarium in which the mice are housed.

The videos were collected with help from veterinarians at the UCSD Animal Care Program, who also advised on how to classify and label the data by hand.

In order to be able to do a full comparison of methods, we also created a greatly simplified, small scale version of the mouse dataset. While the mouse eats, it tends to sit still, and on occasion when it explores it sniffs around but remains stationary. From two different mouse videos we extracted a number of examples of these two behaviors, all of the same (short) duration, and made sure the mouse is spatially centered in each. Data in this form does not benefit our algorithm in any way, however, it is necessary to get results for some of the methods we test against.

The human activity data comes from the dataset collected by [26]. There are 25 individuals engaged in the following activities: walking, jogging, boxing, clapping and waving. We use a subset of the dataset which includes each person repeating each activity 8 times for about 4 seconds each, wearing different clothing (referred to scenarios $s1$ and $s3$), for a total of almost 1,200 clips. The clips have been sub-sampled (people are approximately 80 pixels in height) and contain compression artifacts (this is the version of the dataset available online).

4.2. Methodology

We divide each dataset into groups. The groups we chose for the datasets discussed above are as follows: face clips are divided into 4 groups, one group per person per lighting setup; mouse clips are divided into 7 groups, corresponding to each of the source videos; human activity clips are divided into 25 groups, one per person. We analyze the performance of various algorithms trained on a subset of the groups and tested on a different subset. Often, because of the limited amount of data, we use leave one out cross validation to get an estimate of performance.

All algorithms have parameters that need tuning. In all cases that we report results we report the best performance achieved by a given algorithm – parameter sweeps were done for all the algorithms. As can be seen in figure 5 our method is not very sensitive to the exact parameter settings, in fact, aside from the scale of the cuboids we used the same parameter settings on all three datasets. Some of the algorithms also have a random component (for example a clustering phase), in this case any experiment reported is averaged over 20 runs.

When applicable, we focus on reporting relative performance of the algorithms so as to avoid questions of the absolute difficulty of a given dataset.

4.3. Algorithms for Comparison

We compare our approach to three other methods. Each of these is a general purpose behavior recognition algorithm

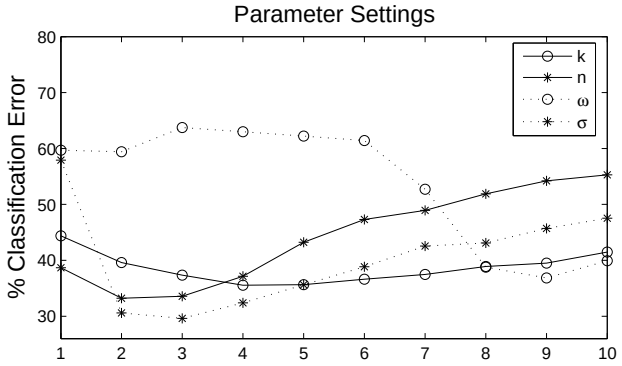

Figure 5: We tested how sensitive the performance of our method is to various parameter settings on the face dataset. In each of the above curves we plot classification error for 10 different settings of a given parameter with all other parameters kept constant at default, 'reasonable' values. The thing to note is that the overall shape of each curve is very smooth and tends to be bowl shaped. The four parameters shown are: k, $50 < k < 500$, the number of clusters prototypes, n, $10 \leq n \leq 200$ the number of cuboids detected per face clips, ω, $0 < \omega < 1$ the overlap allowed between cuboids, and σ, $.2 < \sigma < 9$, the spatial scale of the detector (which also determines the size of the cuboid). Optimal settings were approximately: $k = 250$, $n = 30$, $\omega = .9$ and $\sigma = 2$.

that is capable of dealing with low resolution and noisy data. We implement the algorithms of Efros et al. [5] and Zelnik-Manor and Irani [30], we refer to these as EFROS and ZMI, respectively. We also use a variation of our framework based on the Harris 3D corner detector, described previously. We refer to our framework as CUBOIDS and to the variation using the Harris detector as CUBOIDS+HARRIS[1]. Unless otherwise specified we use 1-nearest neighbor classifier with the χ^2 distance on top of the cuboid representation. We describe EFROS and ZMI in more detail below.

EFROS is used to calculate the similarity of the activity of two subjects using a version of normalized cross correlation on optical flow measurements. Subjects must be tracked and stabilized. If the background is non uniform this can also require figure-ground segmentation. However, when these requirements are satisfied the method has been shown to work well for human activity recognition and has been tested on ballet, tennis and football datasets[2]. EFROS tends to be particularly robust to changes in appearance and has shown impressive results even on very low resolution video.

ZMI works by histogramming normalized gradient measurements from a spatio-temporal volume at various temporal scales, resulting in a coarse descriptor of activity. No assumptions are made about the data nor is tracking or stabilization required. The method's strength lies in distinguishing motions that are grossly different; promising results have been shown on human activities such as running, waving, rolling or hopping. In some sense ZMI and EFROS are complementary algorithms and we could expect one to

[1]This algorithm is very different from the work of [26], the only similarity is that both use features detected by the Harris corner detector.

[2]Unfortunately, these datasets are no longer available.

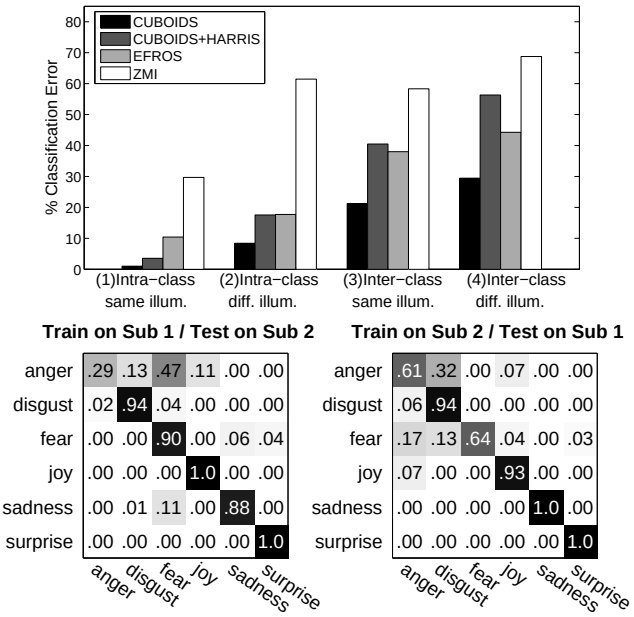

Figure 6: FACE DATASET *Top row:* We investigated how identity and lighting affect each algorithm's performance. In all cases CUBOIDS gave the best results. EFROS and CUBOIDS+HARRIS had approximately equal error rates, except that EFROS tended to perform better under changes in illumination. ZMI was not well suited to discriminating between facial expressions, performing only slightly better than chance . Random guessing would result in 83% error. All algorithms were ran with optimal parameters. *Bottom row:* Inter-class confusion matrices obtained using our method under the first illumination setup on the face data. A majority of the error is caused by anger being confused with other expressions. Subjectively, the two subjects' expression of anger is quite different.

perform well when the other does not.

4.4. Results

In the following sections we show results on the datasets described above: facial expressions, human activity and mouse behavior. In all experiments on all datasets, CUBOIDS had the highest recognition rate, often by a wide margin. Typically the error is reduced by at least a third from the second best method.

4.4.1. Facial Expression

In each experiment, training is done on a single subject under one of the two lighting setups and tested on: (1) the same subject under the same illumination[3], (2) the same subject under different illumination, (3) a different subject under the same illumination, and (4) a different subject under different illumination. Results are shown in figure 6. In all cases CUBOIDS had the highest recognition rates.

4.4.2. Mouse Behavior

The mouse data presents a highly challenging behavior recognition problem. Differences between behaviors can be subtle, optical flow calculations tend to be inaccurate, the mouse blends in with the bedding of the cage, and there are no easily trackable features on the mice themselves (the eyes of the mouse are frequently occluded or closed). The pose of the mouse w.r.t. the camera also varies significantly.

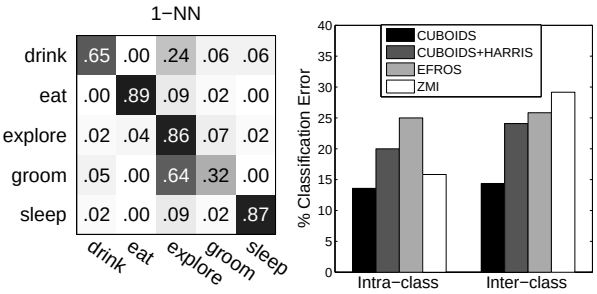

Figure 7: MOUSE DATASET *Left:* Confusion matrix generated by CUBOIDS on the full mouse dataset. As mentioned, this dataset presents a challenging recognition problem. Except for a few difficult categories, recognition rates using our method were fairly high. *Right:* Due to the form of the data, a full comparison of algorithms was not possible. Instead, we created a simple small scale experiment and ran all four algorithms on it. CUBOIDS had the lowest error rates, ZMI was a near second on intra-class error.

Results on the full dataset are presented in figure 7, on the left. The overall recognition rate is around 72%. As mentioned, we also used a simplified, small scale version of the mouse dataset in order to do a full comparison of methods[4]. In both experiments CUBOIDS had the lowest errors, see figure 7, on the right.

4.4.3. Human Activity

For the human activity dataset we used leave one out cross validation to get the overall classification error. Due to the large size of this dataset, we did not attempt a comparison with other methods[5]. Rather, we provide results only to show that our algorithm works well on a diverse range of data. Confusion matrices for the six categories of behavior are shown in figure 8; the overall recognition rate was over 80%.

5. Conclusion

In this work we have shown the viability of doing behavior recognition by characterizing behavior in terms of spatio-temporal features. A new spatio-temporal interest point detector was presented, and a number of cuboid descriptors were analyzed. We showed how the use of cuboid prototypes gave rise to an efficient and robust behavior descrip-

[3] In this case we use leave one out cross validation.

[4] EFROS requires a stabilized figure, and with a non-uniform background stabilization requires figure-ground segmentation, a non-trivial task.

[5] Although the confusion matrices in figure 8 are better than those reported in [26], the results are not directly comparable because the methodologies are different

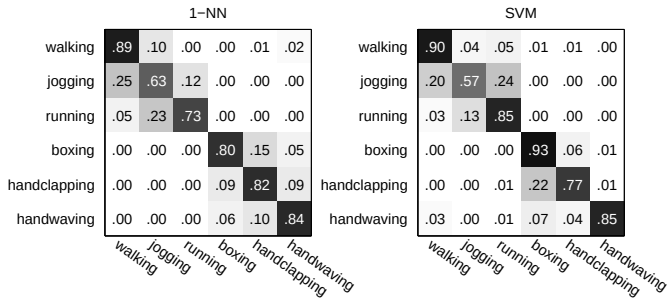

Figure 8: HUMAN ACTIVITY DATASET Shown are confusion matrices generated by CUBOIDS. Two classifiers were used: 1-nearest neighbor and Support Vector Machines with radial basis functions [12]. Using SVMs resulted in a slight reduction of the error. Note that most of the confusion occurs between jogging and walking or running, and between boxing and clapping, most other activities are easily distinguished.

tor. We tested our algorithm in a number of domains against well established algorithms, and in all tests showed the best results.

Throughout we have tried to establish the link between the domains of behavior recognition and object recognition, creating the potential to bring in a range of established techniques from the spatial domain to that of behavior recognition.

Future extensions include using the spatio-temporal layout of the features, extending such approaches as [2] or [1] to the spatio-temporal domain. Using features detected at multiple scales should also improve performance. Another possible direction of future work is to incorporate a dynamic model on top of our representation.

Acknowledgements

The authors wish to thank Kristin Branson, Sameer Agarwal, Josh Wills and Andrew Rabinovich for valuable discussions. We would like to give special thanks to John Wesson for his patience and help with gathering video footage, and also to Keith Jenne, Phil Richter, and Geert Schmid-Schoenbein for plentiful advice. This work was partially supported through subcontract B542001 under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48 and partially by the UCSD division of Calit^2 under the Smart Vivarium project.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 26(11):1475–1490, Nov 2004.

[2] Yali Amit, Donald Geman, and Kenneth Wilder. Joint induction of shape features and tree classifiers. *PAMI*, 19(11):1300–1305, 1997.

[3] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *IJCV*, 56(3):179–194, Feb 2004.

[4] J.W. Davis and A.F. Bobick. The representation and recognition of action using temporal templates. In *CVPR*, pages 928–934, 1997.

[5] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, pages 726–733, Nice, France, 2003.

[6] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[7] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points. In *Intercommission Conf. on Fast Processing of Photogrammetric Data*, pages 281–305, Switzerland, 1987.

[8] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, 2004.

[9] D. M. Gavrila. The visual analysis of human movement: A survey. *CVIU*, 73(1):82–98, January 1999.

[10] G. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[11] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Alvey Conf.*, pages 189–192, 1988.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer Verlag, Basel, 2001.

[13] M.A. Isard and A. Blake. A mixed-state Condensation tracker with automatic model switching. In *ICCV*, pages 107–112, 1998.

[14] T. Kadir and M. Brady. Saliency, scale and image description. *IJCV*, 45(2):83–105, Nov 2001.

[15] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR*, pages 506–513, 2004.

[16] J. Koenderink and A. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–75, 1987.

[17] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003.

[18] B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. In *DAGM*, Aug. 2004.

[19] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, Nov 2004.

[20] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[21] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, pages I: 525–531, 2001.

[22] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, pages II: 257–263, 2003.

[23] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *ECCV*, page III: 666 ff., 2002.

[24] C. Rao and M. Shah. View-invariance in action recognition. In *CVPR*, pages II:316–322, 2001.

[25] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *IJCV*, 37(2):151–172, June 2000.

[26] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, pages III: 32–36, 2004.

[27] Y. Song, L. Goncalves, and P. Perona. Unsupervised learning of human motion. *PAMI*, 25(7):814–827, 2003.

[28] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *ECCV*, page I: 629, 2002.

[29] Y. Yacoob and M.J. Black. Parameterized modeling and recognition of activities. *CVIU*, 73(2):232–247, February 1999.

[30] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *CVPR*, pages II:123–130, 2001.