Learning a Hierarchy of Discriminative Space-Time Neighborhood Features for Human Action Recognition

Adriana Kovashka and Kristen Grauman
Department of Computer Science
University of Texas at Austin
{adriana, grauman}@cs.utexas.edu

Abstract

Recent work shows how to use local spatio-temporal features to learn models of realistic human actions from video. However, existing methods typically rely on a predefined spatial binning of the local descriptors to impose spatial information beyond a pure "bag-of-words" model, and thus may fail to capture the most informative space-time relationships. We propose to learn the shapes of space-time feature neighborhoods that are most discriminative for a given action category. Given a set of training videos, our method first extracts local motion and appearance features, quantizes them to a visual vocabulary, and then forms candidate neighborhoods consisting of the words associated with nearby points and their orientation with respect to the central interest point. Rather than dictate a particular scaling of the spatial and temporal dimensions to determine which points are near, we show how to learn the class-specific distance functions that form the most informative configurations. Descriptors for these variable-sized neighborhoods are then recursively mapped to higher-level vocabularies, producing a hierarchy of space-time configurations at successively broader scales. Our approach yields state-of-the-art performance on the UCF Sports and KTH datasets.

1. Introduction

Automatic recognition of human activities in video would be useful for surveillance, content-based summarization, and human-computer interaction applications, yet it remains a challenging problem. Some approaches seek ways to measure directly how humans are moving in the scene, using techniques for tracking, body pose estimation, or space-time shape templates [27, 28, 24, 10, 29], while others aim to categorize activities based on the video's overall pattern of appearance and motion, often using spatio-temporal interest operators and local descriptors to build the representation [30, 20, 5, 15, 31, 18, 9].

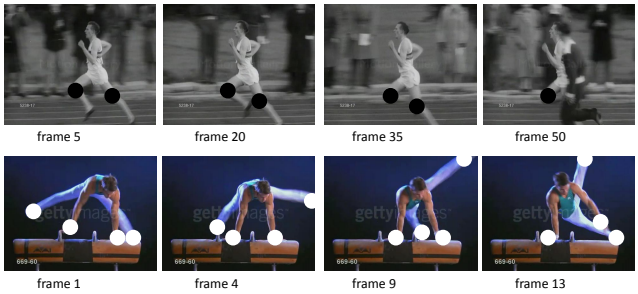

Figure 1. The most discriminative space-time neighborhoods of local descriptors (denoted by circles) may depend on the activity category. For example, for running (first row) a larger temporal extent and smaller spatial extent is most useful, whereas for swinging (second row), the reverse is true. The main idea of the proposed method is to learn class-specific vocabularies of variable-shaped space-time neighborhoods.

In particular, recent work has shown promising results using local spatio-temporal descriptors together with bag-of-words models, where the local features are quantized to form a "visual vocabulary", and a video clip is summarized by the histogram of its feature occurrences [30, 5, 15, 18, 21, 9, 19, 11]. The representation has a number of advantages: being local, the features have robustness to viewpoint changes and occlusions; being relatively sparse, they can be stored and manipulated efficiently. Further, by including both dynamic and static components (e.g., optical flow and gradient histograms), they can capture not only what kind of motion occurs, but also what kind of context and actors are present, without requiring reliable tracks on a particular subject. Various developments building on this general framework have yielded impressive results for realistic activities in Hollywood movies or YouTube videos.

However, a key limitation of spatio-temporal interest point representations is that they can be *too* local, failing to capture adequate spatial or temporal relationships. In the extreme, the orderless bag-of-words lacks cues about

motion trajectories, before-after relationships, or the relative layout of objects and actions. In an attempt to overcome this problem, several alternatives have been proposed to capture mid-level structure using space-time bins of points, with partitions formed either *globally* at the level of the entire clip (e.g., a histogram for the upper third of the frames is recorded separately from one for the lower third) [15, 4, 31, 19, 35, 11], or else in a *feature-centered* manner where a cuboid with multiple sub-bins is used to describe a point's neighborhood [6, 9]. Unfortunately, a global binning makes the representation sensitive to position or time shifts in the clip segmentation, and using predetermined fixed-size space-time grid bins (whether global or feature-centered) assumes that the proper volume scale is known and uniform across action classes. Such uniformity is not inherent in the features themselves, given the large differences between the ways in which they are laid out for different activities (see Figure 1).

To address this problem, we propose to learn the shapes of space-time feature neighborhoods that are most discriminative for a given action category. The idea is to form new features composed of the neighborhoods around the raw initially-detected interest points, taking into account the visual words to which the neighboring features correspond and their orientation with respect to the central interest point. We quantize the resulting neighborhood descriptors to form a higher-level vocabulary in which each word encodes an interest point and the loose configuration of neighbors; repeating the process recursively, we compute a hierarchy of words that capture space-time configurations at successively broader scales.

When identifying which features are nearest to one another to build a neighborhood, there is an important scaling ambiguity—particularly between the spatial and temporal dimensions: i.e., is an interest point three pixels away closer or further than an interest point that is three video frames away? To address this, we generate several candidate distance metrics to evaluate the proximity of points, and learn the combination of variable-sized neighborhood features among all vocabulary levels that is best for the given recognition task. The selected shapes allow our method to capture varying extents of appearance and motion cues. The ultimate classifier used for recognition combines the neighborhoods of different sizes and vocabulary levels to arrive at a rich description of the actions performed in the videos.

We apply the approach to learn human activity categories with the UCF Sports [29] and KTH [30] datasets, and show that it improves the state of the art. We further analyze the advantages of the proposed variable-sized neighborhoods and hierarchy compared to a traditional fixed binning, and examine the types of discriminative neighborhoods our method discovers.

2. Related Work

Activities can be analyzed based on tracked humans and their shapes and limb motions (e.g. [27, 28, 24, 10, 29]), or alternatively, by forgoing direct body tracking and describing the overall appearance and motion patterns within a video clip (e.g. [30, 20, 5, 15, 31, 18, 9]). Work on the activity recognition problem in general is too broad to cover here; we therefore focus the discussion below on the most relevant techniques using interest points, neighborhood features, feature selection, and/or hierarchical representations.

A number of current approaches entail the use of local space-time interest points [30, 5, 20, 15, 3, 18, 31, 4, 19, 11]. Many build representations using visual vocabularies computed with gradient-based descriptors extracted at the interest points [5, 15, 4, 30, 31], while others build descriptors from the point positions themselves [3, 9]. The advantages of combining both static and dynamic descriptors have also been demonstrated [20, 18, 19, 11].

The strategy of generating compound neighborhood-based features—explored initially for static images and object recognition [34, 26, 16, 17, 25]—has since been extended to video. One approach is to subdivide the space-time volume globally using a coarse grid of histogram bins [15, 4, 31, 11]. A second approach is to place grids around the raw interest points, and compute a new representation using the positions of the interest points that fall within the grid cells surrounding that central point [9]. In contrast to these methods, our feature-centered neighborhood descriptors are variable in size and shape, with their extent determined discriminatively per class; further, each compound feature captures both the appearance/motion and relative orientation of the surrounding points.

Feature selection techniques allow activity models to emphasize the most relevant cues. To identify informative local video descriptors, boosting [12, 6], PageRank [18], and item-set mining [23, 9] have all been explored. When using the global grid-based histograms, performance improves when one chooses or learns the most discriminative bins [15, 31, 11]. In particular, the authors of [31, 11] incorporate multiple kernel learning (MKL) to optimize the combination of grid channels; our method also integrates MKL, though in our case it is for the sake of determining which combination of distance metrics between interest points forms the most discriminative neighborhoods.

Aside from discriminative local neighborhoods, the other main theme in our work is to develop a *hierarchy* of descriptors for activity recognition. Related ideas have been considered with static images for object recognition: the "hyperfeatures" of [1] divide the image into a grid of overlapping tiles, with increasing coarseness of the grids at higher levels in the hierarchy; the models in [2, 25] combine raw interest points into local feature types and subsequently parts or objects. The success of these methods

helps motivate our strategy; however, the problem is distinct in video, mainly due to issues of computing neighbors in a joint multi-feature space (both space and time).

Hierarchical representations for local-feature activity recognition have only been explored to a limited extent [20, 9]. The authors of [20] construct a constellation model for actions, where each part is itself a bag-of-words, and show how to use probabilistic latent topic models to introduce a layer between the visual words and video sequences in [21]. Most recently, the authors of [9] explore a recursive use of frequent item-set mining to efficiently obtain informative 2-d interest points within fixed-size quadrants at multiple scales. In contrast, our method finds neighborhoods of varying shape, depending on which layouts enhance the differences between action classes, and our feature encoding relies on the motion and appearance of local features rather than the distribution of interest point positions.

3. Approach

Our approach develops a richer vocabulary for bag-of-words-based activity recognition. The process involves constructing a hierarchy of vocabularies using neighborhoods of spatio-temporal feature points, where the neighborhoods themselves are feature-centered, and their variable shape in the space and time dimensions is automatically learned.

In the following, we define our initial descriptors (Section 3.1), explain how we generate candidate neighborhoods and record their descriptions (Section 3.2), how to construct a hierarchy of those neighborhoods (Section 3.3), and, finally, how we discriminatively learn the combination of neighborhood extents and levels (Section 3.4).

3.1. Interest Points and Initial Descriptors

The inputs to our algorithm are space-time interest points and their associated local descriptors. To detect interest points, we use one of two methods: either a dense sampling throughout the video, or else a sparse sampling using the method of [14], which is a space-time extension of the Harris operator. For the initial descriptors, we use histograms of oriented spatio-temporal gradients, which characterize the motion and appearance within a volume surrounding the interest point. Specifically, for the dense interest points we extract HoG3D descriptors [13], and for the sparse points we use histograms-of-optical-flow (HoF) and histograms-of-oriented-gradients (HoG), as described in [15].

We use standard procedures to form what we call the "level-0" vocabulary: we sample a random set of descriptors from training videos, cluster each feature type separately using k-means, and use the k centers as the visual words. At this point, each feature in a video can be mapped to a level-0 word. Thus, each video clip V is initially described by a set of point-word tuples:

$$V = \{\langle x_1, y_1, t_1, w_1^{(0)}\rangle, \ldots, \langle x_{n_v}, y_{n_v}, t_{n_v}, w_{n_v}^{(0)}\rangle\}, \quad (1)$$

where each $\langle x_i, y_i, t_i, w_i\rangle$ records the spatial position, frame number, and word index, respectively, for one interest point, n_v denotes the video's total number of interest points, $w^{(0)}$ signifies that we are looking at the first-level vocabulary, and $w_i^{(0)} \in \{1, \ldots, k\}$.

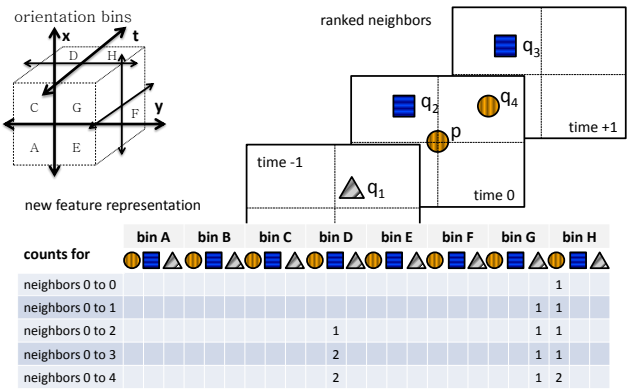

Figure 2. Neighborhood formation overview. The 3d axes in the upper-left depict the 8 orientations relative to the central point. Each letter denotes a space-time orientation. In the center, we depict three frames with their features and corresponding visual words (shapes represent word identity). The histogram at the bottom is our neighborhood representation. For example, neighbor 1 (q_1) is (G=*above, right*, and *before*) the central feature p, and its word is of type "triangle", so we increment the bin for word "triangle" and orientation G in the first row. We also increment the count in the cells directly below, recording the neighbors cumulatively.

3.2. Spatio-Temporal Neighborhood Formation

To make the next level of the vocabulary ("level-1"), we must first generate compound descriptors. Each one is formed from the neighborhood around a central interest point $p = \langle x, y, t, w\rangle$. For a given space-time point, we collect its N closest interest points, where nearness is measured by a normalized Euclidean distance on its 3d position coordinates (i.e., a Mahalanobis distance restricted to a diagonal covariance matrix):

$$D_\sigma(p, q) = \left(\sum_{i=1}^{3} \frac{1}{\sigma_i}(p(i) - q(i))^2\right)^{\frac{1}{2}}, \quad (2)$$

where each σ_i is a weight that scales the x, y, or t dimensions, and will be defined below in Section 3.4.

Let $\mathcal{N}(p) = \{p, q_1, \ldots, q_{N-1}\}$ denote the N nearest neighboring points for central point p; for each, we know its level-0 visual word from above. Additionally, each can be placed in one of $2^3 = 8$ orientation bins, depending on

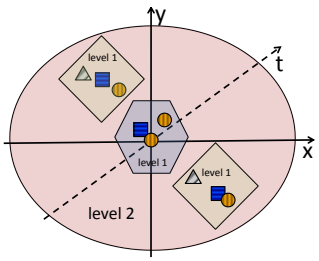

Figure 3. Illustration of the feature / vocabulary hierarchy. The hexagon and diamonds are level-1 visual words composed of level-0 words. The red ellipse is a level-2 word composed of level-1 words (here $N = 3$).

its location in time and space with respect to the central point—to the left or right, below or above, before or after.

We form a new representation for the central interest point by creating a matrix of size N-by-$8k$: the r-th row is a cumulative histogram corresponding to the point's first r ranked neighbors, $p, q_1, \ldots, q_{r-1}$, and the entries in a row record how many of those neighbors fall into each of the orientation bins, separated out by word type (see Figure 2). In other words, for each neighboring point, we increment bins according to both its orientation relative to the central point and which level-0 word it corresponds to, and accumulate these counts as we move further away from the central point. Note that the number of columns in this matrix, $8k$, reflects the number of possible word-orientation combinations. This matrix of histograms is reshaped to a single $8kN$-dimensional vector to yield a single level-1 descriptor (Figure 2 depicts the matrix in 2d for presentation purposes only). In experiments, we let $N = 5$, and reduce the compound descriptors' dimensionality with PCA.

Note that the orientation bins do not have a predetermined scale (i.e., no outer boundaries), since the neighbors are entered into the descriptor according to their distance from the central point; using rank rather than fixed distances also means that we will form similar descriptors for configurations that are similar aside from a scale change or internal shifts and stretching. Furthermore, the cumulative nature of the histogram means that closer interest points have more influence on the feature, and we can mitigate sensitivity to exact relative ordering of the neighborhood, as has been shown useful for related features in static images [17, 16]. By centering these neighborhoods at each interest point, we also maintain translation invariance within the clip between the neighborhoods, unlike the global grid-based methods [15, 4, 31, 11].

3.3. A Hierarchy of Composite Vocabularies

Having formed the neighborhood descriptors, we repeat the process using the neighborhoods themselves as the central points, as follows. We run k-means clustering on a sample of the (reduced dimension) descriptors outlined above, quantizing the "neighborhoods" into their own visual vocabulary. This way, the appearance and layout of each level-1 neighborhood can be succinctly represented by a word

Figure 4. Examples of level-1 words. The most relevant word for the hand-clapping (top) and horse-riding (bottom) actions, as determined by mutual information. (To save space, we omit some intermediate frames). Best viewed in pdf.

$w^{(1)}$, and when forming the neighborhoods for the next higher-level, we can bin them accordingly. For example, what *was* a level-1 neighborhood formed as in Figure 2 becomes a discrete word type (say "diamond") in the level-2 neighborhood formation (see Figure 3). We let the position for a feature $\langle x, y, t, w^{(\ell)} \rangle$ be the position where its neighborhood originated, for all $\ell \geq 1$. Thus, the only difference for forming vocabularies beyond level-1 is that the neighbors are identified according to distances between the compound features' central points. Example instances of "level-1" words are shown in Figure 4.

This process continues L times to form $L + 1$ total vocabularies. In this manner, we generate a hierarchy of composite feature configuration types, each of which loosely encodes the space-time layout of the component features. Note that unlike a "vocabulary tree" [22], which forms a hierarchical quantization of the appearance descriptor space, our method is assembling a hierarchy of composite neighborhood features, where each compound feature captures a particular type of space-time layout. At this point, we can map an input video's raw features V to a set of $L + 1$ bag-of-word histograms:

$$\mathcal{H}_\sigma(V) = \{H_0(V), H_1(V), \ldots, H_L(V)\} \qquad (3)$$

where each $H_i(V)$ is a k-bin histogram and counts the frequency with which each level-i word occurs in the set. The subscript σ reflects that these histograms stem from a particular distance function parameterization (we return to this point in Section 3.4). In practice, we set $L = 2$ (3 levels); we have not experimented with larger values, simply because the neighborhoods begin to cover most of the clips.

3.4. Discriminative Space-Time Neighborhoods

By using the nearest ranked points to form neighborhoods rather than a fixed grid or cuboid, we already are generating variable-sized configurations. However, the ranking itself depends on how we parameterize the distance in Equation 2. We will get a different set of neighbors for each point depending on the scaling this distance applies to each dimension (x, y, t). In a sense, the "correct" distance between interest points is not well-defined: how should one

compare pixel units to frame units? Is a gap of two pixels bigger or smaller perceptually than a gap of two frames? Even within the two spatial dimensions, we expect that a non-uniform scaling may be useful. For example, it may be better to capture a larger horizontal extent of actions in a single neighborhood when categorizing instances of *hand-waving*, but a larger vertical extent to distinguish instances of *diving*. Thus, rather than pick a single distance function, we consider those functions that yield the most informative neighborhoods when used in a learned combination.

In order to do so, we first compose a series of M candidate neighborhoods using different distance functions—meaning M different settings for each of the three σ_i parameters. These in turn produce a series of M feature hierarchies, or $ML + 1$ total vocabularies.[1] We use a separate kernel for each vocabulary type. Then, for each action class, we use multiple kernel learning (MKL) to determine the weighted combination of those neighborhoods that yields the most discriminative means of comparing video clips. MKL algorithms seek the weights w_c for some set of component kernels K_c such that the final combined kernel $K = \sum_{c=1}^{|\mathbf{C}|} w_c K_c$ is most aligned with the "ideal" kernel matrix reflecting the data's true labels [7]. Finally, the clips are categorized according to support vector machine (SVM) classifiers built with the learned kernels. The remainder of this section fleshes out the above description in more detail.

To train the SVMs we employ multi-channel generalized Gaussian kernels with the χ^2 distance, following [15] and others. These kernels sum over the distances between a set of component histograms, appropriately scaled. The χ^2 distance between any two bag-of-words histograms H_i and H_j is defined as:

$$\chi^2(H_i, H_j) = \frac{1}{2}\sum_{b=1}^{k}\left(\frac{(H_i(b) - H_j(b))^2}{H_i(b) + H_j(b)}\right) \quad (4)$$

where b indexes over each of the k histogram bins.

Each example is represented by multiple histograms, and the component comparisons are combined via the kernel:

$$K(H_i, H_j) = \sum_{c \in \mathbf{C}} w_c \exp\left(-\frac{1}{A_c}\chi^2(H_i^c, H_j^c)\right), \quad (5)$$

where $\mathbf{C}$ denotes the set of all channels, and H_i^c and H_j^c denote the two inputs' c-th channel histograms, respectively. The scalar A_c is the kernel's scale parameter, and is simply set to the mean distance between training examples along the given feature channel.

In our case, a "channel" refers to a combination of feature type (HoG and HoF, or HoG3D), a distance function (as parameterized by σ), and a vocabulary level (ranging from $0, \ldots, L$). Any choice of these three items[2] yields a single bag-of-words histogram for a video, and thus $|\mathbf{C}| = F(ML+1)$, where F is the number of feature types used. For the training videos, we compute all choices. We then supply the pool of kernels to MKL to learn the scalar weights $w_1, \ldots, w_{|\mathbf{C}|}$ to combine them. We use the efficient SKMsmo method of [7].

The results from MKL provide a form of feature selection and automatic scaling among the vocabulary levels, descriptor types, and neighborhood shapes. We find that while some distance functions are eliminated by MKL's feature selection, those kernels with non-zero weights are typically balanced across the different levels of the hierarchy.

Note that in this setting it would not be possible to directly learn a Mahalanobis metric for the interest points, for two reasons: firstly, while our training videos have class labels associated with them, the neighborhood features themselves do not have labels; secondly, we want to compose neighborhoods based on the ranked distances from the central point, rather than use their absolute distances.

4. Results

Our experiments demonstrate the proposed approach for action recognition with a variety of categories. In addition to reporting overall accuracy, we also study the empirical tradeoffs between grid-based vs. variable-shaped neighborhoods, and analyze the kinds of discriminative neighborhoods selected by the algorithm.

Datasets and Implementation Details: We evaluate our approach on two benchmark datasets for human activity recognition: the KTH actions dataset [30], and the UCF Sports dataset [29]. Both the labeled examples and test video clips contain primarily a single action of interest. KTH consists of 6 actions (e.g., boxing, hand-clapping, running), each of which is performed four times by each of 25 people, for a total of 600 video clips. The UCF Sports dataset consists of 150 videos with 10 action classes taken from real sports broadcasts (e.g., swinging, weight-lifting, horse-riding), with a wide range of viewpoints and scene backgrounds. We augment the dataset with horizontally flipped versions of each video, following [32]. See Figures 1 and 4 for example frames from a few classes.

The KTH dataset entails a 6-way multi-class recognition task, and is scored by the average recognition rate per class. We use the standard partition, following [30]. Almost all reported numbers use this setup, though [9] also report leave-one-out accuracy, while [3, 5, 21] exclusively use leave-one-out. The UCF Sports dataset is tested in a 10-way recognition task in a leave-one-out manner, cycling each example in as a test video one at a time, following

[1] It is $ML + 1$ rather than $M(L + 1)$ since level-0 is a traditional vocabulary and requires no distance function to compute. Please note each $\vec{\sigma} = [\sigma_1, \sigma_2, \sigma_3]$ is a scaling on the (x, y, t) dimensions in the distance function in Equation 2 used to rank interest points; it does *not* denote a scaling factor on a grid width.

[2] For level 0, the distance function is not needed.

Approach	Year	Accuracy
Schüldt *et al*. [30]	2004	71.72%
Dollar *et al*. [5]	2005	81.17%
Ke *et al*. [12]	2005	62.96%
Nowozin *et al*. [23]	2007	84.72%
Fathi *et al*. [6]	2008	90.50%
Gilbert *et al*. [8]	2008	89.92%
Laptev *et al*. [15]	2008	91.80%
Niebles *et al*. [21]	2008	81.50%
Bregonzio *et al*. [3]	2009	93.17%
Liu *et al*. [18]	2009	93.80%
Gilbert *et al*. [9]	2009	94.50%
Our method		94.53%

Figure 5. Comparison of recognition accuracy on the KTH data.

[32, 29, 33]. As in [32], we withhold the flipped version of the test clip from the training set.

We extract sparse Harris3D points for KTH, and perform dense sampling followed by HoG3D feature extraction for UCF, using code kindly provided by the authors of [15] and [13], with the default parameter settings.[3] We fix the vocabulary size at each level to $k = 300$ (except level-0 in UCF, where $k = 4000$), and $N = 5$ (see below for experiments testing this parameter's sensitivity). For MKL, we use the SKMsmo software.[4] We use $L = 2$ higher-level vocabularies, and consider distance functions parameterized by all combinations of $\frac{1}{\sigma_i} = \{1, 5, 10, 50\}$ for KTH and $\frac{1}{\sigma_1} = \{1\}$, $\frac{1}{\sigma_2} = \frac{1}{\sigma_3} = \{1, 10\}$ for UCF. These values were chosen fairly arbitrarily, with the intent of capturing scaling factors of varying orders of magnitude and degrees of precision for each, and knowing that MKL can "de-select" features by assigning zero weight. The primary cost in computing our neighborhood descriptors is finding the closest neighbors for each interest point, which takes time quadratic in the number of features within the clip when searched exhaustively (however faster implementations would be possible with k-d trees).

4.1. Action Recognition Performance

First we report overall accuracy on both datasets. KTH is a standard benchmark for human action recognition. Figure 5 compares our results (bottom row) to those from previous work. Our method outperforms previously published results, and at 94.53% is equal to the very best accuracy we are aware of, due to Gilbert *et al*. [9]. Most classes are almost perfectly predicted, except for running and jogging, which are frequently confused. The recognition accuracy using only level-0 is 93.05%, which is roughly comparable to a similar baseline reported in [15].

Figure 6 shows our results on the UCF Sports videos. To our knowledge, the accuracy of our method is the best on this dataset thus far with 87.27% per-class average recognition accuracy. Our accuracy is directly comparable to the 85.6% reported in [32], but not to other numbers on the UCF dataset (69.2% [29], 79.3% [33]).[5] Our experiments use the version of the dataset available on the author's website [29] at the time of writing. The results clearly show that our approach performs accurate recognition with rather challenging, realistic actions. A direct comparison to the level-0 baseline (using the identical interest points and features) confirms that our higher-level neighborhood descriptors add useful information. Furthermore, we see that MKL does well in selecting the useful distance combination, improving over a simple average of all candidate kernels.

Approach	Accuracy/Class
Our method	87.27%
Average of all kernels	84.43%
Level-0 baseline	85.49%

Figure 6. Results on the UCF Sports dataset.

4.2. Analysis of Variable-Sized Neighborhoods

Next we run experiments to support our claim that space-time feature configurations captured at variable sizes and shapes can offer more robust descriptions than a rigid fixed gridding of nearby points. We compare our approach side-by-side with our own implementation of a grid-based feature-centered neighborhood descriptor; we model the binning after the grids used in [9, 8], but use visual-word based descriptors to enable the closest comparison. At each interest point, we extract a $3c\sigma$ x $3c\sigma$ x $3c\tau$ cube centered at that point and consisting of 27 total uniformly sized bins, where σ and τ denote the spatial and temporal scales of the detected interest point, and c denotes the scaling factor. The descriptor for the cube is a histogram of the visual words within each cell. We compare this baseline to our own features, for two tests: one measuring the discriminative power of the resulting neighborhoods, and another measuring ultimate recognition accuracy, both on the KTH dataset.

While our method requires choosing how many ranked neighbors N to include per point, the grid-based descriptor requires choosing the scaling on the cube side-lengths, c. Thus, we evaluate the two features as a function of their free parameter, letting c and N vary from 1 to 40, in steps of 4 ($c = 1$ is as suggested in [9]).

Feature quality: Figure 7 (top) shows the results analyzing the words' discriminative power. We measure the mutual information of all the words generated with either method, and compute the sum of the highest-scoring five

[3] http://www.irisa.fr/vista/Equipe/People/Laptev/interestpoints.html
http://lear.inrialpes.fr/people/klaeser/software

[4] http://www.stat.berkeley.edu/∼gobo/SKMsmo.tar, by G. Obozinski.

[5] The exact set of videos used by each author differs, due to some apparent copyright issues with a subset of the videos that the creators had to remove from the collection.

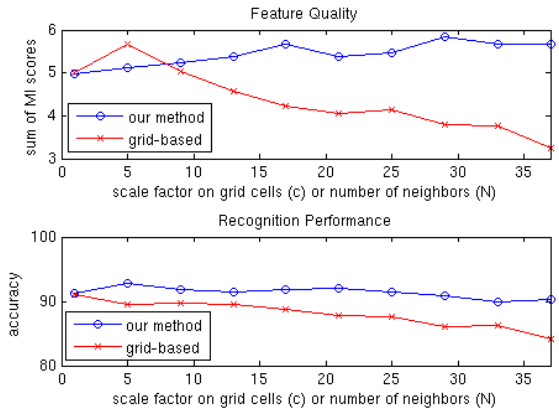

Figure 7. **Top:** Distinctiveness of the neighborhood (level-1) visual words for our features and a grid-based baseline. **Bottom:** Recognition results for the same two methods. Note that a single x, y, t weight combination is used here. See text for details.

words for each of the six categories (30 total per method). We quantify the quality of the resulting neighborhood descriptors in terms of how distinctive they are for the different actions. The figure shows this score plotted as a function of c and N.

What we find is that both methods *can* achieve similar discriminative power, but the parameter selection is much more critical/sensitive for the grid-based features. This fits our intuition that the appropriate neighborhood scale is difficult to set a-priori. Grid-based neighborhoods impose uniformity on the neighbors, which can lead to unbalanced density between classes or omit relevant neighbors. In contrast, our approach yields quite stable accuracy as a function of the number of neighbors included, which we believe is due to the fact that we encode the neighboring points cumulatively, thus emphasizing the more distant neighbors less. Further, the rank-based accumulation of the neighbors means we can capture a similar density of features for different types of interest points.

Recognition performance: Figure 7 (bottom) shows the results analyzing the words' ultimate recognition accuracy, as produced in the usual bag-of-words classifier. Each point on the curves corresponds to an average over 100 runs, with randomly selected train-test partitions. Again, our features are more stable and for this metric perform better overall.

4.3. Impact of Hierarchical Vocabularies

The different vocabulary levels our method generates capture different information, and using a combination of levels can be more effective than using individual levels in isolation. Figure 8 illustrates this empirically, for the UCF Sports data. We see that kernels from all levels contribute to the optimal learned combination.

Levels	Accuracy/Class	Average MKL Weight
0	85.49%	0.63 ($\pm$ 0.3)
1	82.16%	0.10 ($\pm$ 0.2)
2	73.30%	0.10 ($\pm$ 0.2)

Figure 8. Contribution of the vocabulary levels for one x, y, t setting on the UCF Sports dataset. Each level offers discriminative power, and in combination they provide a richer representation that can boost recognition.

4.4. Interpreting the Selected Features

Finally, we conclude by interpreting the sorts of neighborhood shapes that are automatically learned by our method. There is of course no guarantee of finding explainable aspects, but the goal is to get a sense of the space-time scalings that are effective for certain actions.

In general, a high weight on a single dimension in the distance that ranks the neighbors (i.e., low value for σ_i) corresponds to a "stretching" of that dimension, or equivalently, a compressing of distance along the other two. This increases the extent of a neighborhood in the lower-weighted dimensions. For example, with a relatively high weight on t, the size of a feature neighborhood in terms of x and y becomes relatively larger. When the ratio of x and y to t is very small or large, the neighborhoods assume extreme shapes which capture almost exclusively spatial information or temporal information (neighborhoods "longer" in the given dimension).

We observe some trends in the highly weighted kernels found for the KTH dataset. The level-1 kernels generated by HoF features with relatively lower weight on t are used by more classifiers than any other kernels, suggesting that features with larger temporal extent are often informative. Additionally, we observe that the level-1 kernel generated by HoF features with a lower weight on x is primarily used to discriminate between arm-based activities, whereas the kernel with a lower weight on t is mostly used for leg-based activities. This indicates that wider motion/time extents are most helpful for distinguishing between actions which are almost identical to each other in appearance but happen at different speeds—such as *walking*, *jogging*, and *running*. On the other hand, with activities that appear different regardless of speed (such as *boxing*, *clapping*, and *waving*), the features clustering near the same frame in a wider horizontal extent are key. Figure 9 illustrates this point.

5. Conclusions

Our main contribution is a video feature formation technique to learn class-specific vocabularies of space-time neighborhoods. Unlike previous work, our approach allows variable-sized neighborhoods to be selected for different activities. Our experiments demonstrate the positive impact of introducing compound feature-centered descriptors into the

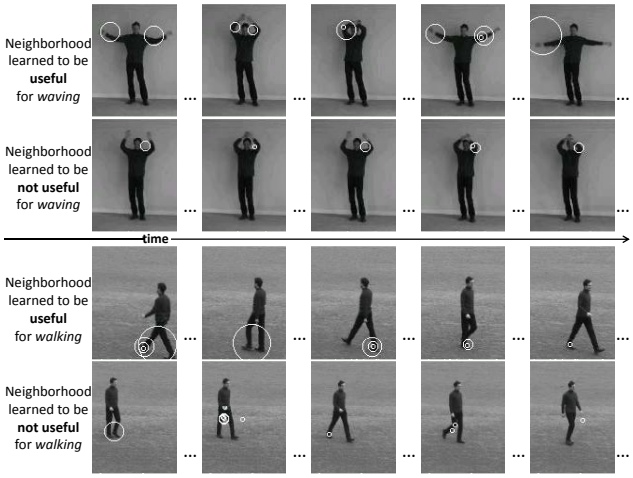

Figure 9. Examples of learned neighborhood feature shapes. The configuration in the **top row** was weighted highly by MKL for arm-based actions, while the one in the **third row** was weighted highly for leg-based actions. The wider horizontal spatial extent is useful for waving, but the neighborhood with a larger temporal extent is discriminative for walking. Second and fourth rows show contrasting (not useful) examples with low weights for the respective categories. Note that each row depicts a single neighborhood feature. Circle sizes denote the scale of level-0 features.

already successful bag-of-words video representation. In future work, we intend to examine how stronger supervision at the local feature level might allow more specific discriminative selection, and to evaluate alternative interest point sampling strategies or descriptors within our framework.

Acknowledgements

This research is supported in part by Texas HECB 003658-01-40-2007, DARPA VIRAT, and the Henry Luce Foundation. We thank the anonymous reviewers for their helpful comments, and the researchers cited above for sharing their code and datasets.

References

[1] A. Agarwal and B. Triggs. Hyperfeatures – multilevel local coding for visual recognition. In *ECCV*, 2006.

[2] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *CVPR*, 2005.

[3] M. Bregonzio, S. Gong, and T. Xiang. Recognizing action as clouds of space-time interest points. In *CVPR*, 2009.

[4] J. Choi, W. Jeon, and S.-C. Lee. Spatio-temporal pyramid matching for sports videos. In *ACM Multimedia*, 2008.

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[6] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *CVPR*, 2008.

[7] F.R.Bach, G.R.G.Lanckriet, and M.I.Jordan. Fast kernel learning using sequential minimal optimization. *Technical report*, 2004.

[8] A. Gilbert, J. Illingworth, and R. Bowden. Scale invariant action recognition using compound features mined from dense spatio-temporal corners. In *ECCV*, 2008.

[9] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV*, 2009.

[10] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *PAMI*, volume 29, pages 2247–2253, 2007.

[11] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009.

[12] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *IEEE Computer Society*, volume 1, pages 166–173, 2005.

[13] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*, 2008.

[14] I. Laptev. On space-time interest points. In *International Journal of Computer Vision*, volume 64, pages 107–123, 2005.

[15] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[16] Y. J. Lee and K. Grauman. Foreground focus: Unsupervised learning from partially matching images. *IJCV*, 85(2), 2009.

[17] H. Ling and S. Soatto. Proximity distribution kernels for geometric context in category recognition. In *ICCV*, 2007.

[18] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009.

[19] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[20] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

[21] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *IJCV*, 2008.

[22] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[23] S. Nowozin, G. Bakir, and K. Tsuda. Discriminative subsequence mining for action classification. In *ICCV*, 2007.

[24] V. Parameswara and R. Chellappa. Human action-recognition using mutual invariants. In *CVIU*, 1998.

[25] D. Parikh, L. Zitnick, and T. Chen. Unsupervised learning of hierarchical spatial structures in images. In *CVPR*, 2009.

[26] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient mining of frequent and distinctive feature configurations. In *ICCV*, 2007.

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

[28] C. Rao and M. Shah. View-invariance in action recognition. In *CVPR*, 2001.

[29] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH: A spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

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

[31] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009.

[32] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[33] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV*, 2009.

[34] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *CVPR*, 2007.

[35] G. Zhu, C. Xu, W. Gao, and Q. Huang. Action recognition in broadcast tennis video using optical flow and support vector machine. In *ECCV*, 2006.