Representing Pairwise Spatial and Temporal Relations for Action Recognition

Pyry Matikainen[1], Martial Hebert[1], and Rahul Sukthankar[2,1]

[1] The Robotics Institute, Carnegie Mellon University
[2] Intel Labs Pittsburgh
{pmatikai,hebert,rahuls}@cs.cmu.edu

Abstract. The popular bag-of-words paradigm for action recognition tasks is based on building histograms of quantized features, typically at the cost of discarding all information about relationships between them. However, although the beneficial nature of including these relationships seems obvious, in practice finding good representations for feature relationships in video is difficult. We propose a simple and computationally efficient method for expressing pairwise relationships between quantized features that combines the power of discriminative representations with key aspects of Naïve Bayes. We demonstrate how our technique can augment both appearance- and motion-based features, and that it significantly improves performance on both types of features.

1 Introduction

It is well known that classification and recognition problems in general cannot be solved by using a single type of feature and that, instead, progress lies in combinations of feature representations. But as there is as vast an array of ways to combine and augment features as there are features themselves, the development of such higher order methods is as difficult (and potentially rewarding) an endeavor as direct feature construction. These meta-methods span a range of approaches from those that make absolutely no assumptions on their base features, and thus are consigned to black-boxes operating on vectors of numbers, to those that are so intimately tied to their base features as to be virtually inseparable from them. The former types of methods, such as multiple kernel learning (MKL) techniques are attractive for their broad applicability, but the latter tend to be more powerful in specific applications due to their strong coupling with their underlying features.

However, between those extremes there are still augmentations that compromise between generality and power by being applicable to broad classes of loosely-related base features. The popularity of statistical bag-of-words style techniques [1, 2] for action recognition and related video tasks creates an opportunity to take advantage of the broad similarities in these techniques. In particular, these techniques rely on accumulating histograms of quantized features extracted from video, features that are almost always localized in space and time. Yet these methods typically do not take advantage of the spatial

Fig. 1. Pairs of features probabilistically vote for action classes; pairs voting for the correct action class are shown in yellow, with brighter color denoting stronger (more informative) votes. For "answerPhone", the relative motion of the hands is particularly discriminative. These results employ trajectory fragments on the Rochester dataset [3], but our method works with any localized video feature.

and temporal relationships between features. While it is obvious that in general using such relationships should help, the subtleties of designing appropriate representations have limited their use. Figure 1 shows examples of the types of informative relationships between features that we are interested in.

Pairwise spatial relationships, in the form of star topologies, fans, constellations, and parts models, have seen frequent use in static image analysis [4–6]. Practical limitations have made transitioning these methods to video difficult. Sparse pairwise topologies (stars, fans, parts) often suffer from a lack of appropriately annotated training data, as they often require annotations that specify the topology for training [7, 8]. Alternatively, there are structured methods which can operate without such annotations, but at the cost of significantly more complicated and computationally expensive training or testing [9, 10]. In the special limited case of a fixed camera, the entire topology can be fixed relative to the frame by simply using the absolute positions of features [11, 12].

The key contributions of this paper can be summarized as follows: (1) we propose an efficient method for augmenting quantized local features with relative spatial-temporal relationships between pairs of features, and (2) we show that our representation can be applied to a variety of base features and results in improved recognition accuracy on several standard video datasets.

For the pairwise model, the most direct representation that is compatible with bag-of-words techniques is to simply generate higher-order features by quantizing the possible spatio-temporal relationships between features. However, this results in a significantly larger number of possible codewords; for example, 100 codeword labels and 10 possible relationships, would produce 100,000 possible labels for the pairwise codewords. Attempts to mitigate this have centered on dimensionality reduction and limiting the number of relationships. In the former case, Gilbert *et al.* [13, 14] employ data mining techniques to find frequently-occurring combinations of features. Similarly, Ryoo and Aggarwal [15] use a sparse representation of the resulting high-dimensional histograms, in addition to using a relatively small relationship set. Taken to the extreme, Savarese *et*

al. [16] consider effectively only one relationship: whether two features occur within a fixed distance of each other, and likewise Sun *et al*. [17] also use a simple proximity relationship.

The subtle difficulty is exposing enough information for discriminative machinery to gain traction, but not so much as to overwhelm it in the noise. We strike this balance by selectively organizing estimated pairwise relationships, thereby exploiting fine spatio-temporal relationships without having to resort to an unmanageable representation for the classifier. Inspired by recent work on the max-margin Hough transform [18], we propose accumulating probabilities in a Naïve-Bayes like fashion into a reduced number of bins, and then presenting these binned probabilities to discriminative machinery. By choosing our bins to coincide with codeword labels, we produce vectors with size proportional to the number of codewords while still taking advantage of discriminative techniques.

Since we propose a method for augmenting features with spatio-temporal relationships, we wish to show that this augmentation performs well on a range of features. To this end, we consider two radically different types of base features. First, we consider features built from space-time interest points (STIPs) with associated Histogram of Oriented Gradient (HOG) descriptors, which are sophisticated appearance-based features popularized by Laptev *et al*. [1]. Second, we consider a simple form of trajectory based features similar to those proposed by Matikainen *et al*. [19] and Messing *et al*. [3], quantized through a fixed (training data independent) quantization method. This selection of base features demonstrates the effectiveness of our method on both appearance- and motion-based features, as well as on sophisticated and simple feature extraction methods. In particular, our simplified trajectory method produces a fixed number of features per frame, and the feature labels are not derived from a clustering of training data. The method is virtually certain to produce a large number of extraneous features, and the feature labels are likely to be more sensitive to noise compared to those produced through clustering. In contrast, STIP-HOG produces relatively few features, which tend to be more stable due to the clustering.

As discussed above, our proposed approach formulates the problem in a Naïve Bayes manner, but rather than independently summing per-feature probabilities in log space, we pass them through a discriminative classifier. We train this classifier by estimating all of the cross probabilities for feature labels, that is, for each pair of labels and each action we build a relative location probability table (RLPT) of the observed spatial and temporal relationships between features of those labels under the given action. Then, any feature label can compute its estimate of the distribution over action probabilities using the trained cross-probability maps. These estimates are combined for each feature label, and the final feature vector is presented to a classifier.

2 Base features

The proposed method can augment a variety of common features employed in video action recognition. To demonstrate our method's generality, we describe

how it can be applied to two types of features that represent video in very different ways, as discussed below.

2.1 Base feature: STIP-HOG

Laptev *et al.*'s space-time interest points (STIPs) [1], in conjunction with Histogram of Oriented Gradient (HOG) descriptors have achieved state-of-the-art performance on a variety of video classification and retrieval tasks. A variable number of STIPs are discovered in a single video and the local space-time volume near each interest point is represented using an 72-dimensional descriptor. These HOG descriptors are quantized using a codebook (typically pre-generated using k-means clustering on a large collection) to produce a discrete label and a space-time location (x, y, t) for each STIP.

2.2 Base feature: Quantized trajectories

In previous work [19], we considered trajectory-based features that we coined *trajectons*; these features describe video data in a very different manner from STIP-HOG. First, Harris corner features are tracked in a given video using KLT to produce a set of trajectories. Each trajectory is first converted from a list of (x_t, y_t) position pairs into a list of discrete derivative pairs $(dx_t, dy_t) = (x_t - x_{t-1}, y_t - y_{t-1})$. These trajectories are then broken up into overlapping windows of fixed duration T, each of which is considered a new feature or trajectory fragment. Unlike STIP-HOG, trajectory fragments seek to express longer-term motion in the video. We generally follow our previous work, but substitute a more straightforward quantization method.

Sequencing code map (SCM) quantization. In our earlier work, we quantize trajectories using k-means. Fragments are clustered to produce a codebook. Messing *et al.* [3] also use trajectory features but employ a different quantization strategy in which trajectories are soft-assigned to Markov mixture components; their strategy is similar to that of Sun *et al.* [17] who also consider quantized transitions within a trajectory.

Both Messing *et al.*'s and our earlier approach can be computationally expensive depending on the number of mixture components or k-means centers, respectively. Taking inspiration from both, we propose quantizing fixed-length trajectories using a derivative table similar to Messing *et al.*'s, which we call a sequencing code map (SCM), examples of which can be seen in Figure 3. However, rather than using quantized derivatives to look up probabilities, we simply combine the quantized indices over the fixed length trajectory fragment into a single label encoding quantized derivatives at specific times with each fragment (see Figure 2).

In our method, a trajectory fragment is divided into k consecutive stages of length t frames, such that $kt \leq T$, where T is the total length of the fragment. The total motion, or summed derivative, of each stage is computed as a $(dx, dy)_k$

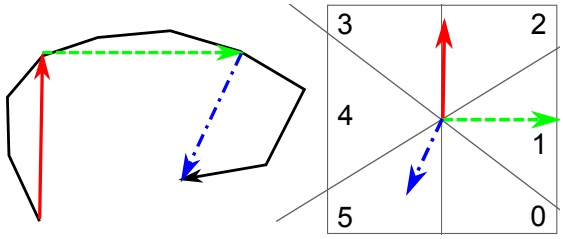

Fig. 2. Sequencing code map (SCM) quantization breaks a trajectory fragment into a number of stages (in this case three) that are separately quantized according to a map (in this case a 6-way angular map). These per-stage labels, called sequence codes, are combined into a final label for the fragment, which in this case would be $215_6 = 83_{10}$.

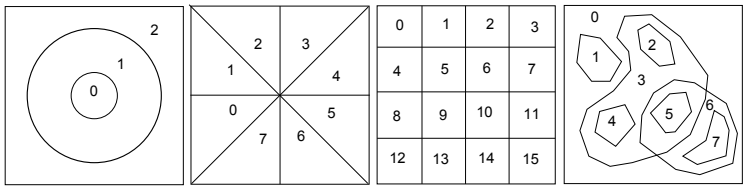

Fig. 3. Examples of possible sequencing code maps (SCMs) or relative location maps (RLMs). Our experiments focus on angular maps (second from left) but our method is completely general.

pair for each stage. This (dx, dy) vector is quantized according to the SCM into one of n stage labels, or sequence codes; the k sequence codes are combined to produce a single combined label that can take on k^n values. For example, with 3 stages and an SCM containing 8 bins, there would be 8^3 or 512 labels total. Since the time to quantize a stage is a single table lookup regardless of how the table was produced, this method is extremely computationally efficient (*i.e.*, the computation time does not grow with an increasing number of quantization bins).

Formally, we denote by $M(dx, dy)$ the SCM function, which is implemented as a two-dimensional lookup table that maps from a dx, dy to an integer label in the range of 0 to $n-1$ inclusive. We denote by $(dx, dy)_k$ the derivative pair for stage k. Then the assigned label is given by $l = \sum_{j=0}^{k} n^j \cdot M(dx_j, dy_j)$.

Besides the convenience of not having to build a codeword dictionary and the reduced computational cost, our introduction of this quantization method is meant to demonstrate that our pairwise features do not depend on data-driven clustering techniques. The quantized labels produced by SCM quantization are unlikely to correspond nicely to clusters of features in the dataset (*e.g.*, , parts), yet the improvement produced by our pairwise relationships persists.

3 Augmenting Features with Pairwise Relationships

In the following subsections we detail our approach for augmenting generic base features with spatial and temporal relationships. While the previous discussions focused on STIP and trajectory fragment features, our proposed method for pairwise spatio-temporal augmentation applies equally well to any type of feature that can be quantized and localized. In the remainder of this section, a "feature" is simply the tuple (l, x, y, t): a codeword label in conjunction with its spatio-temporal position. The set of all observed features is denoted F.

3.1 Pairwise discrimination with relative location probabilities (RLPs)

Starting with the observation that Naïve Bayes is a linear classifier in log space, in this section we formulate the pairwise representation first in the familiar terms of a Naïve Bayes classifier, and then demonstrate how to expose more of the underlying structure to discriminative methods.

We start with the assumption that all pairs are conditionally independent given the action class. Then, if features are quantized to L labels and the spatial relationships between features are quantized to S labels, we could represent the full distribution over pairs of features with a vector of L^2S bins. Unfortunately, for even as few as $L = 100$ trajectory labels and $S = 10$ spatial relationships, there would be $(100^2)(10) = 100,000$ elements in the computed feature vector, far too many to support with the merely hundreds of training samples typically available in video datasets.

This feature vector must be reduced, but the direct approach of combining bins is just equivalent to using a coarser quantization level. Instead, taking inspiration from the Max-Margin Hough Transform [18] and Naïve Bayes, we build probability maps of the spatial relationships between features, and instead of summing counts, we accumulate probabilities, allowing pairs to contribute more information during their aggregation.

Specifically, we produce a feature vector B of length AL, where A is the number of action classes (*e.g.*, walk, run, jump, *etc.*), and where each entry $B_{a,l}$ corresponds to the combined conditional probability of all pairs containing a feature with label l given the action class a. In other words, a bin contains a feature label's probabilistic vote for a given action class, and we could compute a Naïve Bayes estimate of the probability of all the observed features given an action by summing all the votes for that action: $\log P(F|a) = \sum_{l \in L} B_{a,l}$. However, instead of summing these in a Naïve Bayes fashion, we present the vector as a whole to discriminative machinery, in our case a linear SVM. We now describe how we accomplish this feature vector reduction.

Notation. Formally, a video segment has a number of quantized features computed from it. A feature $f_i \in F$ is associated with a discrete quantized label $l_i \in L$ as well as a spatio-temporal position (x_i, y_i, t_i) indicating the frame and location in the frame where it occurs. For a pair of features within the same frame and a

given action $a \in A$, there is a probability of the two features occurring together in a frame $P(l_i, l_j|a)$ as well as the relative location probability (RLP) for their particular spatial relationship $P(x_i, x_j, y_i, y_j|l_i, l_j, a)$. We make the simplifying assumption that RLPs depend only on the relative spatial relationship between the two features, so that $P(x_i, x_j, y_i, y_j|l_i, l_j, a) = P(dx, dy|a, l_i, l_j)$, where dx and dy are slight abuses of notation that should be understood to mean $x_i - x_j$ and $y_i - y_j$ where appropriate. This assumption enforces the property that the computed relationships are invariant to simple translations of the feature pairs.

Probabilistic formulation. The reduction to a feature vector of length AL is done by selectively computing parts of the whole Naïve Bayes formulation of the problem. In particular, a full probabilistic formulation would compute $P(F|a)$ and select the a that maximizes this expression. Since Naïve Bayes takes the form of multiplying a number of features' probabilities, or in this case pair probabilities, we can exploit the distributive and commutative properties of multiplication to pre-multiply groups of pair probabilities together, and then return those intermediate group probabilities rather than the entire sum. This can be seen as binning the pair probabilities.

Assuming feature pairs are conditionally independent, we can compute the probability of a feature set F given an action a according to the equation

$$P(F|a) = \prod_{f_i \in F} P(l_i|a) \prod_{f_j \in F} P(l_j|l_i, a) P(dx, dy|a, l_i, l_j), \tag{1}$$

which strictly speaking double-counts pairs since each pair is included twice in the computation; however, since we are only interested in the most likely action, this is not an issue.

In practice, we employ log probabilities both to avoid issues with numerical precision from extremely small values and to formulate the problem as a linear classifier. In this case the log probability expression becomes

$$\begin{aligned} \log(P(F|a)) = &\sum_{f_i \in F} \log(P(l_i|a)) + \\ &\sum_{f_j \in F} \log(P(l_j|l_i, a)) + \log(P(dx, dy|a, l_i, l_j)). \end{aligned} \tag{2}$$

To simplify the expression we assume uniform probabilities for $P(l_i|a)$ and $P(l_j|l_i, a)$. Later we can include nonuniform label probabilities by simply concatenating the individual label histogram to the pairwise feature vector when both are presented to the classifier. Thus, our probability expression becomes

$$\begin{aligned} &\log(P(F|a)) = \\ &\sum_{f_i \in F} \sum_{f_j \in F} \log(P(dx, dy|a, l_i, l_j)) + C, \end{aligned} \tag{3}$$

which is simply a formal way of stating that the whole log probability is the sum of all the pairwise log probabilities. Since we are only interested in the

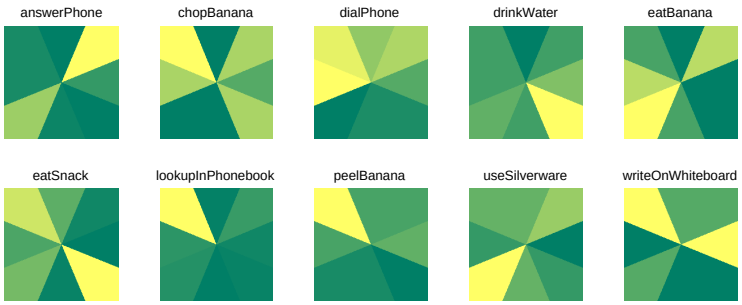

Fig. 4. Relative location probabilities (RLPs) for feature labels 25 and 30 over all actions in the Rochester dataset, using an 8-way angular map. Lighter (yellower) indicates higher probability. We see that for the answerPhone action, features with label 30 tend to occur up and to the right of features with label 25, whereas for useSilverware, features with label 30 tend to occur down and to the left of those with label 25.

relative probabilities over action classes, we collect the uniform probabilities for labels into a constant C which does not depend on the action a, and which is omitted from the following equations for clarity. We now wish to divide this expression into a number of sub-sums that can be presented to a classifier, and this expression leaves us a great deal of flexibility, since we are free to compute and return sub-sums in an arbitrary manner.

Discriminative Form. We rewrite Equation 3 in such a way as to bin probabilities according to individual feature labels. In particular, we can rewrite it in log form as

$$\log(P(F|a)) = \sum_{l \in L} \log(P(b_l|a)), \tag{4}$$

where

$$\log(P(b_l|a)) = \sum_{f_i \in l} \sum_{f_j} \log(P(dx, dy|a, l, l_j)). \tag{5}$$

The expression $\log(P(b_l|a))$ is the bin probability, which directly corresponds to an element of the feature vector according to $B_{a,l} = \log(P(b_l|a))$. Since there are A actions and L labels, this B vector contains AL elements.

3.2 Estimating relative location probabilities from training data

The previous section assumed that the relative location probabilities (RLPs) were simply available. However, these probabilities must be estimated from real data, requiring some care in the representation choice for the relative location probability tables (RLPTs). An RLPT represents an expression of the form $\log(P(dx, dy|a, l_i, l_j))$, where a, l_i, and l_j are considered fixed. In practice this means that it must represent a function from a (dx, dy) pair to a log probability,

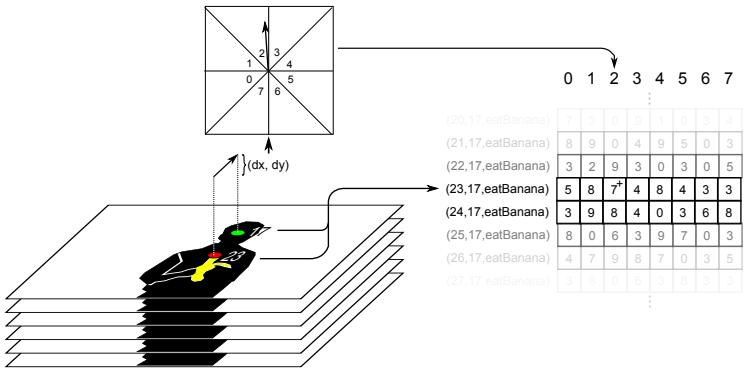

Fig. 5. Features with labels 17 and 23 are observed in a frame of a training video of the class eatBanana. The feature with label 17 has a relative displacement of (dx, dy) from that with label 23, which maps to bin #2 in an 8-way angular RLM. Thus, we increment bin #2 in the corresponding table entry; these counts are all converted to estimated log probabilities after the entire video collection is processed.

and that we must represent one such function for every (a, l_i, l_j) triplet of discrete values. While many representations are possible, we use an approach similar to that used for staged quantization.

We denote by $M(dx, dy)$ a function that maps a (dx, dy) pair to an integer bin label, allowing n such labels. We refer to this map as the relative location map (RLM), possible forms of which can be seen in Figure 3. Then the RLPT for a given (a, l_i, l_j) triplet is a list of n numbers, denoted T_{a,l_i,l_j}. An RLP can then be retrieved according to:

$$\log(P(dx, dy|a, l_i, l_j)) = T_{a,l_i,l_j}[M(dx, dy)]. \tag{6}$$

For example, with 216 labels, 10 actions, and 8 bins in the RLM, storing all the RLPs would require $(216)(10)(8) = 3,732,480$ entries in 466,560 tables.

Estimating the RLPTs is simply a matter of counting the displacements falling within each bin (see Figure 5), and finally normalizing by the total counts in each map. Since some bins may receive zero counts, leading to infinities when the log probability is computed, we use a prior to seed each bin with a fixed number of pseudo-counts. Examples of RLPTs found in this way can be seen in Figure 4.

This method could seemingly generate very sparse probability maps where most bins receive few or no real counts. However, in practice almost all of the bins receive counts. A typical situation (in this case our experiments on Rochester's Daily Living dataset) might have 10 classes, 216 feature labels, and a probability map with 8 bins, for a total of $(216^2)(10)(8) = 3.7 \cdot 10^6$ bins. For Rochester we have approximately 120 training videos, each of which is approximately 600 frames long. If we track 300 features per frame, and only consider in-frame pairs,

then across all videos we will have $(300^2)(600)(120) = 6.5 \cdot 10^9$ pairwise features. Thus, on average each bin will receive over a thousand counts.

3.3 Extension to temporal relationships

The method is naturally extended to include temporal relationships. Rather than representing the relationship between two features as a (dx, dy) pair, it is represented as a (dx, dy, dt) triple. The RLPT then contains entries of the form $\log(P(dx, dy, dt|a, l_i, l_j))$, which are indexed according to a mapping function $M(dx, dy, dt)$, so that

$$\log(P(dx, dy, dt|a, l_i, l_j)) = T_{a,l_i,l_j}[M(dx, dy, dt)]. \tag{7}$$

Previously, the map M could be stored as a simple image, whereas with spatial-temporal relationships this map is a volume or series of images. When counts are accumulated or probabilities evaluated, only pairs of features within a sliding temporal window are considered, since considering all pairs of features over the entire video would both result in a prohibitively large number of pairs and prevent the method from being run online on a video stream. Nevertheless, the change from considering only pairs within a frame to pairs within a temporal window vastly increases the number of pairs to consider, and depending on the number of features generated by a particular feature detector, it may be necessary to randomly sample pairs rather than considering them all. We find that for STIP-HOG we can consider all pairs, while for SCM-Traj we must sample.

3.4 Classification

We train a linear SVM [20] to classify video clips, which is a straightforward matter of presenting computed B vectors and corresponding ground truth classes from training clips. Each bin in B can be interpreted as a particular label's vote for an action, in which case the classifier learns the importance of each label's vote.

Since, when considered in isolation, pairwise relationships are unlikely to be as informative as the base features from which they are derived, we present a simple method for combining the raw base feature histograms with the computed pairwise log probability vectors. We do not present this combination method as the canonical way of combining the two sources of information, but rather as a convincing demonstration that the proposed pairwise relationships provide a significant additional source of information rather than merely a rearrangement of the existing data.

Supposing that H represents the histogram for the base features, and B represents the computed pairwise relationship vector, then one way of combining the two would be to simply concatenate the two vectors into $[H, B]$, and present the result to a linear SVM. However this is unlikely to result in the best performance, since the two vectors represent different quantities. Instead, we separately scale each part, and then simply cross validate to find the scaling ratio p that maximizes performance on the validation set, where the combined vector is $[pH, (1-p)B]$. This scaled vector is simply presented to the SVM.

Table 1. Action recognition accuracy on standard datasets. Adding pairwise features significantly boosts the accuracy of various base features.

Method	UCF-YT	Rochester
STIP-HOG (single) (Laptev *et al.* [1])	55.0%	56.7%
STIP-HOG (NB-pairwise alone)	16.4%	20.7%
STIP-HOG (D-pairwise alone)	46.6%	46.0%
STIP-HOG (single + D-pairwise)	59.0%	64.0%
STIP-HOG-Norm (single) (Laptev *et al.* [1])	42.6%	40.6%
SCM-Traj (single)	42.3%	37.3%
SCM-Traj (NB-pairwise alone)	14.3%	70.0%
SCM-Traj (D-pairwise alone)	40.0%	48.0%
SCM-Traj (single + D-pairwise)	47.1%	50.0%

4 Evaluation

We evaluate our pairwise method on a forced choice action classification task on two standard datasets, the UCF YouTube dataset (UCF-YT) [21] and the recently-released University of Rochester Activities of Daily Living [3]. To evaluate the contribution of our method for generating pairwise relationships, we consider two different types of base features: the trajectory based features we introduced earlier, and Laptev *et al.*'s space-time interest points. We consider both our discriminative formulation (denoted D-pairwise) and a Naïve-Bayes formulation (NB-pairwise) for our pairwise features, where the NB-pairwise results are primarily intended as a baseline against which to compare.

Table 1 summarizes our results. Our experiments are designed to evaluate the effect of adding spatial and temporal relations to the features and to understand in detail the effect of various parameters on the performance of the augmented features. Clearly, significantly more tuning and additional steps would go into building a complete, optimized video classification system. In particular, we do not claim that our performance numbers are the best that can be obtained by using complete systems optimized for these data sets. We use the evaluation metric of total accuracy across all classes in an n-way classification task.

On both datasets we use 216 base feature codewords for both trajectories and STIP-HOG. The number 216 results from the choice of three stages with a 6-way mask for the staged quantization ($6^3 = 216$), and we use the same number for STIP-HOG to make the comparison as even as possible. Likewise, for both datasets we use an 8-way spatial relationship binning for the RLPTs. Combined results are produced by cross validating on the scaling ratio.

UCF-YT consists of 1600 videos in 11 categories acquired from YouTube clips. For evaluation, we randomly split the dataset into a training set of approximately 1200 videos and a testing set of approximately 400 videos. This dataset was chosen for its difficulty, in order to evaluate the performance of pairwise relationships outside of highly controlled environments. In particular,

this dataset is challenging because the videos contain occlusions, highly variable viewpoints, significant camera motion, and high amounts of visual clutter.

On UCF-YT we find that discriminative pairwise features are not as informative as the base features, which is not unexpected since the diversity of the dataset means there are unlikely to be strong, consistent relationships between features. Nevertheless, we still find modest gains for combinations of pairwise and individual features, on the order of 5%. This means that the pairwise features are providing an additional source of information, rather than just obfuscating the information already present in the individual feature histograms. The Naïve Bayes pairwise evaluation performs poorly, but better than chance.

Furthermore, we can see that the performance of our simple fixed quantization on trajectories performs similarly to normalized STIP-HOG features, but significantly worse than non-normalized STIP histograms. This suggests that much of the discriminative power of STIP features might originate from the variation in the quantity of features found in different videos.

The Rochester dataset consists of 150 videos of five individuals performing a series of scripted tasks in a kitchen environment, acquired using a stationary camera. Due to the limited pool of available data, we evaluate using 5-fold cross-validation, using videos from four individuals for training and the fifth for testing, in each fold.

On Rochester we observe that the pairwise features for STIP-HOG do not perform as well as the individual STIP-HOG features, but that the combination outperforms both, which is consistent with the results for UCF-YT. For trajectory features, the pairwise features alone significantly outperforms the base features, a reversal from UCF-YT. The combination of the two outperforms both the individual and pairwise, but adds only a modest gain on top of the pairwise performance.

For both types of features, the gains with pairwise relationships in combination are much larger than for UCF-YT, which is explained by the greater consistency of spatial relationships between codeword labels due to the fixed viewpoint and highly consistent actions. Qualitatively examining which pairs contribute to a correct action identification supports this hypothesis: as can be seen in Figure 1, the pairwise features supporting an action appear to be strongly tied to that action. For STIP-HOG, the Naïve-Bayes pairwise formulation once again performs poorly, however for trajectories the Naïve-Bayes pairwise is the strongest performer. This suggests that for some applications, even simple relationships can give very good performance.

4.1 Effect of Temporal Relationships

The results with using spatial and temporal relationships on UCF-YT are shown in Table 2 in which X-T-Pairwise denotes the classifier (discriminative or Naïve-Bayes) augmented with temporal relations. For these results, we have used the same 8-way spatial relationship binning combined with a 5-way temporal binning, for a total of 40 bins. The pairwise relationships are evaluated over a 30

Table 2. Action recognition accuracy with temporal relationships on UCF-YT

Method	STIP-HOG	Traj-SCM
NB-Pairwise (baseline)	16.4%	14.3%
NB-T-Pairwise	22.2%	31.2%
D-Pairwise (baseline)	46.6%	40.0%
D-T-Pairwise	49.2%	39.7%

frame sliding window. For STIP-HOG, all pairs within the window are considered, but for trajectories we sample 1/20 of the pairs in the interest of tractability. Note that even with this sampling, a four second UCF-YT clip can produce over 100,000,000 pairs when using trajectory features.

The performance of the discriminative pairwise relationships remains virtually unchanged for Traj-SCM, but there is a modest performance boost for STIP-HOG. The Naïve-Bayes versions continue to perform worse than the discriminative ones, however the temporal relationships have a much larger impact on their performance. The difference is especially dramatic for NB-Pairwise vs. NB-T-Pairwise with STIP-HOG, where the temporal relationships have more than doubled the accuracy from 14.3% to 31.2%.

4.2 RLPT sparsity

Earlier we argued that the relative location probability tables should not be sparse based on a simple counting argument. Empirically, we find that for the Rochester dataset 71.6% of the entries receive counts, and that 91.7% of the tables have at least one count in one of the 8 bins. The number of tables containing at least 100 counts is 41.1%, and 15.2% of tables have over 1000 counts. These numbers validate our original claim that the tables are not sparse.

5 Conclusion

We present a simple yet powerful method for representing pairwise spatio-temporal relationships between features in the popular bag-of-words framework. Unlike naïvely expanding codewords to include all possible pairs and relationships between features, our method produces an output whose size is proportional to the number of base codewords rather than to its square, which reduces the likelihood of overfitting and is more computationally efficient. We demonstrate that our method can be used to improve action classification performance with dissimilar STIP-HOG (appearance) and trajectory (motion) based features on two different datasets, and that a discriminative formulation of our pairwise features generally outperforms a Naïve-Bayes classification approach. Although our method takes advantage of spatial relationships, it does not require any additional annotation in the training data, making it appropriate for a wide range of datasets and applications.

As we have only considered simple angular maps, there is potentially still considerable power to be extracted from this method through the careful selection of relative location maps. Additionally, we have presented a binning of probabilities based on codeword label, but an interesting question is whether more intelligent data-driven binnings can be found. We plan to explore these questions in future work.

References

1. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008)
2. Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. IJCV **79** (2008)
3. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV. (2009)
4. Carneiro, G., Lowe, D.: Sparse flexible models of local features. In: ECCV. (2006)
5. Crandall, D.J., Huttenlocher, D.P.: Weakly supervised learning of part-based spatial models for visual object recognition. In: ECCV. (2006)
6. Leordeanu, M., Hebert, M., Sukthankar, R.: Beyond local appearance: Category recognition from pairwise interactions of simple features. In: CVPR. (2007)
7. Zhang, Z.M., Hu, Y.Q., Chan, S., Chia, L.T.: Motion context: A new representation for human action recognition. In: ECCV. (2008)
8. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: ICCV. (2007)
9. Jiang, H., Martin, D.R.: Finding actions using shape flows. In: ECCV. (2008)
10. Junejo, I.N., Dexter, E., Laptev, I., Pérez, P.: Cross-view action recognition from temporal self-similarities. In: ECCV. (2008)
11. Johnson, N., Hogg, D.: Learning the distribution of object trajectories for event recognition. Image and Vision Computing **14** (1996)
12. Makris, D., Ellis, T.: Spatial and probabilistic modelling of pedestrian behaviour. In: BMVC. (2002)
13. Gilbert, A., Illingworth, J., Bowden, R.: Scale invariant action recognition using compound features mined from dense spatio-temporal corners. In: ECCV. (2008)
14. Gilbert, A., Illingworth, J., Bowden, R.: Fast realistic multi-action recognition using mined dense spatio-temporal features. In: ICCV. (2009)
15. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV. (2009)
16. Savarese, S., DelPozo, A., Niebles, J., Fei-Fei, L.: Spatial-Temporal correlatons for unsupervised action classification. In: WMVC. (2008)
17. Sun, J., Wu, X., Yan, S., Cheong, L.F., Chua, T.S., Li, J.: Hierarchical spatio-temporal context modeling for action recognition. In: CVPR. (2009)
18. Maji, S., Malik, J.: Object detection using a max-margin Hough transform. In: CVPR. (2009)
19. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: Action recognition through the motion analysis of tracked features. In: ICCV workshop on Video-oriented Object and Event Classification. (2009)
20. Chang, C.C., Lin, C.J.: LIBSVM – a library for support vector machines (2001)
21. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: CVPR. (2009)