Learning Neighborhood Co-occurrence Statistics of Sparse Features for Human Activity Recognition

Prithviraj Banerjee Ramakant Nevatia
Institute for Robotics and Intelligent Systems
University of Southern California, Los Angeles CA 90089
{pbanerje|nevatia}@usc.edu

Abstract

A common approach to activity recognition has been the use of histogram of codewords computed from Spatio Temporal Interest Points (STIPs). Recent methods have focused on leveraging the spatio-temporal neighborhood structure of the features, but they are generally restricted to aggregate statistics over the entire video volume, and ignore local pairwise relationships. Our goal is to capture these relations in terms of pairwise co-occurrence statistics of codewords. We show a reduction of such co-occurrence relations to the edges connecting the latent variables of a Conditional Random Field (CRF) classifier. As a consequence, we also learn the codeword dictionary as a part of the maximum likelihood learning process, with each interest point assigned a probability distribution over the codewords. We show results on two widely used activity recognition datasets.

1. Introduction

Automated recognition of human activities is a central problem of computer vision with important applications in video surveillance, video retrieval and human computer interaction. There has been considerable research effort in recognizing basic actions (like walking, running, waving and boxing) from a monocular view. While there exist approaches which model human actions as a sequence of key states of a Probabilistic Graphical Model (PGM) [18, 24, 12] , in this paper our focus is on approaches which avoid explicit modeling of the pose and dynamics of the human body, and also do not require the ability to detect and track the actor, which are inherently difficult tasks by themselves. An alternative approach is to directly model the human appearance and motion in the video, using methods based on template matching [1, 19], shape flow correlation [8] and interest point (IP) tracking [4, 13]; these approaches are sensitive to viewpoint variation and background clutter.

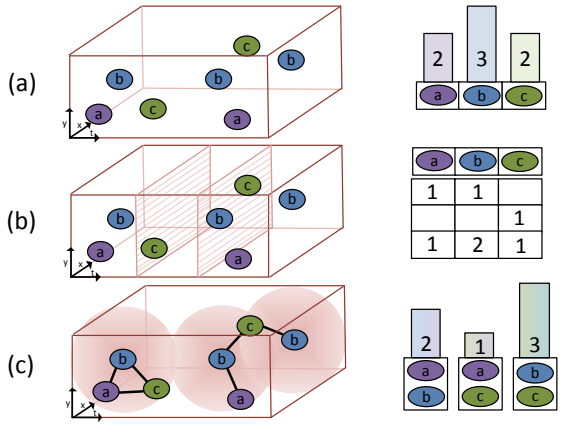

Figure 1. (a) Histogram computed over entire video volume. (b) Histogram computed over global spatio-temporal bins. (c) The feature-centric neighborhood of codeword *b* is shown in red sphere, with co-occurrence relationships transformed into edges of a graph.

Another popular approach is to classify videos based on the spatio-temporal interest points (STIPs) [22, 10, 3, 6, 7, 20]; they do not require person detection results and are robust to background clutter. Due to their localized and unstructured nature, they have been shown to be invariant to temporal and appearance variations across videos. Nevertheless, these methods typically ignore the spatial and temporal distribution of the features and rely only on a bag of features based model. For example, Schuldt et al [22] assume individual feature detections to be independent and construct a single histogram for the entire video (Figure 1.a), and hence capture a single aggregate statistics of the codewords over the entire spatio-temporal volume. To address these issues, there exist methods [11, 23, 25] which use a variety of binning structures to capture the relative layout of the STIPs (Figure 1.b), however the bin partition boundaries are rigid, and hence can be sensitive to spatial or time shifts of the activity segment in the video volume.

Aggregate statistics combined with global bin partitions

ignore local neighborhood relationships, and are only able to capture global relationships between the partitions. Consider the video volume in Figure 1.c, where the feature centric neighborhood of codeword 'b' is shown as a red sphere. We observe that codewords *b* and *c* co-occur quite frequently, whereas *a* and *c* rarely co-occur in the same neighborhood. Such local relationships can be captured by statistics which are pairwise (instead of aggregate), and feature-centric with a notion of local neighborhood associated with them (instead of global clip level partitions).

We aim to address these issues and model the neighborhood relationships in terms of a count function which measures the pairwise co-occurrence frequency of codewords. We describe a transformation to represent the count function in terms of the edge connectivity of latent variables of a CRF [1] classifier, and explicitly learn the co-occurrence statistics as a part of its maximum likelihood objective function. The probabilistic nature of our method allows us to naturally incorporate codebook learning into the CRF learning process, and hence retains the discriminative power of the STIP descriptors. The resulting latent CRF shares the same parametrization as a Hidden Conditional Random Field (HCRF) [17, 27], and hence we can leverage existing training and inference algorithms. Our method is transparent to the type of STIP detector used in the observation layer. We show results using the sparse 3D Harris corner interest points [10]. We evaluate our framework on the Weizmann [1] and KTH [22] activity datasets, and compare with other existing approaches.

2. Related Work

Due to the size of the literature, we focus only on interest point based methods and PGMs for activity recognition.

Numerous interest point based methods have been proposed in the activity recognition community. Recent work has focused on learning spatio-temporal structure of STIPs by a variety of methods like using global grid partitions [11, 23, 25], feature centric grids [9, 5] and latent topic models [16]. However they fail to capture the discriminative pairwise relationships, and only compute aggregate statistics. Bregonzio et al [2] capture the global distribution from a cloud of STIP detections, although they rely solely on the location and scale of the detections. [29] use shape context features to learn the neighborhood structure, however they require accurate motion images.

We briefly survey some of the structural methods for activity recognition using Probabilistic Graphical Models (PGM). Generative models like Bayesian Networks have been used to model key pose changes [12, 18]. Discriminative models like a CRF network [24] have been used to model the temporal dynamics of silhouettes based features, like shape context and pair wise edge features. Wang et al [26] proposed a Hidden Conditional Random Field (HCRF) for gesture recognition, which introduced a latent variable layer in the CRF network. Morency et al [14] model the dynamics between gesture labels by using a Latent-Dynamic CRF model. These methods aim to leverage the temporal structure present in human activities, however they require the ability to detect and track the actor which is an inherently difficult task by itself. Furthermore, models are often represented in terms of 3-D pose sequences which can be difficult to acquire. There exist graphical model based approaches [15, 27, 28] which learn a *part based model* from the underlying STIP detections, but they rely on per-frame classification and can not be extended to a video-wide model in an obvious manner.

3. Learning Co-Occurrence statistics of STIPs

We define the co-occurrence statistic as a count function $C(b, c)$, which counts the number of times codewords b and c co-occur in the same neighborhood. We aim to learn a human activity classifier which leverages these statistics in a discriminative manner. To this end, we introduce a logistic regression model for classifying histogram of codeword based features. We define potential functions over latent variables for learning the codeword assignments, and show a transformation of the $C(b, c)$ function in terms of edge connectivity of the latent variables. We show that the parametrization of the resulting latent CRF model is equivalent to a Hidden Conditional Random Field [17, 27], and hence existing techniques for HCRF can be leveraged to train our model.

3.1. CRF for Bag-of-Words classifier

Let $\mathcal{Y}$ be the set of class labels and $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ be the set of m interest point (IP) detections in the video volume, where $\boldsymbol{x}_j \in R^K$ is a K-dimensional feature descriptor vector of the j^{th} IP. The classical approach is to learn a dictionary of codewords $\mathcal{H}$ using K-Means clustering. The j^{th} IP is assigned a codeword $CW(\boldsymbol{x}_j) \in \mathcal{H}$ based on its nearest cluster center. A histogram of codewords $\mathbf{g} \in R^{|\mathcal{H}|}$ is constructed over the video volume. The histogram $\mathbf{g} = \left[g_1, g_2, \ldots, g_{|\mathcal{H}|}\right]$ can be computed using an indicator function [2] as follows:

$$g_i = \sum_{\boldsymbol{x}_j \in X} \mathbf{1}_{CW(\boldsymbol{x}_j)=i} \tag{1}$$

Traditionally a χ^2 kernel based SVM classifier [22] is learned on the histogram $\mathbf{g}$, however it is not obvious how to incorporate pairwise relationships explicitly in a SVM

[1] CRF is an undirected graphical model representing the conditional probability distribution of the unobserved variables conditioned over the observed variables.

[2] $\mathbf{1}_{a=b}$ returns 1 if $a = b$ is true, otherwise returns 0.

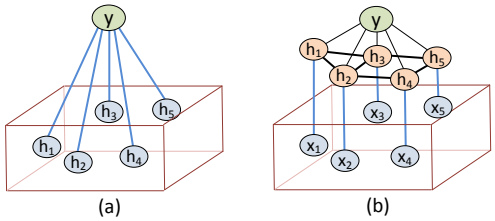

Figure 2. CRF formulation for Bag-of-Words classifier: (a) Logistic Regression model with pre-assigned codewords as observations. (b) CRF with latent variables where the edges represent the co-occurrence relationships between codewords.

framework. We propose a CRF classifier based approach and define appropriate potential functions to capture the desired statistics. We redefine the codeword assignment function in terms of a vector $\boldsymbol{h} = \left[h_1, h_2, \ldots, h_{|m|}\right]^T$. Each element $h_j \in \mathcal{H}$ denotes the codeword associated with the j^{th} interest point, and hence we have $\forall_{\boldsymbol{x}_j \in X} : h_j = CW(\boldsymbol{x}_j)$. The logistic regression classifier (a type of CRF) parametrized by weight vector $\boldsymbol{\beta}$, is defined over the class label $y \in \mathcal{Y}$ and the histogram feature vector $\boldsymbol{g}$ as follows:

$$
\begin{aligned}
P(y|X;\boldsymbol{\beta}) = P(y|\mathbf{h},X;\boldsymbol{\beta}) \quad &\propto exp\left\{\sum_{a\in\mathcal{Y}} \mathbf{1}_{y=a}\boldsymbol{\beta}_a^T\mathbf{g}\right\} \\
&= exp\left\{\sum_{a\in\mathcal{Y}} \mathbf{1}_{y=a} \sum_{i\in|\mathcal{H}|} \beta_{ai} g_i\right\} \\
&= exp\left\{\sum_{a\in\mathcal{Y}} \mathbf{1}_{y=a} \sum_{i\in|\mathcal{H}|} \beta_{ai} \sum_{\boldsymbol{x}_j\in X} \mathbf{1}_{CW(\boldsymbol{x}_j)=i}\right\} \\
&= exp\left\{\sum_{\boldsymbol{x}_j\in X}\sum_{a\in\mathcal{Y}}\sum_{i\in|\mathcal{H}|} \beta_{ai}\mathbf{1}_{y=a}\mathbf{1}_{h_j=i}\right\}\Bigg|_{\forall j:h_j=CW(\boldsymbol{x}_j)} \\
&= exp\left\{\sum_{\boldsymbol{x}_j\in X} \boldsymbol{\beta}^T\Phi_1(y,h_j)\right\}\Bigg|_{\forall j:h_j=CW(\boldsymbol{x}_j)} \qquad (2)
\end{aligned}
$$

where $\Phi_1(y,h_j)$ is a potential function which returns a $|\mathcal{Y}||\mathcal{H}|$ dimensional vector whose elements are the product of the indicator functions $\mathbf{1}_{y=a}\mathbf{1}_{h_j=i}$. The weight vector $\boldsymbol{\beta}$ can be learned by maximizing the conditional likelihood over the training data. Figure 2.a shows the corresponding CRF model, in this case a Logistic Regression classifier.

3.1.1 Latent Variables for Learning Code-Words

The function $\Phi_1(y,h_j)$ takes only the codeword assignment as input parameter, and hence is independent of the actual interest point descriptor $\boldsymbol{x}_j$. We generalize it to incorporate codeword assignment into the maximum likelihood learning process. We remove the restriction of $\forall_{\boldsymbol{x}_j\in X}$: $h_j = CW(\boldsymbol{x}_j)$, and declare $\boldsymbol{h}$ to be a vector of *latent* random variables, whose associated probability mass function $P(\boldsymbol{h}|X;\boldsymbol{\alpha})$ we would like to learn as a part of the training process. An interesting consequence is that each interest point has a distribution of codewords $P(h_j|\boldsymbol{x}_j;\boldsymbol{\alpha})$ associated with it, instead of a single codeword per interest point. We define $P(\mathbf{h}|X;\boldsymbol{\alpha})$ in terms of potential function $\Phi_2(\boldsymbol{x}_j,h_j)$ as follows:

$$
\begin{aligned}
P(\mathbf{h}|X;\boldsymbol{\alpha}) &\propto exp\left\{\sum_{\boldsymbol{x}_j\in X} \boldsymbol{\alpha}^T\Phi_2(\boldsymbol{x}_j,h_j)\right\} \qquad (3)\\
&= exp\left\{\sum_{\boldsymbol{x}_j\in X}\sum_{c\in\mathcal{H}} \mathbf{1}_{h_j=c}\cdot\alpha_c^T\boldsymbol{x}_j\right\}
\end{aligned}
$$

3.1.2 Incorporating Co-Occurrence Statistics

An interest point $\boldsymbol{x}_j$ lies in the neighborhood of $\boldsymbol{x}_k$ iff $\boldsymbol{x}_j \in Nb(\boldsymbol{x}_k)$. We introduce suitable neighborhood functions $Nb(\cdot)$ in Section 3.3. We define an indicator function $\Lambda(\boldsymbol{x}_j,\boldsymbol{x}_k)$:

$$
\Lambda(\boldsymbol{x}_j,\boldsymbol{x}_k) = \begin{cases} 1 & \text{if } \boldsymbol{x}_j \in Nb(\boldsymbol{x}_k) \\ 0 & \text{otherwise} \end{cases} \qquad (4)
$$

We model the indicator function Λ in terms of a graph : let the set of nodes $h_j (j = 1,\ldots,m)$ correspond to the vertices in a graph $G = (E,V)$, and the set of edges E is given as:

$$
E = \{(j,k): \quad j,k \in V, \quad \Lambda(\boldsymbol{x}_j,\boldsymbol{x}_k) = 1\} \qquad (5)
$$

i.e. there is a edge connecting h_j and h_k, iff $\boldsymbol{x}_j$ lies in the neighborhood of $\boldsymbol{x}_k$. Figure 1.b shows an example of the graph transformation of the co-occurrence relationship between the codewords. A count function is defined as:

$$
\begin{aligned}
C(b,c) &= \sum_{(\boldsymbol{x}_j,\boldsymbol{x}_k)\in X} \mathbf{1}_{CW(\boldsymbol{x}_j)=b}\cdot\mathbf{1}_{CW(\boldsymbol{x}_k)=c}\cdot\Lambda(\boldsymbol{x}_j,\boldsymbol{x}_k) \\
&= \sum_{(j,k)\in E} \mathbf{1}_{h_j=b}\cdot\mathbf{1}_{h_k=c} \qquad (6)
\end{aligned}
$$

The conditional class label distribution is defined as:

$$
\begin{aligned}
P(y|\mathbf{h},X;\boldsymbol{\gamma}) &\propto exp\left\{\sum_{a\in\mathcal{Y}} \mathbf{1}_{y=a}\boldsymbol{\gamma}_a^T\mathbf{C}(\cdot)\right\} \qquad (7)\\
&= exp\left\{\sum_{a\in\mathcal{Y}}\sum_{b\in\mathcal{H}}\sum_{c\in\mathcal{H}} \gamma_{a,b,c}\mathbf{1}_{y=a}C(b,c)\right\} \\
&= exp\left\{\sum_{(j,k)\in E}\sum_{a\in\mathcal{Y}}\sum_{b\in\mathcal{H}}\sum_{c\in\mathcal{H}} \gamma_{a,b,c}\mathbf{1}_{y=a}\mathbf{1}_{h_j=b}\mathbf{1}_{h_k=c}\right\} \\
&= exp\left\{\sum_{(j,k)\in E} \boldsymbol{\gamma}^T\Psi\left(y,h_j,h_k\right)\right\}
\end{aligned}
$$

The resulting CRF with $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$ is as follows:

$$P(y|X;\boldsymbol{\theta}) = \sum_{\mathbf{h}\in\mathcal{H}^{|V|}} P(\mathbf{h}|X;\boldsymbol{\alpha})P(y|\mathbf{h},X;\boldsymbol{\beta})P(y|\mathbf{h},X;\boldsymbol{\gamma})$$
$$\propto \sum_{\mathbf{h}\in\mathcal{H}^{|V|}} exp\Big\{ \sum_{j\in V} \boldsymbol{\alpha}^T \Phi_2(x_j,h_j) + \sum_{j\in V} \boldsymbol{\beta}^T \Phi_1(y,h_j) + \sum_{(j,k)\in E} \boldsymbol{\gamma}^T \Psi\left(y,h_j,h_k\right) \Big\} \quad (8)$$

The observation features are conditionally independent given the hidden node values, and hence the dependencies between the observations are modeled through the hidden layer only. In our case, the edge set E directly models the neighborhood function $Nb(\cdot)$. Also the HCRF model parametrization $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$ is independent of the number of observations, and hence independent of the number of hidden nodes, which makes it well suited for classifying varying length videos.

3.2. Hidden Conditional Random Fields

The latent CRF (Eqn. 8) shares the same parametrization as that of a Hidden Conditional Random Field [17, 27], nevertheless there are significant differences in how we define and interpret the potential functions of our respective CRF models. We view our model from a Bag-Of-Words perspective, with each latent variable corresponding to a codeword assignment, and also interpret the edges connecting the latent variables as representing co-occurrence relationships between the interest points. In contrast, [27] treats latent variables as a 'part' belonging to a constellation model (akin to a pictorial structure); an interpretation which does not carry over to sparse STIP features. Furthermore, they use dense optical flow features, require stabilized human detection windows, and perform classification on a per-frame basis. Hence it is not obvious how it can be extended to incorporate sparse features like Harris 3D corners. Extended segments of video can be void of STIPs, which makes it challenging to apply frame-by-frame methods. Inspite of these differences, due to the shared parametrization of the CRF equations, we use the existing HCRF learning and inference procedure.

The model parameters $\boldsymbol{\theta}$ are learned by maximizing the conditional log likelihood on the training data. The gradient expression can be represented as a set of expectations over the posterior probability of the hidden parts and the class labels: $P\left(h_j, y|X;\boldsymbol{\theta}\right)$ and $P\left(h_j, h_k, y|X;\boldsymbol{\theta}\right)$, which are inferred using belief propagation. Exact belief propagation is possible only if the latent layer edges form a forest, and its computational complexity is given as: $O(|E||\mathcal{Y}||\mathcal{H}|^2)$. We train a binary HCRF classifier ($\mathcal{Y} = \{0, 1\}$) for each class label, and the final classification is determined as: $y^* = argmax_{i\in\mathcal{A}}\left\{P(y=1|X;\boldsymbol{\theta}_i)\right\}$, where $\mathcal{A}$ is the set of action categories.

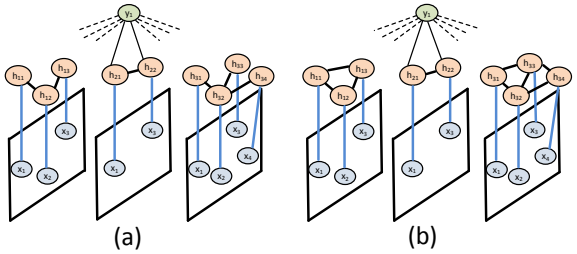

Figure 3. Latent layer edge set defined as: (a) E_{MST} (b) E_{2Con}

3.3. Latent Layer Connectivity

The edge set E models the neighborhood structure as captured by the neighborhood function $Nb(\boldsymbol{x}_j, \boldsymbol{x}_k)$. The simplest neighborhood function is a Euclidean ball function with radius r such that: $Nb\left(\boldsymbol{x}_k\right) = \{\boldsymbol{x}_j : ||p_j - p_k|| < r\}$, where p_j, p_k are the spatio-temporal location of the interest points. This leads to a densely connected edge set E with large number of cycles, and exact belief propagation would be difficult. We propose simpler connectivity models, and show results validating their effectiveness in capturing the neighborhood structure.

[Minimum Spanning Tree] We propose a distance function based on the Minimum Spanning Tree (MST), which allows exact belief propagation for inference during HCRF training. We define the neighborhood function as: $Nb\left(\boldsymbol{x}_k\right) = \{\boldsymbol{x}_j : (j,k) \in \text{MST}(G_{t_k})\}$, where G_{t_k} is a graph with nodes as the set of interest points with the same frame index as node $\boldsymbol{x}_k$, and edge weights equal to their pairwise spatial Euclidean distance. The resulting Euclidean MST has a link between each node and its closest neighbor [3], and hence the neighborhood function captures the relationship of interest points with respect to its closest spatial neighbor. Similar tree structures have been used for learning part based models for object detection [17] and activity recognition [27]. The latent layer edge set E_{MST} is shown in Figure 3.a .

[2-Edge-Connected Graph] A graph is k-edge-connected if it remains connected whenever fewer than k-edges are deleted from the graph. We construct a 2-edge-connected edge set E_{2Con} by adding edges to E_{MST} such that every node is linked to their two closest spatial neighbors. Similar graphs have been used earlier for object recognition [17]. Note that $E_{\text{Mst}} \subset E_{\text{2Con}}$, and hence E_{2Con} should be more robust to perturbations in interest point locations across videos of the same class. Figure 3.b shows the CRF model corresponding to an E_{2Con} edge set. The latent layer contain cycles, and hence requires approximate inference methods like loopy belief propagation.

[3] Nearest Neighbor Graph is a subgraph of the Euclidean MST.

4. Results

We validate our framework on two widely used human activity datasets: Weizmann [1] and KTH [22]. We use the original version of the Weizmann dataset with 9 action categories: walking, running, jumping, sideways, bending, one-hand-waving, two-hands-waving, jumping in place and jumping jack. Each action was performed by 9 different actors resulting in a total of 81 videos in the dataset. The KTH dataset contains 600 videos of 25 actors performing 6 actions: walking, jogging, running, boxing, hand-waving and hand-clapping; and is repeated in different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with clothing variation (s3) and indoors (s4).

In our experiments, we selected the 3D Harris corner [10] as our choice of STIP detector. It has been extensively used in human activity recognition literature, and provides a good baseline for comparison of our framework with existing systems. However, our framework is independent of the type of interest point used in the observation layer. We generate a 144 dimensional HoG-HoF descriptor to capture the shape and flow information in the neighborhood of the STIPs, using code provided by [11] with default parameters.

We train the HCRF classifiers with the interest points as observation features, using our extension[4] of the HCRF library made available by [17]. Our method was implemented on a 3.16 GHz Intel Xeon CPU. The average training time for the KTH dataset (48 videos with ~ 500 frames each) was ~ 30 to 60 hours depending on cardinality of $\mathcal{H}$. The average test time was ~ 30 seconds for a single video given the interest point descriptors.

4.1. Weizmann

We validated our approach on the Weizmann dataset using Leave-One-Out Cross-Validation (LOOCV), which is the standard experimental setup also used by others. The 9 actions are automatically split into two categories based on the net motion of the interest point detections: Mobile actions (walking, running, jogging, gallop-sideways) and Stationary actions (bending, one-hand-waving, two-hands-waving, jumping in place and jumping jack).

Table 1 shows our results with varying codebook size $|\mathcal{H}|$ using E_{MST} and E_{2Con} connectivity matrix. We get good results with a much smaller codebook size because each interest point can be associated with multiple codewords based on the class conditional distribution $P(h|y, X)$. The connectivity matrix describes the neighborhood function being modeled. Almost similar accuracy results were obtained for E_{2Con} connectivity, with a difference of only single video being misclassified. Hence good results can be achieved with relatively simple definitions of neighborhood functions. Figure 4.a shows our confusion matrix on Weizmann dataset using LOOCV, and achieves an accuracy rate of 98.76% using E_{2Con} and $|\mathcal{H}| = 15$. We only make a single classification error out of 81 test samples. Table 2 contrasts our performance with others. We consistently outperform other exisiting approaches which attempt to model the neighborhood structure [2, 16, 29, 15]. There exist approaches [1, 21, 28] which have achieved perfect results on this dataset, however they either require accurate silhouettes, fixed-size image windows centered at the person of interest or track information to stabilize the videos. Our interest point approach does not require such information, but gives comparable performance.

[4]The inference engine is replaced using the libDAI library: *http://people.kyb.tuebingen.mpg.de/jorism/libDAI/*.

$\|\mathcal{H}\| =$	10	15	20
E_{MST}	97.53%	96.30%	97.53%
E_{2Con}	96.34%	$\mathbf{98.76}\%$	97.53%

Table 1. Comparison with varying codebook size $|\mathcal{H}|$ and different edge connectivity on Weizmann dataset.

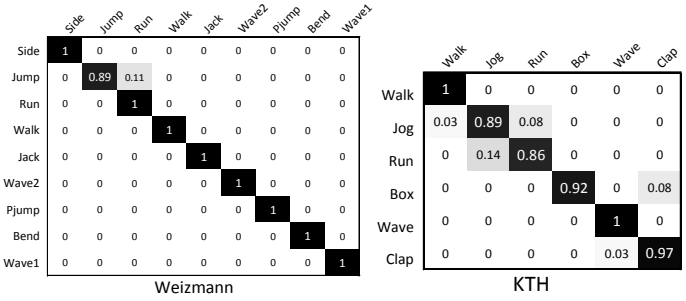

Figure 4. Confusion Matrices: (a) Weizmann dataset with E_{2Con} connectivity and $|\mathcal{H}| = 20$. Average accuracy over all classes is $\mathbf{98.76}\%$. (b) KTH dataset with E_{MST} connectivity and $|\mathcal{H}| = 20$ for s1,s2 and $|\mathcal{H}| = 10$ for s3,s4 scenarios. Average accuracy over all classes is $\mathbf{93.98}\%$

4.2. KTH

Following the experimental setup of [22], we use video clips of 16 actors as our training set, and use the remaining 9 actors as our test set. The actions are automatically split into two categories based on the net motion of the interest point detections: Mobile actions (walking, running, jogging) and Stationary actions (boxing, hand-waving and hand-clapping). The experiments on KTH were run with E_{MST} connectivity and $|\mathcal{H}| = 20$ for s1,s2 and $|\mathcal{H}| = 10$ for s3,s4. Figure 4.b shows our average confusion matrix for all four scenarios: s1, s2, s3 and s4. We note that majority of the confusion is between actions jog and run, which is expected due to their similar nature.

Table 2 compares our average accuracy rate with other methods. We achieve an average accuracy score of 93.98%. A direct comparison should be made carefully as some of the results are reported on the easier $24 : 1$ train-test ratio. Our results are most directly comparable to Schuldt et al [22] as they compute a single aggregate historgam

	Method	Weizmann	KTH	Train/Test
I	Our Approach	98.76%	93.98%	16:9
	Laptev et al [11]	95.06%[5]	91.80%	16:9
	Schuldt et al [22]	-	71.72%	16:9
II	Bregonzio et al [2]	96.66%	93.17%	24:1
	Zhang et al [29]	92.89%	91.33%	24:1
	Niebles et al [15]	72.80%	-	-
	Kovashka et al [9]	-	94.53%	16:9
	Gilbert et al [5]	-	94.50%	16:9
	Ryoo et al [20]	-	91.10%	16:9
	Jingen et al [7]	-	94.16%	24:1
III	Wang et al [28]	100.0%	92.51%	1:1
	Wang et al [27]	97.20%	87.60%	1:1
	Schindler et al [21]	100.0%	92.70%	4:1
	Jhuang et al [6]	98.8%	91.70%	16:9
	Dollar et al [3]	85.20%	81.17%	24:1

Table 2. Comparative results on the Weizmann and KTH datasets. (I) Comparison of our approach with methods using global bin partitions and aggregate statistics. (II) Approaches that model the spatio-temporal relationships between interest points. (III) Approaches from general activity recognition systems, not necessarily based on interest points. Last column shows the train/test split ratio for KTH.

over all the codewords detected in a video, and do not model any spatio-temporal relationships between the interest points. We achieve a significant improvement in performance ($\sim 22\%$) over [22], which we attribute to our learning co-occurrence relationships between interest points. We also outperform [11] which learns relationships between global bin partitions, and hence demonstrate the advantages of learning pairwise local neighborhood statistics like co-occurence features. Note that our performance is significantly better than [29, 20, 27, 6, 3], and is comparable to [9, 5] with the difference of a single misclassification. Furthermore, both [9, 5] use dense features which are much more expensive to compute.

5. Conclusion

We proposed a novel framework for learning the local neighborhood relationships between STIP features, and train a CRF based human activity classifier. The neighborhood relationships are modeled in terms of pairwise co-occurrence statistics. We showed a transformation to represent these statistics as edges between the latent variables of a Conditional Random Field. We validate our framework on two widely used human activity datasets: Weizmann and KTH, and show improvements over other existing approaches. Our method can be naturally extended to capture more complex relationships, *e.g.* temporal relationships, which is a part of our current ongoing work.

[5]The results are from our implementation of [11]. It uses Multiple Kernel Learning [9] instead of greedy kernel selection.

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, Dec. 2005.

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

[3] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior Recognition via Sparse SpatioTemporal Features. In *VSPETS*, 2005.

[4] C. Fanti, L. Zelnik-Manor, and P. Perona. Hybrid Models for Human Motion Recognition. In *CVPR*, 2005.

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

[6] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A Biologically Inspired System for Action Recognition. In *ICCV*, Oct. 2007.

[7] L. Jingen and M. Shah. Learning human actions via information maximization. In *CVPR*, June 2008.

[8] Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal Shape and Flow Correlation for Action Recognition. In *CVPR*, June 2007.

[9] A. Kovashka and K. Grauman. Learning a Hierarchy of Discriminative Space-Time Neighborhood Features for Human Action Recognition. In *CVPR*, 2010.

[10] I. Laptev. On Space-Time Interest Points. *IJCV*, 2005.

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

[12] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, 2007.

[13] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, Sept. 2009.

[14] L.-P. Morency, A. Quattoni, and T. Darrell. Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. In *CVPR*, June 2007.

[15] J. C. Niebles and L. Fei-Fei. A Hierarchical Model of Shape and Appearance for Human Action Classication. In *CVPR*, 2007.

[16] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *IJCV*, Mar. 2008.

[17] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *PAMI*, Oct. 2007.

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

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

[20] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, Sept. 2009.

[21] K. Schindler and L. Van Gool. Action Snippets: How many frames does human action recognition require? In *CVPR*, 2008.

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

[23] C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009.

[24] C. Sminchisescu, A. Kanaujia, Z. Li, and M. Dimitris. Conditional models for contextual human motion recognition. In *ICCV*, 2005.

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

[26] S. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell. Hidden Conditional Random Fields for Gesture Recognition. In *CVPR*, 2006.

[27] Y. Wang and G. Mori. Learning a discriminative hidden part model for human action recognition. In *NIPS*, 2008.

[28] Y. Wang and G. Mori. Max-margin hidden conditional random fields for human action recognition. In *CVPR*, June 2009.

[29] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia. Motion Context : A New Representation for Human Action Recognition. In *ECCV*, 2008.