Pose based Activity Recognition using Multiple Kernel Learning

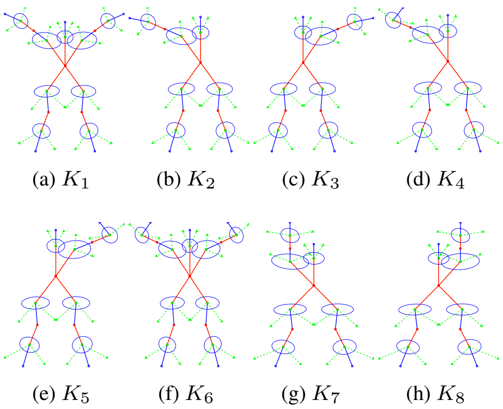
Prithviraj Banerjee Ramakant Nevatia
University of Southern California[*]
{*pbanerje,nevatia*}@*usc.edu*

Abstract

We describe a method for activity recognition based on distribution of human poses in a video. Pose estimation has shown to be sensitive to the priors given to the inference method; we use a collection of distinctive kinematic tree priors to model the variety of pose variations present in a video. Feature histograms are computed from vector quantized descriptors derived from the pose estimates. A learned Multiple Kernel SVM classifier is used to combine the various histograms to give activity classifications. We report results on a publicly available human gesture dataset. [*]

1. Introduction

Recognizing single actor human actions is a central problem of computer vision with important applications in video surveillance, video retrieval and human computer interaction. A common approach is to compute global histogram based statistics of local spatio-temporal features in the video volume, which are used to train a discriminative classifier [6, 7]. These approaches do not require significant annotations or model construction, however they completely ignore the structure present in human activities. A complementary approach is to train a classifier (such as a dynamic bayesian network, hidden markov model etc.) based on the human pose dynamics present in an action, while using pose estimates as observations [5, 12]. Such methods leverage their knowledge of the structure of the activity, however training such models requires semantic decomposition of the activity into key-poses, with interpolations defined for the intermediate states. This imposes significant annotation and modeling requirements on the end user during training. The classifier is also specific to a particular style and duration of the activity.


(a) K_1 (b) K_2 (c) K_3 (d) K_4

(e) K_5 (f) K_6 (g) K_7 (h) K_8

Figure 1: Kinematic tree priors K_1 to K_8 representing distinct mean-pose configurations. The one-σ boundary of the Gaussian distribution for relative location and orientation between part pairs is shown using blue ellipses and green lines respectively.

We are motivated by recent developments in 2D human pose detection [10, 4, 13], and introduce a classification method that uses the structure provided by human pose without constraining it to a specific dynamical model. Pose estimation is more reliable for poses which are similar to the mean-pose configuration represented in the prior, and it is hard to design a single prior which works for all poses present in an activity. Hence we use a collection of pose priors, which helps in detecting the distinct poses present in an activity by at-least one of the pose priors. Figure 1 shows examples of pose priors used in our work. The multiple pose detection responses are combined in a multiple kernel learning (MKL) framework to classify the videos into action categories. We test our framework on a human gesture dataset, and provide convincing results in support of our framework.

[*] Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0063. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. The authors thank Vivek Kumar Singh for useful discussions and insights on our work.

2. Human Pose Estimation

Human poses provide strong semantic cues to the underlying activity. Pictorial structure based object detection techniques have improved the quality and reliability of modern pose detection algorithms [10]. We use pose detection results of the human body, as input observations to our activity recognition module. We represent a human body using a 10-part model: head, torso, and upper/lower limbs of arms/legs. The 2D pose model consists of nodes $\boldsymbol{l}_i$ corresponding to each part i, with edges between the nodes enforcing spatial constraints on their arrangements. The complete human pose is represented as the configuration of the parts given by $L = \{\boldsymbol{l}_0, \boldsymbol{l}_1, \cdots, \boldsymbol{l}_N\}$, where the state of part i is defined as $\boldsymbol{l}_i = (x_i, y_i, \theta_i)$. Given the observed image I, the pose posterior distribution is defined as:

$$\begin{aligned} p(L|I) &\propto p(I|L)p(L) \qquad (1)\\ &\propto exp\left(\sum_i \boldsymbol{\phi}\left(l_i; I_i\right) + \sum_{(i,j)\in E} \boldsymbol{\psi}\left(l_i, l_j\right)\right) \end{aligned}$$

where $(i, j) \in E$ are the pairwise constraints between the parts. In standard implementations [10, 4] a tree graph configuration rooted at the torso is chosen, which ensures exact inference procedures. $p(L)$ represents the prior on the part configurations, and are derived from the kinematic constraints imposed on the parts. Details on constructing meaningful priors for activity recognition are discussed in Section 2.1. $p(I|L)$ is the likelihood of the image observation, given a particular configuration state of the parts. The joint probability is decomposed as $p(I|L) \propto \prod_i p\left(I_i|\boldsymbol{l}_i\right)$, where each term $p\left(I_i|\boldsymbol{l}_i\right)$ is computed from independent part detectors applied over the detection window. We assume human location in the image is known apriori. We use the boundary and region template-based part detectors provided by [10]. The marginal posterior probability for each body-part is inferred using belief propagation over the graphical model defined in equation 1.

2.1 Kinematic Pose Priors

The pose configuration prior is encoded in the distribution $p(L)$. A variety of pose priors have been used in the literature ranging from tree priors set to uniform probabilities within a bounded range [11], non-tree priors with occlusion reasoning built into the model [8] and fully connected graph models using absolute part orientations [3]. We choose to use the kinematic tree priors (KTPs) proposed by Andriluka et al [1], because they are easy to design for the end user, and also exact inference algorithms is possible for such priors. KTPs

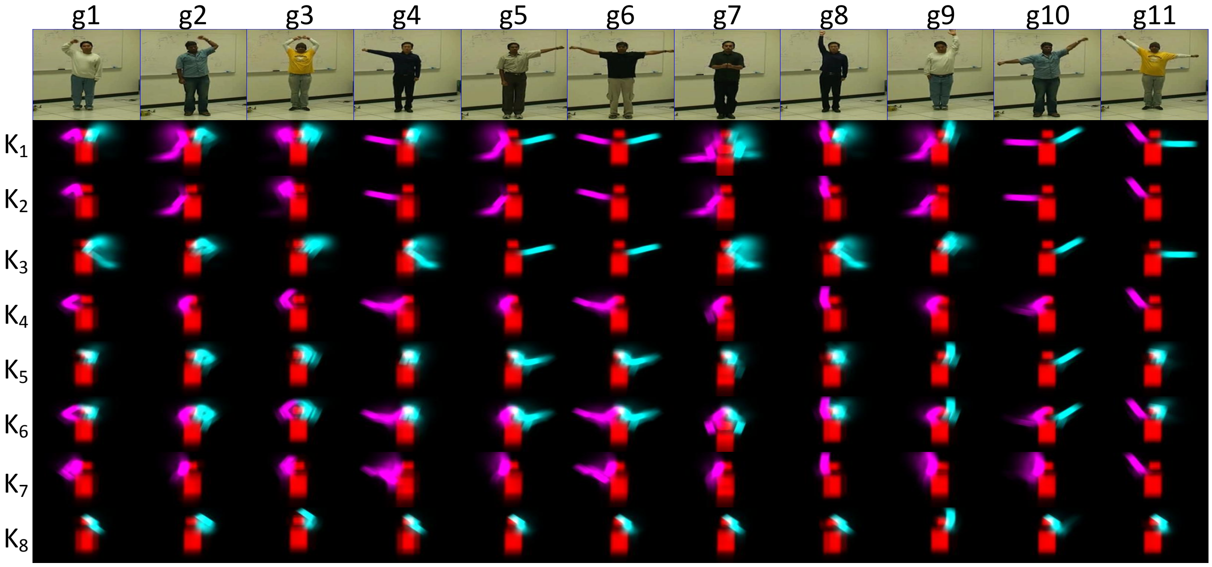


Figure 2: Sample pose estimation results on the USC-Gesture Dataset using kinematic prior trees K_1 to K_8. The results display the part-wise marginal posterior distributions $p(l_i = (x, y, \theta)\,; k)$. The distributions have significantly lower entropy for frames closer to the mean-pose configuration of the KTPs, for example, K_1G_6, K_2G_4, K_6G_3 and others. (This figure is best viewed in color).

decompose the kinematic constraints over L as

$$p(L) = p(\boldsymbol{l}_0) \prod_{(i,j)\in E} p\left(\boldsymbol{l}_i|\boldsymbol{l}_j\right) \tag{2}$$

where the relative location and orientation arrangement of each child part w.r.t. the parent part is encoded in terms of a Gaussian distribution.

A common approach is to learn a single KTP [11, 1], which is representative of the human poses present in the dataset. However detectors based on a single prior tend to return confident estimates only for the poses similar to the configuration represented in the prior. Hence we design a collection of prior trees $K_i \in \mathcal{K}$, representing a variety of mean-pose configurations, which helps in detecting the distinct poses present in an activity by at-least one of the pose priors. Figure 1 shows examples of KTPs defined in our current implementation, each with a distinctive mean-pose configuration. Figure 2 shows results with different KTPs; it is clear that estimates are much better when the actual pose is similar to the prior pose.

The set of Gaussian parameters, corresponding to the mean relative orientation between each pair of parts, defines the mean-pose configuration of the KTP. The parameters are easily set via visual inspection, for example, K_1 is similar to a pose with both hands raised above the head, where as K_5 represents a pose with bent elbows. The rest of the Gaussian parameters are set to generic values independent of the activity dataset, and are determined empirically from a standard pose estimation dataset [10].

The kinematic tree priors are extended to contain a set of flags defining which body parts are important to capture a particular pose. Hence a KTP can have only upper body parts visible, or even a single side (left/right) of the body parts visible (e.g. K_2 and K_3), depending on what is most representative of the pose we want to capture. To speed up the pose search algorithm, we restrict the part object detectors to search in a bounded location and orientation range, determined by the KTPs, similar to the position priors described in [4].

3 Activity Recognition

The pose detectors described above are applied to all the frames in the video. The detector corresponding to the k^{th} kinematic tree prior returns a set of part-wise posterior marginal distributions $E_k = \{E_{k,i}\}_{i=1\cdots N}$ for each frame, where $E_{k,i} = p(l_i = (x, y, \theta)\,; k)$. The pose detectors return confident estimates for the frames containing poses similar to its corresponding mean-pose configuration, and will likely have higher uncertainty for other configurations. The confident estimates can be identified visually, however there exists no obvious algorithm to identify them automatically. A part-wise entropy based scoring may be employed, however we could not determine a set of consistent entropy based thresholds which would select a single confident result across all the detectors. Hence, the activity recognition problem remains quite challenging. We propose computing descriptors from the distributions and combine them in a MKL framework for activity classification.

3.1 Pose Descriptors

A standard approach to summarizing the distributions is to use the maximum a posteriori (MAP) estimate but this ignores the variances of the distribution. We avoid making a hard threshold regarding the final pose; instead, pose descriptor histograms are extracted from the marginal posterior distributions of the parts, as introduced by Ferrari et al [4]. We use only two of the three pose descriptors proposed in [4], namely Descriptor A and Descriptor B (described below). Descriptor C was found to be redundant for classification purposes.

[Descriptor A] The marginal distribution $E_{k,i}$ is quantized to $20 \times 16 \times 24$ bins in the x, y and θ dimensions. The descriptor captures the global distribution of the parts in the detection window in each orientation.

[Descriptor B] The descriptor encodes the part orientations, relative locations and relative orientations in a single concatenated histogram. The following three distributions are computed from $E_{k,i}$:

$$P\left(l_i^{\theta}\right) = \sum_{(x,y)} P\left(l_i = (x, y, \theta)\right) \tag{3}$$

$$P\left(r\left(l_i, l_j\right) = \rho\right) = \sum_{(\theta_i,\theta_j)} P(l_i^{\theta})P(l_j^{\theta})\mathbf{1}_{(r(\theta_i,\theta_j)=\rho)}$$

$$P\left(l_i^{xy} - l_j^{xy} = \delta\right) = \sum_{(x_i,y_i,x_j,y_j)} P(l_i^{xy})P(l_j^{xy})\mathbf{1}_{l_i^{xy}-l_j^{xy}=\delta}$$

where the marginal orientation of each part is given by $P\left(l_i^{\theta}\right)$ and is quantized into 24 bins. The relative orientation marginals between pairs of parts is given by $P\left(r\left(l_i, l_j\right)\right)$ and is quantized into 24 bins. The relative location marginals is given by $P\left(l_i^{xy} - l_j^{xy}\right)$ and is quantized into 7×9 bins. The final descriptor is a concatenation of all the above three histograms computed for each part.

3.2 Multiple Kernel Learning

A vocabulary of codewords is learned for each type of descriptor (A and B) corresponding to each of the kinematic tree priors $K_i \in \mathcal{K}$ by K-Means clustering;

the cluster centers constitute the codewords of the vocabulary. The learned vocabulary is used to compute a set of histogram of codeword features $\boldsymbol{h} \in \mathcal{H}$ for each of the input videos, where each histogram $\boldsymbol{h}$ corresponds to a particular descriptor type and KTP. Hence we have $|\mathcal{H}| = 2 \times |\mathcal{K}|$ number of histogram of codewords for each video. Similarity kernel matrices $\boldsymbol{K}_c$ are computed for each type of histogram using the χ^2 distance function:

$$\boldsymbol{K}_c\left(i,j\right) = exp\left(-\frac{1}{A_c}\chi^2\left(\boldsymbol{h}_i,\boldsymbol{h}_j\right)\right) \tag{4}$$

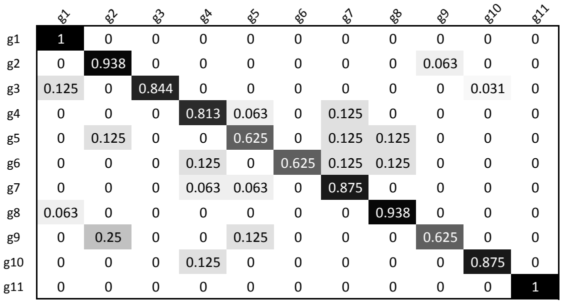
$$\chi^2\left(\boldsymbol{h}_i,\boldsymbol{h}_j\right) = \frac{1}{2}\sum_{b=1}^{D}\left(\frac{\left(\boldsymbol{h}_i(b)-\boldsymbol{h}_j(b)\right)^2}{\boldsymbol{h}_i(b)+\boldsymbol{h}_j(b)}\right)$$

A_c is the kernel scaling parameter and is set to the mean distance value. Multiple kernel learning (MKL) is used to determine the most discriminative combinations of kernels in a max-margin framework. MKL has been successfully used for combining feature channels in computer vision [6]. MKL determines the weights w_c, such that the combined kernel $\boldsymbol{K} = \sum_c^{|2\times\mathcal{K}|} w_c * \boldsymbol{K}_c$ is the best conical combination of the individual kernels for recognizing a given action. Final classification is performed by learning an SVM classifier for each class using the combined kernel $\boldsymbol{K}$. We use a publicly available implementation of the MKL algorithm proposed by Bach et al [2].

4 Results and Conclusion

We tested our framework on the USC-Gesture dataset [9], containing multiple video clips of 11 actions performed by 8 actors. There is very little variation in the clips belonging to the same actor and action, hence we choose only two clips per actor per action, resulting in a total of $2 \times 11 \times 8 = 176$ video clips in our dataset. We use videos from 7 actors as training data, and test on the 8th actor, and repeat for all possible permutations of actors. Pose detectors are applied with a collection of 8 kinematic prior trees shown in Figure 1. The marginal posterior distribution of parts returned by a subset of the pose detectors are shown in Figure 2, where each row corresponds to a different KTP. We observe that at least one of the pose detectors returns a confident pose estimate across all the frames.

We used $K = 40$ for constructing the descriptor codewords. Final classification results using the MKL algorithm are illustrated by a confusion matrix shown in Figure 3. We achieve an average accuracy rate of 83.33% across all folds. Note that [12] report a 92% accuracy rate, however that method requires manual construction of activity models by annotating 2.5D joint locations for selected key poses; the models also contain motion styles and durations. The method presented here does not require any manual modeling effort (assuming that the set of pre-defined KTPs is sufficient) and should be insensitive to styles of motion as dynamics is not modeled specifically. This is not to argue that motion dynamics is not useful, or even critical, for activity recognition but that some analysis that is not dependent on precise dynamical models may be useful prior to the application of the dynamical models.



	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11
g1	1	0	0	0	0	0	0	0	0	0	0
g2	0	0.938	0	0	0	0	0	0	0.063	0	0
g3	0.125	0	0.844	0	0	0	0	0	0	0.031	0
g4	0	0	0	0.813	0.063	0	0.125	0	0	0	0
g5	0	0.125	0	0	0.625	0	0.125	0.125	0	0	0
g6	0	0	0	0.125	0	0.625	0.125	0.125	0	0	0
g7	0	0	0	0.063	0.063	0	0.875	0	0	0	0
g8	0.063	0	0	0	0	0	0	0.938	0	0	0
g9	0	0.25	0	0	0.125	0	0	0	0.625	0	0
g10	0	0	0	0.125	0	0	0	0	0	0.875	0
g11	0	0	0	0	0	0	0	0	0	0	1

Figure 3: Confusion Matrix for USC-Gesture dataset.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009.

[2] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Fast Kernel Learning using Sequential Minimal Optimization. Technical report, UC-Berkeley, 2004.

[3] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr. A Study of Parts-Based Object Class Detection Using Complete Graphs. *IJCV*, 2009.

[4] V. Ferrari, M. Jimenez, and A. Zisserman. Pose search: retrieving people using their pose. In *CVPR*, 2009.

[5] N. Ikizler and D. Forsyth. Searching Video for Complex Activities with Finite State Models. In *CVPR*, 2007.

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

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

[8] G. Mori, X. Ren, and A. Efros. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, 2004.

[9] P. Natarajan, V. Singh, and R. Nevatia. Learning 3D Action Models from a few 2D videos. In *CVPR*, 2010.

[10] D. Ramanan. Learning to parse images of articulated bodies. *NIPS*, 2007.

[11] D. Ramanan, D. Forsyth, and A. Zisserman. Strike a Pose: Tracking People by Finding Stylized Poses. In *CVPR*, 2005.

[12] V. K. Singh and R. Nevatia. Action recognition in cluttered dynamic scenes using Pose-Specific Part Models. *ICCV*, 2011.

[13] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *CVPR*, 2011.