A Framework for Indexing Human Actions in Video

Kaustubh Kulkarni, Srikanth Cherla, Amit Kale, V. Ramasubramanian

A Framework for Indexing Human Actions in Video

Kaustubh Kulkarni Srikanth Cherla Amit Kale V. Ramasubramanian

Siemens Corporate Technology,
SISL - Bangalore, India
{kulkarni.kaustubh, srikanth.cherla, kale.amit,
v.ramasubramanian}@siemens.com

Abstract. Several researchers have addressed the problem of human action recognition using a variety of algorithms. An underlying assumption in most of these algorithms is that action boundaries are already known in a test video sequence. In this paper, we propose a fast method for continuous human action recognition in a video sequence. We propose the use of a low dimensional feature vector which consists of (a) the projections of the width profile of the actor on to a Discrete Cosine Transform (DCT) basis and (b) simple spatio-temporal features. We use an earlier proposed average-template with multiple features for modelling human actions and combine it with One-pass Dynamic Programing (DP) algorithm for continuous action recognition. This model accounts for intra-class variability in the way an action is performed. Furthermore, we demonstrate a way to perform noise robust recognition by creating a noise match condition between the train and the test data. The effectiveness of our method is demonstrated by conducting experiments on the IXMAS dataset of persons performing various actions and an outdoor Action database collected by us.

1 Introduction

With vast amounts of visual data generated in television broadcast, surveillance camera networks, as well as more recent online repositories such as YouTube, the problem of organizing and tagging videos based on content has assumed great importance. Of particular interest is the subset of these videos which contain human actions. Several challenges arise when recognizing human actions from video. Firstly, there is the issue of action recognition independent of viewing direction. Secondly, there are intra-class variations in the way an action is performed which must be accounted for by the system. Thirdly, the amount of training data is usually limited which precludes usage of sophisticated statistical methods such as Hidden Markov Models.

Several researchers have addressed the problem of human action recognition. In interests of brevity, we cite only relevant references here. Weinland et. al. [1] address view invariant action recognition using a 3D occupancy grid as a feature to learn a set of exemplars and a HMM. For recognition, these 3D exemplars

are used to produce 2D image information for matching with the observations. The work of Lv and Nevatia [2] is similar in spirit to this. One of the difficulties with these methods is that synthesizing 2D projections from 3D exemplars and comparing them with the observed 2D image is computationally intensive. Earlier approaches similar to the one described in this paper include Veeraraghavan et al. [3] who propose the use of shape features and a Procrustes distance between human silhouettes as a distance metric to distinguish between actions. They propose the use of constraints on the warping region around the nominal activity trajectory in order to model intra-class variations. More recently, Cherla et al. [4] proposed the use of a low-dimensional feature set comprising of projections of silhouette width on an *action basis* and simple spatio-temporal features. Recognition was performed using DTW classifier that made use of the average template with multiple features (ATMF) model to encompass intra-class variabilities in actions. An important consideration in learning human actions is to define analogues of phonemes for human actions to divide a complex human action into atomic actions. Vitaladevuni et al. [5] introduced the notion of ballistic verbs to this end.

An important shortcoming of the above methods is that they do not explicitly deal with the problem of continuous action recognition and can only be used if the boundaries between human actions in a video sequence are known. What makes the problem of continuous action recognition difficult is that, unlike speech, where word boundaries are delimited by pauses, simple abrupt changes do not always exist between two actions in the case of video. Turaga et al. [6] propose a method for unsupervised segmentation of videos by exploiting the conformance of the interframe optical flow to a linear dynamical system model between action boundaries. Such an approach will have difficulties identifying boundaries between two actions, the transition between which, is smooth. Furthermore since the approach is unsupervised, the system can come up with its own action definition which could lack semantic significance.

Considering the above facts, we outline here what we feel are the important building blocks for a continuous action recognition system. In order to come up with a semantically meaningful parsing of the human action video, it is necessary to generate an action vocabulary. The second important issue in building a continuous action recognition system is a clear definition of delimiters between actions. While silence or pauses are obvious choices for delimiters between spoken words when performing continuous speech recognition, the choice in the case of continuous action recognition is not so obvious. Finally, given this vocabulary and delimiters, the system must be able to parse a video into an action sequence.

In this paper, we develop the building blocks of a continuous action recognition system. The first step is to perform reliable isolated recognition of actions in the vocabulary. An important step in this regard is choosing features which are discriminative between actions and low-dimensional. We choose the width of the outer contour of the person's silhouette as the basis for generating our feature. Our feature comprises of the first 13 DCT components of the width vector along with simple spatio-temporal features. For temporal matching, we

use a non-parametric, generative ATMF model as proposed by [4]. This model models intra-class variability well and performs better in the presence of limited training data. In our choice of dataset, since stand is the base pose that is assumed by a subject while performing most of the actions in our vocabulary, stand was chosen as the delimiter. The stand action can have large variations in length as well as appearance. To deal with this problem we model it by a single frame model arising from a multimodal distribution. To perform continuous action recognition we combine the ATMF model with the one-pass DP decoding algorithm [7]. Furthermore, we propose a simple method to do noise robust recognition by adding synthetic noise to the train data. We demonstrate the ability of our approach to do reliable continuous action recognition on IXMAS dataset which is publicly available on INRIA's Perception Laboratory webpage[1] and also on an outdoor action database captured by us.

2 Features

Given the video of an action, background subtraction [8] can be applied to obtain the binarized silhouettes of the person. The largest blob in the binary image is assumed to be the subject. The binarized silhouette of the person provides a reasonable starting point for performing action recognition and has been used in [9–11]. While approaches which do not rely on background subtraction are also prevalent [5, 12] usually they end up being more computationally intensive since they rely on computing optical flow. We propose to use only the outer contour of the binarized silhouette, specifically the width of the outer contour, as we believe that it contains adequate information for recognizing actions. As we shall demonstrate, this feature captures both structural and dynamic information for an action. Width features for gait representation have been used by [10, 11, 13].

2.1 Width Features

Width Extraction The first step in generating the width vector corresponding to the binary silhouette is to place a bounding box around the silhouette. The size of the bounding box is set dynamically based on the size of the silhouette. However, there is little uniformity in the size of the silhouette as different people have different heights and also, its size varies with the person's distance from the camera. It is necessary that our width vector is of uniform size for later calculations. In order to normalize the size of the width vector, we uniformly scale the size of the bounding box so that it has a fixed height of hundred pixels, keeping the aspect ratio constant. The width along a given row is simply the difference in the locations of the right-most and left-most silhouette boundary pixels in that row. The advantage in using the width profile of a person as a feature is that it encompasses structural information peculiar to each action well. Also, use of the width feature provides uniformity to feature representation across different individuals. We denote the hundred dimensional width vector of

[1] https://charibdis.inrialpes.fr/html/sequences.php

a given action at time t as $v(t) \in \mathbf{R}^{100}$. The width vector can be computed as a special case of $\mathcal{R}$-transform in [13]. The $\mathcal{R}$-transform is given by

$$g(\rho, \theta) = \sum_x \sum_y I(x, y)\delta((x \sin\theta + y \cos\theta) - \rho) \tag{1}$$

In our case, the $\mathcal{R}$-transform computed with $\theta = 0, \rho = 0, 0 < y < 100$.

Dimensionality Reduction While $v(t)$ theoretically can span all of $\mathbf{R}^{100}$, [4] showed that a five dimensional *action basis* suffices to capture variations in $v(t)$ corresponding to a person's limb movements. The walk action, which is the most dynamic of all with respect to movements of hands and legs, was chosen as the action to generate the 5D *action basis*. This basis was obtained by taking the top five eigenvectors of Principal Component analysis of the width vectors of walk sequences of different people. However, the following are drawbacks of using the action basis:

1. The *action basis* is built using only the walk sequences of people. However, action specific basis functions are the more natural choice for certain actions. For example, actions such as picking up or sitting down are very different from walking. For such actions, the projections on the action basis can be low and therefore more susceptible to noise.
2. While projecting the width vector on the *action basis*, the mean of the vector is subtracted. The mean contains static information related to the person's posture. We experimentally demonstrate that including this static information increases recognition accuracy.

In this paper we propose the use of the Discrete Cosine Transform (DCT) to perform dimensionality reduction on the width feature. The main advantage of using the DCT for dimensionality reduction over the action basis is that the former is data independent, unlike the latter. The first coefficient of the DCT vector would contain the static information which is lost while computing the PCA coefficients from the action basis. We demonstrate in Section 5 that our low dimensional model to represent human appearance substantially increases recognition scores when compared with simple optical flow features used in [6].

The one dimensional DCT for the width vector v_i for the i^{th} person is given by the equation:

$$u_i(k) = \alpha(k) \sum_{n=0}^{N-1} v_i(n) \cos[\frac{\pi(2n+1)k}{2N}],$$
$$\text{where } 0 \leq k \leq N-1, \alpha(0) = \sqrt{\frac{1}{N}}, \alpha(k) = \sqrt{\frac{2}{N}} \; for \; 1 \leq k \leq N-1 \tag{2}$$

In (2) above, the index n refers to the row index of v_i. We found in our experiments that using the first 13 DCT coefficients sufficed to provide good recognition accuracy.

2.2 Spatio-temporal Features

In addition to the DCT of the width, we also include simple spatio-temporal features as described in [4]: displacements of the centroid c_x and c_y of the silhouette and the standard deviations σ_x and σ_y in both the X and Y directions respectively. These features give us information about posture and global motion of the human silhouette e.g. significant centroid motion in one direction can suggest that the person is walking, while a significant change in the value of standard deviation could indicate whether a person is sitting or standing. We augment our initial 13 dimensional DCT features by adding the four spatio-temporal features to obtain a seventeen dimensional feature vector

$$x = [v_1 \dots v_{13}\ c_x\ c_y\ \sigma_x\ \sigma_y]^T \tag{3}$$

3 Average-templates with multiple features

Given the features, the next consideration is the choice of the temporal matching method. Since the training data is limited, we opt for a template based representation for each activity. The computational complexity of the recognition process is highly dependent on these template representations.

3.1 Computing average-templates

One way to reduce the complexity of recognition is to compute an average or nominal template [14] over all the instances of that action. The average pattern or average-template R is computed by mapping available training instances $T = \{T_1, T_2, \dots, T_n, \dots\}$ using DTW. We use Euclidean distance as the local distance $d(i, j)$ between the frame i of R and frame j of T. If I is the length of R and J is the length of T, the path is forced to begin at the point $D(1, 1)$ and end at $D(I, J)$. The accumulated distance $D(i, j)$for the DTW is defined as:

$$\min[D(i-2, j-1)+3d(i, j), D(i-1, j-1)+2d(i, j), D(i-1, j-2)+3d(i, j)] \tag{4}$$

where i is the frame index of the average reference pattern R and j is the frame index of the train pattern T.

Backtracking from the point $D(I, J)$ yields the optimal path $p = [i_k, j_k]$ and the corresponding mapped set of feature vectors $[R(i_k), T(j_k)]$. Here k, is the index of a point on the optimal path p. The average reference pattern R_n for an activity is computed by the successive weighted averaging of n instances as follows:

$$R_n(k) = (1 - \frac{1}{n})R_{n-1}(i_k) + (\frac{1}{n})T_n(j_k), k = 1 \dots K \tag{5}$$

where K is the number of points on the optimal path p and $R_{n-1}(i_k)$ is the average of the previous $n - 1$ templates. The new time axis for the instance R_n is computed as:

$$p_1(k) = (1 - \frac{1}{n})i_k + (\frac{1}{n})j_k, k = 1 \dots K \tag{6}$$

We linearly transform this new time axis to a constant length P where P is the average length of all instances of an activity. The transformation is done as follows:

$$p_2(k) = \frac{P}{K} p_1(k) \tag{7}$$

as $p_2(k)$ would have non-integer values we define a time axis $p_3(k')$ where $k' = 1, 2, 3 \ldots P$. The feature values of the average pattern $R_n(k)$ are interpolated to get the new average pattern $R_n(k')$.

3.2 Combining average templates with multiple features

While the average activity template is clearly the best choice in terms of computational complexity, it is unable to account for intra-class variability in the way an action is performed. Therefore, we use the ATMF model proposed in [4]. In this method, the average template is first computed as discussed in Section 3.1. Using this information, all frames corresponding to each frame of the average-template are binned together. Following this, the local DTW distance between a frame of test data and a frame of the average-template is computed as the minimum of the distances between the respective test frame and the multiple feature vectors in the bin corresponding to that frame of the average. In this way, all the variations seen in a particular action class can be combined. To summarize the above steps, every k' in an average pattern for a category $R_n(k')$, where $k' = 1, 2, 3 \ldots P$, is associated with a bin of frames of size M.

3.3 Building a delimiter model

Since the goal of this work is continuous action recognition, an analogy to continuous speech recognition is instructive. Continuous speech recognition entails recognizing sequences of words spoken in succession, delimited by pauses or silences. When we consider a sequence of human actions, the direct analog of such delimiters do not exist. In the IXMAS dataset, we observe that stand is the base pose that is assumed by a subject while performing most of the actions in our vocabulary. This makes it a good candidate for being a delimiter. One of the problems in using stand action as the delimiter is as regards the large variability in its prevalence. For example, it is plausible that if a person is just waiting for something, the length of the stand action is a lot longer than those stand actions that truly represent the pauses between actions. Due to this large variability in length, it is unrealistic to warp these disparate instances of the stand action to one single average entity. To circumvent this problem, we propose to represent the stand action by a single frame. This choice can be justified by noting that stand is not an articulated action and each frame of stand can be treated independently. The second problem with using stand as delimiter is the large variability in its appearance across people. To deal with this we model the single frame delimiter as arising from a multimodal distribution which can be learned using kernel methods. For the sake of simplicity, we simply performed a $k-$means clustering over the stand poses and built a codebook of size M. In this

paper we choose $M = 8$. While standing was the natural choice for the IXMAS dataset, in general any pose which can persist for a long time such as sitting or sleeping can be easily added to the single frame delimiter model.

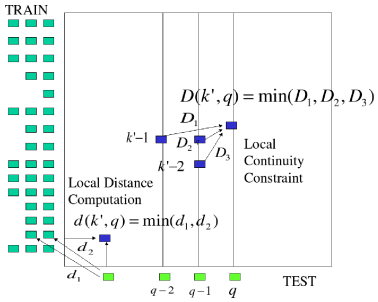

Fig. 1. The Local Continuity constraint and the local distance filling for DTW for Isolated Action Recognition

4 Continuous Activity Recognition using One-pass-DP Decoding

In continuous activity recognition, we have a test sequence $\mathbf{O}$ that is assumed to consist of certain number of different actions performed in succession, each action drawn from a vocabulary of N_w actions $W = \{w_1, w_2, \ldots, w_{Nw}\}$. Continuous recognition of human actions is a difficult task to do online, primarily because this involves the problem of jointly determining the optimal number of actions K^* in the test sequence $\mathbf{O}$, their boundaries $B^* = \{b_0^*, b_1^*, b_{k-1}^*, b_k^*, \ldots, b_{K^*}^*\}$ and associated optimal action indices $I^* = \{i_1^*, i_2^*, \ldots, i_k^*, \ldots, i_{K^*}^*\}$ (where $w_{i_k^*} \in W$), by minimizing a measure of distance $D(\mathbf{O}, \mathbf{R})$ between the test sequence $\mathbf{O}$ and a typical reference action template sequence $\mathbf{R} = \{w_{i_1}, w_{i_2}, \ldots, w_{i_k}, \ldots, w_{i_K}\}$ each drawn from W. This is typically done in the form of 'connected action decoding' based on the one-pass DP, well known in speech recognition [7]. The decoding problem of determining (K^*, B^*, I^*) is solved by minimizing $D(\mathbf{O}, \mathbf{R})$ over the variables (K, B, I) using the time-synchronous one-pass DP decoding algorithm.

To compute the optimal cumulative distance, we use two types of transition rules (a) for action interior i.e. Within Action Recursion (b) for action boundary i.e Cross-Action Recursion. These recursions are computed for all frames of the test video sequence w.r.t the all frames of all ATMF action models in a left to right time synchronous manner. These recursions would then result in many possible paths. The optimal action sequence or path will be the one which corresponds to the minimum cumulative distance (Termination and Backtracking). A diagrammatic representation of all steps in the algorithm is given in Fig. 2.

We now provide the mathematical details pertaining to the above intuitive explanation of the algorithm. The action vocabulary of size N_w is given by $W = \{w_1, w_2, \ldots, w_{Nw}\}$. Each action corresponds to a reference pattern $R_w(k')$, where $k' = 1, 2, 3 \ldots P_w$; P_w is the number of frames in the w^{th} action and each k' has a codebook of size M. The test frame index is given by q and Q is the

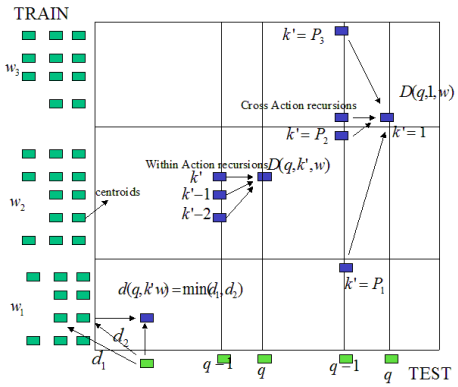

Fig. 2. Illustration of Onepass-DP transitions for Continuous action Recognition

length of the test data. During the recognition pass the sequence of warping is given by the average-templates. The multiple features contribute to the local distance in the following way:

$$d(q, k', w) = \min_{m=1\ldots M}(d(q, m, k', w)) \tag{8}$$

where $d(q, m, k', w)$ is the local distance between test frame q and the m^{th} centroid of the k' frame of the average-templates corresponding to the action w.

This method combines all possible combinations of features which were observed during the training phase. Let D denote the global accumulated distance between the test frame and the reference pattern frame. The one-pass DP decoding would look to minimize the global accumulated distance over all the frames of the test pattern. The following steps give a method to accumulate the global distance between a given test frame and a frame of the reference pattern to find a globally optimal path:

1. Within action recursion: This recursion is computed for all frames Q of the test pattern and all frames k' of all reference patterns except for $k' = 1$ i.e. the recursions are applied to to all frames except at the action beginning. This recursion can be denoted as:

$$D(q, k', w) = d(q, k', w) + \min_{k'-2 \le r \le k'}(D(q-1, r, w)) \tag{9}$$

2. Cross-action Recursions: This recursion is computed for all Q test frames and for $k' = 1$ frames of all reference patterns. This recursion allows a transition into the first frame of a given reference pattern from the last frame of all other reference pattern including the given reference pattern or it allows the path to be in the last frame of that given reference pattern i.e. the algorithm either stays in the particular action or transits into the first frame on any other action depending on which of the two paths yields a minimum score. It can be denoted as:

$$D(q, 1, w) = d(q, 1, w) + \min[\min_{1 \le w \le N_w}[D(q-1, P_w, w)], D(q-1, 1, w)] \tag{10}$$

3. Termination and Backtracking: To find the best action sequence the algorithm uses the following termination condition at the test frame Q:

$$D^* = \min_{1 \leq w \leq N_w} [D(Q, P_w, w)] \tag{11}$$

The algorithm checks for the minimum accumulated distance for the best path at the last frame of every reference pattern at the test frame Q. The best path is backtracked from that point through back-pointers stored during the Within Action and Cross Action recursions.

It is important to note that the 'stand' model is only one frame long, therefore only cross action recursion is possible which allows you stay in that one frame model transit into any other action including the stand model itself. This is equivalent to a single state HMM(Hidden Markov Models) which can stay in the same state, exit that state or transit back into the same state. Also, if we switch of the cross action transition the algorithm automatically reduces to an isolated action recognition.

Robustness to noise: The accuracy of recognition depends on the the noise resilience of the feature vector. Since our features are derived from the outer contour of the silhouette extracted using background subtraction, it is important to have some measures to deal with noise. In this paper, we propose a simple approach to deal with noise: specifically if we expect noise corruption by $n(t) \sim p(.)$ we simulate another noise process $\tilde{n}(t) \sim p(.)$ which we add to our noise free data. Following [15], consider the training and testing sets when no noise is added to the training data. Let $r(t)$ denote the training sequence and let $s(t)$ denote the test case. Let us first consider the case when the training and testing sequences are identical $x_1(t)$ Then

$$r(t) = x_1(t), \qquad s(t) = x_1(t) + n(t) \tag{12}$$

Transforming each of them to the frequency domain and using Parsevals identity we can write

$$|R(f)|^2 = |X_1(f)|^2, \qquad |S(f)|^2 = |X_1(f)|^2 + |N(f)|^2 \tag{13}$$

If we add synthetic noise with the same pdf to the training data viz $\tilde{r}(t) = x_1(t) + \tilde{n}(t)$ it is easy to see that $\frac{|\tilde{R}(f)|^2}{|S(f)|^2} \rightarrow 1$, thereby improving the recognition. When the training and testing actions are not identical, it is not clear whether addition of noise increases discriminability. However in our experiments we found that addition of synthetic noise matching with the test set increased recognition rates. Furthermore, also note that the spatio-temporal features are already noise robust because they are computed for the largest blob in the given frame.

5 Experimental Results

We present experimental results on the IXMAS dataset as well as an outdoor action dataset collected by us. The multi-view data in the IXMAS dataset was

obtained with the help of five synchronized cameras placed at different positions around the region where several actors performed the set of actions to be identified. We observed that, though the cameras might have been static, not always did they have the same view of all the persons performing the actions. That is, if a camera, say camera 1, viewed a person from a certain angle, it is not always the case that the same camera had the same view of another person. In order for our algorithm to perform recognition, view consistency was necessary while training. We re-organized the camera views in the dataset into 6 categories, depending on the direction in which the person performing the actions faces as given by the Figure 3:

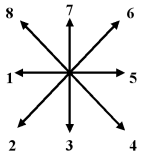

Fig. 3. Directions corresponding the the views 1-8 in the experiments

For performing experiments we use view 1, which contains a total number of 108 templates, as the train set and view 5, which contains a total of 241 actions, as the test set. The test set is obtained from 19 video sequences.

5.1 Importance of Different Feature Components in Isolated Action Recognition

In order to assess the relative importance of different feature components we used the average template with view 1 as train set and view 5 as test set for isolated action recognition with various combinations of the feature components. The result with 12 DCT feature excluding the first energy term is 67.22%. Adding the first energy term to the feature increases the recognition to 71.78% which shows that the static information contained in the posture helps in recognition. The result by adding the spatio-temporal terms to the feature increases the recognition accuracy to 76.76%. For actions such as sitting down or getting up, as we might expect, the spatio-temporal terms provide reliable recognition even in the face of significant pose changes and noise. We also experimented with using only the global motion feature, similar to [6]. In this case, the recognition accuracy falls to 24.07%. This demonstrates that pure global motion features do not suffice as a rich descriptor of human actions and how our lower dimensional DCT feature helps to capture nuances of hard-to-recognize actions.

5.2 Continuous Recognition Results

The isolated recognition (Figure: 1) task assumes that the boundaries of the actions in the test data are known. Under this assumption only one kind of error i.e. a substitution error is possible. The action label selected is the one that

gives a minimum DTW score. In continuous action recognition there are two more errors that are introduced - (a) Deletions i.e. when the action is classified as a delimiter and (b) Insertions i.e. when an action is broken into multiple fragments. Our continuous recognition labels each frame of the test sequence as one of the actions in the vocabulary. The top row of Figure 4 shows the ground truth of the action sequence while the bottom row shows the labels identified by our algorithm. The white color in the picture corresponds to the delimiter action (stand). What we observed was that, for correctly recognized actions the labels generated are very close to the ground truth.

Fig. 4. Ground truth of action sequence (top row) Action sequence recognized by our algorithm (bottom row). As we can see our algorithm provides reasonable labeling of the action sequence.

To compute the accuracy of the labelling we use a text matching algorithm based on Levenshtein distances [16]. This algorithm matches the generated action string with the the original ground truth labeling to compute the number of insertions, deletions and substitutions. The delimiters are not included in the accuracy computation. This algorithm is commonly used in spell checkers. Based on the string matching we compute the accuracy of action recognition as:

$$\text{Accuracy} = 100 - \frac{(\text{substitutions} + \text{insertions} + \text{deletions})}{\text{total number of actions}} \times 100 \qquad (14)$$

Based on this we compute the recognition accuracy for continuous action recognition given in the Table 1.

Table 1. *Accuracy Results for Isolated and Continuous Activity Recognition*

Type of Recognition	Type of Template	Accuracy [%]	subs [%]	ins [%]	dels [%]
Isolated	Average	76.76	23.24	0.0	0.0
Isolated	ATMF	82.16	17.84	0.0	0.0
Continuous	ATMF	68.05	16.18	5.81	9.96

As we can see from the figure, the as we switch the model from a single average template to ATMF for an action class there is considerable increase in the recognition accuracy as the ATMF model accounts for the intra-class variability of the action class.

Outdoor Action Database To verify our feature building and the ATMF model and the Onepass-DP algorithm we also ran experiments on an outdoor database we built. In this database, we asked 10 actors to perform the action vocabulary of the IXMAS database in a random order. All actions were recorded from a side view. The foreground silhouettes corresponding to the actors were extracted using [8]. We then used actions from 5 actors as a train set and the

remaining 5 as test set. The train set consists of 66 templates for all actions. The test set consists of 65 actions. We performed both isolated and continuous action recognition on the collected data. The results are described in Table 2.

Fig. 5. An example action from our Outdoor Action Database with the corresponding background subtracted silhouettes.

Table 2. *Accuracy Results for Isolated and Continuous Activity Recognition on Outdoor Action Database*

Type of Recognition	Type of Template	Accuracy [%]	subs [%]	ins [%]	dels [%]
Isolated	ATMF	81.54	18.46	0.0	0.0
Continuous	ATMF	72.30	23.08	0.0	4.62

Results on Noisy Data: We perform the experiments on the IXMAS dataset by adding salt and pepper noise of variance 0.04 (Using MATLAB imnoise function)to the test set. We filter the width vector generated from the noisy data by using a traditional 5-dimensional mean and median filter. We also add the same salt and pepper noise to the training set. We show that the noise matching method works the best and gives results comparable to the clean data results. The recognition accuracies achieved are tabulated in Table 3. The clean and the noise template are represented in Figure 6

Table 3. *Accuracy Results for Isolated and Continuous Activity Recognition on Noisy Data.*

Type of Recognition	Type of Template	Clean [%]	Noise [%]	Mean Filter [%]	Median Filter [%]	Noise Match [%]
Isolated	Average	76.76	53.94	51.45	55.19	73.86
Isolated	ATMF	82.16	57.26	58.09	55.19	78.84
Continuous	ATMF	68.05	42.0	38.17	41.91	65.98

5.3 Comparison with MHIs, HMMs and Action Basis

We also conducted a comparative study between our method and the method proposed by Bobick et al. [9]. We chose this method for comparison as it is well-known and considered seminal in the area of action recognition. This approach

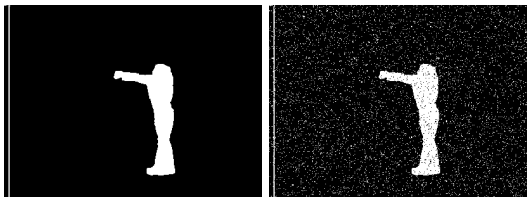

Fig. 6. Figure depicting the extent of noise corruption of the original test data during our noise robustness test.

uses Hu Moment Invariants of the Motion History Images (MHIs) of various activities as features for recognition. We performed the experiment with view 1 as train set and view 5 as test set with our implementation of this technique. For each action in the gallery we computed the mean of the Hu moments across people and used the Mahalanobis distance based vector quantization to classify each instance from the test set. We found that this system achieved a recognition rate of 33.20% as compared to 82.16% rate of our system. The recognition performance was slightly better than the one using action basis and ATMF where the number was 80.05% .

As noted before, our choice of template based representation was motivated by the limited training data available. Nevertheless we built a HMM model for isolated action recognition using HTK [17] (Version 3.4). Each action was modeled by a 5 state left to right HMM with a mixture of 3 Gaussians chosen for the observation probability within each state. Furthermore we chose the diagonal covariance matrices corresponding to the Gaussians. The recognition performance using HMMs was 42% for isolated action recognition. As noted before, since continuous recognition entails two additional types of errors viz. insertion and deletion, the performance of HMMs for continuous action recognition will be necessarily worse than that of one-pass DP ATMF. The same argument holds for the case of using simple global motion features used by [6].

6 Conclusion and Future Work

In this paper, we proposed a novel method for continuous human action recognition. The method uses a 17-dimensional feature vector which comprises of (a) DCT of the width of the outer contour of the human silhouette and (b) simple spatio temporal features. To deal with intra-class variability, we used the average template with multiple features representation in the DTW framework. We proposed a multimodal single frame model for delimiters in human action sequences. We combined the isolated action recognition with this delimiter model for continuous human action recognition in a one pass DP framework. We also introduced a method to improve recognition performance by adding synthetic noise to the training sequences. Future work consists of improving continuous action recognition by augmenting the present feature vector with additional features. Furthermore we also wish to study how continuous action recognition can be carried out when multiple cameras are available.

References

1. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3d exemplars. Proc. of IEEE Intl. Conf. on Computer Vision (2007)
2. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and viterbi path searching. Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (2007)
3. Veeraraghavan, A., Chellappa, R., Roy-Chowdhury, A.K.: The function space of an activity. Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (2006)
4. Cherla, S., Kulkarni, K., Kale, A., Ramasubramanian, V.: Towards fast, view-invariant human action recognition. Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (2008)
5. Vitaladevuni, S.N., Kellokumpu, V., Davis, L.S.: Action recognition using ballistic dynamics. Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (2008)
6. Turaga, P.K., Veeraraghavan, A., Chellappa, R.: From videos to verbs: Mining videos for events using a cascade of dynamical systems. Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (2007)
7. Ney, H.: The use of one-stage dynamic programming algorithm for connected word recognition. IEEE Trans. on Acoustic Speech and Signal Processing **32(2)** (1984) 263–270
8. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. Proc. of European Conf. on Computer Vision (2000)
9. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence (2001)
10. Kale, A., Cuntoor, N., Yegnanarayana, B., Rajagopalan, A.N., Chellappa, R.: Gait analysis for human identification. Proc. of Intl. Conf. on Audio and Video Based Person Authentication (2003)
11. Liu, Y., Collins, R.T., Tsin, Y.: Gait sequence analysis using frieze patterns. Proc. of European Conf. on Computer Vision (2002)
12. Natarajan, P., Nevatia, R.: View and scale invariant action recognition using multiview shape-flow models. Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (2008)
13. Souvenir, R., Babbs, J.: Learning the viewpoint manifold for action recognition. Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (2008)
14. Zelinski, R., Class, F.: A learning procedure for speaker-dependent word recognition systems based on sequential processing of input tokens. ICASSP (1983)
15. Morales, N., Gu, L., et. al.: Adding noise to improve noise robustness in speech recognition. INTERSPEECH (2007)
16. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10 (1966) 707–710
17. Young, S., et.al.: The htk book. Technical Report Cambridge University (2006)