Learning Maximum Margin Temporal Warping for Action Recognition

Jiang Wang, Ying Wu
Northwestern University
2145 Sheridan Rd, Evanston IL, 60201
wangjiangb@gmail.com, yingwu@eecs.northwestern.edu

Abstract

Temporal misalignment and duration variation in video actions largely influence the performance of action recognition, but it is very difficult to specify effective temporal alignment on action sequences. To address this challenge, this paper proposes a novel discriminative learning-based temporal alignment method, called maximum margin temporal warping (MMTW), to align two action sequences and measure their matching score. Based on the latent structure SVM formulation, the proposed MMTW method is able to learn a phantom action template to represent an action class for maximum discrimination against other classes. The recognition of this action class is based on the associated learned alignment of the input action. Extensive experiments on five benchmark datasets have demonstrated that this MMTW model is able to significantly promote the accuracy and robustness of action recognition under temporal misalignment and variations.

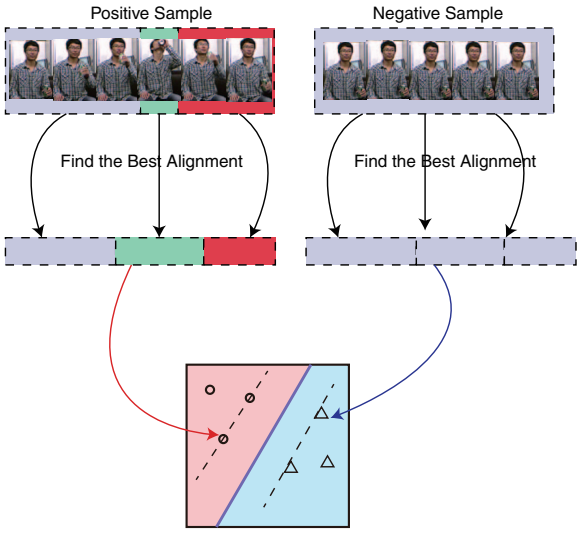

Figure 1. The video action are temporally aligned to a phantom action template. We learn a separating hyperplane such that the positive and negative examples are separated with the largest margin when the best alignment is applied.

1. Introduction

A fundamental yet challenging problem in human action recognition is to deal with its temporal variations. In addition to the compositional variance (i.e., the way of performing an action), the action can be performed at difference paces and thus spanning different time durations. Moreover, in practice, action video data may not be accurately localized along the time axis, and the starting and ending of an action are not provided. If used in training, such action videos can only be regarded as weakly labeled. If used as inputs for recognition, they bring extra work of action localization, explicitly or implicitly. Effective handling of such temporal variations is important to the performance of action recognition.

One approach to handle the the temporal structure is based on statistical generative models, such as HMM [12], dynamic Bayes nets [29], stochastic grammar [17] and CRFs [15]. These methods attempt to model the generative process of actions so as to perform action inference and learning. As they exploit the structural or compositional information in modeling, they may produce effective representations for action parsing and interpretation. However, learning the right structure can be very difficult.

Another approach is to perform explicit temporal alignment and localization. This facilitates discriminative models for action classification, whose training may be simpler than generative models. Dynamic time warping (DTW) has been used to align videos for recognition [32], time series classification [13] and action retrieval [14]. However, DTW's performance heavily depends on a good distance to measure the frames' similarity, especially when the dimension of the frame-level features are high. Generally such distances are heuristically defined and specified in advance. As a result, action alignment and classification are treated independently.

In this paper, we propose to learn action alignment so that we can unify action alignment and classification.

Specifically, the proposed method, called maximum margin temporal warping (MMTW), learns temporal action alignment for max margin classification. For each action class, an MMTW model is learned to achieve maximum margin separation from the rest action classes. This learned MMTW model can be treated as a *phantom action template* for representing this action class. The learning is formulated as a latent structural SVM, which can be efficiently solved with the cutting plane algorithm. Comparing with DTW, the proposed learning-based alignment leads to much better recognition performance. In addition, the inference of the proposed MMTW can be efficiently solved via dynamic programming, which makes the algorithm capable of processing very long videos. An illustration of the proposed method is shown in Fig. 1.

The contributions of this work include the following. First, the proposed maximum margin temporal warping (MMTW) is a novel approach to both action alignment and action recognition. It learns to align action videos and to model actions. Second, we find an innovative method to achieve computationally efficient action alignment and MMTW inference based on dynamic programming, which also enables effective learning. Third, we give a new formulation of latent structural SVM learning.

We evaluate the proposed approach on five benchmark datasets: MSR Sport Action3D dataset [11], MSR-DailyActivity3D dataset[26], Action Pair 3D dataset [16], Olympic Sports dataset [15], and UCF-sports dataset [18]. Because the action models are discriminatively learned, and the temporal deformation is explicitly modeled, the proposed approach achieves excellent results on action recognition tasks, as demonstrated by our extensive experiments on these five benchmark datasets.

2. Related Work

Actions usually exhibit complex temporal structures. Representing the temporal structure is crucial for successful action recognition. Spatio-temporal pyramids [19] divides the video into a pyramid of cells in the spatial and temporal dimensions, and represents the video as bag-of-words or max-pooling of the local features in each cell. This representation achieves good balance between the invariance to the spatio-temporal distortion and the discrimination to other classes. However, it only roughly characterizes the temporal structure of the actions. Fourier temporal pyramid [26] exploits the magnitudes of the low-frequency Fourier coefficients of the features as the representation. This representation is robust to temporal misalignment because it discards the phase information, but may fall short when the phase information may be important for action classification. The temporal structure of an action can also be modeled based on hidden Markov models [10, 11]. Learning a hidden Markov model for actions is challenging because the frame-level labels is not available in the training data. Other temporal structure models include temporal AND-OR graph [17], actom sequence model [5] and spatio-temporal graphs [2]. The proposed method is a novel learning approach to learning the temporal structure for action alignment and classification. This new method has exhibited good robustness to misalignment and good discriminativeness for classification.

Structural max-margin learning has recently been introduced to computer vision tasks to discriminatively learn the relationship between the structural variables. Recently, structural max-margin learning has been applied to action detection [21, 23]. These models represent the bounding box as a structured output, and employ structural output SVM for model learning. [6] employs the structural output SVM to learn a well shaped predictive function for early action detection. [13] models the temporal structure of the action with maximum margin temporal clustering. [22] and [15] use a latent graphical model to represent the temporal structure. These models typically requires careful initialization for the latent graphical model. The MMTW approach proposed in this paper is much simpler than the above mentioned graphical models, and it enables easier learning and results in better recognition accuracy.

3. Action Classification with Maximum Margin Temporal Warping

In this section, we propose maximum margin temporal warping (MMTW) approach to integrate action temporal alignment and action classification. The proposed MMTW approach is robust to the temporal deformation and misalignment in action recognition tasks, and has the discriminative power of the max margin methods.

A video action is represented as a sequence of frame-level features $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L)$, where $\boldsymbol{x}_i$ is the visual descriptor extracted at the i-th frame. The details of such features will be discussed in Sec. 6.1. We denote an action dataset by $\{(\boldsymbol{X}_1, y_1), (\boldsymbol{X}_2, y_2), \cdots, (\boldsymbol{X}_N, y_N)\}$, where $\boldsymbol{X}_i \in \mathcal{X}$ is a video action, and $y_i \in \mathcal{Y}$ is its action category labels. Action classification is to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$. Here we assume binary classification i.e., $Y = \{+1, -1\}$ for simplicity, because we can easily convert the multi-class classification problem to binary classification problem with one-vs -the-rest approach.

For each action class, we define a *phantom action template* $\boldsymbol{T}$ that consists of a sequence of *atomic actions*:

$$\boldsymbol{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_{L_T}\}, \tag{1}$$

where L_T is the length of the atomic action sequence. $\boldsymbol{t}_j$ denotes the frame-level features for the j-th *atomic action*. In addition, the expected length of an *atomic action* $\boldsymbol{t}_j$ is μ_j, but it can deform under warping. Its variation is captured by

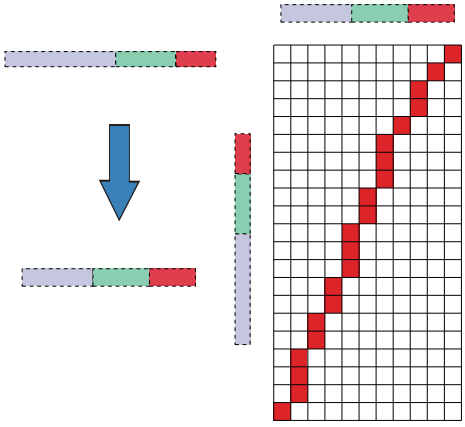

Figure 2. The warping alignment matrix. The long sequence above is aligned to the short sequence below red element (i, j) in the alignment matrix means the i-th element in the long sequence is aligned to the j-th element in the short sequence.

its deformation parameters a_j and d_j (details will be provided shortly). In the binary classification setting, the phantom action template is associated with the positive class. In the multi-class setting, each action class is associated with a phantom action template. The phantom template and its deformation parameters are learned from training data (details will be discussed in Sec. 5).

In order to deal with misalignment, we align an action $\boldsymbol{X}$ of length L to the *phantom action template* $\boldsymbol{T}$ with a warping function. The alignment can be represented by a $L \times L_T$ matrix, as shown in Fig. 2. Notice that the length of the input action L and the length of the phantom template L_T are not necessarily the same. A *warping path* P is a contiguous set of matrix elements that defines a mapping between $\boldsymbol{X}$ and $\boldsymbol{T}$. For example, an element $p = (i, j)$ means the i-th element in $\boldsymbol{X}$ is mapped to the j-th element in $\boldsymbol{T}$. We have the warping path:

$$P = p_1, p_2, \cdots, p_M. \tag{2}$$

where M is the length of the warping path. One constraint for the alignment is the boundary condition, i.e., $p_1 = (1, 1)$ and $p_M = (L, L_T)$. This is similar to dynamic temporal warping [14].

We define the cost function of aligning the action $\boldsymbol{X}$ to the *phantom action template* $\boldsymbol{T}$ under a warping path P as:

$$g(\boldsymbol{X}, P) = \frac{1}{L} \sum_{j=1}^{L_T} \boldsymbol{t}_j^T \sum_{i=b_j}^{e_j} \boldsymbol{x}_i + C(P) \tag{3}$$

where the $\{b_j, b_j + 1, \cdots e_j\}$ elements in $\boldsymbol{X}$ are aligned to the j-th element in $\boldsymbol{T}$, and $C(P)$ is the cost of length deformation of the *atomic actions* under the the warping P. We denote the number of the elements in $\boldsymbol{X}$ that are aligned to the j-th element in the *phantom action template* $\boldsymbol{T}$ by $l_j = e_j - b_j + 1$. The deformation cost $C(P)$ is defined as:

$$C(P) = \frac{1}{L_T} \sum_{j=1}^{L_T} \left(d_j (\frac{L_T}{L} l_j - \mu_j) + a_j (\frac{L_T}{L} l_j - \mu_j)^2 \right) \tag{4}$$

where μ_j is the expected length of the j-th *atomic action* in the *phantom action template* $\boldsymbol{T}$, and d_j, a_j model its length variation. This cost function can be regarded as a soft-version of the commonly used Sakoe-Chiba Band constraint [4].

The predictive mapping function is evaluated by finding the optimal warping path P that maximizes the cost function Eq. (3).

$$f(\boldsymbol{X}) = sign(\max_P g(\boldsymbol{X}, P)) \tag{5}$$

where $sign(x) = +1$ if $x > 0$ and -1 otherwise. The solution of $\max_P g(\boldsymbol{X}, P)$ will be given in Sec. 4. Then the binary classification of the action can be simply based on $f(\boldsymbol{X})$.

The proposed method has two advantages. First, it finds the optimal alignment of the input action to the *phantom action template* of a particular action class. Thus, it is robust to temporal misalignment. Second, since both the phantom templates and their deformation parameters are learnt from the training data, the proposed method is more discriminative and adaptive than the traditional dynamic temporal warping.

4. Inference: Action Alignment and Classification

In order to predict the class label of an input action $\boldsymbol{X}$, we perform the following steps. First, we compute the optimal warping path P to obtain $f(\boldsymbol{X})$, i.e., action alignment. Then, we determine the action class label of $\boldsymbol{X}$ by $f(\boldsymbol{X})$, i.e., action classification. As the second task is straightforward, here we focus on the first task.

We define a score function $S(i, j, l)$ that indicates the cost of warping the $\{1, 2, \cdots, i\}$-th elements of the input $\boldsymbol{X}$ to the $\{1, 2, \cdots, j\}$-th elements of the phantom action template $\boldsymbol{T}$, where l elements are aligned to the j-th element of the template.

This score function $S(i, j, l)$ can be computed recursively:

$$S(i,j,l) = \begin{cases} c(i,j) + \delta(j,1) & , l = 1 \text{ and } i, j = 1 \\ c(i,j) + max_l(S(i, j-1, l), & \\ S(i-1, j-1, l)) + \delta(j,1) & l = 1 \\ c(i,j) + S(i-1, j, l-1) + & \\ \delta(j,l) - \delta(j, l-1) & \text{otherwise} \end{cases} \tag{6}$$

where $c(i,j) = \frac{1}{L}\boldsymbol{t}_j^T\boldsymbol{x}_i$, $\delta(j,l)$ is the deformation cost of aligning the l elements to the j-th element in $\boldsymbol{T}$:

$$\delta(j,l) = d_j(\frac{L_T}{L}l - \mu_j) + a_j(\frac{L_T}{L}l - \mu_j)^2 \tag{7}$$

Then the maximum alignment score in Eq. (5) can be easily obtained by $f(\boldsymbol{X}) = max_l S(L_T, L, l)$ at the end of the recursion. In addition, the optimal warping path can be computed via back-tracking.

We can then compare the matching scores of all the action categories for action recognition.

5. Learning via Latent Structural SVM

Since the warping path P is not observable in the traning data, we formulate the learning problem as a latent structural SVM [33], with the warping path P as the latent variable.

Given two warping path P and P', we define the loss function $\Delta(P,P')$ as the loss of classifying P to P'. Suppose we have l_j and l'_j elements in the feature sequence $\boldsymbol{X}$ aligned to the j-th element of the action template sequence $\boldsymbol{T}$ in P and P', respectively. $\Delta(P,P')$ can be expressed as:

$$\Delta(P,P') = \frac{1}{L_T}\sum_{j=1}^{T}(l_j - l'_j)^2 \tag{8}$$

Denote by $\boldsymbol{w} = (\boldsymbol{t}_1, \cdots, \boldsymbol{t}_{L_T}, d_1, \cdots, d_{L_T}, a_1, \cdots, a_{L_T})$ the concatenation of all parameters to learn in Eq. (3) and (4). The training data are $\{(\boldsymbol{X}_1, y_1), \cdots, (\boldsymbol{X}_N, y_N)\}$, where $\boldsymbol{X}_i \in \mathcal{X}$ is the sequence of features, and $y_i \in \mathcal{Y}$ is the action category labels. The latent structural SVM can be formulated as

$$\begin{aligned}
&\min_{\boldsymbol{w},\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i=1}^{N}\xi_i \\
&s.t. \Delta(P,P^i) + g(\boldsymbol{X}_i,P) - g(\boldsymbol{X}_i,P^i) \le \xi_i, \forall P, \forall y_i = -1 \\
&\quad 1 - g(\boldsymbol{X}_i,P^i) \le \xi_i, \forall y_i = 1 \\
&\quad \xi_i > 0, \forall i
\end{aligned} \tag{9}$$

where P^i is the warping path for the i-th training data, P can be any feasible warping path. The optimization specifies that, for the negative training data, applying any warping path P to $\boldsymbol{X}_i$ should result in a score function $g(\boldsymbol{X}_i,P)$ that satisfies the margin constraint; and for the positive tracing data, applying the current warping path P^i should result in a score function that satisfies the margin constraint.

This optimization problem is challenging because it contains a huge number of constraints in Eq. (9), corresponding a lot of possible warping paths P. We can solve this optimization problem via the cutting plane algorithm [33]. The cutting plane algorithm solves an optimization problem with many constraints by iteratively solving the relaxed optimization problem with only a subset of the most violated active constraints. Since $\Delta(P,P')$ can also be decomposed according to each p_j of the warping path P, the most violated constraints can be found with the dynamic programming algorithm in Sec. 4.

In addition, since P^i are not observable in the training data, we iteratively solve the warping path P^i, the expected length μ_j^n, and the parameters $\boldsymbol{w}$ in our optimization. The optimal warping path P^i is solved via the dynamic programming algorithm in Sec. 4 for the positive data of each class. The parameters $\boldsymbol{w}$ is solved via linear SVM because the cost function $g(\boldsymbol{X}_i,P)$ is a linear function with respect to the parameters $\boldsymbol{w}$. The expected length μ_j for the j-th atomic action element in the *phantom action template* is computed as the average number of elements matched to it in the positive training data:

$$\mu_j = \frac{1}{N_+}\sum_{i:y_i=1}\frac{L_T}{L_i}(e_j^i - b_j^i + 1) \tag{10}$$

where b_j^i and e_j^i are the beginning and ending of the elements warped to the j-th atomic action in the phantom action template for $\boldsymbol{X}_i$, N_+ is the number of the positive training data, L_i is the length of the i-th training sequence, $j = 1, \cdots, L_T$ is the index of the atomic action in the phantom action template.

In the beginning of the algorithm, we initialize P^i to be a uniform warping, which aligns the same number of elements to each atomic action in the phantom action template. For example, if we align a length-4 sequence to a length-2 sequence, $P^i = ((1,1),(2,1),(3,2),(4,2))$.

Finally, in order to deal with multi-class classification, we apply the one-vs-the-rest approach to convert multi-class classification to a set of binary classification problems. We learn a *phantom action template* and the associated score function $f(\boldsymbol{X}_i)$ for each class. The class having the highest score function is regarded to be the predicted class. We require the length of *phantom action template* L_T to be the same for all the classes. The outline of the whole optimization algorithm is given in Alg. 1.

6. Implementation Details

6.1. Frame Feature Description

We represent an action by a sequence of high-dimensional frame descriptors, as described in this section. We are interested in action recognition from both the depth sequences captured by Kinect cameras and conventional RGB videos. The Kinect cameras can capture the depth sequences and track human skeleton joints. The depth sequences give the distance of the object to the camera at each pixel, and the tracked human skeleton joints contain the 3D

Take a set of training data $\{(\boldsymbol{X}_1, y_1), (\boldsymbol{X}_2, y_2), \cdots, (\boldsymbol{X}_N, y_N)\}$ and the number of the classes C.
Initialize the warping path P^i to be the uniform warping for all the training data, and initialize the mean length of the template sequences according to Eq. (10).
for $iter = 1$ *to* $maxiter$ **do**
  for $c = 1$ *to* C **do**
    (1) Find the most violated warping path P for all the negative training data (class label $y_n \neq c$);
    (2) Solve the parameters $\boldsymbol{w}$ with stochastic gradient, with the most violated warping path P specifying the constraints;
    (3) Solve the optimal warping path P_n for all the positive training data (class label $y_n = c$);
    (4) Estimate the expected length of the atomic actions in the *phantom action template* according to Eq. (10).
  end
end
return parameters $\boldsymbol{w}$ and the expected lengths μ_j for all the classes.

Algorithm 1: Latent Structural SVM Learning

locations of the joints. One pixel in the RGB video contains the RGB values of the corresponding point in the scene.

For the 3D human skeleton joint locations, we employ the pairwise joint position feature [26]. This feature first normalizes the joint locations so that it is invariant to the absolute body position, the initial body orientation and the body size. Then, for each joint, we compute its 3D relative positions to all the other joints. The relative positions of all the joints are utilized as the frame descriptor to represent the 3D human skeleton configuration. This representation is a very intuitive way to represent human motion.

For the depth sequences, we employ the local HON4D feature [16]. HON4D feature treats the 3D depth sequence as a surface in the 4D spatio-temporal space, and employs the distribution of the surface normal orientation as a shape descriptor. The local HON4D features are computed around the 3D locations of each human skeleton joint. For each human skeleton joint, we divide its local neighbors as a $N_x \times N_y \times N_z$ 3D spatial grid, and compute the HON4D histograms in all the cells. The concatenation of the HON4D histograms in all the cells for all the human skeleton joint are used as the frame descriptor. This descriptor can roughly characterize the local spatial shape around each joint to represent the human-object interactions.

For RGB videos, we employ widely used HOG [3] and HOF [8] features. The dense HOG and HOF features are extracted at a regular grid for all the frames. We employ the k-means clustering to learn a codebook for all HOG/HOF features. Then each HOG/HOF feature can be quantized by the nearest visual word in the codebook. Finally, the histogram of the visual words in one frame is employed as the frame descriptor. Because the HOG and HOF features are histograms and we are using linear classifiers, we employ the root histograms of the HOG and HOF as the frame descriptors, as suggested in [1].

Our proposed method can use several frame descriptors together. We simply concatenate the different frame descriptors if we use more than one frame descriptor to represent a frame. We will be explicit on this when describing our experiments.

6.2. Other Treatments

First, we observe that there may exist some other (non-temporal) variations in some actions. For example, for the action "call cellphone", some people tend to use their right hand, while some people use their left hand. Learning a mixture of MMTW can help in this situation. We cluster the training data of each action category via k-means clustering using the video-level descriptors (such as bag-of-words and Fourier temporal pyramid). We learn a *phantom action template* for each cluster as a sub-category of a conceptual action class.

Second, in order to avoid the trivial alignment, i.e., aligning most of the sequence into the same *atomic action*, we restrict a_j and d_j in Eq. (4) to be larger than a threshold $\eta = 0.1$ for all j. If the optimization results in a_j or d_j that is smaller than η, we cap them by 0.1.

7. Experiments

We evaluate the proposed MMTW method on five benchmark datasets. The first three dataset are: MSR Sport Action 3D dataset [11], MSR-DailyActivity3D dataset [26] and 3D ActionPair dataset [16]. These datasets contain the depth sequences captured with Kinect cameras and the tracked human skeleton joint positions. We also evaluate the proposed MMTW approach on two RGB video dataset, Olympic sports dataset [15] and UCF-sports dataset [18], to validate its performance on RGB videos.

In the following experiments, unless specified, we use a mixture of two MMTW models for each action class, as described in Sec. 6.2.

7.1. MSR Sports Action3D dataset

MSR Sports Action3D dataset [11] is an action dataset of depth sequences captured by Kinect camera. It contains twenty actions: *high arm wave*, *horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *side-boxing*,

Figure 3. Example frames from different actions from MSR Sports Action dataset [11], MSR-DailyActivity3D dataset [26], 3D Action Pair dataset [16], adn UCF Sports dataset [18]

bend, *forward kick*, *side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing*, *pick up & throw*. Every action was performed by ten subjects three times each. Example depth sequences from this dataset are shown in Fig. 3. This dataset also contains the human skeleton joint positions tracked by the algorithm in [20].

We employ the relative joint positions as the frame descriptors for this dataset, and set the length of the *phantom action template* $L_T = 11$ in this experiment. The accuracy of different methods is shown in Table 1. The proposed MMTW approach achieves a state-of-the-art 92.67% accuracy with the same experimental setup as in [26]. Moreover, compared with the 71.79% accuracy of using the uniform warping (no action alignment), the proposed MMTW approach achieves much better accuracy because it discriminatively aligns the sequences.

We also evaluate the dynamic temporal warping (DTW) in our dataset using the Euclidean distance of the skeleton joint positions as the frame matching. We found that DTW algorithm does not perform well in our experiments, because the Euclidean distance can not discriminatively characterize the similarity of two human skeleton configurations. In contrast, as the proposed MMTW method learns the best alignment from the training data, it can better distinguish different actions.

Finally, we study the robustness of the proposed method to temporal misalignment and phantom action template length, shown in Fig. 4. In this experiment, we circularly shift half of the training data and testing data, and keep the rest of the data the same. The accuracy of the proposed MMTW approach is compared with that of the uniform warping and Fourier Temporal Pyramid [26]. We find that the MMTW approach is much more robust than

Method	Accuracy %
Action Graph on Bag of 3D Points [11]	74.7
Histogram of 3D Joints [28]	78.9
Eigenjoints [30]	82.3
HON4D + D_{desc} [16]	88.9
Actionlet Ensemble [26]	88.2
Random Occupancy Pattern [25]	86.5
Depth HOG [31]	85.5
Dynamic Temporal Warping	54.0
Hidden Markov Model	63.0
Uniform Warping	71.8
MMTW	**92.7**

Table 1. The performance of the methods on Action3D dataset.

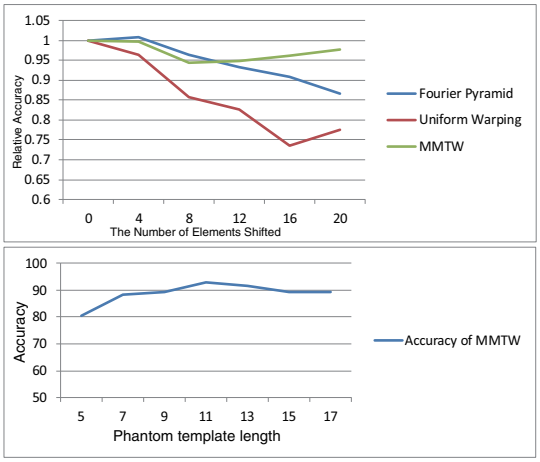

Figure 4. The robustness of the methods to temporal shifts and phantom template length.

the uniform warping approach because of the explicit action alignment in MMTW. We also find that the MMTW approach is more robust than the Fourier Temporal Pyramid approach under large temporal misalignment. Moreover, the proposed method is insensitive to the length of the phantom action template. An example alignment can be found in the supplemental materials.

7.2. MSR-DailyActivity3D dataset

DailyActivity3D dataset is a daily activity dataset captured by a Kinect device. There are 16 activity types:*drink*, *eat*, *read book*, *call cellphone*, *write on a paper*, *use laptop*, *use vacuum cleaner*, *cheer up*, *sit still*, *toss paper*, *play game*, *lay down on sofa*, *walk*, *play guitar*, *stand up*, *sit down*. If possible, each subject performs an activity in two different poses: "sitting on sofa" and "standing". Some example frames are shown in Fig. 3. This dataset also provides the human skeleton joint positions tracked via [20].

Modeling human-object interaction is very important for this dataset. Thus, in addition to the relative joint posi-

Method	Accuracy %
HON4D + D_{desc} [16]	80.00
Actionlet Ensemble [26]	85.75
Dynamic Temporal Warping	34.45
Uniform Warping	69.38
MMTW	**88.75**

Table 2. The performance of the methods on Sports Action 3D dataset.

Method	Accuracy %
HON4D + D_{desc} [16]	96.67
Skeleton + LOP + Pyramid [26]	82.22
Depth HOG [31]	66.11
Uniform Warping	90.00
MMTW	**97.22**

Table 3. The performance of the methods on 3D action pairs dataset.

tions, we also use the local HON4D features [16] extracted at each human skeleton joint, as well as the human skeleton joint positions per-frame features. We use a patch size of $12 \times 12 \times 6$, and divide it into a $3 \times 3 \times 1$ grid for HOV4D features. We set the length of the *phantom action template* $L_T = 12$ in this experiment. Table 2 shows the performance of different methods. The proposed MMTW method achieves 88.75% accuracy. It outperforms the state-of-the-art methods.

7.3. 3D ActionPair dataset

3D ActionPair dataset [16] is an action dataset captured by a Kinect camera. This dataset contains six pairs of actions: '*Pick up a box/Put down a chair*, *Lift a box/Place a box*, *Push a chair/Pull a chair*, *Wear a hat/Take off hat*, *Put on a backpack/Take off a backpack*, *Stick a poster/Remove a poster*. Since the motion cue is usually similar for a pair of the actions, modeling the temporal structure is crucial for successful action recognition. The example frames are shown in Fig. 3.

We employ the relative joint positions and the local HOV4D features [16] extracted at each human skeleton joint and the human skeleton joint positions per-frame features. We use a patch size of $12 \times 12 \times 6$, and divide it into a $3 \times 3 \times 1$ grid for HOV4D features. We set the length of the *phantom action template* $L_T = 16$ in this experiment. The experimental setting of [16] is used in our experiments. The result is shown in Table 3. The proposed MMTW method achieves excellent accuracy (97.22%) on this dataset because it can model the temporal order of the time sequence very well, and its performance is much better than uniform warping.

Method	Accuracy %
STIP [8]	62.0
Decomposable Motion Segments [15]	72.1
Latent Temporal Structure [22]	66.8[a]
Uniform Warping	52.9
MMTW	**73.8**

Table 4. The performance of the methods on Olympic Sports dataset.

[a]This result is obtained under a different experimental setting.

7.4. Olympic Sports dataset

The Olympic Sports dataset [15] is captured by RGB cameras. It contains the sports actions from 16 sport classes: *basketball layup*, *bowling*, *clean and jerk*, *discus throw* , *diving platform 10m*, *diving springboard 3m*, *hammer throw*,*high jump*, *javelin throw*, *long jump*, *pole vault*, *shot put*, *snatch*, *tennis serve*, *tripe jump*, *vault* with 50 sequences per class. The actions in this dataset usually exhibit the complex temporal structure and temporal misalignment. The sequences are collected from YouTube, and the class label annotations are obtained using Mechanical Turk.

We extract dense HOG/HOF features for all the frames, and employ the bag-of-words representation of the HOG/HOF features in one frame as frame-level descriptor. The length of the *phantom action template* is set to be $L_T = 30$. The experimental setting suggested by [15] is employed in this experiment. Table 4 shows the experimental results. The proposed method archives better accuracy than [15] because the proposed approach is more flexible than [15]. In the proposed MMTW method, one atomic action can occur at any place of the input sequence, as long as the order of the atomic action is preserved, while [15] restricts the position of the atomic action.

7.5. UCF-sports datasets

The UCF-Sports dataset [18] is captured by RGB cameras. It contains the sports actions from 12 categories: *Diving-side*, *Golf-swing-back*, *Golf-swing-front*, *Golf-swing-side*, *Kicking-front*, *Kicking-side*, *Riding-horse*,*Run-side*, *skateboard*, *swing-bench*, *Swing-sideangle*, *walking*. Each action is performed 5-12 times.

We extract dense HOG/HOF features for all the frames, and employ the bag-of-words representation of the HOG/HOF features in one frame as frame-level descriptor. The length of the *phantom action template* is set to be $L_T = 25$ in this experiment. We employ the leave-one-out cross validation experimental setting. Table 5 shows the accuracy of different methods. The proposed MMTW approach archives 90.00% accuracy despite the fact that MMTW merely uses dense HOG/HOF features here. Although other methods can achieve slightly better recognition accuracy by modeling the spatio-temporal context [27]

Method	Accuracy %
Dense HOG/HOF [24]	81.6
Dense HOG3D [24]	85.6
Feature Learning [9]	86.5
Hierarchical spatio-temporal context [7]	87.3
Context and appearance distribution [27]	91.3
Action Bank [19]	95.0
Uniform Warping	55.56
MMTW	**90.00**

Table 5. The performance of the methods on UCF-Sports dataset.

or using detection responses [19], since this paper is mainly focuses on temporal structure modeling, we simply use widely used HOG/HOF features and have already obtained comparable performance to [27] and [19]. This experiment also shows that the proposed MMTW can achieve much better recognition accuracy than the bag-of-words representation when dense HOF/HOF features are employed [24].

8. Conclusion

This paper proposes a novel unification of action alignment and classification, called maximum margin temporal warping (MMTW). MMTW method integrates the advantages of the dynamic temporal warping and discriminative max-margin learning. Due to the learned action alignment, it is robust to the temporal variations and misalignment, while at the same time maximizes the margin among different action classes. Extensive experiments have demonstrated the robustness and superior performance of the proposed approach on five benchmark datasets. In the future, we plan to apply the proposed approach to other sequential data classification applications, such as handwriting recognition.

9. Acknowledgement

This work was supported in part by National Science Foundation grant IIS-0916607, IIS-1217302, and DARPA Award FA 8650-11-1-7149.

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[2] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*. Ieee, Nov. 2011.

[3] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, pages 886–893. IEEE, 2005.

[4] H. Ding, G. Trajcevski, and P. Scheuermann. Querying and mining of time series data: experimental comparison of representations and distance measures. In *PVLDB*, volume 1, 2008.

[5] A. Gaidon, C. Schmid, and L. I. Grenoble. Actom Sequence Models for Efficient Action Detection. In *CVPR*, 2011.

[6] M. Hoai and F. De la Torre. Max-margin early event detectors. In *CVPR*, pages 2863–2870. Ieee, June 2012.

[7] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, pages 2046–2053, 2010.

[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, volume 1, pages 1–8, 2008.

[9] Q. Le, W. Zou, S. Yeung, and A. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[10] K. Li, J. Hu, and Y. Fu. Modeling Complex Temporal Composition of Actionlets for Activity Prediction. In *ECCV*, pages 286–299, 2012.

[11] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *Human Communicative Behavior Analysis Workshop (in conjunction with CVPR)*, 2010.

[12] F. Lv and R. Nevatia. Recognition and Segmentation of 3-D Human Action Using HMM and Multi-class AdaBoost. In *ECCV*, pages 359–372, 2006.

[13] Minh Hoai and F. D. Torre. Maximum Margin Temporal Clustering. In *ICML*, volume XX, 2012.

[14] M. Muller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 137–146. Eurographics Association, 2006.

[15] J. C. Niebles, C.-w. Chen, and L. Fei-fei. Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In *ECCV*, pages 1–14, 2010.

[16] O. Oreifej and Z. Liu. HON4D : Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In *CVPR*, 2013.

[17] M. Pei, Y. Jia, and S.-c. Zhu. Parsing Video Events with Goal inference and Intent Prediction. In *ICCV*, 2011.

[18] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH:a spatio-temporal Maximum Average Correlation Height filter for action recognition. In *CVPR*, pages 1–8. Ieee, June 2008.

[19] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, number May, 2012.

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[21] T. Simon, M. H. Nguyen, F. D. La, and J. F. Cohn. Action Unit Detection with Segment-based SVMs. In *CVPR*, 2010.

[22] K. Tang and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, pages 1250–1257. Ieee, June 2012.

[23] D. Tran and J. Yuan. Max-Margin Structured Output Regression for Spatio-Temporal Action Localization. In *NIPS*, 2012.

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

[25] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3D Action Recognition with Random Occupancy Patterns. In *ECCV*, pages 1–14, 2012.

[26] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In *CVPR*, 2012.

[27] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In *CVPR*. Ieee, June 2011.

[28] L. Xia, C.-c. Chen, and J. K. Aggarwal. View Invariant Human Action Recognition Using Histograms of 3D Joints The University of Texas at Austin. In *CVPR 2012 HAU3D Workshop*.

[29] T. Xiang and S. Gong. Beyond Tracking: Modelling Activity and Understanding Behaviour. *International Journal of Computer Vision*, 67(1):21–51, Apr. 2006.

[30] X. Yang and Y. Tian. EigenJoints-based Action Recognition Using Naïve-Bayes-Nearest-Neighbor. In *CVPR 2012 HAU3D Workshop*, 2012.

[31] X. Yang, C. Zhang, and Y. Tian. Recognizing Actions Using Depth Motion Maps-based Histograms of Oriented Gradients. In *ACM Multimedia*, 2012.

[32] B. Yao and S.-C. Zhu. Learning deformable action templates from cluttered videos. In *ICCV*, pages 1507–1514. IEEE, Sept. 2009.

[33] C. Yu and T. Joachims. Learning structural SVMs with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, New York, New York, USA, 2009. ACM.