Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

Minimum-risk temporal alignment of videos

Zhen Wang · Massimo Piccardi

Received: date / Accepted: date

Abstract Temporal alignment of videos is an important requirement of tasks such as video comparison, analysis and classification. Most of the approaches proposed to date for video alignment leverage dynamic programming algorithms whose parameters are manually tuned. Conversely, this paper proposes a model that can learn its parameters automatically by minimizing a meaningful loss function over a given training set of videos and alignments. For learning, we exploit the effective framework of structural SVM and we extend it with an original scoring function that suitably scores the alignment of two given videos, and a loss function that quantifies the accuracy of a predicted alignment. The experimental results from four video action datasets show that the proposed model has been able to outperform a baseline and a state-of-the-art algorithm by a large margin in terms of alignment accuracy.

Keywords Sequence alignment · action video alignment · dynamic time warping · extended hidden Markov model · structural SVM

1 Introduction and Related Work

When dealing with sequential data, one of the urgent problems is how to align multiple sequences to allow their meaningful comparison. This problem, known as sequence alignment or warping, concerns fields as diverse as bioinformatics, finance, climate series analsyis and meteorology, and multimedia signal processing at large. The problem is often framed as the alignment of two given sequences, with the first being used as reference and the second being aligned,

Z. Wang
University of Technology Sydney
E-mail: zhen.wang-3@student.uts.edu.au

M. Piccardi
University of Technology Sydney
E-mail: massimo.piccardi@uts.edu.au

or "warped", onto the first. In the case of video clips, the goal of sequence alignment is that of finding corresponding frames in two given videos to be used for comparison, analysis and, possibly, classification.

The most well-known sequence alignment technique is dynamic time warping (DTW). Its main idea is to scan both sequences while looking for local correspondences of minimum cost, where the cost is a function that reflects both the similarity between the frames and their indices [20]. The outputs of DTW are a *path*, i.e., a set of index correspondences in the two sequences, and a total cost which can be interpreted as an overall dissimilarity between the sequences. DTW is an instance of dynamic programming algorithms and, as such, the returned path is guaranteed to be globally optimal. While DTW was originally proposed for the alignment of time series, it has later found use in a number of other applications including data mining [14], speech processing [18], medicine [5] and classification of genomic signals [23]. Over the years, many extensions have been proposed. Windowing DTW restricts the corresponding frames to fall within a given window [4]. Slope-weight DTW restricts the search to paths within a given slope [22]. Keogh and Pazzani [15] used derivatives of the original signal to improve the alignment before applying DTW. In computer vision, Gong and Medioni have extended DTW by integrating it with manifold learning [7]. Hsu *et al.* have augmented warping along the time dimension with smooth spatial warping to align actions performed with different styles [10]. Gritai *et al.* have exploited anthropometric and epipolar constraints to improve the alignment of human actions [9]. Junejo *et al.* have used DTW to recognize human actions under view changes. Amongst the many algorithms, the state of the art is likely held by the generalized canonical time warping (GCTW) that applies canonical correlation analysis (CCA) alongside DTW to perform the alignment in a subspace [30]. The role of CCA is to analyze the two sets of multivariate measurements and extract the most informative linear combinations of their dimensions [2]. GCTW iteratively alternates between CCA and a weighted version of DTW to simultaneously find an optimal linear subspace and a path, and it has recently reported the best performance over a variety of video datasets [30].

However, while DTW and its variants provide inference of the optimal alignment path, they do not provide explicit procedures for the training of the cost function. To amend this issue, it is possible to adopt an *extended hidden Markov model* and learn the cost function under maximum likelihood from a set of manually-aligned sequence pairs [6,3]. This extended HMM is a graphical model consisting of the two sequences of measurements and a Markov chain of latent states that encodes their alignment path. Further details are provided in the next section. The main advantage of this extended HMM is that it is a proper probabilistic model that can be trained in a maximum-likelihood framework. However, in recent years, training objectives other than the likelihood function such as the cross-entropy and the regularized empirical risk have proved to lead to more accurate models and appear promising also for the alignment problem.

The goal of our paper is to propose an alignment approach that can outperform the state of the art in video alignment. To this aim, we integrate the extended hidden Markov model with regularized risk minimization (i.e., the structural support vector machine [25]) and dedicated dissimilarity functions. Our approach is inspired by the work in [13] on protein alignment. However, the application to video has required us to recast the features and the dissimilarity functions to the frame domain. While an initial version of this work was presented at a recent conference [27], this submission has been substantially rewritten and extended, and it presents these original contributions:

- Two dedicated cost functions (one stricter, one more lenient) that can be used to describe desirable performance for the sequence alignment task;
- A training algorithm for structural SVM that can minimize these dedicated loss functions over any given training set;
- A performance analysis of various dissimilarity functions for the comparison of video frames;
- An extensive experimental evaluation that includes four action video datasets and quantitative and qualitative comparisons.

The proposed model has been tested against DTW (as baseline) and GCTW (state of the art) in a set of experiments on action alignment over selected actions from the Weizmann dataset [8], the Olympic Sports dataset [19], the UCF101 dataset [24] and the MSR Daily Activity 3D dataset [28]. The experimental results show that the proposed approach has been able to outperform the compared approaches by at least 10 percentage points of alignment accuracy in all the experiments.

2 Sequence Alignment and Minimum-Risk Training

In this section, we first formally describe the task of sequence alignment, and then we describe the extended hidden Markov model (EHMM) and the structural SVM framework for the training of its parameters.

2.1 Sequence Alignment

Given two generic sequences of multidimensional measurements, $s = \{s_1, \ldots, s_i, \ldots, s_{L_s}\}$ and $t = \{t_1, \ldots, t_j, \ldots, t_{L_t}\}$, the alignment task is formally defined as providing a set of monotonically- increasing index pairs over the two sequences. However, to simplify both notations and operations, the alignment can be redefined as a sequence of only three types of symbols: M ("match"), S ("insert a gap on sequence s") and T ("insert a gap on sequence t"). These symbols have the following meaning: assuming i and j to be the current indices over sequences s and t, respectively, 1) symbol M pairs frames s_i and t_j and then increments both indices; 2) symbol S pairs no frames and only increments index j; and, likewise, 3) symbol T pairs no frames and only increments index

i. From an alignment path made of these symbols, it is possible to sequentially reconstruct the set of paired indices. As a toy example, we show below a possible alignment path for two short sequences, $s = \{6.1, 10.5, 9.2, 10.0, 8.4, -5.2\}$ and $t = \{6.3, 5.8, 11.0, 9.5, -4.8\}$:

$$\begin{array}{lcccccc}
s = 6.1 & - & 10.5 & 9.2 & 10.0 & 8.4 & -5.2 \\
t = 6.3 & 5.8 & 11.0 & 9.5 & - & - & -4.8 \\
y = M & S & M & M & T & T & M
\end{array}$$

In the above example, sequence y encodes the alignment path, with the M symbols showing the matched frames and the S and T symbols accounting for the required gaps. The corresponding set of paired frames is: $\{(s_1, t_1), (s_2, t_3), (s_3, t_4), (s_6, t_5)\}$. Intuitively, a good alignment path will pair frames of similar values without inserting unneeded gaps. The optimal (i.e., minimum-cost) path can only be identified once a precise cost function is given to account for the differences between frame pairs and the cost of gap insertions. Given a cost function, the optimal path can be easily inferred by using a dynamic programming algorithm of $O(L_s L_t))$ complexity; we present this algorithm in Section 2.2. The length of an alignment path encoded in terms of these matching symbols is always bounded between $\max(L_s, L_t)$ and $L_s + L_t$.

2.2 Extended Hidden Markov Model

In probability notation, the extended HMM for sequence alignment is a model for the joint probability, $p(s, t, y)$, of the two sequences and their alignment path. Such a model can be used to infer an optimal alignment, $\bar{y}$, for any two given sequences as $\bar{y} = \text{argmax}_y\, p(s, t, y)$. Like for a conventional HMM, the joint probability of an EHMM factorizes into a set of transition and emission probabilities. The transition probabilities include: (1) the probabilities to transition from state M to either S or T; (2) the probabilities to transition from either S or T to M; and (3) the probabilities to stay in S or T. We note these transition probabilities as $p(y_k|y_{k-1})$. Note that this model bars direct transitions from S to T and the vice versa assuming that a pair of matched frames will always follow a run of gaps.

To complete the model, we also need to define the emission probabilities. To this aim, we note the probability of emitting a matched pair of measurements, (a, b), as $p_{a,b}$ and the probability of emitting measurement a against a gap as q_a. In the common case of numerical measurements, both p and q will be multivariate likelihoods such as Gaussian distributions or mixture models. Figure 1 shows a graphical model representation of the EHMM.

Using an EHMM, the optimal alignment for a pair of sequences can be found via a dynamic programming algorithm reminiscent of the well-known Viterbi algorithm [21]. The main steps of the algorithm are given below, where the probability of reaching state $* = \{M, S, T\}$ at indices i and j over s and t is noted compactly as $p^*(i, j)$.

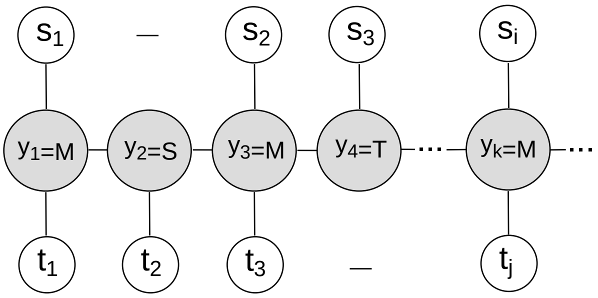
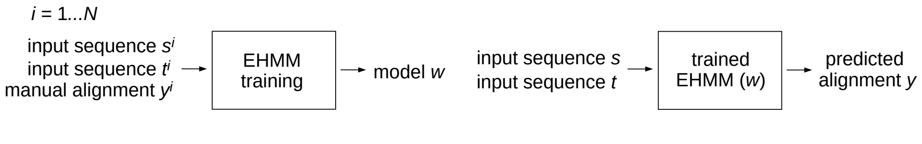


Fig. 1: The extended hidden Markov model for sequence alignment (represented as an undirected graphical model).



a) Training

b) Inference

Fig. 2: Main steps of the proposed approach: a) training (with regularized empirical risk minimization, i.e., structural SVM - Section 2.3); inference (Section 2.2).

Initialization: $p^M(1,1) = p^S(1,1) = p^T(1,1) = p^*(0,j) = p^*(i,0) = 1$.

Recurrence: $i = 1, \ldots, L_s, j = 1, \ldots, L_t$:

$$p^M(i,j) = p_{s_i,t_j} \max \begin{cases} p(y_k = M|y_{k-1} = M)\ p^M(i-1,j-1) \\ p(y_k = M|y_{k-1} = S)\ p^S(i-1,j-1) \\ p(y_k = M|y_{k-1} = T)\ p^T(i-1,j-1) \end{cases} \tag{1}$$

$$p^S(i,j) = q_{s_i} \max \begin{cases} p(y_t = S|y_{t-1} = M)\ p^M(i-1,j) \\ p(y_k = S|y_{k-1} = S)\ p^S(i-1,j) \end{cases} \tag{2}$$

$$p^T(i,j) = q_{t_j} \max \begin{cases} p(y_k = T|y_{k-1} = M)\ p^M(i,j-1) \\ p(y_k = T|y_{k-1} = T)\ p^T(i,j-1) \end{cases} \tag{3}$$

Termination:

$$p(s,t,\bar{y}) = \max(p^M(L_s,L_t), p^S(L_s,L_t), p^T(L_s,L_t)) \tag{4}$$

Probability $p(s,t,\bar{y})$ is the maximum probability for the two given sequences, s,t, and the optimal alignment, $\bar{y}$, can be easily obtained by storing the corresponding state sequence.

2.3 Minimum-Risk Classification and Structural SVM

Empirical minimum risk (EMR) classifiers learn the classifier's parameters by minimizing a chosen loss function over a given training set. To avoid overfitting the model onto the training data, regularization terms are also often added to the minimization objective. The most famous member of ERM is the support vector machine which has also been extended to the case of structured prediction, i.e., the classification of structures such as sequences and graphs, and tasks such as ranking and alignment (structural SVM [25]). In the alignment case, the problem is to learn a scoring function, $F(s,t,y)$, that quantifies the compatibility of measurement sequences s and t and alignment path y based on training samples of manually-aligned sequence pairs. The scoring function typically takes the form of a linear discriminant, $F(s,t,y) = w^\top \psi(s,t,y)$, that can be extended to non-linear mappings by the use of kernels. ψ is a function that maps an alignment y of s and t to a so-called feature vector (a suitable numerical vector that is independent of the model's parameters; details are provided in the following subsection), and w is the vector of the model's parameters. Such a linear model is completely equivalent to the full probabilistic model in the case of distributions that belong to the exponential family (Gaussian, categorical, Gamma, chi-squared and many others) through the simple position $w^\top \psi(s,t,y) \propto \ln p(s,t,y)$ and it is therefore suitable to represent the EHMM. In addition, the assumption does not require the probability distribution to be normalized and therefore the w parameters can be chosen from a larger domain.

Given a model, w, inference of the optimal aligment path, $\bar{y}$, for two given input sequences can thus be formally obtained as:

$$\bar{y} = \underset{y \in Y}{\operatorname{argmax}} \; w^\top \psi(s,t,y) \tag{5}$$

where Y is the set of all possible alignments between s and t. If the scoring function, $w^\top \psi(s,t,y)$, can be decomposed as a sum over the frames of the sequences, its maximum in y can be efficiently found using a dynamic programming algorithm akin to a modified Viterbi algorithm (1) in logarithmic scale.

At its turn, the model can be learned by minimizing the risk over a given training set of supervised sequence-path triplets using structural SVM. By noting the training set as $(s^i, t^i, y^i), i = 1 \dots N$, the learning objective of structural SVM can be written as:

$$\begin{aligned} &\underset{w,\xi}{\operatorname{argmin}} \; \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \xi^i \quad s.t. \\ &w^\top \psi(s^i,t^i,y^i) - w^\top \psi(s^i,t^i,y) \geq \mathit{\Delta}(y^i,y) - \xi^i, \\ &\qquad i = 1 \dots N, \;\; \forall y \in \mathcal{Y} \end{aligned} \tag{6}$$

Like for a conventional SVM, objective (6) aims to minimize a trade-off between an upper bound of the classification loss over the training set ($\sum_{i=1}^{N} \xi^i$) and a regularization term ($\|w\|^2$). The constraints ensure that the score assigned to the ground-truth alignment, y_i, is higher than that assigned to any other alignment, y, by a margin equal to the classification loss for that alignment, $\Delta(y^i, y)$. At its turn, $\Delta(y^i, y)$ is a loss function that can be arbitrarily chosen to quantify the inaccuracy of incorrect alignments.

The challenge with structural SVM is that the number of possible alignments for a given sequence pair is exponential in their length. This, in turn, leads to a highly-constrained learning objective that proves computationally infeasible even for relatively short sequences. However, Tsochantaridis *et al.* in [25] have shown that an ϵ-close, controlled approximation to the solution of (6) can be obtained by using only a polynomial (i.e., easily feasible) number of constraints, and Joachims *et al.* in [13] have shown that this approach can also be used for the sequence alignment problem. The constraints are chosen as the "most-violated constraints", i.e., the constraints that set the value of variable ξ^i for each sample, $i = 1 \ldots N$. Let us consider the constraints in (6) and rearrange their terms:

$$\begin{aligned} & w^\top \psi(s^i, t^i, y^i) - w^\top \psi(s^i, t^i, y) \geq \Delta(y^i, y) - \xi^i \;\; \forall y \\ & \rightarrow \xi^i \geq -w^\top \psi(s^i, t^i, y^i) + w^\top \psi(s^i, t^i, y) + \Delta(y^i, y) \;\; \forall y \\ & \rightarrow \xi^i = \max_y(-w^\top \psi(s^i, t^i, y^i) + w^\top \psi(s^i, t^i, y) + \Delta(y^i, y)) \\ & \rightarrow y^{*i} = \operatorname*{argmax}_y(w^\top \psi(s^i, t^i, y) + \Delta(y^i, y)) \end{aligned} \tag{7}$$

Equation (7) shows that the alignment y^{*i} setting the value of variable ξ^i can be found by a modified version of the inference, known as the "loss-augmented" inference since it adds up the loss function to the score. If the loss function, too, can be evaluated frame-by-frame, the maximum of the loss-augmented inference can still be found by the same algorithm used for the inference by adding the loss to the emission scores.

Algorithm 1 shows the main steps of the training procedure. In the pseudocode, ϵ is a small constant that sets the accuracy of the approximation (set to 0.01 in the experiments), and $\mathcal{W}$ is the set of the most-violated constraints. As for similar quadratic programs, the training algorithm enjoys convergence to a global optimum [25]. Its computational complexity has been proven to be only $O(\frac{N}{\epsilon})$, where ϵ is the accepted tolerance from the exact solution [12].

3 The Proposed Model

The model proposed for the alignment of video pairs consists of: a) a linear scoring function that embeds the graphical structure of the EHMM; b) various dissimilarity functions that measure the dissimilarity of any two video frames;

Algorithm 1: Structural SVM training algorithm: main steps.

```
Input  : Measurement sequences $s^i, t^i$ and ground-truth alignment $y^i$, $i = 1 \dots N$;
         parameter $\epsilon$
$\mathcal{W} = \emptyset,\ w = 0,\ \xi = 0$
repeat
    foreach $i = 1 \dots N$ do
        $y^{*i} \leftarrow \mathrm{argmax}_y (w^\top \psi(s,t,y) + \Delta(y^i, y))$;
        if $\xi^i = [w^\top (\psi(s^i, t^i, y^{*i}) - \psi(s^i, t^i, y^i)) + \Delta(y^i, y^{*i})] > \xi^{i\ prev} + \epsilon$ then
            $\mathcal{W} \leftarrow \mathcal{W} \cup y^{*i}$;
        end
    end
    $(w, \xi) = \mathrm{argmin}_{w,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi^i$ s.t. $\mathcal{W}$;
until $\xi$ unchanged;
Output: Model $w$
```

and c) two loss functions capable of properly quantifying alignment/misalignment. These components are presented in the following subsections.

3.1 Scoring Function

In the structural SVM framework, the score for a sample (s, t, y) is obtained from the product of a parameter vector, w, and a feature function, $\psi(s, t, y)$, that provides a re-mapping of the given sequences and path. Both the parameter vector and the feature function contain two sections: one accounting for the transitions between the states of the alignment path, and one accounting for the emission of the measurements from the two sequences. The transition parameters, noted as w^{tr}, are a 3×3 matrix indexed by states y_{k-1} and y_k (note that transitions between symbols $S \rightarrow T$ and $T \rightarrow S$ are not allowed). The transition features are therefore just indicator functions that take a value of one for states y_{k-1} and y_k and zero otherwise. As emission features, we use a dissimilarity function over measurements s_i and t_j emitted by a matching state. Therefore, the emission parameters, w^{em}, are a vector with the same dimensionality as the dissimilarity function. With these assumptions, the scoring function can be written as:

$$
\begin{aligned}
& w^\top \psi(s, t, y) = \sum_{k=1}^{|y|} w^{tr}_{y_{k-1}, y_k} + w^{em\top} d(s_i - t_j) \mathbf{I}[y_k = M] \\
& w^{tr}_{0,*} = 0; \quad \mathbf{I}[y_k = M] : i{+}{+}, j{+}{+}; \\
& \mathbf{I}[y_k = S] : j{+}{+}; \quad \mathbf{I}[y_k = T] : i{+}{+}
\end{aligned}
\tag{8}
$$

The notations in (8) read as: $|y|$ is the length of the alignment path; $d(s_i, t_j)$ is the dissimilarity function between measurements s_i and t_j (details in Section 3.2); $\mathbf{I}$ is an indicator function that takes a value of one when its argument is true; and indices i and j, initially set to 1, are post-incremented according to the value of state y_k. It is evident that the scoring function decomposes over the individual states of the alignment path, $y_k, k = 1 \dots |y|$, in a form of the type $\sum_{k=1}^{|y|} w^\top \phi(s, t, y_k)$, thus permitting a Viterbi-style inference.

3.2 Dissimilarity Functions

To quantify the dissimilarity of two frames, we have employed three popular dissimilarity measurements: the cosine distance, the Euclidean distance and the Euclidean squared distance. The cosine distance is defined as one minus the cosine between two vectors [1]. As such, it ranges between zero (the vectors have the same orientation) and two (the vectors have opposite orientation, i.e., are most dissimilar). The Euclidean distance and the Euclidean squared distance measure the dissimilarity as the element-wise sum of squared differences between the two vectors, with and without a final square root, respectively. These distances are increasingly sensitive to the difference in magnitude between the two vectors, and the choice between them should reflect whether this difference is informative or not. In particular, the cosine distance is completely insensitive to the variations in the magnitude of the two input vectors and can therefore focus on differences in direction.

In the experiments described in Section 4, we have used four different datasets. For the first and fourth datasets, we have utilized specific measurements (PCA of background-subtracted frames and 3D skeletons, respectively) and the differences in magnitude seemed a-priori important. In fact, for these datasets the Euclidean distance delivered the highest accuracy. For the second and third datasets, we have used bag-of-words histograms of dense HOG/HOF features [16]. Since these histograms are normalized, the cosine distance seemed the most appropriate, and the experimental results have confirmed it.

3.3 Loss Function and Loss-Augmented Inference

The loss function, $\Delta(y^g, y)$, assigns a penalty for predicting alignment y when the annotated ground-truth alignment is y^g. This function must be able to gradually quantify what we regard as a "bad" or a "good" prediction. In turn, the reciprocal of the loss function can be used as the main measurement of accuracy. In our model, we have adopted two types of loss functions, nicknamed as Q-loss and Q_4-loss.

To describe these loss functions, we first need to expand each match state of an assignment into its corresponding index pair. We then introduce an indicator function, $\mathbf{I}[y^g_h = (m, n), y_k = (i, j)]$, that takes value one if the index pairs in its arguments are the same, and zero otherwise. We use this function to define a partial matching function:

$$\delta(y^g, y_k) = \sum_{h=1}^{|y^g|} \mathbf{I}[y_h^g = (m,n), y_k = (i,j)] \tag{9}$$

that checks whether index pair (i, j) in the predicted assignment matches any index pair in the ground-truth assignment. Since indices are allowed to only appear once in an assignment, this function can only return one if a match is found and zero otherwise. Eventually, the Q-loss function is defined as:

$$\Delta_Q(y^g, y) = 1 - \frac{\sum_{k=1}^{|y|} \delta(y^g, y_k)}{N}. \tag{10}$$

where N is the number of index pairs in the ground-truth alignment. The Q-loss is a recall-like measure which returns zero if the predicted alignment contains all the index pairs of the ground truth, and proportionally up to one in case of missed pairs. Function $1 - \Delta_Q(y^g, y)$ is therefore a measurement of accuracy, and we refer to is as Q-accuracy hereafter.

The key step of the training of structural SVM is the loss-augmented inference in (7). Equation (10) shows that, like the scoring function, also the Q-loss decomposes over the individual states of the predicted alignment, $y_k, k = 1 \ldots |y|$. Thus, the loss-augmented inference takes the form:

$$\underset{y \in Y}{\operatorname{argmax}} \sum_{k=1}^{|y|} \left[w^\top \phi(s, t, y_k) - \delta(y^g, y_k) \right] \tag{11}$$

which can, again, be computed by the same algorithm used for the inference by adding the loss to the emission scores.

Another useful loss function is the Q_4-loss. This is a more lenient loss function that counts a match as correct even if the paired indices in the prediction are shifted by ±2 compared to those in the ground truth (i.e., "close enough"). In other terms, given $y_k = (i, j)$, a match with (m, n) in the ground truth is stated if $|i - m| \leq 2$ and $|j - n| \leq 2$. The Q_4-loss is, too, decomposable, and we define the Q_4-accuracy as $1 - Q_4$-loss.

In the experiments, we perform training using (11) and report results in terms of both Q-accuracy and Q_4-accuracy. In addition, during the annotation of the training set we have annotated the ground-truth alignments only for "key" frames that we have been able to pair with high confidence (e.g., apex phases of movements). The accuracy is measured only against such key frame pairs. Figure 2 summarizes the two main steps of the proposed approach.

4 Experimental Results

In order to evaluate the performance of the proposed model, we have carried out experiments against the baseline model, DTW [20], and a state-of-the-art algorithm, GCTW [30], over selected actions from four action video datasets. The first experiment has compared the performance in aligning the "jump"

	GCTW	DTW	Proposed approach		
			Cosine	Euclidean Squared	Euclidean
Q-accuracy	60.8%	41.2%	52.9%	51.1%	**72.5%**
Q_4-accuracy	96.1%	72.6%	**98.0%**	96.1%	**98.0%**

Table 1: Experimental results for action "jump" from the Weizmann dataset.

action from 9 subjects of the Weizmann dataset [8]. In the second experiment, we have compared the "clean and jerk" (weightlifting) action performed by 11 subjects from the challenging Olympic Sports dataset [19]. The third experiment has aligned instances of "body weight squats" from the UCF101 dataset. These three datasets all consist of RGB videos. However, since the proposed model can align generic data streams, in the forth experiment we have aligned instances of 3D joint sequences from action "stand up" in the MSR Daily Activity 3D dataset [28].

As software for structural SVM, we have used the SVM^{struct} package of Joachims [11]. For training, we have set parameters C to 10 and ϵ to 0.01 from a preliminary analysis, noting very limited sensitivity. As software for GCTW and DTW, we have used the package of Zhou [29], using the author's values for the parameters. The accuracy is reported in terms of both Q-accuracy and Q_4-accuracy (see Section 3.3).

4.1 Weizmann Dataset

The Weizmann dataset is a staged action dataset of 9 people performing 10 actions. In this experiment, we have used the proposed model to align video sequences of the "jump" action from different performers. As measurements, we have first subtracted the background from each frame and then preprocessed the resulting frames by the Euclidean distance transform [17], retaining 416 principal components (99% of the energy) as in [29,30]. For alignment, we have selected 13 video pairs and annotated their corresponding key frames manually. We have then randomly picked 6 pairs as training set and the remaining as test set.

Table 1 shows the alignment results for the "jump" action. The proposed model with the Euclidean distance has achieved a Q-accuracy that is 31.3 percentage points higher than that of DTW, and 11.7 higher than that of GCTW. Although these differences reduce significantly in terms of the more lenient Q_4-accuracy, the proposed model has still outperformed both GCTW and DTW. In terms of dissimilarity functions, the Euclidean distance has proved remarkably better than the other two, although results are again more similar in terms of Q_4-accuracy. This is likely due to the fact that the features themselves are based on the Euclidean distance. It is also interesting to note that the accuracy of GCTW has been much higher than that of DTW, as expected from a state-of-the-art algorithm.

	GCTW	DTW	Proposed approach		
			Cosine	**Euclidean Squared**	**Euclidean**
Q-accuracy	42.4%	36.9%	**53.0%**	45.5%	50.8%
Q_4-accuracy	70.5%	69.6%	**78.0%**	72.7%	76.5%%

Table 2: Experimental results for action "clean and jerk" from the Olympic Sports dataset.

4.2 Olympic Sports Dataset

For the second experiment, we have chosen the "clean and jerk" action from the Olympic Sports dataset since manual alignment of videos pairs could be done with good confidence. This dataset is much more challenging since it consists of real (unstaged) videos from YouTube, taken from very different scenes and viewpoint. We have manually annotated 55 video pairs and split them into 27 as training set and 28 as test set. We have then extracted dense STIP features from all the videos [16] and used the VLFeat library [26] to compute bag-of-words histograms with $1,000$ bins from each frame. although these are popular features in the action recognition literature, we stress that the choice of the specific features is not the focus of this paper.

Table 2 shows the alignment results for the "clean and jerk" action. The proposed model with the cosine distance has achieved a Q-accuracy that is 16.1 percentage points higher than DTW and 10.6 than GCTW. The higher performance with the cosine distance is likely due to the fact that the magnitude of bag-of-words features is not informative. Marked improvements are also reported in terms of Q_4-accuracy. In general, the accuracies are lower than for the Weizmann dataset since this dataset is more realistic and probing. While the accuracy has decreased with the other dissimilarity functions, the proposed model has still outperformed DTW and GCTW in all cases.

4.3 UCF101 Dataset

UCF101 is a large dataset of videos collected from YouTube with 101 action categories. For this experiment, we have chosen action "body weight squats" which was performed by 23 people under different scenes and viewpoints. As frame measurements, we have extracted the same features as the previous dataset (dense STIPs, bag-of-words encoded). We have then manually aligned the key frames of 253 video pairs and split them into 126 pairs for training and 127 for testing.

Table 3 shows the alignment results for the "body weight squats" action. The proposed model with the cosine distance has outperformed DTW in terms of Q-accuracy by 15.8 percentage points, and GCTW by 10.3 percentage points. The differences remain remarkable also in terms of Q_4-accuracy. Like for the Olympic Sports dataset, the cosine distance has achieved the highest accuracy, confirming that it is the most suitable for bag-of-words features.

	GCTW	**DTW**	**Proposed approach**		
			Cosine	**Euclidean Squared**	**Euclidean**
Q-accuracy	47.2%	41.7%	**57.5%**	52.9%	56.7%
Q_4-accuracy	69.3%	66.2%	**81.1%**	77.9%	78.7%

Table 3: Experimental results for action "body weight squats" from the UCF101 dataset.

	GCTW	**DTW**	**Proposed approach**		
			Cosine	**Euclidean Squared**	**Euclidean**
Q-accuracy	65.2%	43.2%	67.8%	68.1%	**76.4%**
Q_4-accuracy	86.9%	69.6%	94.3%	93.2%	**95.7%**

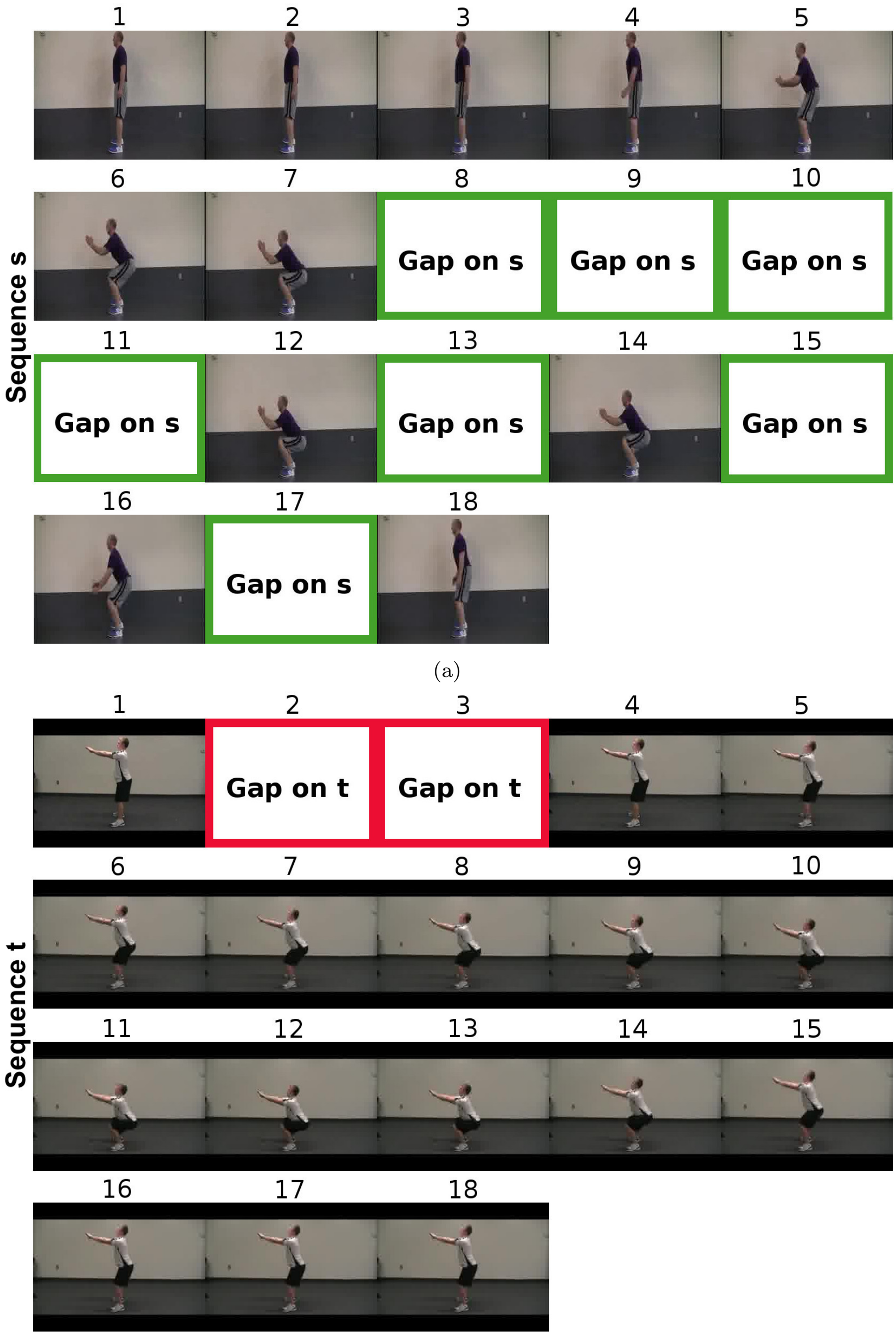
Table 4: Experimental results for action "stand up" from the MSR Daily Activity 3D dataset.

However, the proposed model has, again, outperformed DTW and GCTW also with the other distances. To further illustrate the results, Figure 3 shows a visual example of the alignment of two videos with the proposed model. The correspondence between key frames seems remarkably accurate. To give an idea of the computational times, training the proposed model on this training set has taken only 68.74 seconds on a PC with a 2.80 GHz i7-7600U CPU and 16 GB of RAM.

4.4 MSR Daily Activity 3D Dataset

This dataset is a popular activity dataset captured by a Microsoft Kinect device. It contains 16 activities of daily living performed by 10 subjects in two different poses: a standing position and a sitting position. The data included in this dataset comprise three channels: RGB, depth and "skeleton". The skeleton channel encodes 20 joints per frame as displayed in Table 5.a. Each joint is represented by its estimated real world coordinates (x, y, z) and its screen coordinates plus depth $(u, v,$ depth). The skeleton information is derived from the depth channel and is therefore naturally synchronized with it. The RGB channel runs instead on a separate thread and can be slightly shifted in time. For annotation, we have decided to annotate the alignment of the key frames using first the RGB channel, and then correct possible synchronization errors using the skeleton sequences. Table 5 shows an example of skeleton sequence superimposed, respectively, to its depth and RGB frames (rendered in gray-levels for greater clarity).

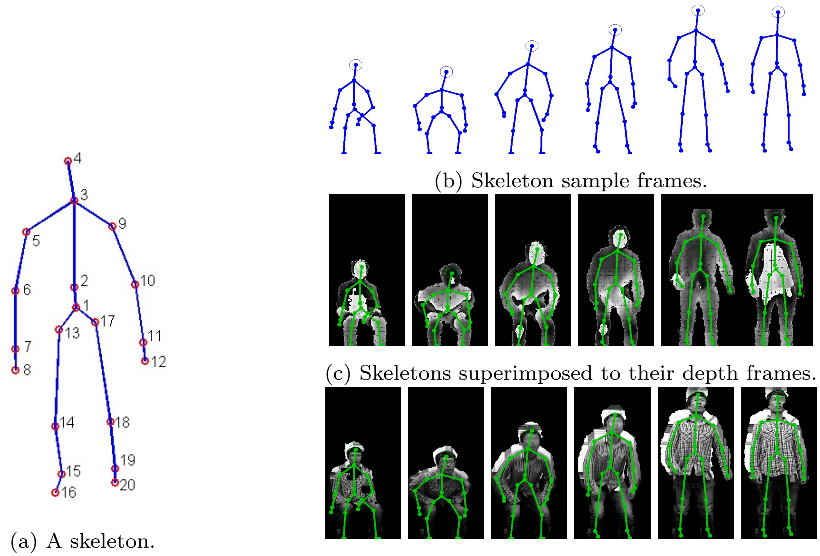
For this experiment, we have chosen action "stand up", manually annotating 45 pairs and using 22 as training set and 23 as test set. The 3D joint positions in the skeleton data were first preprocessed with the invariant features from [28]. Table 4 shows the alignment results for the "stand up" action. The proposed model with the Euclidean distance has outperformed DTW and



(a)

(b)

Fig. 3: Example of alignment obtained with the proposed approach for two videos of action “body weight squats” from UCF101.



(a) A skeleton.

(b) Skeleton sample frames.

(c) Skeletons superimposed to their depth frames.

(d) Skeletons superimposed to their RGB frames.

Table 5: Example data from the MSR Daily Activity 3D dataset.

GCTW, respectively, by 33.2 and 11.2 percentage points of Q-accuracy and 26.1 and 8.8 percentage points of Q_4-accuracy. Again, the proposed model has reported higher accuracies with all the distances. For a visual assessment, Table 6 shows an example of predicted skeletons from the proposed model and the compared algorithms, with arrows pointing to noticeable inaccuracies. The skeletons predicted by the proposed model look the closest to the ground truth.

5 Conclusion

In this paper, we have presented a novel approach for video alignment that can learn an optimal, minimum-risk model from training sets of manually-aligned video pairs. The approach integrates an extended hidden Markov model to provide the video alignments and the structural SVM framework to optimally train its parameters. The main contributions of our paper have been: a) a generalized linear scoring function suitable to score alignments paths; b) various distance functions to assess the dissimilarity of any two video frames; and c) two loss functions that gradually assess the quality of a predicted alignment against a given ground truth. In addition, since both the score and loss functions are decomposable frame-by-frame, we have been able to retain an efficient, dynamic programming approach for the loss-augmented inference required by structural SVM for training.

The proposed model has been tested over selected actions from four, popular action video datasets against a baseline algorithm (DTW) and a state-of-

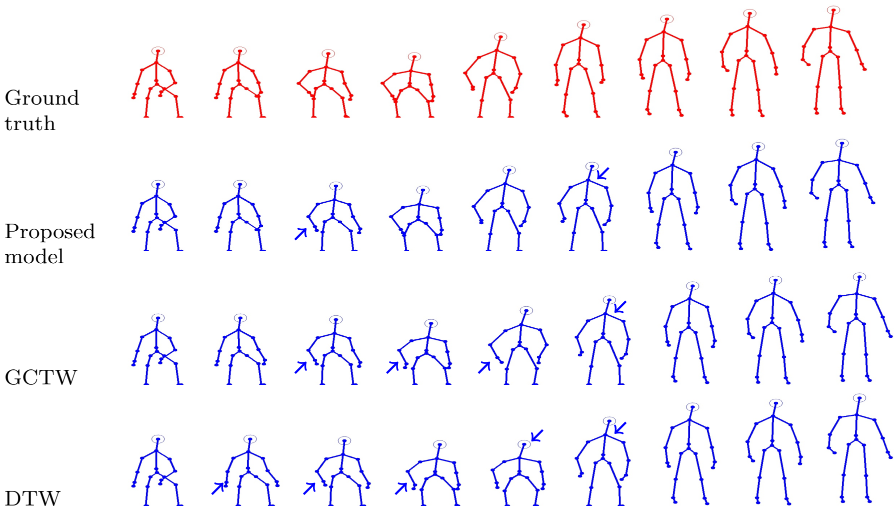


Table 6: Example of skeleton alignment from action "stand up" in the MSR Daily Activity 3D dataset. The arrows highlight noticeable inaccuracies.

the-art algorithm (GCTW [30]). In the experiments, we have used diversified measurements to characterize the frames of the various datasets, including PCA of background-subtracted pixel values, spatio-temporal descriptors, and skeletons. The experimental results can be regarded as very encouraging since the proposed model has outperformed the compared algorithms by a large margin in all experiments, both for the stricter and the more lenient alignment accuracies.

In the future, we plan to assess the ability of the proposed method to act as a generalized distance measurement between action videos and use it with minimum-distance classifiers for action classification. In addition, we will try to introduce a CCA step to perform the alignment in an optimized measurement subspace similarly to GCTW. Lastly, given that manual alignment of the training videos is significantly time-consuming, we will attempt to automatically detect the "key frames" in each video to facilitate the ground-truth annotation.

References

1. Cosine distance. http://reference.wolfram.com/language/ref/CosineDistance.html.
2. T. W. Anderson. *An Introduction to Multivariate Statistical Analysis.* Wiley, 1984.
3. Y. Bengio and P. Frasconi. An input output HMM architecture. In *Proceedings of the 7th International Conference on Neural Information Processing Systems (NIPS)*, pages 427–434, 1994.
4. D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of KDD-94, AAAI-94 Workshop on Knowledge Discovery in Databases*, pages 359–370, 1994.

5. E. Caiani, A. Porta, G. Baselli, M. Turiel, S. Muzzupappa, F. Pieruzzi, C. Crema, A. Malliani, and S. Cerutti. *Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume.* 1998.
6. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids.* Cambridge University Press, 1998.
7. D. Gong and G. G. Medioni. Dynamic manifold warping for view invariant action recognition. In *ICCV*, pages 571–578, 2011.
8. L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.
9. A. Gritai, Y. Sheikh, and M. Shah. On the use of anthropometry in the invariant analysis of human actions. In *17th International Conference on Pattern Recognition (ICPR'04)*, pages 923–926, 2004.
10. E. Hsu, K. Pulli, and J. Popović. Style translation for human motion. *ACM Trans. Graph.*, 24(3):1082–1089, July 2005.
11. T. Joachims. SVM struct. https://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html.
12. T. Joachims, T. Finley, and C. J. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.
13. T. Joachims, T. Galor, and R. Elber. Learning to align sequences: A maximum-margin approach. In *New Algorithms for Macromolecular Simulation, B. Leimkuhler, LNCS Vol. 49, Springer*, pages 57–69, 2005.
14. E. Keogh and M. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pages 239–241, 1998.
15. E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *Proceedings of First SIAM International Conference on Data Mining (SDM2001*, 2001.
16. I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8, 2008.
17. C. R. Maurer, R. Qi, V. Raghavan, and S. Member. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, 2003.
18. C. Myers, L. Rabiner, and A. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(6):623–635, 1980.
19. J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc. 11th European Conf. Comput. Vision*, pages 392–405, 2010.
20. L. Rabiner and B. Juang. *Fundamentals of Speech Recognition.* Prentice-Hall Signal Processing Series, Englewood Cliffs, New Jersey, 1993.
21. M. S. Ryan and G. R. Nudd. The Viterbi algorithm. Technical report, Coventry, UK, 1993.
22. H. Sakoe and S. Chiba. Readings in speech recognition. chapter Dynamic Programming Algorithm Optimization for Spoken Word Recognition, pages 159–165. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.
23. H. Skutkova, M. Vtek, P. Babula, R. Kizek, and I. Provaznik. Classification of genomic signals using dynamic time warping. *BMC Bioinformatics*, 14(S-10):S1, 2013.
24. K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.
25. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.
26. A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1469–1472, 2010.
27. Z. Wang and M. Piccardi. A pair hidden Markov support vector machine for alignment of human actions. In *Proceedings of the 2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 800–805, 2016.

28. Y. Wu. Mining actionlet ensemble for action recognition with depth cameras. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297, 2012.
29. F. Zhou. Software for canonical time warping. http://www.f-zhou.com/ta_code.html.
30. F. Zhou and F. De la Torre. Generalized canonical time warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):279–294, 2016.