Video temporal alignment for object viewpoint

Anestis Papazoglou[1], Luca Del Pero[1,2], and Vittorio Ferrari[1]

[1] University of Edinburgh [2] Blippar
a.papazoglou@sms.ed.ac.uk, luca.delpero@blippar.com, vittorio.ferrari@ed.ac.uk

Abstract. We address the problem of temporally aligning semantically similar videos, for example two videos of cars on different tracks. We present an alignment method that establishes frame-to-frame correspondences such that the two cars are seen from a similar viewpoint (e.g. facing right), while also being temporally smooth and visually pleasing. Unlike previous works, we do not assume that the videos show the same scripted sequence of events. We compare against three alternative methods, including the popular DTW algorithm, on a new dataset of realistic videos collected from the internet. We perform a comprehensive evaluation using a novel protocol that includes both quantitative measures and a user study on visual pleasingness.

1 Introduction

Temporal alignment of videos is often a key step in several popular tasks, such as video morphing [1], video mosaicking and stitching [2], video compositing [3], video summarisation [4], action recognition and video retrieval [5] and High Dynamic Range (HDR) video [6]. Much previous work on temporal alignment focuses on videos of the same scene recorded from multiple cameras [7–14]. Instead, we want to align videos that are only weakly related: we simply require that their main object belongs to the same semantic class. For example, two videos of different cars driving along different tracks, and backgrounds.

Our alignment method establishes frame-to-frame correspondences such that the two cars are seen from a similar viewpoint (e.g. facing right) while also enforcing temporal smoothness, i.e. we preserve the temporal order of the frames in the original videos as much as possible (fig.1). Our key intuition is that the object viewpoint is a good indicator of whether two individual frames showing different cars are aligned correctly. Temporal smoothness promotes consistency at a larger temporal scale (i.e. an entire left turn, fig. 1), which is more robust to noise in individual frames, and also makes the alignments more visually pleasing.

A few previous works [15–17] have tackled aligning semantically similar videos. However, they typically assume that the videos show a scripted sequence of events (e.g. drinking motion [17], hand waving [15]), possibly out of phase (*i.e.* the events occur at different, varying speeds). Under this assumption, finding an optimal alignment can be solved using Dynamic Time Warping (DTW) [18] (as in [17, 15]). However, this assumption is unrealistic for most real-world videos,

Fig. 1. ***Viewpoint-driven temporal alignment.*** *The goal of this task is to align the two videos so that both of them show the object from the same viewpoint frame-by-frame as shown above. This example alignment was produced by our method.*

where events may occur in a different order, or some occuring in only one of the videos.

Here, we present a method that is able to cope with such challenging videos. Our assumption is that we can decompose videos into contiguous temporal segments, and put them into correspondence so that each pair of corresponding segments (rather than the entire videos) show the same sequence of events (fig. 3). The main contribution of our approach is to solve the temporal segmentation and the correspondence problems jointly. For this, we use a principled probabilistic model defined over the space of all possible temporal segmentations and correspondences (sec. 3). A likelihood function promotes putting in correspondence segments showing similar viewpoints, while other components favour temporal consistency and smoothness. Inference in our model is a computationally intractable combinatorial problem. Therefore, we present a Markov Chain Monte Carlo (MCMC) sampling [19] procedure to search its complex parameter space efficiently (sec. 4).

We test our method on a set of 22 videos of cars racing in rally competitions collected from the internet, where we have manually annotated the viewpoint in each frame for evaluation (sec. 6.1). These videos are challenging, showing fast motion, complex backgrounds and different car models. We automatically split them into different shots using [20], but they are otherwise untrimmed and unedited. This is different from videos used in previous work [15–17], which are trimmed so that they show the exact same sequence of events in their entirety. We release this dataset at http://calvin.inf.ed.ac.uk/datasets/videoalignment.

In our videos, events are often in a different order and occur a different number of times. Hence, determining their optimal alignment can be ambiguous, i.e. we cannot define a unique ground-truth alignment as in [17]. For instance, if a certain viewpoint appears only once in a video and multiple times in the other, there are multiple valid ways of aligning them (e.g. fig. 2). To address this, we perform a comprehensive evaluation that takes into account several different factors: viewpoint similarity, temporal consistency and visual pleasingness. We evaluate these factors quantitatively on our dataset using a new carefully designed evaluation protocol, as well as with a substantial user study

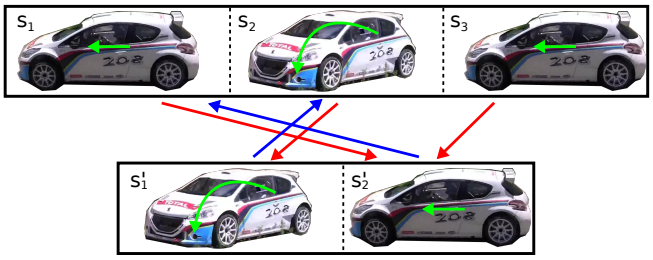

Fig. 2. *Example alignment between two videos. The first video shows the same events as the second (going straight, turning left) but in a different sequence. These videos cannot be aligned by just stretching and shrinking the time domain of the videos (as DTW does), but our method can cope with it.*

on visual pleasingness (in contrast to previous works that are mostly evaluated qualitatively on a few videos, *e.g.* [15, 16]). Our results show that our method is superior to three alternative alignment methods (sec. 6.2), including the popular DTW [18].

2 Related work

Previous works on temporal alignment can be categorised based on their assumptions about the input videos.

Videos of the same scene from different views. Most previous works, e.g. [7–12] focus on joint spatio-temporal alignment of videos of the same dynamic scene, recorded by two uncalibrated cameras placed at different viewpoints (typically stationary). [21] also attempts to spatio-temporally align videos of a single dynamic scene, but they jointly process videos from multiple cameras instead of just two. The work of [22] also assumes a single dynamic scene recorded by multiple cameras, but focuses on temporal alignment only.

Videos of the same scene at different times. A few works [23, 13, 14] focus on spatio-temporal alignment of videos of the same scene, but taken at a different time. To compensate for the lack of temporal overlap between the input videos, these works assume the cameras follows roughly the same trajectory.

Videos of semantically similar scenes. Our work falls in the category of temporal alignment of videos that do not show the same scene, but rather semantically similar content (e.g. drinking motion [17], hand waving [15]). Typically, the videos depict people performing some scripted sequence of actions, such as drinking or waving [15–17], and the goal is to put in correspondence frames showing the same body pose. These approaches typically align short videos showing the exact same sequence of events (possibly at different speed) and cannot handle challenging videos showing events in a different order like we do.

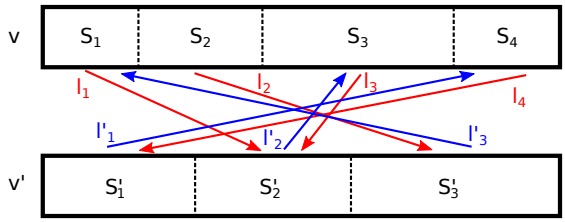

Fig. 3. *One possible configuration (sec. 3) of temporal segmentations S, S' and correspondences l between two videos v, v'. Each video is partitioned into a series of contiguous temporal segments: $S = \{s_1, s_2, s_3, s_4\}$ and $S' = \{s'_1, s'_2, s'_3\}$. Each segment has a correspondence in the other video, denoted by l (arrows). Note that the correspondences are not necessarily mutual (e.g. $s_2 \rightarrow s'_3$, but $s'_3 \rightarrow s_1$).*

Frame descriptors for video retrieval and classification. Some works [24–26] design good frame descriptors for video retrieval and classification. These methods do not consider aligning videos like we do (but [25] considers the somewhat related problem of recovering the temporal order of the jumbled frames of a single video). While the frame descriptor we use focuses on viewpoint similarity (sec.5), our video alignment formulation is general and can use any other frame descriptor. In the experiments (sec. 6.3) we compare our viewpoint descriptor to the descriptor from [25], which achieves state-of-the-art on several retrieval tasks by encoding temporal context.

3 Temporal alignment model

Our goal is to align two videos where different events may appear in a different order. Fig. 2 shows a simple example, featuring two types of events: going straight (s_1, s_3, s'_2) and turning left (s_2, s'_1). Ideally, we would like to match s'_1 to s_2 and s'_2 to either s_1 or s_3 (both would be valid). Note that the problem is not symmetric: when aligning the second video to the first, we would like to align s_1 to s'_2, s_2 to s'_1, and s_3 to s'_2 again. Aligning this example requires shuffling the temporal order of the videos, and re-using some of its segments.

An additional challenge is that the temporal segmentation of the videos into different events is also not known in advance. Our method solves the temporal segmentation and the segment correspondence problems jointly, using a single probabilistic model over the two tasks, which we now define formally.

Let v and v' be the two videos we want to align. $S = \{s_1, ..., s_N\}$ is the set of *contiguous* temporal segments composing v. The temporal segmentation S' of v' is defined analogously. The correspondence l_i indicates which segment from v' is matched to the i-th segment in v (l'_j is defined analogously; note that $l_i = j \implies l'_j = i$, fig. 3). The model parameters Θ include the temporal segments of both videos $\mathcal{S} = \{S, S'\}$ and the set $\mathcal{L}$ containing all correspondences (fig. 3). Note how both the segmentations S, S' and the correspondence $\mathcal{L}$ have a variable number of elements, as the number of segments in each video is not predefined. It is another parameter to be searched over during inference.

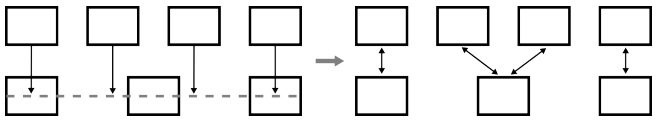

Fig. 4. *Our appearance distance d (4) measures the similarity in appearance between two segments of potentially different length (sec. 3). We first put the segment frames in one-to-one correspondence. For this, we project the longest segment onto the shorter one (top), and put each frame in the longest segment in correspondence with the frame closest to the projection (bottom). d is the distance in appearance averaged over all corresponding frames (4).*

We define the posterior distribution over the parameters to be

$$p(\mathcal{L}, \mathcal{S}|D) = p(\mathcal{L}|\mathcal{S}, D) \cdot p(\mathcal{S}|D) \tag{1}$$

where D are appearance descriptors extracted for all frames in the videos. Since we want to align the videos so that they show the same viewpoint, we use state-of-the-art CNN descriptors [27] which we specifically fine-tuned to classify different viewpoints (sec. 5). The two factors in the posterior compete to allow our model to find alignments which put similar viewpoints in correspondence, while also being temporally smooth. The *correspondence likelihood* $p(\mathcal{L}|\mathcal{S}, D)$ promotes putting into correspondence temporal segments (across videos) that are consistently similar in appearance through time. The *temporal segmentation likelihood* $p(\mathcal{S}|D)$ promotes having few temporal segments. Having too many segments can cause the alignment to look jerky due to the frequent segment switches over time, which is not visually pleasing. Furthermore, it promotes that each temporal segment is homogeneous in appearance (within a video). A homogeneous segment is likely to contain a single viewpoint, which makes it a good unit for matching across videos. We now discuss each factor in more detail.

Correspondence likelihood. We define the correspondence likelihood to be

$$p(\mathcal{L}|\mathcal{S}, D) = \prod_i p(l_i|\mathcal{S}, D) \cdot \prod_j p(l'_j|\mathcal{S}, D) \tag{2}$$

where each $p(l_i = k|\mathcal{S}, D)$ evaluates the likelihood of $l_i = k$ according to the appearance similarity of s_i and s'_k (these factors are conditionally independent). We define the probability of one correspondence l_i to be

$$p(l_i = s'_j|\mathcal{S}, D) \propto \exp\left(-\alpha_M \frac{||s_i||}{||v||} d(s_i, s'_j)\right) \tag{3}$$

where α_M is a scalar weight, $||s_i||$ is the length of segment s_i (*i.e.*, the number of frames in it), $||v||$ is the length of video v, and $d(s_i, s'_j)$ denotes the appearance distance between the segments s_i and s'_j.

We designed d so that it can evaluate whether the appearance of the segments is consistently similar through time. Since the segments can have different length

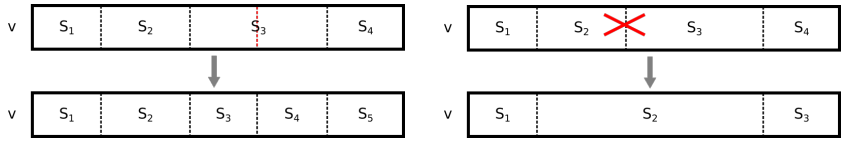

Fig. 5. (Left) Split move (sec. 4). In this example, temporal segment s_3 is split in half, creating two new segments. **(Right)** Merge move (sec. 4). In this example, temporal segments s_2 and s_3 are merged into a single segment.

(*i.e.*, different speed), we first put their frames in one-to-one correspondence, denoted by $(f \rightarrow f')$ (see fig. 4). We can now compute

$$d(s_i, s'_j) = \frac{\sum_{f \rightarrow f'} \mathrm{a}(f, f')}{\max(||s_i||, ||s'_j||)} \tag{4}$$

where $\mathrm{a}(f, f')$ denotes the appearance distance between frames f and f' (sec. 5). Note that DTW [18] is a reasonable alternative segment distance, as it also measures similarity through time. However, we found that d produces comparable results to DTW, while being computationally more efficient.

Temporal segmentation likelihood. The temporal segmentation likelihood $p(\mathcal{S}|D)$ promotes having a small number of segments that are homogeneous in terms of appearance.

$$p(\mathcal{S}|D) \propto \exp\left(-\alpha_T \sum_i \frac{||s_i||}{||v||} \Delta_i\right) \exp\left(-\alpha_P (||S||^2 + ||S'||^2)\right) \tag{5}$$

where α_T, α_P are scalar weights, Δ_i is the appearance distance a averaged over all pairs of frames within s_i (sec. 5), and $||S||$ and $||S'||$ are the number of segments in v and v', respectively. Note that we can compute Δ_i in constant time by using summed area tables [28]. The ratio $\frac{||s_i||}{||v||}$ ensures that the contribution of each temporal segment is proportional to its length.

The first factor in eq. 5 promotes segments that are homogeneous in terms of appearance. The second factor acts as a prior, promoting having a small number of segments. These two factors and the correspondence likelihood compete in order to strike a balance on the optimal number of segments. On one hand, having many short segments results in a high $p(\mathcal{S}|D)$, which is trivially maximised when each frame forms its own segment (which is maximally homogeneous in appearance). In this limit case, $p(\mathcal{L}|\mathcal{S}, D)$ reduces to a nearest-neighbour matching between individual frames in the two videos, which results in a low average appearance distance, but is also sensitive to noisy appearance descriptors. On the other hand, having a few long segments brings temporal smoothness and produces a more visually pleasing alignment. However, having corresponding segments that show the same sequence of viewpoints is more unlikely.

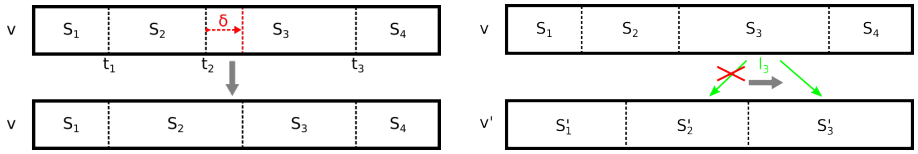

Fig. 6. (Left) Perturbation move (sec. 4). In this example, we propose moving the delimiter t_2 between temporal segments s_2 and s_3 by an offset δ (with the constraint that $t_1 < (t_2 + \delta) < t_3$). **(Right)** Correspondence move (sec. 4). In this example, the correspondence l_3 for temporal segment s_3 (green arrow) changes from s'_2 to s'_3.

4 Inference

Maximising the posterior (eq. 1) with respect to Θ is a hard combinatorial problem, since we jointly optimise over the number of segments N, N', the position of their delimiters S, S', as well as the set of correspondences $\mathcal{L}$. Furthermore, the posterior (eq. 1) is a complex distribution which we cannot evaluate analytically. Thus, we use Markov Chain Monte Carlo (MCMC) sampling [19] to search the parameter space.

Following the standard formulation, at each iteration we propose a new sample Θ' from the current sample Θ using a proposal distribution $q(\Theta'|\Theta)$. Θ' is then accepted with probability

$$A(\Theta', \Theta) = \min\left(1, \frac{p(\Theta'|D) \cdot q(\Theta'|\Theta)}{p(\Theta|D) \cdot q(\Theta|\Theta')}\right) \tag{6}$$

If Θ' is accepted, it becomes the current sample, otherwise we keep Θ. Our proposal distribution uses four different kind of moves, each sampling over a subset of Θ. For each move, we change a single model parameter while keeping all other parameters fixed. Finally, we select the sample with the highest posterior (eq. 1) as our output.

Perturbation move. We define a perturbation as changing the position of one of the current delimiters t by an offset δ (fig. 6 left). We construct Θ' from the current sample with $f(\Theta, t, \delta)$, which replaces t with $t + \delta$. We choose (t, δ) from the space of all possible perturbations (t', δ') conditioned on the current positions of the delimiters in Θ. For this, we sample from

$$q_P(t, \delta|\Theta) = \frac{p(f(\Theta, t, \delta)|D)}{\sum_{(t', \delta')} p(f(\Theta, t', \delta')|D)} \tag{7}$$

Merge and split move. The merge move proposes merging a pair of subsequent segments into a single one (fig. 5 right). We select which segments to merge from all possible merges given the delimiters in the current sample, using a proposal constructed analogously to (7). The complementary split move proposes splitting a segment in half, yielding two segments (fig. 5 left). Both merge and split moves change the number of segments in a video. We note that

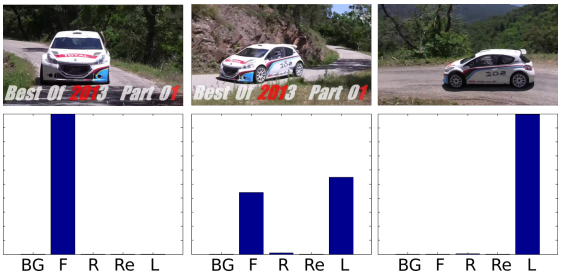

Fig. 7. *Three example outputs of the softmax layer from cars in* 90*,* 150 *and* 180 *degree viewpoints respectively. The labels correspond to background (BG), front* 90 *degrees (F), right* 0 *degrees (R), rear* 270 *degrees (Re) and left* 180 *degrees (L).*

merge/split moves are quite a standard tool in MCMC methods for solving association problems, for example in the domain of tracking multiple objects (*e.g.* MCMCDA [29] or [30]).

Correspondence move. This move chooses a segment s_i in a video and proposes to change its matching segment in the other video (*i.e.*, it changes l_i, fig. 6 right). We choose s_i and the new value for l_i from all possible alternatives given the current segmentation $\mathcal{S}$. Again, we use a proposal constructed analogously to (7).

The way we constructed the proposals above increases the acceptance ratio of the moves, which improves mixing. For example, choosing (t, δ) in the perturbation move from a uniform distribution would result in a low acceptance ratio (which significantly improves using q_P). Note that, while our proposals need to compute a large number of posteriors during each move, they can still do it efficiently since most of the terms are shared between these computations, and need to be computed only once.

Initialisation. Starting from a Θ sample with a reasonably good posterior reduces the amount of time wasted in regions of low probability at the beginning of the sampling process (compared to random initialisation). We begin by individually decomposing each video into homogeneous temporal segments, without considering any correspondences. This is achieved by optimising the temporal segmentation likelihood $p(\mathcal{S}|D)$ using just perturbation, merge and split moves. Since there is no correspondence likelihood involved, $p(\mathcal{S})p(\mathcal{S}|D)$ can be optimised independently for each video efficiently.

Having the initial temporal segmentation $\mathcal{S}$, we then find the optimal correspondence between these segments. This corresponds to optimising the correspondence likelihood $p(\mathcal{L}|\mathcal{S}, D)$. Since the correspondences are conditionally independent under our model, we can find the exact optimal set of correspondences with a nearest neighbour approach.

Fig. 8. ***A visualisation of the segmentation pipeline*** *We segment the cars in the videos using video foreground segmentation [34] (sec. 5). Here we show: object proposals from selective search [35] (top left), the pixel-wise probability map produced by the car detector [36] (top right), the resulting segmentation (bottom left), and the bounding box of the segmentation is used to extract the viewpoint descriptor (bottom right).*

5 Appearance descriptors

We now discuss the appearance distance $\mathrm{a}(f, f')$ that we use to compute the distance between frames as part of our likelihood (sec. 3). We designed it to capture the difference in viewpoint between two frames.

Appearance distance. Modern appearance classifiers based on Convolutional Neural Networks (CNN) are state-of-the-art for whole image classification [31] and object detection [27]. However, they are optimised to differentiate between objects of different classes, and they actually strive for invariance to viewpoint differences. Therefore they are not ideal for our problem. Instead, we train a CNN classifier, based on the AlexNet architecture [31], to distinguish among viewpoints of the car class (which we use in our experiments). We start from a CNN pre-trained for image classification on the ILSVRC 2012 [31, 32] and fine-tune it to classify 4 car viewpoints (front, left, right, rear) and the background. As training data we use the PASCAL VOC 2012 [33] dataset which has car images with viewpoint annotations. In order to focus on the appearance of the cars and not on the background, we crop the cars from the image using the provided bounding-box annotations.

After training, we apply the CNN viewpoint classifier on a video frame and use the output of the softmax layer as our frame descriptor (i.e. a 5D vector, summing to 1). The intuition is that if the viewpoint of the input frame matches one of the training viewpoints from PASCAL VOC closely, the softmax output vector will be peaked on one of the 4 viewpoints. Instead, if the viewpoint of the frame lies in-between the training viewpoints, the output probability mass should be spread between two viewpoints (fig. 7). We then define the distance $\mathrm{a}(f, f')$ between two frames as the histogram intersection of their frame descriptors.

Object localisation in the videos. To compute the viewpoint descriptor on a video frame, we first need to localise the car up to a bounding-box. This focuses the descriptor on the car and matches the kind of data the CNN was trained on.

We start from the video segmentation technique of [34], which can handle unconstrained video and reliably segments objects even under rapid motion and against cluttered backgrounds. This method uses a spatio-temporal Markov Random Field with unary potentials derived from motion, and a pairwise potential enforcing spatial and temporal smoothness. Here, we add a unary potential de-

rived from a car detector trained on PASCAL VOC 2012 dataset [33]. While [34] is class agnostic and segments arbitrary foreground objects, the car detector injects domain-specific knowledge to anchor the segmentation on cars.

More precisely, for each video frame we extract object proposals using Selective Search [35] and score them with an R-CNN [27] car detector pre-trained on PASCAL VOC. Next, we score each pixel by the sum of the scores of all the proposals containing it. This results in a pixel-wise 'heatmap' that we use as the additional unary potential (fig. 8).

We evaluated the method on our dataset using the CorLoc [37] performance measure as in [34]. Adding the car detector performs significantly better than [34], which is class agnostic with (88.5% accuracy versus 73.2%) respectively. Furthermore, the class detector [27] alone achieves just 69.8%, which shows that using video object segmentation can significantly improve the accuracy over using just a detector.

6 Experimental evaluation

In this section we first introduce the data used for evaluation (sec 6.1). Second, we present the methods we compare against (sec. 6.2). Next, we present our evaluation protocol (sec. 6.3) and finally discuss our results (sec 6.4).

6.1 Data

We assembled a novel dataset of 22 video sequences depicting cars on racing sequences collected from YouTube, each 5-30 seconds long. These videos are challenging, showing different cars in different races, with fast motion, fast moving camera and cluttered backgrounds.

Our data contains viewpoint annotations for each frame. For this, we use an annotation protocol that reduces the amount of manual effort as follows. We first define a set of 16 canonical viewpoints, spaced by 22.5 degrees (starting from full frontal). We manually annotate all the frames showing one of them. Then, we automatically annotated the rest of the frames by linearly interpolating the manual annotations.

We evaluate our model on a pair of videos v, v' only if at least 50% of frames in v show a viewpoint that also appears in v' (up to a difference of 10 degrees). This leads to 251 pairs of videos out of the 484 possible pairs. We plan to release this dataset along with the ground truth annotations upon acceptance.

6.2 Alternative methods

We compare our model against three alternatives: a nearest neighbour model, an MRF model promoting temporal smoothness, and Dynamic Time Warping (DTW). Note that as input, all methods use the same appearance distance a($f, f\prime$) used in our model.

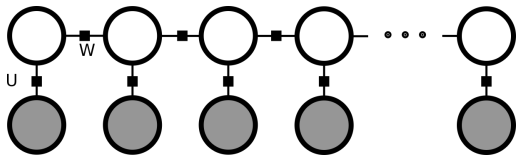

Fig. 9. The MRF model corresponding to eq. 9. Each hidden node (in white) corresponds to a single frame in video v. The observable states (in grey) correspond to the frames in video $v\prime$.

Nearest neighbour. The nearest neighbour model matches each frame in v to its closest neighbour in $v\prime$ according to the appearance distance a (sec. 5). This simple model has no notion of temporal smoothness and it allows us to verify what we can achieve using appearance alone.

This approach corresponds to the following model:

$$\mathcal{L}^* = \underset{\mathcal{L}}{\operatorname{argmin}} \sum_f U(l_f) \tag{8}$$

where U is a unary cost of matching frame f in v to a frame $f\prime$ in $v\prime$. Note that in this case, l denotes the correspondence between individual frames instead of entire temporal segments. We set U between $f, f\prime$ to be equal to their appeance distance a$(f, f\prime)$.

MRF model. As a second method, we augment the nearest neighbour model by adding temporal smoothness between subsequent frames. For this we use a Markov Random Field (MRF) with pairwise terms that promote that consecutive frames in v are in correspondence with frames in v' that are also close in time (fig. 9). More precisely, we solve the following optimisation problem:

$$\mathcal{L}^* = \underset{\mathcal{L}}{\operatorname{argmin}} \sum_f U(l_f) + \alpha_W \sum_f W(l_f, l_{f+1}) \tag{9}$$

where α_W is a weight term, and U is the same unary term used in the nearest neighbour model. We define the pairwise potential W to be:

$$W(l_f, l_{f+1}) = (|l_f - l_{f+1}| - 1)^2 \tag{10}$$

Dynamic Time Warping. Dynamic Time Warping (DTW) [18] is a popular sequence alignment algorithm. Assuming that the two sequences show the same event transition with the only variable being the speed of each, DTW can compute an optimal alignment between them. However, this assumption does not necessarily hold true for realistic video sequences. As input, we used the appearance distance a to compute the distance between individual frames.

6.3 Evaluation protocol

Comparative user study. We performed a substantial user study to verify that our results are indeed visually more appealing to humans compared to the

Method	NN	MRF	DTW	ours
NN	-	12.7	34.8	9.1
MRF	87.3	-	47.2	15.6
DTW	64.2	52.8	-	23.4
ours	**90.9**	**84.4**	**76.6**	-

Table 1. Comparative user study. *The table shows the comparative results between the different methods: nearest neighbour (NN), the MRF model, DTW and our model. The value in a cell shows the percentage of videos for which the participants preferred the method of the corresponding row over that of the column. For example, the participants preferred our method over the nearest neighbour approach for* 90.9% *of the videos.*

alternative methods. We performed a "blind taste test" in which participants are presented with the same pair of sequences aligned by two different methods, and asked which alignment they think is better, *i.e.*, it is consistent in terms of viewpoint and also looks realistic. In our setup the participant is shown an original video and how it was aligned to a second video by two different alignment methods. The original video is displayed in the centre of the screen, the two alignments are on each side, being played simultaneously (we randomly choose on which side we put them). The participant then has to decide which one they think is better. Note that we never reveal to the user which method was used to produce the alignments we display.

We use this protocol to compare all of the alignment methods in pairs (*e.g.* our full method vs DTW, DTW vs MRF, etc.). We ensure that each pair of methods is shown to at least 3 different participants for each pair of videos and we aggregate the results (table 1).

Quantitative criteria. We identify several properties that correspond to what humans perceive as attractive alignments. First, frames in correspondence should be displaying the same viewpoint. Second, the viewpoint transitions should be temporally consistent. Third, long sequences of correct correspondences are preferable. Based on this observation, we propose two measures to evaluate an alignment quantitatively:

Percentage of correct correspondences: This measures the percentage of frames that are in correct correspondence, *i.e.* difference in viewpoint is 22.5 degrees or less (which is equal to the spacing we use to manually annotate keypoints, sec. 6.1), and the difference in the viewpoint derivative is 5 degrees/frame or less. Intuitively, the viewpoint difference ensures that aligned frames show the same viewpoint, while the derivative difference ensures that the viewpoint transition is smooth.

Longest correct sequence: This measures the length of the longest sequence of correct correspondences in each video, normalised by the length of the video. Intuitively, alignments that are correct for large periods of time are visually preferable to alignments with alternating correct and erroneous correspondences.

Quantitative criteria vs user study. We analyse how well these two evaluation criteria capture what humans perceive as a good temporal alignment, by verifying how accurately they can predict the results of the user study. We

Fig. 10. ***Qualitative results.*** *Each pair of rows (a-d) shows an original video (top) and a second video aligned to it by our method (bottom).*

do this as follows. Given two methods, we predict that the user will choose the alignment found by the method that scores higher according to the evaluation criteria. We then report the *prediction accuracy* of each criterion, *i.e.* the number of times the prediction made using that criterion is correct, averaged over all possible pairs of methods and videos (table. 3).

6.4 Results and discussion

Comparative user study. Table 1 shows the results of the user study. The value in a cell shows the percentage of videos for which the participants preferred the method of the corresponding row over that of the column. Our method substantially outperforms all three alternative methods (last row).

The nearest neighbour model produces very jittery alignments, as it does not enforce any temporal smoothness. As a consequence, the participants do not find the output visually pleasing. Thanks to pairwise temporal smoothness, the MRF model partly alleviates this problem. However, the smoothness is promoted only at a local level (between consecutive frames). Hence, the MRF is unable to capture smooth transitions of viewpoints on a larger time scale. Instead, our model enforces smoothness at the level of temporal segments, leading to large, piece-wise smooth alignments.

	Correct correspondence %		Longest correct sequence	
	our descriptor	descriptor [25]	our descriptor	descriptor [25]
NN	29.8	16.8	9.4	8.7
MRF	46.0	18.9	27.5	12.7
DTW	40.8	15.8	26.4	11.7
our alignment model	**47.8**	20.0	**30.3**	13.6

Table 2. Quantitative results. *Comparison of the different video alignment methods: nearest neighbour (NN), MRF model, DTW and our model. The first two columns show the percentage of correct correspondences when we use our appearance descriptor (sec. 5) and when we use the descriptor from [25]. The next two columns show performance on longest correct sequence (sec. 6.3).*

	Human agreement
Correct correspondence %	70.7
Longest correct sequence	77.7

Table 3. Evaluation of criteria. Each value corresponds to the accuracy of a quantitative criterion when trying to predict the results of our user study (sec. 6.3).

Interestingly, the participants clearly prefer DTW over the nearest neighbour model, but results are comparable with respect to the MRF model. As mentioned before, DTW makes the strong assumption that both videos show the exact same sequence of events, possibly occurring at varying speeds. When this assumption holds, DTW can produce an optimal temporal alignment, and the participants prefer it over MRF. However, in the scenario where this assumption does not hold, the participants consistently prefer MRF. They however clearly prefer our method over both DTW and MRF, as our model can handle both scenarios thanks to the temporal segmentation.

Quantitative criteria. Table 2 shows the performance of each method according to our two quantitative criteria (sec. 6.3) using our descriptor (sec. 5) and the state-of-the-art descriptor [25]. Our alignment method outperforms all of the alternatives for both criteria and both descriptors. Moreover, our descriptor outperforms [25] on both criteria, probably because object viewpoint is a powerful cue for our task. In contrast to the user study results, DTW performs significantly worse than the MRF model with respect to percentage of correct correspondence. This indicates that humans tend to prefer smoother alignments, even if the aligned frames exhibit a larger difference in viewpoint. While the quantitative performance difference between the MRF model and ours is rather mild, users prefer our method over the MRF on 84% of the videos, showing that our alignments are much more visually pleasing. Fig. 10 shows some qualitative results produced by our method.

Quantitative criteria vs user study. Table 3 shows an analysis of how well our evaluation criteria can predict what humans perceive as visually pleasing alignments. As can be seen from the results, both criteria show a strong correlation to what the participants prefer, in particular the longest correct sequence.

References

1. Liao, J., Lima, R.S., Nehab, D., Hoppe, H., Sander, P.V.: Semi-automated video morphing. In: Eurographics Symposium on Rendering. (2014)
2. Agarwala, A., Zheng, K.C., Pal, C., Agrawala, M., Cohen, M., Curless, B., Szeliski, R.: Panoramic video textures. In: SIGGRAPH. (2005)
3. Ruegg, J., Wang, O., Smolic, A., Gross, M.: Ducttake: Spatiotemporal video compositing. Comput. Graphics Forum (Proc. Eurographics) **32** (2013)
4. Ngo, C., Ma, Y., Zhang, H.: Video summarization and scene detection by graph modeling. IEEE Transactions on Circuits and Systems for Video Technology **15** (2005)
5. Jiang, Y., Ngo, C., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: International Conference on Image and Video retrieval. (2007)
6. Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High dynamic range video. ACM Transactions on Graphics (2007)
7. Caspi, Y., Irani, M.: A step towards sequence-to-sequence alignment. In: CVPR. (2000)
8. Caspi, Y., Irani, M.: Spatio-temporal alignment of sequences. IEEE Trans. on PAMI (2002)
9. Caspi, Y., Irani, M.: Alignment of non-overlapping sequences. In: ECCV. (2001)
10. Caspi, Y., Simakov, D., Irani, M.: Feature-based sequence-to-sequence matching. IJCV **68** (2006) 53–64
11. Wolf., L., Zomet, A.: Wide baseline matching between unsynchronized video sequences. IJCV (2006)
12. Tuytelaars, T., van Gool, L.: Synchronizing video sequences. In: CVPR. (2004)
13. Evangelidis, G.D., Bauckhage, C.: Efficient subframe video alignment using short descriptors. IEEE Trans. on PAMI (2013)
14. Wang, O., Schroers, C., Zimmer, H., Gross, M., Sorkine-Hornung, A.: Videosnapping: Interactive synchronization of multiple videos. ACM Transactions on Graphics (2014)
15. Rao, C., Gritai, A., Shah, M.: View-invariant alignment and matching of video sequences. In: ICCV. (2003)
16. Ukrainitz, Y., Irani, M.: Aligning sequences and actions by maximizing space-time correlations. In: ECCV. (2006)
17. Dexter, E., Perez, P., Laptev, I.: Multi-view synchronization of human actions and dynamic scenes. In: BMVC. (2009)
18. Sakoe, H., Chiba, S.: Object segmentation by alignment of poselet activations to image contours. In: IEEE Trans. Acoustics, Speech, and Signal Proc. (1978)
19. Neal, R.M.: Probabilistic inference using markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto (1993)
20. Kim, W.H., Kim, J.N.: An adaptive shot change detection algorithm using an average of absolute difference histogram within extension sliding window. In: ISCE. (2009)
21. F. L. C. Padua, R.L.C.: Linear sequence-to-sequence alignment. IEEE Trans. on PAMI (2009)
22. Douze, M., Revaud, J., Verbeek, J., Jegou, H., Schmid, C.: Circulant temporal encoding for video retrieval and temporal alignment. IJCV (2016)
23. Diego, F., Serrat, J., Lpez, A.M.: Joint spatio-temporal alignment of sequences. In: IEEE Transactions on Multimedia. (2013)

24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv **abs/1409.1556** (2014)
25. Ramanathan, V., Tang, K., Mori, G., Fei-Fei, L.: Learning temporal embeddings for complex video analysis. In: ICCV. (2015)
26. Zha, S., Luisier, F., Andrews, W., Srivastava, N., Salakhutdinov, R.: Exploiting image-trained cnn architectures for unconstrained video classification. In: BMVC. (2015)
27. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)
28. C., F.: Summed-area tables for texture mapping. In: SIGGRAPH. (1984)
29. Oh, S., Russell, S.J., Sastry, S.: Markov chain monte carlo data association for multi-target tracking. IEEE Transactions on Automatic Control **54** (2009) 481–497
30. Brau, E., J., G., Simek, K., Del Pero, L., Dawson, C.R., Barnard, K.: Bayesian 3d tracking from monocular video. In: ICCV. (2013)
31. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)
32. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L.: ImageNet large scale visual recognition challenge. IJCV (2015)
33. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (2012)
34. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: ICCV. (2013)
35. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV (2013)
36. Girshick, R.: Fast R-CNN. In: ICCV. (2015)
37. Prest, A., Leistner, C., Civera, J., Schmid, C., Ferrari, V.: Learning object class detectors from weakly annotated video. In: CVPR. (2012)