

Slice Matching for Accurate Spatio-Temporal Alignment

Georgios Evangelidis, Ferran Diego, Joan Serrat, Antonio López

▶ **To cite this version:**

Georgios Evangelidis, Ferran Diego, Joan Serrat, Antonio López. Slice Matching for Accurate Spatio-Temporal Alignment. ICCVW - Workshops of International Conference on Computer Vision - 2011, Nov 2011, Barcelona, Spain. pp.1941-1946, 10.1109/ICCVW.2011.6130486 . hal-00936021

HAL Id: hal-00936021

https://inria.hal.science/hal-00936021v1

Submitted on 24 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Slice Matching for Accurate Spatio–Temporal Alignment

Georgios D. Evangelidis[1] Ferran Diego[2] Joan Serrat[2] Antonio M. López[2]
[1]Dept. Computer Engineering & Informatics
University of Patras
26500 Rio-Patras, Greece
evagelid@ceid.upatras.gr

[2]Dept. Ciencies Computador
Computer Vision Center
Universitat Autonoma de Barcelona, Spain
{fdiego,joans,antonio}@cvc.uab.es

Abstract

Video synchronization and alignment is a rather recent topic in computer vision. It usually deals with the problem of aligning sequences recorded simultaneously by static, jointly– or independently–moving cameras. In this paper, we investigate the more difficult problem of matching videos captured at different times *from independently–moving cameras, whose trajectories are approximately coincident or parallel. To this end, we propose a novel method that pixel-wise aligns videos and allows thus to automatically highlight their differences. This primarily aims at visual surveillance but the method can be adopted as is by other related video applications, like object transfer (augmented reality) or high dynamic range video. We build upon a slice matching scheme to first synchronize the sequences, while we develop a spatio-temporal alignment scheme to spatially register corresponding frames and refine the temporal mapping. We investigate the performance of the proposed method on videos recorded from vehicles driven along different types of roads and compare with related previous works.*

1. Introduction

Video alignment aims to relate two video sequences in both their spatial and temporal dimensions so that they can be compared pixel–wise. By designating one of them as *observed* and the other as *reference* sequence, video alignment consists of mapping the reference spatio-temporal coordinates to the observed ones. That mapping thus decomposes in a temporal and a spatial component (Fig. 1). The former, or synchronization, estimates a frame correspondence $(t_{i,o}, t^*_{i,r})$ that associates the frame $t^*_{i,r}$ in the reference sequence to the frame $t_{i,o}$ in the observed sequence. Once the temporal mapping has been estimated, the latter, usually called image registration, estimates a geometric transformation that provides dense matches of corresponding frames. Both mappings usually count on optimizing

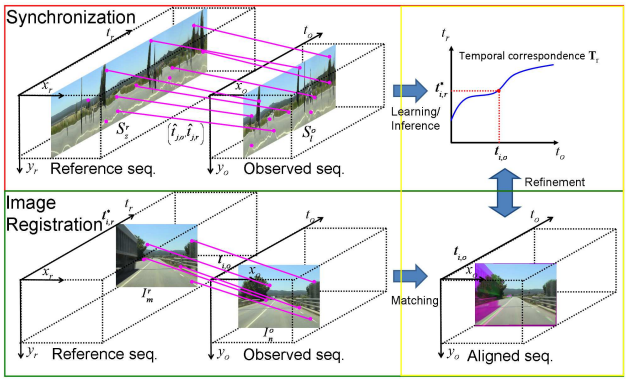


Figure 1. Slice matching provides temporal correspondences by analogy to image matching that provides spatial correspondences.

an appropriate measure. The same problem is addressed by various computer vision applications like wide baseline matching [3, 17], high dynamic range video and video matting [13], action recognition and sensor fusion [16], difference spotting [5], video-copy detection [2] and frame dropping prevention [10].

Static or jointly moving cameras come with a fixed inter-sequence geometric transformation (i.e. homography), while simultaneous recording amounts to a fixed temporal mapping across the sequences [3, 17, 10, 16, 9, 12] like constant time-offset and frame rate ratio. But when video acquisition takes place at different times, from independently moving cameras following nearly coincident trajectories [13, 5], the problem becomes much more challenging, as the above constancy for both spatial and temporal mapping is not active.

The latter scenario is what we consider in this paper. Specifically, we propose a novel video alignment method based on slice[1] matching to align sequences recorded at different times from independently moving cameras whose trajectories can be more or less coincident *or* parallel. The

[1]A slice is defined as a 'cut' of a video sequence seen as a spatio-temporal volume in a XYT system, normal to the X and parallel to the Y axis for vertical Y–T slices, and the opposite for X–T slices.

key idea of the algorithm is to exploit the analogy between image matching and slice matching (Fig. 1). As image matches are processed to obtain the geometric transformation between images, slice matching provides temporal correspondences that can be further processed towards temporal mapping. Although image matching via retrieval can also lead to synchronization [7, 2], by slice matching we work directly on time domain. Furthermore, putative matches are viewed as samples of a frame correspondence pdf described by a Gaussian mixture model (GMM). The GMM parameters are learnt through a *Maximum Likelihood Estimation*. We then formulate the temporal mapping estimation as a *maximum a posteriori* (MAP) inference problem based on probabilities extracted by the learnt pdf, in contrast to [5] where they are estimated empirically.

Unlike [3], independently moving cameras imply that each frame, or at most short-time sequence, must be separately registered in space, as well as that each observed (reference) frame ideally corresponds to a reference (observed) subframe. Thus, instead of spatially registering corresponding single frames [5], a spatio-temporal alignment applies to short subsequences in turn towards spatial registration and synchro refinement. A common choice for this would be the spatio-temporal extension of the Lucas-Kanade algorithm [3]. However, different recording times come with variant illumination and outliers. To handle the former we extend in time the recently proposed ECC image alignment algorithm [6] that offers robustness to appearance variation.

1.1. Related works

Caspi and Irani [3] present video alignment solutions for static or jointly moving cameras. They align feature trajectories, as [11, 15, 17] also do, or register direct the whole intensity manifolds, in order to estimate homographies or fundamental matrices and affine temporal models. Tuytelaars and VanGool [15] consider moving cameras that capture the same event and synchronize the videos by registering backprojected lines. Our work is more closely to [13, 5] where different recording times are supposed. Sand and Teller [13] propose an exhaustive search between frames looking for motion-consistent pixel matches, while Diego *et al*. [5] globally solve the temporal mapping by fusing the information obtained from camcorders and GPS receivers. In the context of video alignment, Liu *et al*. [7] recently proposed a dense alignment scheme for retrieving and registering still images from different scenes. That solution easily adapts to our problem by considering each observed frame as a query and the same goes for [2] where video copy detection is addressed as retrieval based on the standard bag-of-keypoints paradigm [14]. Finally, Pundik and Moses [10] solve the synchronization problem by exploiting temporal signals along epipolar lines, but considering static cameras.

2. A slice matching to video synchronization

Suppose we are given two spatio-temporal volumes that are represented with an observed and reference image sequence, let $\mathcal{I}^o = \{I^o_n(x_o, y_o)\}_{n=1}^N$ and $\mathcal{I}^r = \{I^r_m(x_r, y_r)\}_{m=1,}^M$ respectively, where N, M are their number of frames. These volumes can be also represented by YT slice sequences, $\{S^o_l(y_o, t_o)\}_{l=1}^L$ and $\{S^r_z(y_r, t_r)\}_{z=1}^Z$, being L, Z the width of observed and reference frames, respectively (Fig. 1, Fig. 2).

Our scenario involves sequences recorded at different times from cameras that follow similar or approximately parallel trajectories. Lateral displacements like those due to lane changes lead to partial overlap in the camera field of view. As the speed of cameras can vary, we need a non–parametric model to describe the temporal mapping from an observed frame t_o to a reference frame $t_r = f_t[t_o]$, with $t_o = 1, \ldots, N$ and $f_t : \{1, \ldots, N\} \rightarrow \{1, \ldots, M\}$ a discrete mapping. To this end, the estimation of temporal mapping is posed here as a MAP Bayesian inference problem:

$$\mathbf{T}^*_r = \underset{\mathbf{T}_r \in \mathcal{M}}{\operatorname{argmax}}\, p(\mathbf{T}_r | \mathbf{T}_o; \mathcal{I}^r, \mathcal{I}^o), \tag{1}$$

where $\mathbf{T}^*_r = (t^*_{1,r}, \ldots, t^*_{i,r}, \ldots, t^*_{N,r})$ is the most likely image indexes of f_t given the input $\mathbf{T}_o = (1, \ldots, t_{i,o}, \ldots, N)$, $t^*_{i,r}$ is the reference index that corresponds to the $t_{i,o}$ observed index frame, $\mathbf{T}_r$ is a sequence of N random variables and $\mathcal{M}$ is the set of all possible temporal mappings. The posterior probability distribution $p(\mathbf{T}_r | \mathbf{T}_o; \mathcal{I}^r, \mathcal{I}^o)$ describes the probability that the observed frames $\mathbf{T}_o$ correspond to the frames $\mathbf{T}_r$ in the reference sequence. For conciseness, from now on the arguments $\mathcal{I}^r, \mathcal{I}^o$ are omitted. The most likely temporal alignment between the observed and reference sequences is inferred by optimizing Eq. (1). For simplicity, each random variable $t_{i,r}$ is conditionally independent given their respective observed frame $t_{i,o}$. Hence, $p(\mathbf{T}_r | \mathbf{T}_o)$ decomposes as the product of frame correspondence probabilities $p(t_{i,r} | t_{i,o})$ for all frames in the observed sequences. Therefore, the most likely temporal alignment in Eq. (1) is inferred by associating the observed frame $t_{i,o}$ to the frame in the reference sequence with the highest frame correspondence probability as follows:

$$t^*_{i,r} = \underset{t_{i,r} \in [1,M]}{\operatorname{argmax}}\, p(t_{i,r} | t_{i,o}), \quad i = 1, \ldots, N. \tag{2}$$

To estimate $p(t_{i,r} | t_{i,o})$, or in short $p(t_r | t_o)$, and achieve synchronization, we proceed as follows. The most likely reference slice S^r_z is retrieved for each observed slice S^o_l and a matching scheme between corresponding slices provides putative temporal matches (Sec. 2.1). Based on these matches (samples), we learn the joint p.d.f. of frame correspondence $p(t_o, t_r)$ modeled by a Gaussian mixture model instead of the posterior probability distribution $p(t_r | t_o)$

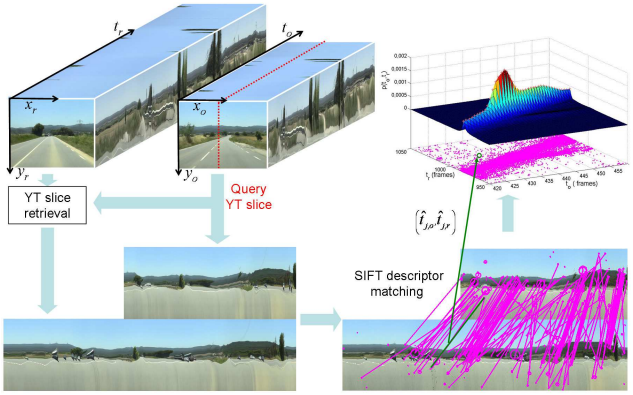


Figure 2. Slice retrieval and matching lead to putative temporal correspondences $(\hat{t}_{j,o}, \hat{t}_{j,r})$ between observed and reference frames.

(Sec. 2.2) because there is no distinction between reference and observed sequence, the samples are continuous, and also the latter can be easily derived by the former for all observed frames t_o like in [1]. Fig. 2 describes visually the above procedure of slice retrieval, matching and learning.

2.1. Slice retrieval & matching

Slice retrieval aims at efficiently associating a slice S_l^o to the most similar S_z^r in the reference sequence. Therefore, in order to match all observed slices, we run a retrieval algorithm for all slices in the observed sequence. To do this, we follow an approach similar to [14]. In short, we first enable the SIFT algorithm to localize keypoints in all the reference slices and describe the area around them [8]. Next, we build a visual vocabulary and an inverted index list. Given an observed slice, we extract its SIFT descriptors and look for the closest visual word, voting thus for the assigned slices stored in the inverted file through the inverse-document-frequency weighting scheme [14]. The reference slice with the highest score is assigned to the query slice. Note that we do not make use of any a priori knowledge about partial (lateral trajectories) of full (almost coincident trajectories) overlap between sequences, but, rather, we obtain this information (horizontal overlap) by slice retrieval.

Having corresponding slices at our disposal, we follow a matching scheme to aggregate temporal correspondences, since each descriptor is assigned to continuous y and t locations. The matching procedure is performed similar to [8], using a distance ratio between nearest and second-nearest neighbor. As a result, pairs of temporal coordinates for matched descriptors reflect putative matches $(\hat{t}_{j,o}, \hat{t}_{j,r})$.

2.2. Learning frame correspondence pdf

Slice matching provides a set of putative matches $\mathcal{T} = \{(\hat{t}_{j,o}, \hat{t}_{j,r})\}_{j=1}^{J}$ that reflects the frame correspondence between observed and reference sequence. Since the matching scheme provides mismatches too, $\mathcal{T}$ is noisy. In this sense, putative matches are considered as samples on $\mathbb{R}^2$ of a frame correspondence pdf $p(t_o, t_r)$. Hence, our goal here is to find the density function $p(t_o, t_r)$ that is most likely to have generated the set $\mathcal{T}$. To this end, we propose the use of a Gaussian mixture model (GMM) and we model the density function as a mixture of K two–dimensional Gaussians, that is,

$$p(t_o, t_r) = \sum_{k=1}^{K} \pi_k \Phi(t_o, t_r; \boldsymbol{\mu}_k, \Sigma_k), \tag{3}$$

where $\Phi(t_o, t_r; \boldsymbol{\mu}_k, \Sigma_k)$ denotes the evaluation of the Gaussian pdf $\mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$ at (t_o, t_r), and π_k, $\boldsymbol{\mu}_k = [\mu_{t_o,k}, \mu_{t_r,k}]^T$ and Σ_k are the prior, the mean and covariance of the k^{th} posterior Gaussian pdf, respectively. The parameters of GMM $\{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}_{k=1}^{K}$ are learnt using *Maximum Likehood Estimation* that is solved by *Expectation—Maximization* (EM) algorithm [4]. The likelihood increase of frame correspondence is guaranteed during optimization. In order to avoid over–fitting, the number of Gaussian components K is chosen evaluating the Bayesian information criterion (BIC) from a set of possible number of components with $K \in \{N/2, N/2 + 10, \ldots, N\}$. That criterion penalizes models with a large number of parameters.

3. Spatial registration and synchro refinement

Now that we have synchronized the sequences up to frame accuracy, our goal reduces to the alignment in space. However, due to unsynchronized acquisition, observed frames optimally match to reference subframes. To achieve simultaneously spatial registration and synchro refinement, we propose the use of a spatio-temporal alignment scheme that applies to short subsequences (say 3 frames long) in turn and assumes homographies in space and affinities in time. Homographies approximate the inter-sequence motion since our scenario assumes a short baseline, while temporal affinities provide subframe accuracy and compensate for different frame rates and/or speed of cameras.

Let us suppose that $\mathbf{q}_o = [x_o, y_o, t_o]^t$ and $\mathbf{q}_r = [x_r, y_r, t_r]^t$ denote space-time points in the observed and reference sequences respectively. Since we are interested in dense correspondences, we adopt a parametric model $\mathbf{q}_o = \gamma(\mathbf{q}_r; \mathbf{h})$ parameterized as follows:

$$\begin{bmatrix} \tilde{x}_o \\ \tilde{y}_o \\ \tilde{w}_o \\ t_o \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & 0 \\ h_4 & h_5 & h_6 & 0 \\ h_7 & h_8 & 1 & 0 \\ 0 & 0 & \beta & \alpha \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ 1 \\ t_r \end{bmatrix}, \tag{4}$$

where $x_o = \tilde{x}_o/\tilde{w}_o$ and $y_o = \tilde{y}_o/\tilde{w}_o$, and $\mathbf{h} = [h_1, \ldots, h_8, \alpha, \beta]^T$. Essentially, the parameters h_i, $i = 1, ..8$ describe the motion between corresponding frames,

the parameter α adjusts the foreshortening in time between sequences and the parameter β provides the subframe correction. The goal of parametric alignment is the estimation of the above transformation matrix by defining an objective function and solving the appropriate optimization problem. To this end, we extend the Enhanced Correlation Coefficient (ECC) algorithm [6] to space–time dimensions, since ECC offers robustness to appearance variations.

Let us assume that we are interested in correspondences of a Group Of Locations (GOL) in the input sequence, being L their number. In our case, GOL reflects all space-time points, otherwise it could be a group of sparse points or a sub-region. By stacking the image intensities of GOL, we form the observed vector and the reference counterpart $\mathbf{b}$ and $\mathbf{r_h}$ respectively, denoting as $\bar{\mathbf{b}}$ and $\bar{\mathbf{r}}_\mathbf{h}$ their zero-mean versions. Note that reference vector is parameterized by $\mathbf{h}$ since $\gamma()$ applies to the reference sequence. Then, ECC alignment algorithm aims at solving the following problem

$$\max_{\mathbf{h}} f(\mathbf{h}) = \max_{\mathbf{h}} \frac{\bar{\mathbf{b}}^t \bar{\mathbf{r}}_\mathbf{h}}{\|\bar{\mathbf{b}}\|\|\bar{\mathbf{r}}_\mathbf{h}\|} , \tag{5}$$

where $f(\mathbf{h})$ is the enhanced correlation coefficient between the two vectors (sequences). By assuming a forwards additive rule $\mathbf{h}^j = \mathbf{h}^{j-1} + \Delta\mathbf{h}^j$, $j = 1, 2, ...$, and after Taylor expanding the reference vector, $f(\mathbf{h})$ is approximated by the function

$$f(\Delta\mathbf{h}^j; \mathbf{h}^{j-1}) = \frac{\bar{\mathbf{b}}^t [\bar{\mathbf{r}}_{\mathbf{h}^{j-1}} + \mathrm{G}_{\mathbf{h}^{j-1}} \Delta\mathbf{h}^j]}{\|\bar{\mathbf{b}}\|\|\bar{\mathbf{r}}_{\mathbf{h}^{j-1}} + \mathrm{G}_{\mathbf{h}^{j-1}} \Delta\mathbf{h}^j\|} , \tag{6}$$

where $\mathrm{G}_\mathbf{h}$ is the $L \times 10$ Jacobian of $\mathbf{r}$ w.r.t. $\mathbf{h}$. Specifically, each row of G is obtained by the product of spatio-temporal gradient of reference image at some location and the Jacobian of warp in (4) for this location, given by

$$\mathrm{J}_\gamma = \frac{1}{\tilde{w}_o} \begin{bmatrix} x_r & y_r & 1 & 0 & 0 & 0 & -x_r x_o & -y_r x_o & 0 & 0 \\ 0 & 0 & 0 & x_r & y_r & 1 & -x_r y_o & -y_r y_o & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{w}_o t_r & \tilde{w}_o \end{bmatrix} . \tag{7}$$

Hence, by the above iterative framework, we ideally expect that $f(\Delta\mathbf{h}^j; \mathbf{h}^{j-1})$ approaches $f(\mathbf{h})$ as j increases.

By dropping the indices $_\mathbf{h}$ and ^{j}, when $\bar{\mathbf{b}}^t \mathrm{A} \bar{\mathbf{r}}_\mathbf{h} > 0$ [2] with $\mathrm{A} = \mathrm{I} - \mathrm{G}(\mathrm{G}^T\mathrm{G})^{-1}\mathrm{G}^T$ being an orthogonal projection operator and I the identity matrix, it has been proved [6] that the function $f(\Delta\mathbf{h}; \mathbf{h})$ attains a global maximum at

$$\Delta\mathbf{h} = (\mathrm{G}^t\mathrm{G})^{-1}\mathrm{G}^t \left\{ \frac{\bar{\mathbf{r}}_\mathbf{h}^t \mathrm{A} \bar{\mathbf{r}}_\mathbf{h}}{\bar{\mathbf{b}}^t \mathrm{A} \bar{\mathbf{r}}_\mathbf{h}} \bar{\mathbf{b}} - \bar{\mathbf{r}}_\mathbf{h} \right\} . \tag{8}$$

The complexity of ECC algorithm is $O(LN_h^2)$ per iteration where N_h is the number of parameters.

The partial overlap of frames caused by parallel camera trajectories can be easily extracted by slice matching, initializing thus appropriately the parameter h_3 of the warp.

[2] This condition degenerates only when totally unrelated profiles are compared [6]. Though, we consider here highly correlated image profiles.

4. Experimental results

In this section qualitative and quantitative results are presented to validate the proposed approach. Specifically, we evaluate the performance of different counterparts of the proposed algorithm and compare them with the most related works [5, 7]. The evaluation counts on experimenting with real sequences recorded by in-vehicle cameras, when they are following approximately coincident trajectories [5]. The alignment of these sequences implies a quite challenging task, since the speed of vehicles varies too irregularly. The average length of the reference and observed sequences is approximately 2000 and 1400 frames respectively, while their spatial resolution is 720×540 pixels. These datasets are provided with their ground–truth, i.e. the reference intervals $[l_i, u_i]$ that each observed frame must correspond to; the length of these intervals is 3 frames on average. Similar to [5], the synchronization error of a candidate pair $(t_{i,o}, t_{i,r})$ is defined as

$$\mathrm{err}(t_{i,o}, t_{i,r}) = \begin{cases} 0 & \text{if } l_i \le t_{i,r} \le u_i \\ \min\left(|l_i - t_{i,r}|, |u_i - t_{i,r}|\right) & \text{otherwise} \end{cases} \tag{9}$$

The performance of synchronization is quantified through the percentage $1 - \sum_i (err(t_{i,o}, t_{i,r}) > \epsilon)/N$ for $\epsilon = 0, 1$. However, since it comes to real datasets, we qualitatively compare the performance of the methods regarding the spatial registration.

4.1. Performance Evaluation

In this section, we evaluate the different components of the proposed algorithm: slice retrieval & matching (SRM) (Sec. 2.1), learning frame correspondence pdf (SRM+GMM) (Sec. 2.2), and finally, subframe video alignment (SVA) (Sec. 3). The first two components infer the temporal mapping maximizing Eq. (2) (pure SRM builds on empirical probabilities), while SVA refines the temporal mapping obtained by SRM+GMM and register spatially the corresponding frames based on the ECC outcome.

In the context of pure SRM, we could obviously obtain temporal matches by directly matching spatio-temporal descriptors through retrieval, without mapping the slices before; we call this scheme as direct temporal matching (DTM). This way, we empirically estimate the needed probabilities, that is, by counting the number of times that a candidate pair (t_o, t_r) appears in $\mathcal{T}$ after its rounding. That counting goes for all possible frame correspondences within the proper normalization.

Table 1 shows the synchronization scores achieved by the investigated methods. We provide results for $\epsilon = 0$ and $\epsilon = 1$ to show the error variance. As we can see, SRM achieves higher synchronization scores than DTM across all sequences. It is very important to note that no geometric constraints about matched descriptors are taken into account

with both methods. Moreover, the contribution of learning the pdf (GMM) instead of the empirical estimation of probabilities (DTM) is clearly evident too. Specifically, GMM remarkably increases the performance of SRM, that is, by 9% on average.

Except for the contribution in spatial alignment, we achieve further improvements in synchronization with the help of ECC (SVA). Subsequences of 3 frames were adopted, permitting ECC to execute 15 iterations per subsequence. However, experiments showed us that misalignment in space comes usually with misalignment in time. Therefore, and taking into account the assumption of similar trajectories, it is reasonable to not refine frame pairs when SVA returns an extreme homography; this can be easily checked by the value of parameters h_7 and h_8. As a result, SVA reaches more higher levels as it increases the synchronization score by 7% on average.

4.2. Comparison

We compare the proposed algorithm with the two closest related works [5, 7]. Diego *et al*. [5] estimate a complete temporal mapping by maximizing an image– and location–similarity based on global image descriptors and GPS data respectively. Note that this method exploits prior information as it assumes forward motion only. Besides, we adapt the scene alignment algorithm proposed by Liu *et al*. [7] to video alignment, i.e. we solve the problem in turn for all observed frames. This reflects a reasonable comparison between frame and slice retrieval. That method consists of retrieving the short-list (i.e. top-20 [7]) of reference frames by spatial histogram matching of quantized SIFT. Then, a spatial coherence step using the SIFT-flow algorithm re-ranks the list w.r.t. the flow energy, thus emerging the nearest reference frame.

From the results of Table 1, we derive that SVA outperforms its competitors, SIFT-flow [7] and BN [5], in all cases. Specifically, SVA improves by 13% and 6% the scores obtained by BN and SIFT-flow respectively. We recall that this comparison does not favor our method in the sense that our method does not count on geometric constraints, since we aim at investigating the performance of the net algorithm. However, it is obvious that SRM scheme would be benefitted by such constraints.

By putting aside the training time of the algorithms, the main drawback of SIFT-flow algorithm is its complexity, since it requires more than 30 sec to compute the flow between two frames at half resolution. This leads to a heavy task since the alignment of two sequences with 1000 frames takes approximately 7 days. On the other hand, synchronization by SRM+GMM takes 15 sec and the ECC algorithm, implemented in Matlab, requires 14 sec per subsequence. BN method adopts an image alignment scheme based on Lucas-Kanade framework with 3 parameters (3D rotation) that captures obviously the registration faster than ECC at the cost of lower accuracy. However, the synchronization problem is solved globally and execution times are not given in [5]. Note that ECC and Lucas-Kanade attain the same complexity for the same number of parameters and pixels [6]. Although, we consider here an offline application, we must stress that SIFT-flow requires a lot of time for typical current architectures, even if we drastically reduce the short-list.

As we deal with real data, we give in Fig. 3 some representative results that capture the pros and cons of SVA and its strong competitor SIFT-flow. To qualitatively assess the spatial registration, a simple image fusion is established by replacing the green component of the observed frame with the *warped* green component of the reference frame. As a consequence, any misalignment or difference is marked with green and pink colors [3]. A motorbike in (Fig. 3a), two persons in (Fig. 3b) and various vehicles in (Fig. 3c-f) appear in one of the frames must be aligned. We observe that Liu et al.'s algorithm achieves pixel-wise correspondences at the expense of misalignment when non-scene objects appear (outliers). SIFT-flow creates splashes in their locations while the proposed algorithm seems to be robust to these outliers and provides accurate registration results. When the homography does not fit well, SVA may cause minor local misalignments, as in road–lines in Fig. 3b. Note that SIFT-flow returns pixel–wise flows instead of estimating a global transformation as SVA does. The performance of the spatial registration and the final video alignment is clearly more evident in http://www.cvc.uab.es/~fdiego/VS2011/.

5. Conclusions

A novel spatio-temporal alignment method was proposed to align video sequences recorded at different times from independently moving cameras whose trajectories can be nearly coincident or parallel. In order to avoid exhaustive cross-frame search, video synchronization builds upon matches between corresponding spatio-temporal slices. These matches are considered as samples of frame correspondence pdf modeled by a GMM whose parameters are learnt, and a MAP inference problem is solved based on this pdf. Next, a spatio–temporal alignment is adopted to refine the synchronization up to subframe resolution, and at the same time, to spatially align subsequences. Experiments on real video sequences recorded by moving vehicles on different road types show that the proposed algorithm outperforms state–of–the–art methods. As future work, we envisage dealing with crossed, or at least more random, trajectories by working with sub-slices or appropriately oriented slices and piece-wise homographies.

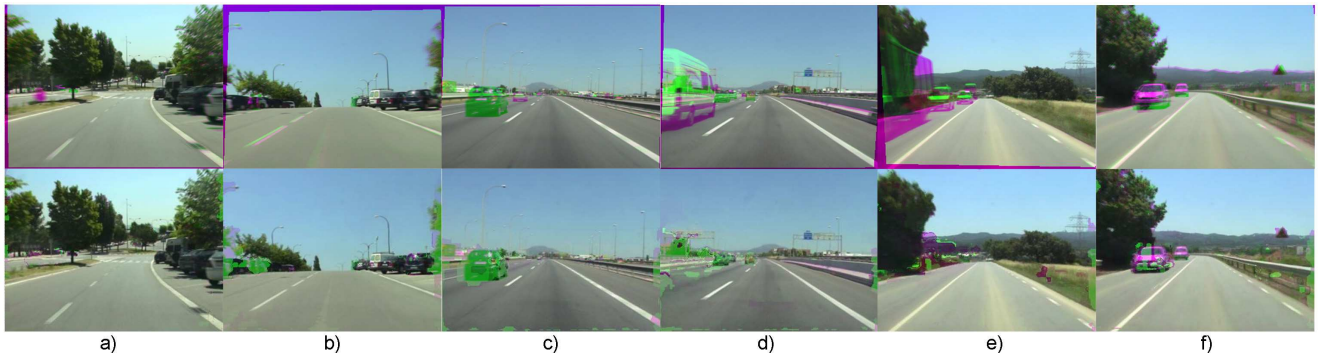


Figure 3. Example results of SVA algorithm (top) and SIFT-flow (bottom). More results of SVA can be viewed at the supplemental material. Please refer to http://www.cvc.uab.es/~fdiego/VS2011/ for video results.

	Highway ($\epsilon = 0 \backslash \epsilon = 1$)	Campus ($\epsilon = 0 \backslash \epsilon = 1$)	Backroad ($\epsilon = 0 \backslash \epsilon = 1$)	Average ($\epsilon = 0 \backslash \epsilon = 1$)
DTM	67.7\81.5	69.6\78.9	50.2\62.1	62.5\71.2
SRM	72.2\83.2	73.4\85.3	63.0\77.4	69.5\81.9
SRM+GMM	83.2**92.6**	75.8\88.0	69.2\85.1	76.1\88.6
SVA	**84.5**\92.3	**82.3\91.5**	**78.1\88.7**	**81.6\90.8**
SIFT-flow [7]	74.2\87.2	82.0\89.7	72.4\86.4	76.2\87.7
BN [5]	67.6\73.1	83.1\90.5	66.8\70.5	72.5\78.0

Table 1. Synchronization scores (%) obtained by the proposed methods and the competitors for two values of error tolerance ϵ.

6. Acknowledgments

This work was supported by the Spanish MICINN under Project TRA2010-21371-C03-01, Consolider Ingenio 2010: MIPRCV(CSD200700018), and FPU MEC grant AP2007-01558 of Ferran Diego.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1st edition, 2007. 3

[2] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. The video genome. *CoRR*, abs/1003.5320, 2010. 1, 2

[3] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, 2002. 1, 2, 5

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. 3

[5] F. Diego, D. Ponsa, J. Serrat, and A. M. Lopez. Video alignment for change detection. *Image Processing, IEEE Transactions on*, Preprint(99), 2010. 1, 2, 4, 5, 6

[6] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008. 2, 4, 5

[7] C. Liu, J. Yuen, A. Torralba, and W. T. Freeman. Sift flow: dense correspondence across different scenes. In *Proc. European. Conf. on Computer Vision (ECCV)*, 2008. 2, 4, 5, 6

[8] D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110, 2004. 3

[9] F. L. Padua, R. L. Carceroni, G. A. Santos, and K. N. Kutulakos. Linear sequence-to-sequence alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:304–320, 2010. 1

[10] D. Pundik and M. Y. Video synchronization using temporal signals from epipolar lines. In *Proc. of European. Conf. on Computer Vision (ECCV)*, 2010. 1, 2

[11] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood. View-invariant alignment and matching of video sequences. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2003. 2

[12] A. Ravichandran and R. Vidal. Video registration using dynamic textures. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2011. 1

[13] P. Sand and S. Teller. Video matching. *ACM Transactions on Graphics*, 22(3):592–599, 2004. 1, 2

[14] J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 31(4):591–606, 2009. 2, 3

[15] T. Tuytelaars and L. C. Gool. Synchronizing video sequences. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004. 2

[16] Y. Ukrainitz and M. Irani. Aligning sequences and actions by maximizing space–time correlations. In *ECCV06*, volume 3953, pages 538–550, Graz, Austria, 2006. 1

[17] L. Wolf and A. Zomet. Wide baseline matching between unsynchronized video sequences. *Int. Journal on Computer Vision*, 68(1):43–52, 2006. 1, 2