Efficient and Robust Alignment of Unsynchronized Video Sequences

Georgios D. Evangelidis[1] and Christian Bauckhage[2]

[1] Department of Computer Engineering & Informatics,
University of Patras, Rio-Patras, 26500, Greece
[2] Fraunhofer IAIS, Schloss Birlinghoven, St. Augustin, Germany

Abstract. This paper addresses the problem of aligning two unsynchronized video sequences. We present a novel approach that allows for temporal and spatial alignment of similar videos captured from independently moving cameras. The goal is to synchronize two videos of a scene such that changes between the videos can be detected automatically. This aims at applications in driver assistance or surveillance systems but we also envision applications in map building. Our approach is novel in that it adapts an efficient information retrieval framework to a computer vision problem. In addition, we extend the recent ECC image-alignment algorithm to the temporal dimension in order to improve spatial registration and enable synchro refinement. Experiments with traffic videos recorded by in-vehicle cameras demonstrate the efficiency of the proposed method and verify its effectiveness with respect to spatio-temporal alignment accuracy.

1 Introduction

Video alignment requires matching scene points in both space and time. Given two or more video sequences, the goal is to find correspondences between projections of the same scene point in a time-coherence framework so that frames from the different videos can be registered.

Most related contributions either assume stationary cameras or consider settings of jointly moving cameras in a fixed relative orientation [2,9,14]. With the exception of [9], these works also consider a linear model for temporal displacements between videos. Independently moving cameras have been studied either in the context of a constant temporal offset between sequences (overlap in time) [13] or of a dynamic time shift (no overlap in time) [3,10]. Since the latter poses difficult problems when moving cameras accelerate irregularly, related contributions assumed nearly coincident camera trajectories or the availability of metadata such as GPS coordinates [3,10]. While most approaches to video synchronization attempt to align trajectories of interest points [2,9,13,14], other methods rely on spatial intensity information [2,3,10]. To establish the geometry between synchronized frames, models such as 2D homographies [2], fundamental matrices [2,9], 3D rotations [3], or affine projections [14] have been used.

In this paper we consider independently moving un-calibrated cameras whose trajectories are similar. In particular, we consider in-vehicle cameras that are mounted behind the windshield and record everyday street scenes. We aim at aligning videos that are recorded on *different days* from within different vehicles driving the same route and

R. Mester and M. Felsberg (Eds.): DAGM 2011, LNCS 6835, pp. 286–295, 2011.

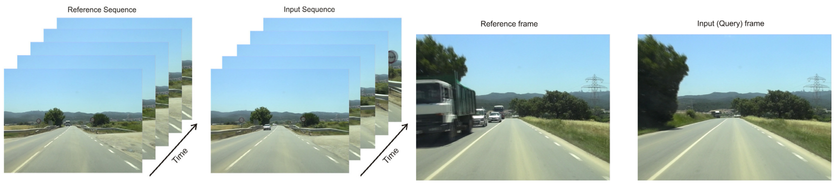


Fig. 1. *Top*: An example of two video sequences [3]. Due to non-overlapping capture times, different moving objects appear in the sequences. *Bottom*: Examples of corresponding frames with noticeably different scene content.

following approximately the same lane (see Fig.1). In this scenario, velocity and acceleration of the cameras naturally vary and the corresponding temporal mapping is highly non-linear. Unlike previous works, the method we propose in this paper can even deal with backward motion of cameras. It is fast enough to allow for online application and the recorded 3D scene is not required to be static.

Our scenario is closely related to [3,10], yet, we consider completely different algorithmic approach: we treat video synchronization as an information retrieval problem where we apply highly efficient low-level descriptors and efficient subsequent matching steps. As our video data sets are captured at sensibly different times, the first recorded sequence can be preprocessed and indexed before the second sequence becomes available for analysis. This mimics a recent trend in computer vision where computations are pushed back to an off-line task in order to accelerate online procedures [6,12]. In our case, pre-processing focuses on efficiently storing the frames of the first sequence in a database, indexing the database, and structuring the index appropriately. This way we can handle the subsequent synchronization problem by means of querying the database for content that is similar to a given frame in the second video sequence. Having thus obtained a rough synchronization, we then address the spatial registration between synchronized frame and the problem of subframe correction and propose a space-time extension of the recently introduced ECC algorithm [4].

Our presentation proceeds as follows: Next, we formalize the video alignment problem. Section 3 casts video synchronization as an information retrieval problem and Section 4 presents our extension of the ECC algorithm to the space-time dimension. In Section 5, we discuss efficiency and, in Section 6, we evaluate our approach on real world sequences. Finally, Section 7 concludes this contribution.

2 Problem Formulation

Suppose we are given two image sequences $S_r = I_r(\hat{\mathbf{x}})$ and $S_q = I_q(\mathbf{x})$, where the first is a reference and the second is a query sequence and $\hat{\mathbf{x}} = [\hat{x}, \hat{y}, \hat{t}]^t$, $\mathbf{x} = [x, y, t]^t$ denote space-time points. The goal of video alignment is to match space-time points in the two sequences. We are interested in a spatio-temporal mapping $W(\mathbf{x}; \mathbf{p})$ where $\mathbf{p}$ is a space-time parameter vector, such that $\hat{\mathbf{x}} = W(\mathbf{x}; \mathbf{p})$. Following [2], we define the mapping model as $W(\mathbf{x}; \mathbf{p}) = [W_s([x, y]^t; \mathbf{p}_s)^t, W_t(t, \mathbf{p}_t)]^t$ where $W_s()$ is the spatial- and $W_t()$ is the time-warp parameterized by $\mathbf{p}_s$ and $\mathbf{p}_t$ respectively, and $\mathbf{p} = [\mathbf{p}_s^t,\ \mathbf{p}_t^t]^t$. For independently moving cameras, both parameter vectors $\mathbf{p}_s$ and $\mathbf{p}_t$ vary along S_q. Yet, in the case of irregular and backward motion, both vectors must be re-estimated for all query frames.

In order to efficiently handle such cases, we propose a new approach to video synchronization that can also be viewed as an initialization scheme for the spatio-temporal alignment. Let us suppose that the time mapping is roughly expressed through a finite discrete-time signal $T : \mathbb{N} \rightarrow \mathbb{N}$, such that $t' = T(t)$ and t' is close to $\hat{t}$. Towards the goal of finding *integer* values $T(t)$ for all time indices t, we consider this signal to be the outcome of an information retrieval step. More specifically, we consider the whole set of reference frames as a *database* of images and all input frames as *query* frames. Then, by querying the database with an input frame assigned to time index t_0, we retrieve the corresponding frame assigned to time index $t_0' = T(t_0)$.

Given the pair $(t_0, T(t_0))$ we adopt a *time-local* spatio-temporal model $W()$, which permits us not only to spatially align synchronized frames, but to refine the time alignment result, thus providing subframe accuracy. Note that this model does not imply a short-time sequence-to-sequence alignment but an image-to-sequence, or better *frame-to-subframe*, alignment. Given a query frame $I_q(x,y,t_0)$ and the mapped pair $(t_0, T(t_0))$, we are looking for a *spatio-temporally warped* image (subframe) from the short-time sequence $I_r(\hat{x}, \hat{y}, T(t_0) \pm \mu)$, where μ is a small integer so that a predefined error criterion between corresponding frames is satisfied. As a result, we obtain subframe accuracy without using expensive spatio-temporal manifold computations [2]. This is due to the space-time extension in parameter-domain only. Next, we discuss how to determine the time-mapping $T()$ and the spatio-temporal model $W()$.

3 An IR Approach to Video Synchronization

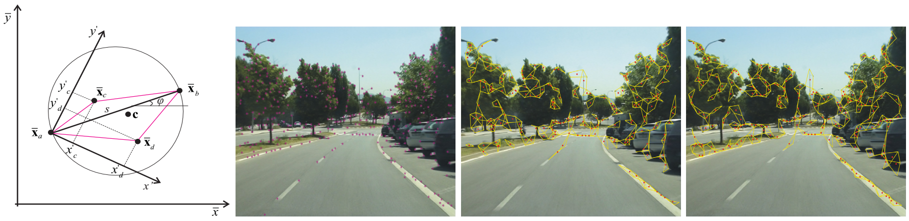
We adopt an information retrieval approach to deal with the video synchronization problem. This allows us to preprocess the reference data *without any knowledge of the query sequence* and to devise an efficient synchronization step. Similar to modern information retrieval methods [8,12] we apply inverted index lists and weighted voting scores in order to improve the reliability of the retrieval process.

Although most retrieval works in computer vision society rely on multidimensional descriptors [7,12], our scenario permits the use of short-length descriptors. In order to *describe* image patches we apply a geometric hashing method introduced in [6] for astrometry. Specifically, let us assume that we have applied an interest point detector [11] in an image and the locations of the interest points are available. Then we consider quadruples of nearby interest points to characterize local image structures.

Suppose a quadruple (*quad*) of interest points $\bar{\mathbf{x}}_i = [\bar{x}_i, \bar{y}_i]^t, i = \{a,b,c,d,\}$ as shown in Figure 2. $\bar{\mathbf{x}}_a, \bar{\mathbf{x}}_b$ are the *control points* which are defined by the most widely separated pair of points. By s we denote the distance (*diameter*) between control points; φ denotes the *orientation* of the diameter vector and $\mathbf{c}$ denotes the *centroid* of the quad. That is

$$s = \|\bar{\mathbf{x}}_a - \bar{\mathbf{x}}_b\|, \quad \varphi = \tan^{-1}\frac{\bar{y}_b - \bar{y}_a}{\bar{x}_b - \bar{x}_a}, \quad \mathbf{c} = \frac{1}{4}\sum_i \bar{\mathbf{x}}_i\ , \qquad \text{(1a-c)}$$

where $\|\cdot\|$ denotes the Euclidean distance. We then consider a local coordinate system oriented and centered with respect to the control points $\bar{\mathbf{x}}_a,\ \bar{\mathbf{x}}_b$, so that their locations



(a) (b) (c) (d)

Fig. 2. (a) Geometric hashing using a quad structure; (b) query frame with extracted Harris points; (c) and (d) valid quads of the query and corresponding reference frame; red dots are quad centers

coincide with $(0,0)$ and $(1,1)$, respectively. This allows for *hashing* the remaining points $\bar{\mathbf{x}}_c, \bar{\mathbf{x}}_d$ in the local coordinate system through their new coordinates (x'_c, y'_c), (x'_d, y'_d). Accordingly, any quad of nearby features can be coded using a length-four hash-code (x'_c, y'_c, x'_d, y'_d). In other words, each quad is represented as a 4D point space and similar quads correspond to nearby points in this space. Similar to [6], we only consider quads where the points $\bar{\mathbf{x}}_c, \bar{\mathbf{x}}_d$ lie inside a circle of diameter s. Any different order of points in pairs $(\bar{\mathbf{x}}_a, \bar{\mathbf{x}}_b)$ and $(\bar{\mathbf{x}}_c, \bar{\mathbf{x}}_d)$ creates a different symmetry which can be easily resolved [6].

This novel local descriptor is translation-, scale-, and rotation invariant which is required to match quads between frames. Also, small localization errors from interest point detection entail only small displacements of the hash code in the 4D feature space.

3.1 Indexing, Structure, and Retrieval

Once the reference sequence is available, we store each frame $I_r(\hat{x}, \hat{y}, \hat{t}_n)$ as an image I_n in a database where $n = 1, 2, \cdots, N$. We apply an interest point detector (e.g. Harris) to all images, extract all valid quads and assign to the j^{th} quad of the n^{th} image its hash code $q_{nj} = (x'_c, y'_c, x'_d, y'_d)_{nj}$, where $j = 1, 2, \cdots, J_n$. Since the discriminative power of the quad descriptor is low, we do not apply vector quantization [12] but keep working with continuous hash-codes. In addition, the short-length descriptor allows us to store all hash-codes q_{nj} and create an inverted index list assigning to every record its reference set $R_{nj} = \{n, \mathbf{c}_{nj}, s_{nj}, \varphi_{nj}\}$.

Given a query quad, we do not search for the nearest neighbor but look for similar quads inside a range. This implies a *range search* problem and in order to quickly answer a query we apply a kD-tree structure ($k = 4$). Searching for a corresponding frame to a query frame can then be cast as a voting approach. Given a query image and its quads $q_k, k = 1, 2, \cdots, K$, we query the database with all q_k and any quad q_{nj} which is ε-close to q_k votes for the n^{th} image. By initializing all image scores v_n to 0, we increase the score of each retrieved image by $v_n \leftarrow v_n + f(q_k, q_{nj})$, where

$$f(q_k, q_{nj}) = \begin{cases} w_n & \text{if} \quad \|q_k - q_{nj}\| < \varepsilon \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The weights w_n could be chosen to be the *terms frequency - inverse document frequency* (TF-IDF) scores used in text retrieval [8]. However, since quads correspond to continuous

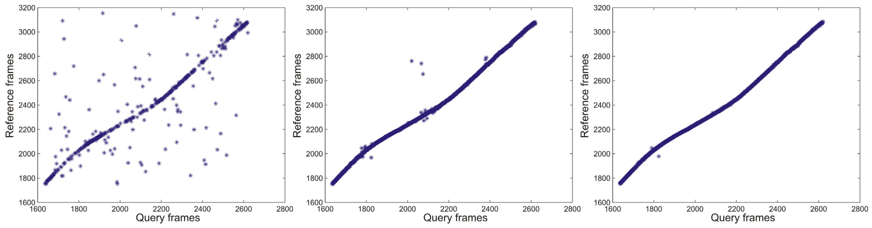


Fig. 3. Frame synchronization for the *Rural* sequence [3] based on pure retrieval results (*left*), after enforcing spatio-temporal consistency with $R_0 = 50$ (*middle*) and after additionally enforcing rotation consistency constraints with $|\varphi_k - \varphi_{nj}| < \pi/12$ (*right*)

vectors and thus are unique with high probability, the TF factor does not add to the precision. The IDF factor, on the other hand, improves the retrieval precision since quads that appear in a similar form in many images are not indicative of image content. Hence, we choose $w_n = log\frac{N}{N_k}$, where N_k is the number of the retrieved images after querying q_k.

3.2 Spatio-temporal Coherence

In order to reject false positive matches before voting, we enforce a spatio-temporal coherence constraint which agrees with our basic assumption that the trajectories of two cameras are approximately coincident. Since we would like to retrieve that frame which has been captured from the closest point to the viewpoint of the query frame, it is justified to not allow matches between far apart quads. Therefore, for correspondence, we require a quad in the database image to be inside a circular region whose center coincides with the centroid of the query quad, i.e. $\|\mathbf{c}_k - \mathbf{c}_{nj}\| < R_0$. Due to large overlaps between images this constraint favors both spatial and temporal coherence.

We can also enforce additional constraints like scale- and rotation-consistency by enabling appropriate coarse coherence measures for s and φ respectively. However, we found such constraints not to be as vital as the spatio-temporal one. Fig 3 shows the synchronization result before and after enabling constraints.

4 Spatial Alignment and Synchro Refinement

The above rough video synchronization step results in a sequence $T : \mathbb{N} \to \mathbb{N}$ and matched frames $(t, T(t))$. Ideally, however, synchronization would yield a sequence $T : \mathbb{N} \to \mathbb{R}_+$ providing subframe accuracy. To further refine synchronization results and to spatially align synchronized frames, we extend a recent, robust image alignment algorithm [4].

The Enhanced Correlation Coefficient (ECC) scheme as reported in [4] supposes that $I_q(x, y, t_0)$ is the template image and $I_r(\hat{x}, \hat{y}, T(t_0))$ is the input image that must be warped towards the alignment. If $A = \{\mathbf{x}_m | m = 1, 2, \ldots, M\}$ is the set of space-time points of the query image, ECC then determines the corresponding set $\hat{A} = \{\hat{\mathbf{x}}_m | \hat{\mathbf{x}}_m = W(\mathbf{x}_m; \mathbf{p}),\ m = 1, 2, \ldots, M\}$ in the other sequence. This requires to explicitly define the spatio-temporal mapping $W()$. Although the fundamental matrix would apply to our scenario, its use only characterizes pixel motions up to an epipolar line and entails extra effort for computing dense correspondences [5]. Moreover, estimating the fundamental matrix is susceptible to errors and moving cameras may increase this uncertainty. Therefore, we approximate the spatial motion using a 2D homography model. Incorporating only temporal shifts for the time warping and using homogeneous spatial coordinates, we can write the space-time model as

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \\ \hat{t} \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & 0 \\ h_4 & h_5 & h_6 & 0 \\ h_7 & h_8 & 1 & 0 \\ 0 & 0 & \tau & 1 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ 1 \\ t_0 \end{bmatrix} , \tag{3}$$

where $x = \tilde{x}/\tilde{w}$, $y = \tilde{y}/\tilde{w}$, $\mathbf{p}_s = [h_1, \ldots, h_8]^t$ and $\mathbf{p}_t = \tau$, being τ appropriately initialized via the synchronization task.

ECC alignment aims at estimating the optimal parameter vector such that the correlation coefficient between the query image and the *warped* retrieved image is maximized. Stacking the intensities of the points contained in A and $\hat{A}$ we form the vector $\mathbf{i}_q = [I_q(\mathbf{x}_1), I_q(\mathbf{x}_2), \cdots, I_q(\mathbf{x}_M)]^t$ and the warped vector $\mathbf{i}_\mathbf{p} = [I_r(\hat{\mathbf{x}}_1), I_r(\hat{\mathbf{x}}_2), \cdots, I_r(\hat{\mathbf{x}}_M)]^t$, and let $\bar{\mathbf{i}}_q$ and $\bar{\mathbf{i}}_\mathbf{p}$ be their zero mean counterparts. Then, the objective function that must be maximized is the *enhanced correlation coefficient* defined as

$$\rho(\mathbf{p}) = \frac{\bar{\mathbf{i}}_q^t \bar{\mathbf{i}}_\mathbf{p}}{\|\bar{\mathbf{i}}_q\| \, \|\bar{\mathbf{i}}_\mathbf{p}\|} . \tag{4}$$

In order to solve the maximization problem, we assume similar to [4] that a nominal parameter vector $\tilde{\mathbf{p}}$ is known, such that $\mathbf{p} = \tilde{\mathbf{p}} + \Delta\mathbf{p}$. Then, using a first order Taylor expansion on $\bar{\mathbf{i}}_\mathbf{p}$, the ECC function amounts to

$$\rho(\Delta\mathbf{p}; \tilde{\mathbf{p}}) = \frac{\bar{\mathbf{i}}_q^t [\bar{\mathbf{i}}_{\tilde{\mathbf{p}}} + \mathrm{J}_{\tilde{\mathbf{p}}} \Delta\mathbf{p}]}{\|\bar{\mathbf{i}}_q\| \sqrt{\|\bar{\mathbf{i}}_{\tilde{\mathbf{p}}}\|^2 + 2\bar{\mathbf{i}}_{\tilde{\mathbf{p}}}^t \mathrm{J}_{\tilde{\mathbf{p}}} \Delta\mathbf{p} + \Delta\mathbf{p}^t \mathrm{J}_{\tilde{\mathbf{p}}}^t \mathrm{J}_{\tilde{\mathbf{p}}} \Delta\mathbf{p}}} , \tag{5}$$

where $\mathrm{J}_{\tilde{\mathbf{p}}}$ is the Jacobian of the vector $\bar{\mathbf{i}}_\mathbf{p}$ with respect to parameters evaluated at $\tilde{\mathbf{p}}$. However, our extension requires the redefinition of this matrix. Its size is $M \times 9$ and the m^{th} row is formed by the product $\nabla \mathrm{I}_r^t \mathrm{J}_W$ where $\nabla \mathrm{I}_r = [\frac{\partial I_r}{\partial \hat{x}}, \frac{\partial I_r}{\partial \hat{y}}, \frac{\partial I_r}{\partial \hat{t}}]^t$ is the spatio-temporal gradient of image I_r evaluated at point $W(\mathbf{x}_m; \tilde{\mathbf{p}})$ and J_W is the Jacobian of the transformation in (3) evaluated at $\tilde{\mathbf{p}}$. Note that both spatial and temporal gradients build on first-order central differences of smoothed intensities. As far as J_W is concerned, based on (3) we have

$$\mathrm{J}_W = \begin{bmatrix} \frac{\partial W_s}{\partial \mathbf{p}_s} & \mathbf{0} \\ \mathbf{0}_{1\times 8} & \frac{\partial W_t}{\partial \tau} \end{bmatrix} = \frac{1}{\tilde{w}} \begin{bmatrix} \hat{x} & \hat{y} & 1 & 0 & 0 & 0 & -\hat{x}x & -\hat{x}y & 0 \\ 0 & 0 & 0 & \hat{x} & \hat{y} & 1 & -\hat{y}x & -\hat{y}y & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{w} \end{bmatrix} . \tag{6}$$

Despite the non-linearity of the function $\rho(\Delta\mathbf{p}; \tilde{\mathbf{p}})$, its maximization results in the following closed form solution

$$\Delta\mathbf{p} = (\mathrm{J}_{\tilde{\mathbf{p}}}^t \mathrm{J}_{\tilde{\mathbf{p}}})^{-1} \mathrm{J}_{\tilde{\mathbf{p}}}^t \left\{ \lambda \bar{\mathbf{i}}_q - \bar{\mathbf{i}}_{\tilde{\mathbf{p}}} \right\} , \tag{7}$$

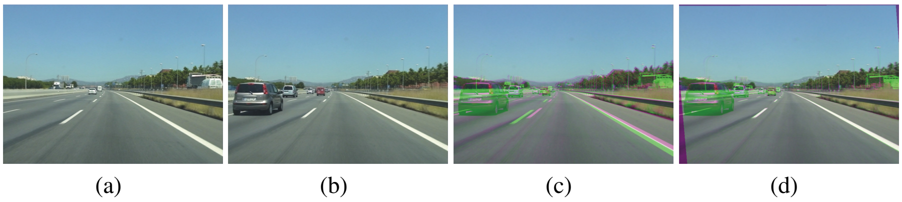


Fig. 4. (a) A query frame and (b) the best retrieved frame; (c) the space-time alignment after 2 and (d) after 10 iterations; differences between frames are indicated in lawn-green and hot-pink

with λ being given by

$$\lambda = \frac{\bar{\mathbf{i}}^t_{\tilde{\mathbf{p}}}(\mathrm{I}_M - \mathrm{P}_\mathrm{J})\bar{\mathbf{i}}_{\tilde{\mathbf{p}}}}{\bar{\mathbf{i}}^t_q(\mathrm{I}_M - \mathrm{P}_\mathrm{J})\bar{\mathbf{i}}_{\tilde{\mathbf{p}}}}\,, \tag{8}$$

where I_M is the identity matrix and $\mathrm{P}_\mathrm{J} = \mathrm{J}_{\tilde{\mathbf{p}}}(\mathrm{J}^t_{\tilde{\mathbf{p}}}\mathrm{J}_{\tilde{\mathbf{p}}})^{-1}\mathrm{J}^t_{\tilde{\mathbf{p}}}$ is a projection operator.

By translating this solution into an iterative scheme $\mathbf{p}^{\{i\}} = \mathbf{p}^{\{i-1\}} + \Delta\mathbf{p}^{\{i\}}$, we can approximate the solution of the highly non-linear problem of maximizing the function in (4). This yields the optimum parameter vector for dense spatio-temporal correspondences of subpixel and subframe accuracy. The complexity per iteration of this scheme can be shown to be $O(M\eta^2)$, where η is the number of parameters [4]. Figure 4 shows an example of the resulting spatio-temporal alignment.

5 Efficiency

An important characteristic of our proposed framework is that we can exploit the sequential nature of video data which implies a *coarse* time-consistency for synchronizing successive frames. We thus propose to split the database of frames into β subsets of successive frames and use a separate *k*D-tree for the quads of each subset. For a regular split (Fig.5 *left*), we would need to investigate two adjacent subtrees whenever the current results are inside a transition area. To avoid this, we allow overlap between adjacent subtrees in the forest (Fig.5 *right*). This way, we need to query only one sub-tree and have to change the tree index if the current retrieval results are above a threshold (e.g. the median of the overlap area).

For range search problems, querying a 4D-tree structure requires $O(n^{\frac{3}{4}} + \kappa)$ where n is the number of points and κ is the number of neighbors within range [1]. Adopting the above splitting method, the query time reduces to $O((n/\beta)^{\frac{3}{4}} + \kappa/\beta)$, which accelerates the synchronization process without affecting its precision. For spatial alignment, we can apply a pyramid based scheme [2] which not only accelerates the alignment algorithm but also compensates for large displacements. Additionally, since gradient-based alignment schemes mainly rely on high frequent parts of a signal, we ignore low-frequency pixels and aggregate only those pixels around key points. Taking into account the complexity $O(M\eta^2)$, where $M \gg \eta$, the computational burden of spatio-temporal alignment drastically reduces via these two modifications.

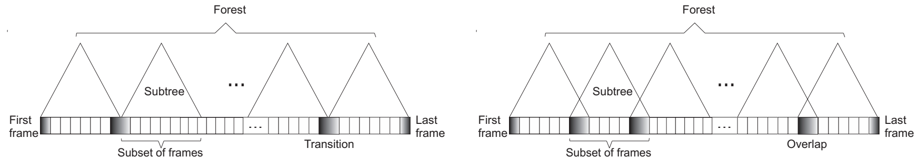


Fig. 5. Subtrees of quads that belong to subsets of reference frames. Regular split (*left*) and split with overlap (*right*).

6 Results

Following the methodology of [3], we evaluate the accuracy of the proposed synchronization method via the resulting synchronization error. As we adopt an IR approach, we compare our method with the recently proposed SIFT-flow algorithm [7] and the method presented in [3]. SIFT-flow estimates temporal alignments by histogram matching whereas spatial correspondences result from a pixel-based flow algorithm. The work in [3] models synchronization as a MAP inference problem in a Bayesian network and considers the common least-squares framework for spatial registration.

We experiment with three real-world video sequences recorded from within moving vehicles at different times, namely the *Backroad*, the *Campus* and the *Highway* sequences [3]. Each dataset shows footage from *accelerating and decelerating* cars. Ground truth is available in form of lower and upper bounds of synchronization indices. If the sequence $f_t(t)$ represents any synchronization result and $L(t)$ and $U(t)$ are the sequences of the lower and upper bounds respectively, the synchronization error is

$$e(f_t(t)) = \begin{cases} 0 & \text{if } L(t) \leq f_t(t) \leq U(t) \\ \min\{|f_t(t) - L(t)|, |f_t(t) - U(t)|\} & \text{otherwise} \end{cases} . \tag{9}$$

The resolution of sequences is 540 × 720 pixels in space and 1500 frames on average in time. The interest point detector we used is the Harris detector as described in [11]. We also tested other detectors, but our results were in accordance with the results of [11] verifying the favorable repeatability of Harris detector. Each subtree of the forest structure (Fig.5 *left*) efficiently stored the descriptors of 250 successive frames, being the overlap equal to 20 frames. Based on equation (2) we considered a tolerance threshold with $\varepsilon = 0.07$ while retrieval results were re-ranked by the space-time coherence constraint with $R_0 = 50$ pixels (the latter should be defined with respect to the video resolution). Finally, ECC run within a coarse-to-fine framework in spatial domain only, using a 4-level gaussian pyramid and running 5 iterations per level.

Table 6 shows the performance of the methods in terms of the synchronization error, i.e. the percentage of values where $e(f_t(t)) > \delta$. We provide results for $\delta = 0$ and $\delta = 1$ to indicate the error variance. We observe that the proposed method performs better for *Highway* dataset since the vehicle follows an almost straight road with high velocity; the latter leads to fewer reference frames as candidate matches to a query. Moreover, the low error variance favors the refinement, as ECC cannot cancel out strong synchronization errors. In other words, even if the quad-based alignment returns false positives, what seems to be important is the distribution of errors, being their concentration around

Table 1. Synchronization Error (%)

	Backroad		Campus		Highway		Average	
	$\delta = 0$	$\delta = 1$	$\delta = 0$	$\delta = 1$	$\delta = 0$	$\delta = 1$	$\delta = 0$	$\delta = 1$
Quad-based	29.4	15.4	26.4	13.5	25.3	8.7	**27.0**	**12.5**
Quad-based-ECC	25.4	8.4	23.8	11.4	8.3	2.9	**19.1**	**7.5**
Diego *et al.* [3]	37.4	31.9	17.7	9.17	32.6	27.7	**29.2**	**22.9**
SIFT-flow [7]	27.7	13.6	18.5	11.7	25.7	12.9	**23.9**	**12.7**

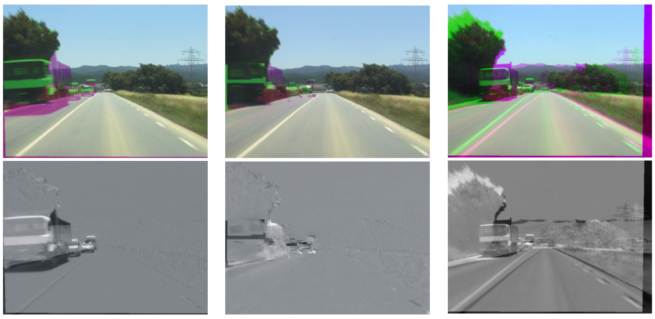


Fig. 6. (*Top*) Alignment results and (*bottom*) pixel-wise differences after alignment by applying (*left*) the proposed approach, (*middle*) SIFT-flow and (*right*) the method in [3]

zero particularly desired. On the other hand, in *Campus* and *Backroad* sequences there appear near-camera "objects" and road turns; the former affects the quad-based alignment while the latter gives rise to homography uncertainties. The SIFT-flow method provides slightly higher error scores while it obviously requires many more operations due to the descriptor's size (a 128-dimensional vector). Still, our method also exhibits better performance than the method in [3] which actually incorporates GPS data.

Figure 6 illustrates change detection results obtained from the three approaches. The proposed method detects scene changes with higher accuracy. SIFT-flow seems to be affected by the presence of moving cars and creates artifacts and truncated objects. The method in [3] performs poorly. As far as the complexity is concerned, the average synchronization time of the proposed method is $\mathbf{0.22}$ **sec** per frame (Matlab implementation on a 3GHz Pentium) and the space-time alignment requires $\mathbf{1.12}$ **sec**. As a result, we envision online execution of the proposed algorithm in a GPU-based environment. The retrieval time of the SIFT-based histogram matching is 9.46 sec per frame, while SIFT-flow re-ranks the top-5 list in terms of the flow energy and register the frames in 160.5 sec (5×32.1). The method in [3] compares each input image to all reference images and the comparison is meaningless.

Please refer to **http://xanthippi.ceid.upatras.gr/people/evangelidis/DAGM2011/** for alignment videos of the real sequences.

7 Conclusions

A novel method for video alignment with applications in change detection was presented. This method enables the spatio-temporal alignment of similar videos captured from independently moving cameras. We proposed an efficient method adopted from information retrieval that applies short-length descriptors of frame content for video synchronization and a spatio-temporal alignment scheme for accurate change detection between synchronized frames. We experimented with a series of real world traffic videos captured from within moving vehicles. Our results verified both the efficiency and the effectiveness of the proposed method. Although we aim at driver assistance and security scenarios, the proposed framework obviously also applies to problems such as automated 3D map building or visual odometry.

Acknowledgements. This work has been funded by ERCIM. We also thank the ADAS group of the Computer Vision Center of Barcelona (Spain) for video data sharing and Ferran Diego for discussion.

References

1. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications, 2nd edn. Springer, Heidelberg (2000)
2. Caspi, Y., Irani, M.: Spatio-temporal alignment of sequences. IEEE Trans. Pattern Analysis and Machine Intelligence 24(11), 1409–1424 (2002)
3. Diego, F., Ponsa, D., Serrat, J., Lopez, A.M.: Video alignment for change detection. IEEE Trans Image Processing PP(99) (2010) (preprint)
4. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. IEEE Trans. Pattern Analysis and Machine Intelligence 30(10), 1858–1865 (2008)
5. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
6. Lang, D., Hogg, D.W., Mierle, K., Blanton, M., Roweis, S.: Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. The Astronomical J. 37, 1782–2800 (2010)
7. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: Dense correspondence across different scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008)
8. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
9. Rao, C., Gritai, A., Shah, M., Syeda-Mahmood, T.: View-invariant alignment and matching of video sequences. In: Proc. ICCV (2003)
10. Sand, P., Teller, S.: Video matching. ACM Trans. Graphics 22(3), 592–599 (2004)
11. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. Int. J. Computer Vision 37(2), 151–172 (2000)
12. Sivic, J., Zisserman, A.: Efficient visual search of videos cast as text retrieval. IEEE Trans. Pattern Analysis and Machine Intelligence 31(4), 591–606 (2009)
13. Tuytelaars, T., Gool, L.C.: Synchronizing video sequences. In: Proc. CVPR (2004)
14. Wolf, L., Zomet, A.: Wide baseline matching between unsynchronized video sequences. Int. J. Computer Vision 68(1), 43–52 (2006)