Efficient Subframe Video Alignment Using Short Descriptors

Georgios Evangelidis, Christian Bauckhage

Efficient Subframe Video Alignment Using Short Descriptors

Georgios D. Evangelidis* and Christian Bauckhage‡

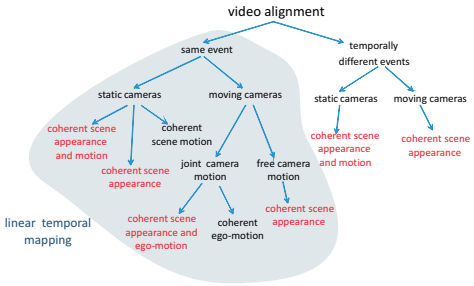
***Abstract*— This paper addresses the problem of video alignment. We present efficient approaches that allow for spatiotemporal alignment of two sequences. Unlike most related works, we consider independently moving cameras that capture a 3D scene at different times. The novelty of the proposed method lies in the adaptation and extension of an efficient information retrieval framework that casts the sequences as an image database and a set of query frames, respectively. The efficient retrieval builds on the recently proposed quad descriptor. In this context, we define the 3D *Vote Space* (VS) by aggregating votes through a multiquerying (multiscale) scheme and we present two solutions based on VS entries; a causal solution that permits online synchronization and a global solution through multiscale dynamic programming. In addition, we extend the recently introduced ECC image-alignment algorithm to the temporal dimension that allows for spatial registration and synchronization refinement with subframe accuracy. We investigate full search and quantization methods for short descriptors and we compare the proposed schemes with the state of the art. Experiments with real videos by moving or static cameras demonstrate the efficiency of the proposed method and verify its effectiveness with respect to spatio-temporal alignment accuracy.**

***Index Terms*— Video synchronization, spatio-temporal alignment, image/video retrieval, short image descriptors**

I. Introduction

Video alignment aims at finding point correspondences between two video sequences, namely a *reference* and an *input* sequence. It, therefore, extends the image alignment problem because correspondences have to be established in both space and time. That is, if $\mathbf{x} = [x, y, t]^\top$ and $\hat{\mathbf{x}} = [\hat{x}, \hat{y}, \hat{t}]^\top$ denote corresponding spatiotemporal input and reference points, henceforth called *vixels* (Video Picture Elements), a mapping has to be determined such that $\mathbf{x} = \Phi(\hat{\mathbf{x}}; \mathbf{p}_s, \mathbf{p}_t)$, with $\mathbf{p}_s$ and $\mathbf{p}_t$ being the spatial and temporal parameters, respectively. Accordingly, video alignment can be cast as a parameter estimation problem, but, in contrast to the related 3D alignment problem, it merges spatial and temporal aspects, which should be treated differently [1].

Fig. 1a summarizes assumptions and constraints that are commonly adopted in video alignment applications. A critical distinction is whether or not a temporal overlap between the two recorded sequences exists. If reference and input sequence show the same event and are recorded simultaneously (same event), there will be a global affine temporal transformation $t = \alpha\hat{t} + \tau$ that determines correspondences between the indices t and $\hat{t}$, regardless of scene content or camera motion. In addition, for static or jointly moving cameras there will be a spatial transformation between pixels of *temporally corresponding* frames that remains fixed for the duration of the videos. If there is no spatial overlap between the fields of view (FOV) of the cameras, e.g. because they are facing each other or observe adjacent non-overlapping FOVs [2], any alignment algorithm has to track moving scene objects or ego-motion in order to be able to align the sequences.



(a)

rigidity of spatio-temporal parameters

	same event	different events
static cameras	fixed $\mathbf{p}_s$ fixed $\mathbf{p}_t$	fixed $\mathbf{p}_s$ varying $\mathbf{p}_t$
moving cameras	varying $\mathbf{p}_s$ fixed $\mathbf{p}_t$	varying $\mathbf{p}_s$ varying $\mathbf{p}_t$

(b)

Fig. 1: (a) Taxonomy of assumptions (internal nodes) and constraints (leaves) in video alignment. Paths to red leaves are covered by the proposed algorithm. Note that if videos are recorded simultaneously (shadowed area), the temporal mapping between two sequences is linear; if videos are recorded at different points in time, frame-wise mapping is required. (b) Rigidity of spatial and temporal parameters $\mathbf{p}_s$ and $\mathbf{p}_t$ of the vixel-correspondence model $\mathbf{x} = \Phi(\hat{\mathbf{x}}; \mathbf{p}_s, \mathbf{p}_t)$ along the sequences (*see text*).

Video alignment becomes more challenging if the two sequences are recorded at different points in time (temporally different events). In this case, there is no rigid temporal mapping between frames anymore. Nevertheless, as long as both videos show similar content and egomotion, they can still be synchronized based on an identification of similar frames.

A. Objective

In this paper, we address the problem of aligning videos of the same scene, which are captured by independently moving cameras that follow similar trajectories at different times. A canonical example for this setting are videos obtained from in-vehicle cameras mounted behind the windshield (see Fig. 2). Our goal is a vixel-wise alignment of videos for which a pixel-wise alignment of corresponding frames is technically possible because they were recorded from cameras of comparable FOV and viewing direction. This essentially generalizes the case of simultaneously recording the same scene but is critically different in that temporally unrelated videos may show different scene objects so that corresponding frames can actually look rather

* Georgios D. Evangelidis is with Perception Team, INRIA Rhone-Alpes, 38330, France, email:georgios.evangelidis@inria.fr

‡Christian Bauckhage is with Fraunhofer IAIS, 53754 St. Augustin, Germany, email:christian.bauckhage@iais.fraunhofer.de

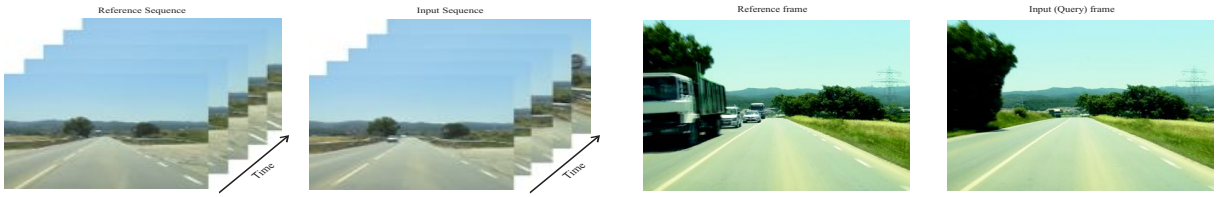


Fig. 2: (*Left*) An example of two video sequences (*Backroad* dataset [8]). Due to non-overlap in capture time, different moving objects appear in sequences. (*Right*) Corresponding frames of substantially distinct visual appearance.

different. For moving cameras, this problem can be addressed by considering the static background in order to guide the alignment; for stationary cameras, it is the motion of scene objects that informs the solution. Note that in the pathological case of a static scene and unrelated cameras trajectories, a reasonable spatiotemporal alignment cannot be expected to be obtained.

In what follows, we assume uncalibrated cameras with fixed but unknown internal parameters. While the technical specification of the cameras might differ, they should be comparable in the sense that videos which they record are of comparable quality and appearance. Moreover, we do not consider any calibration step and the computation of temporal alignments disregards timestamp information and is guided *solely* by visual similarities between frames. Our prerequisite of similar trajectories guarantees that frames which are captured from similar viewpoints have sufficient overlap in their FOVs; the extent of this overlap is discussed in section VII.

In addition to subpixel accuracy in spatial alignment, a temporal alignment with subframe accuracy is an important goal in video alignment [1]. In order to register two sequences with sub-vixel accuracy (subpixel in space and subframe in time), we present an approach that extends a gradient-based image alignment scheme to spatiotemporal dimensions [3]. Although, under certain circumstances, direct methods may provide accurately aligned videos, they generally cannot cope with very large temporal displacements even if a multi-resolution scheme is used. Moreover, in our scenario, the temporal aspect is particularly important because the common FOV of cameras inherently initializes the spatial transformation and the initialization of $\mathbf{p}_t$ essentially provides the initialization of the parameters $\mathbf{p}_s$. Furthermore, even though $\mathbf{p}_t$ and $\mathbf{p}_s$ will vary over time and thus may require frequent (re-)estimation, we aim at real-time capability where, given a prerecorded reference video, we want to align each frame of the input video just after its acquisition. Consequently, we investigate the possibilities of very efficient schemes.

B. Overview

We formulate the synchronization of two temporally unrelated video sequences as an Information Retrieval (IR) problem where the reference sequence is preprocessed and indexed before the input sequence becomes available. This way, we make efficient use of the time that elapses between the two recordings; computations are pushed back to an offline procedure thus rendering the online task considerably less demanding [4]–[7]. Our approach is inspired by recent work of Lang *et al.* [7] who introduced a geometric hashing method to encode the relative position of neighboring stars for satellite imagery. Here, we apply a similar coding scheme to encode and characterize sets of neighboring interest points within a video frame. This coding enables us to efficiently index the reference sequence, to store the indices appropriately, and, finally, to synchronize the videos by querying the reference sequence using individual frames of the input sequence.

Furthermore, we extend this novel IR approach to synchronization towards a multiscale (multi-level) framework that provides high synchronization accuracy. In short, we aggregate votes for correspondences obtained from multiple queries (each per scale) and build a 3D Vote Space (VS). An appropriate temporal synchronization is then determined from VS entries either by investigating VS slices that consider only the input (query) frame or by taking into account the whole VS. By considering VS slices and counting only on the query frame, we obtain a causal (online) solution which is useful for real-time application. On the other hand, the consideration of the whole VS implies a global solution which is preferable for offline applications.

Having obtained a rough solution for the synchronization problem, we proceed toward the spatiotemporal alignment of the sequences with sub-vixel accuracy. That is, each input frame is spatially registered with a *subframe* of the reference video. Since similar sets of interest points may be visible in several successive reference frames, the best frame retrieved in the synchronization step may not be the *visually closest* reference frame. In addition, since video acquisition takes place at different times (e.g. different days) so that variations in visual appearance are to be expected, we refine the retrieval result by adopting a recently proposed image alignment scheme (ECC algorithm) [3]. The ECC algorithm offers the desired robustness and we extend it towards the space-time dimension to obtain an approach that allows for simultaneous spatial registration and synchronization refinement with subframe accuracy.

C. Contribution

The video alignment problem arises in many applications of computer vision. Examples include vehicle detection for advanced driver assistance systems (ADAS) [8], high dynamic range video and video matting [9], frame dropping prevention [10], action recognition and sensor fusion [11], video-copy detection [12], and wide baseline matching [1], [13]. Moreover, spatiotemporal alignment can resolve several ambiguities of standard image alignment techniques [1] and, although we focus on registering similar videos captured at different times (e.g. captured by in-vehicle cameras that follow similar trajectories on different days (Fig. 2)), our solutions can be adopted to various other settings that require the spatiotemporal alignment of video sequences.

Our contributions in this paper are summarized as follows:

1) The proposed causal solution for video alignment can deal with dynamic camera motion. Unlike [8], we do not require assumptions or prior information as to the motion of cameras except for roughly similar trajectories. Since we synchronize each frame separately, our approach allows for

cameras to stop at any time or even to move backwards. Moreover, our approach enables online synchronization since the reference sequence can be queried while the input sequence is still being recorded.

2) We apply the recent *quad descriptor* [7] for rough video synchronization. This expressive, low dimensional local feature descriptor allows for very efficient retrieval schemes based on kD-trees and thus establishes frame correspondences much faster than standard bag-of-words (BoW) approaches with high-dimensional descriptors [5], [6], [12].
3) In contrast to the spatiotemporal extension of [1] which leads to a sequence-to-sequence alignment scheme, we also investigate an extension in the *parameter space only*. This is to say that the mapping $\Phi()$ applies temporal parameters determined from just a single frame and its temporal gradient. In other words, we develop a *frame-to-subframe* alignment scheme of low computational complexity since it supersedes considering many frames from each sequence.
4) The proposed scheme can easily be adopted *as is* to computer vision applications such as video copy detection [12] or object transfer (augmented reality) [9] which require spatiotemporal alignment. In addition, it provides a new retrieval solution for various systems (e.g. [6]) through application of multi-querying frameworks that fuse retrieval results obtained with different parameter settings. Finally, given pre-recorded reference videos, our framework can assist visual detection [14] by automatically marking areas of interest for further processing.

A preliminary solution to the problem in question was presented in [15]. Unlike the new multiscale approach, the quad descriptor was tested under a naive single-scale retrieval scheme. A local solution was only presented and tested on three driving sequences. Despite the non-in-depth evaluation in [15], the limitation of the single-scale framework becomes evident.

The remainder of this paper is organized as follows. In the next section, we review related approaches. Then, in Section III, we formulate the video alignment problem. In Section IV, we cast the video synchronization problem as an IR task and, based on this idea, we present a multiscale framework and two solutions in Section V. Section VI presents our spatiotemporal extension of the ECC alignment algorithm. In Section VII, we evaluate our scheme on several real world video sequences and compare the proposed solutions to standard methods. Finally, Section VIII concludes this work.

II. Related work

Most related contributions either assume stationary cameras or consider settings where cameras move jointly and are rigidly attached to each other [1], [2], [10], [11], [13], [16], [17]. Such scenarios are simpler than our setting, because a fixed spatial transformation between corresponding frames is guaranteed and need not be re-estimated at runtime. Once an event has been identified in two such videos, a temporal mapping between the sequences can be globally described by simple parametric models. Examples include a time offset model [1], [2], [10] to cope with unsynchronized acquisition, or a 1D affine model to account for different frame rates [1], [11], [13], [17], [18]. Assuming simultaneous recording [18]–[20], this kind of temporal rigidity is preserved even for independently moving cameras. If the acquisition of related videos takes place at different points in time, previous work was concerned with nearly coincident camera trajectories [8], [9], [21], while in outdoor scenarios GPS data are integrated [8], [21].

Video synchronization of temporally independent recordings is an important problem, because, once synchronized sequences are available, the problem of video alignment reduces to several spatial image alignment tasks. Here, offline solutions align trajectories of tracked interest points along the sequences [1], [13], [16], [19], [20]. Feature-based matching [9], [18], [21] or direct methods [1], [2], [8], [11], too, have been extended to video matching. Typically, corresponding approaches consider a spatial mapping model such as a 2D homography [1], [2], [21], a fundamental matrix [1], [2], [10], [16], [20], affine transforms [11], 3D rotations [8], or the trifocal tensor [19] to describe the relation between corresponding frames.

Our scenario is most closely related to the work in [8], [9]. Sand and Teller [9] proposed an exhaustive search between frames by looking for motion-consistent pixel matches using a regression model and Diego *et al.* [8] cast video synchronization as a MAP inference problem. The latter adopts the Lucas-Kanade alignment algorithm to spatially register synchronized frames. In a similar manner, Caspi and Irani [1] extended feature-based and area-based image alignment schemes to the space-time dimension, and, in [2], they used intra-sequence transformations to recover spatial and temporal parameters in non-overlapping sequences. Our framework differs from these approaches in that it provides a more efficient scheme for (online) sub-vixel video alignment.

Video alignment scenarios where cameras are moving bear a certain similarity to the problem of robot localization based on video data. In essence, video synchronization can be viewed as a by-product of vision-based simultaneous localization and mapping (SLAM) [22], [23]. Approaches to video-based SLAM attempt to link novel frames to previously recorded ones that were captured from nearby viewpoints. The methods in [22] and [23] use visual words representations [6] to extract similar frames from a database of landmark images. The BoW paradigm is also used in [12] where the video alignment is applied to detect copied video material, and in [5] in order to retrieve images of similar but different scenes. In particular, the latter proposes a flow-based alignment, called SIFT-flow, to spatially register corresponding frames. However, while SIFT-flow and similar methods can be used for computing general non-rigid alignment between two images [5], they are sensitive to visual occlusions and too computationally demanding to allow for robust and fast video alignment.

III. Problem Formulation

Suppose we are given a reference sequence $I_r(\hat{\mathbf{x}})$ and an input sequence $I_q(\mathbf{x})$ contained within the reference one, where $\hat{\mathbf{x}} = [\hat{x}, \hat{y}, \hat{t}_n]^\top$, $\mathbf{x} = [x, y, t_m]^\top$ are their vixel coordinates and $n = 1, .., N$ and $m = 1, ..., M$ are their frame indices respectively. Further suppose that vixel correspondences are described by a mapping $\mathbf{x} = \Phi(\hat{\mathbf{x}}; \mathbf{p})$, $\mathbf{p} = [\mathbf{p}_s^\top,\ \mathbf{p}_t^\top]^\top$, which fuses the spatial and temporal warps parameterized by $\mathbf{p}_s$ and $\mathbf{p}_t$, respectively. Since we permit an irregular motion for the cameras, both $\mathbf{p}_s$ and $\mathbf{p}_t$ vary along the sequences and must be re-estimated for all input frames. For simplicity, we consider for the integer frames that $t_m = m$ and $t_n = n$, and we refer to the m^{th} input and n^{th} reference frame as I_m and I_n, respectively.

Fig. 3: (a) Geometry hashing via the quad structure and (b) a query frame with its extracted Harris points. (c) The valid quads of the query and (d) the reference corresponding frame; red dots denote the centroids of the quads.

We assume that the temporal mapping with subframe accuracy is expressed in terms of a discrete-time signal $T : \mathbb{N} \rightarrow \mathbb{R}$, such that $(m, T(m))$ is an assignment of an input frame to a reference subframe. Given I_m, our primary goal is to find the frame I_n whose index n is as close as possible to the non-integer value $T(m)$, and then to recover the subframe index through the model $\Phi()$. To efficiently address the former, the task of roughly synchronizing two videos is cast as a retrieval problem. In particular, the reference frames define an *image database* and each input frame represents a *query image*. By querying the database with I_m, we retrieve the visually closest reference frame I_n (Sec. IV, V). Such a solution does not guarantee that $|n - T(m)| < 1$, since the visual features we use for indexing may be visible in several successive frames and we may not retrieve the correct frame. We address this issue by means of a subsequent spatiotemporal alignment step which refines the mapping (Sec. VI).

Given a pair (m, n), we consider the *temporally-local* subsequence $I_{n-\mu}, ..., I_{n+\mu}$ where μ is a small integer. After defining $\Phi()$ we look for the image warped in *space and time* from the above subsequence that aligns with I_m. To this end, we extend the ECC alignment algorithm [3] to the space-time dimensions, i.e. the extended scheme estimates the spatiotemporal parameters $\mathbf{p}$ that maximize the correlation coefficient between the input frame $I_q(\mathbf{x})$ and the warped reference subframe $I_r(\Phi(\hat{\mathbf{x}}; \mathbf{p}))$. It is important to note that such an extension can yield an image-to-sequence alignment rather than a sequence-to-sequence alignment as proposed by Caspi and Irani [1]. The extension in [1] considers both the pixel and the parameter domains, thus linking many input frames to many reference subframes. Our image-to-sequence extension, however, regards only the parameter domain since it considers pixels contained in one input frame only, which is linked to a subframe of the subsequence. This way, we do not sensibly increase the complexity of the image alignment problem. If the cameras have different frame rates, we have to appropriately scale the reference temporal axis with the ratio of the rates. Note, however, that if the input frames are weakly-textured, sequence-to-sequence schemes may be preferable to image-to-sequence counterparts.

To summarize, once a temporal registration has been initialized by means of the efficient retrieval step, the ECC algorithm converges to that subframe that maximizes the value of the enhanced correlation coefficient [3]. In other words, the highest number of feature matches provides a rough synchronization, and the maximum similarity defines the final corresponding reference frame.

IV. An Information Retrieval approach to Video Synchronization

In this section, we adopt an IR approach to address the video synchronization problem. This approach allows us to preprocess the reference data in an offline step that does not require any knowledge as to the input sequence. Once the input sequence is available, this idea allows for fast yet reliable synchronization. In specific, the proposed method mainly critically depends on sets of short descriptors of the content of individual video frames. We adopt a geometric hashing method introduced in [7] for applications in astronomy. Instead of clusters of stars, we use the method to represent sets of spatially adjacent interest points. Such points are determined using the Harris detector [25] because of its favorable behaviour with respect to speed and repeatability [27]. That is, given a grayscale image $I(\mathbf{y})$, $\mathbf{y} = [x, y]^\top$, we determine local maxima of the cornerness measure $C(\mathbf{y}; \sigma_D) = \det(\Sigma(\mathbf{y}; \sigma_D)) - 0.04 \cdot \mathrm{trace}(\Sigma(\mathbf{y}; \sigma_D))$, where Σ is a matrix of second order moments computed at pixel $\mathbf{y}$ where moments are parameterized by the differentiation scale σ_D [28], [29].[1]

Once interest points have been computed, we determine quadruples of nearest neighbors of such points in order to describe local image structures. This is done as follows: Suppose a quadruple ("quad") of interest points $\mathbf{y}_i, i = \{1, 2, 3, 4\}$ as shown in Fig. 3. The points $\mathbf{y}_1, \mathbf{y}_2$ are the *control points* defined as the most widely separated pair of points. Let d denote the distance (*diameter*) between the control points, φ the *orientation* of the diameter and $\mathbf{c}$ the *centroid* of this quad, that is

$$d = \|\mathbf{y}_1 - \mathbf{y}_2\|, \qquad \varphi = \tan^{-1}\frac{y_2 - y_1}{x_2 - x_1}, \qquad \mathbf{c} = \frac{1}{4}\sum_i \mathbf{y}_i \ , \quad \text{(1a-c)}$$

where $\| \cdot \|$ denotes the L_2 norm. We then consider a local coordinate system O_{XY} oriented and centered with respect to the control points $\mathbf{y}_1$, $\mathbf{y}_2$, so that they coincide with the points $(0, 0)$ and $(1, 1)$, respectively. This allows us to encode the quad structure in terms of the new coordinates of the remaining points $\mathbf{y}_3, \mathbf{y}_4$. Accordingly, any quad of four points can be represented by means of a 4D vector $\mathbf{q}$ which is called a *quad descriptor*, or simply a *quad*. In essence, such a coding realizes a *similarity normalization transform*, i.e. the descriptor is invariant to any scale, rotation and translation of points. We refer the reader to Appendix I for the detailed definition of this transform.

Similar to [7], we only regard quadruples where $\mathbf{y}_3, \mathbf{y}_4$ lie inside the circle of diameter d. Any permutation of the order of points in the pairs $(\mathbf{y}_1, \mathbf{y}_2)$ and $(\mathbf{y}_3, \mathbf{y}_4)$ creates a symmetry that can be easily resolved. In addition, any location error of the interest point detector yields a small error in the position of

[1]The second-moment matrix is parameterized by an integration scale as well, but the scale ratio is usually constant and only one independent scale can be considered.

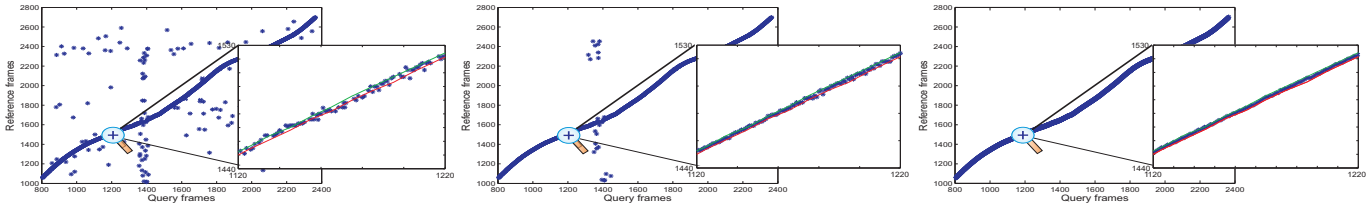


Fig. 4: (a) Synchronization for the *Campus* sequences [8] based on the initial votes and (b) after filtering for spatiotemporal consistency ($R_c = 100$, frame resolution: 720×540). In (c), the temporal mapping obtained by the proposed multiscale approach (Sec. V) is shown; the zoom-ins also show the bounds of the available ground-truth for this data.

the code inside the 4D quad space. Finally, note that geometric hashing based on quads benefits from uniformly distributed descriptors [7]; applying non-maximum suppression during interest point computation leads to well distributed interest points [30] and thus to well distributed quads. Fig. 3 shows the quads of two corresponding frames.

A. Indexing, Structure and Retrieval

In order to be able to determine reference frames that are similar to a given input frame, we compute Harris points (for a given scale σ_D) for all reference frames I_n and extract all valid quads. We denote by $\mathbf{q}_{nj}$ the j^{th} descriptor of I_n, where $j = 1, 2, \cdots, L_n$, and by $r_{nj} = \{n, \mathbf{c}_{nj}, d_{nj}, \varphi_{nj}\}$ its characteristics. Since the discriminative power of the quad descriptor is low, we do not apply any vector quantization step [6] but work with all available continuous hash codes. This is feasible, because the short length of descriptor length allows us to actually store all quads $\mathbf{q}_{nj}$ and their characteristics r_{nj}.

Any interest point detector will inevitably produce location and repeatability errors [27]. Therefore, given a query quad, we look for similar reference quads using a *near neighbor* (*range search*) approach. Since quad descriptors are of low dimensionality, it is possible to store them in a kD-tree structure so that near neighbor searches can be done efficiently.

The frame correspondence can be interpreted as a voting scheme: given frame I_m, let $\mathbf{q}_{mi}, i = 1, 2, \ldots, L_m$ denote its quads. By querying the database with $\mathbf{q}_{mi}$, any quad $\mathbf{q}_{nj}$ which is ε-close to $\mathbf{q}_{mi}$ in terms of the Euclidean distance is retrieved and votes for the image I_n. If v_{nm} denotes the votes of I_n when the database is queried by I_m and is initialized to 0, the vote score is updated by $v_{nm} \leftarrow v_{nm} + f(\mathbf{q}_{mi}, \mathbf{q}_{nj})$, where

$$f(\mathbf{q}_{nj}, \mathbf{q}_{mi}) = \begin{cases} \lambda_w, & \text{if } \|\mathbf{q}_{nj} - \mathbf{q}_{mi}\|_2 < \varepsilon \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

and ε is the distance tolerance for similar quads. The score λ_w can be chosen according to best practices in IR [24]. Here, we choose the *inverse document frequency* (IDF) as very common quads are not indicative of the image content. That is, we set $\lambda_w = log\frac{N}{N_{mi}}$, where N_{mi} is the number of retrieved images after querying $\mathbf{q}_{mi}$.

B. spatiotemporal Coherence

In order to eliminate false positive matches, we enforce a spatiotemporal coherence constraint during voting that relies on the assumption of approximately coincident trajectories. If the cameras do not have too different settings, it is reasonable to reject matches between quads whose centroids are not spatially close. To do this, we only accept those $\mathbf{q}_{nj}$ that satisfy $\|\mathbf{c}_{mi} - \mathbf{c}_{nj}\| < R_c$, where R_c is a radius threshold that defines a valid search area around $\mathbf{c}_{mi}$. Provided that the FOVs are sufficiently overlapped, this constraint favors both spatial and temporal coherence due to camera motion. For instance, the projection of a scene point undergoes a strong displacement when a vehicle turns, hence the temporal coherence is a by-product of the spatial one. Moreover, the constraint is not too strict as it concerns the centroids of the quads and not the interest points per se. If the images show partial overlap, this constraint can be replaced by a weak geometry constraint (Sec. VII-C). Depending on the intended application scenario, it would be easy to also incorporate scale- and/or rotation-consistency measures.

Fig. 4a and Fig. 4b show examples of synchronization results before and after enabling constraints. It is obvious that the spatial consistency leads to a less erroneous temporal mapping, but mismatches still remain. Aside from randomly distributed mismatches, we observe systematic errors at some intervals like the one around the 1400-th input frame. Such intervals contain highly-textured frames that provide a large number of non-discriminative quads, thus leading to mismatches even after filtering for spatial consistency. Clearly, the temporal mapping obtained by the local multiscale approach (Fig. 4c), that is described in the next section, is more robust and provides a more reliable and smoother synchronization.

V. A MULTISCALE APPROACH

In the single-scale framework presented in the previous section, interest points are detected using a Harris detector with a fixed scale parameter. Accordingly, the corresponding quad descriptors cannot faithfully describe scene content at different scales because interest points locations do vary with scales. Obviously, our framework would perform better if the optimum scale for each frame was known. Since estimating the optimal scale is a formidable task, one might resort to using a scale-invariant [31] or an affine detector [32] thus assigning a characteristic scale to each interest point. On the other hand, this idea would lead to quad descriptors containing points of different scales. As we found experimentally that quads consisting of scale-invariant or affine points led to a noticeably reduced synchronization accuracy, we do not consider scale-invariant interest point detectors but propose the use of a multiscale method.

Given that scene depth and content may drastically vary, different local areas in a frame might be *better* described at different scales. However, since a naive implementation of this idea may lead to different false positives (mismatched query frames) at different scale levels, we propose to combine the voting results

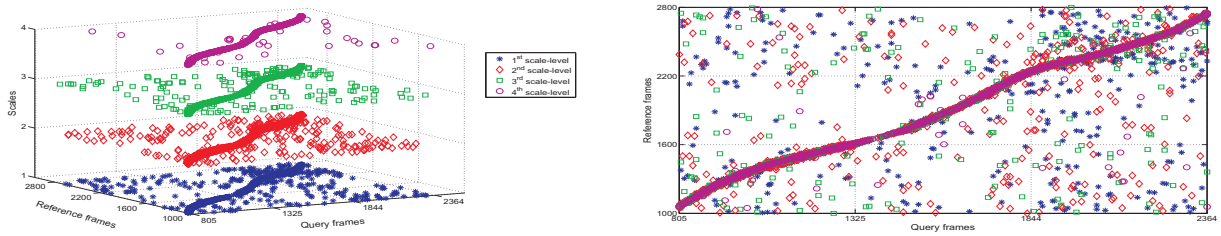


Fig. 5: (*Left*) A multiscale representation of votes in a multi-querying for the *Campus* dataset and (*right*) view from the top.

at all levels to improve retrieval precision and synchronization accuracy.

Suppose that we consider K scale levels with $\sigma_D = \sigma_k$, $k = 1, \ldots, K$, and that $\mathcal{O}_k$ and $\mathcal{I}_k$ are sets of false (FP) and true positive (TP) matches at the k^{th} level, respectively. That is

$$\mathcal{O}_k = \{m \mid m \in \mathcal{M} \text{ and } I_m \text{ is FP at the } k^{th} \text{ scale level}\}$$
$$\mathcal{I}_k = \{m \mid m \in \mathcal{M} \text{ and } I_m \text{ is TP at the } k^{th} \text{ scale level}\},$$

where $\mathcal{M} = \{m \mid m = 1, \cdots, M\}$, $\mathcal{I}_k \bigcap \mathcal{O}_k = \emptyset$ and $\mathcal{I}_k \bigcup \mathcal{O}_k = \mathcal{M}$. We then would *ideally* expect that

$$\bigcup_k \mathcal{I}_k = \mathcal{M} \quad \text{and} \quad \bigcap_k \mathcal{O}_k = \emptyset. \tag{3}$$

Although (3) does not imply that the final synchronization error will vanish, it provides a straightforward approach to its reduction and the final error takes into account votes across all scales.

Figure 5 illustrates a 4-level voting. Note that this representation does not indicate total numbers of votes and only displays instances where the more than one vote was cast. Apparently, there are less outliers and less votes in total for larger scales.

A. Vote Space

Having defined a set of $\sigma_k, k = 1, \cdots, K$, we apply the multiscale Harris detector, compute quad descriptors independently for each scale, and aggregate the votes per scale. These votes are stored in a discrete 3D *Vote Space* (VS). Let $V^{(3)}$ of size $N \times M \times K$ be such a space, so that each entry v_{nmk} defines the votes for reference frame I_n when the database is queried by input frame I_m at the k^{th} scale level. We also define a 2D vote space $V^{(2)}$ of size $N \times M$, which is essentially a projection of $V^{(3)}$ in two dimensions such that each element v_{nm} defines the total support for I_n given the query frame I_m.

To construct the VS, we investigate three different methods: 1) a method where all reference descriptors are stored in a kD-tree, 2) a visual dictionary (VD) method where each reference descriptor is assigned to a cluster (visual word), and 3) a bag-of words (BoW) scheme. Note that in 2) we only make use of inverted files, without representing images as vectors. By separately working at each scale, we have to build kD-trees, inverted indices, visual vocabularies or bag-of-words reference representations for all scale levels during the preprocessing step. However, since interest points are more frequently found on lower scales, the sizes of the corresponding databases or data structures drastically decrease for growing scale values.

What we describe below for each method is executed K times, thus setting the corresponding entries of the VS. The tree-based method is the procedure described in Sec. IV. The VD method relies on clustering for efficiency reasons, i.e. a *k-means* algorithm that uses Euclidean distances clusters the reference quads into visual words. An inverted file then summarizes the instances and their characteristics (frame index, location, diameter, and orientation) of all visual words. Once the query quad is mapped to the closest visual word, we vote for the linked reference frames using the inverted file. As with the tree-based method, we again apply IDF weighting and spatial-consistency checks.

The BoW model [4]–[6], [33] is widely recognized as a state-of-art retrieval method. Similar to [6], we represent each image by a weighted histogram of features whose bins correspond to visual words of the dictionary. The similarity between normalized input and reference histograms provides the VS entries. The short-list of the retrieval results is further filtered for spatial consistency, i.e. the percentage of putative matches between quads that are spatially consistent is added to the similarity value for the final vote score. This approach differs from the previous methods, because, now, the spatial coherence constraint is activated after voting.

As long as our primary goal is a *causal* system for rapid online alignment, we cannot synchronize frames after constructing the VS. However, if a non-causal solution for more accurate alignments is appropriate, we can estimate the temporal mapping once VS is available. Therefore, we also present a global solution based on the whole VS. In later sections, this enables us to assess the loss in performance when the synchronization is based on a single query frame only.

B. A Causal solution

When a causal system is required, we propose to only consider a subpart of a given VS. Recall that we generally treat each frame separately, in order to obtain robustness against effects of camera motion. Accordingly, we choose the subpart of VS to be that vertical slice of size $N \times K$ that corresponds to the query frame.

Since the number of interest points decreases with scale [30] and thus also the number of descriptors, we observe fewer votes at higher scale levels than at lower ones (see again Fig. 5). In order not to favor lower levels, we apply a *scale-adapted* voting scheme: Suppose that we query the databases with the m_0^{th} frame and concentrate the votes v_{nm_0k}. Then, the total support v_{nm_0} of the n^{th} reference image is given by the following convex combination of votes (linear projection of $V^{(3)}$)

$$v_{nm_0} = \sum_{s=1}^{K} \zeta_k v_{nm_0k} \ , \qquad n = 1, \cdots, N, \tag{4}$$

where $\sum_k \zeta_k = 1$, $\zeta_k > 0$, $k = 1, \cdots, K$. Convexity implies that $\zeta_k = \sigma_k / \sum_k \sigma_k$, which is the case we consider here. Generally, we could search more analytically for the optimum weights (e.g. by mixture models), but this is beyond the scope of this paper. The votes in the BoW model are, in a sense, scale-invariant because of the vector normalization. This suggests to only use a simple summation for this case, i.e. $\zeta_k = 1/K$. Once the votes v_{nm_0}

have been computed, the optimum reference index n^o is found to be

$$n^o = \arg\max_n v_{nm_0}, \tag{5}$$

and the same applies to all the other query frames of the input sequence.

C. A Global solution

When the online capability is not the most important practical requirement, a global solution can take advantage of the successive nature of video frames. Here, we propose a multiscale Dynamic Programming (mDP) method to estimate the optimal path (polyline) between reference and query indices.

Assume that each candidate correspondence (m, n) between I_m and I_n denotes an endpoint $e_i,\ i = 1 \cdots, N_E$ of a polyline $\mathfrak{L}$, that is $\mathfrak{L} = \{e_1, e_2, \cdots, e_{N_E}\}$ in the grid $N \times M$. Polyline $\mathfrak{L}$ ideally contains one endpoint per column, i.e. $N_E = M$, where there are, theoretically, N candidates. Since we are concerned with votes here, our DP variant looks for the *maximum-vote path* across a vote-plane. If the function $F_v(e_1, e_2, \ldots, e_i)$ denotes the vote-score of the path from e_1 to e_i, the goal is to estimate the polyline $\mathfrak{L}^o$ such that the vote-score is maximized, i.e.,

$$\mathfrak{L}^o = \arg\max_{\mathfrak{L}} F_v(\mathfrak{L}). \tag{6}$$

The function $F_v(e_1, \ldots, e_{N_E})$, or $F_v(e_1, e_{N_E})$, is defined by the aggregation of local votes along the path but is penalized by the inverse distance of subsequent endpoints, i.e.

$$F_v(e_1, e_{N_E}) = v(e_1) + \sum_{i=2}^{N_E} \big(v(e_i) + \frac{1}{\hat{d}(e_i, e_{i-1})}\big) \tag{7}$$

where $v(e_i)$ corresponds to the votes of e_i and $\hat{d}(e_i, e_{i-1})$ is the Euclidean distance between e_i and e_{i-1}. Based on DP theory [34], the problem reduces to a sequence of subproblems through the following recursion mode. Equation (7) can be rewritten as

$$F_v(e_1, e_{N_E}) = F_v(e_1, e_{N_E-1}) + v(e_{N_E}) + \frac{1}{\hat{d}(e_{N_E}, e_{N_E-1})} \tag{8}$$

and, accordingly, (6) reduces to a multistage problem

$$\mathfrak{L}^o = \arg\max_{\ell} \{F_v(\ell) + v(e_{N_E}) + \frac{1}{\hat{d}(e_{N_E}, e_{N_E-1})}\},\ \mathfrak{L} = \{\ell, e_{N_E}\}. \tag{9}$$

Our mDP scheme first suggests to project $V^{(3)}$ into a 2D space $\widehat{V}^{(2)}$ in a non-linear manner and then to search for the maximum-vote path based on the above recursion. The projection amounts to the suppression of the scale dimension resulting in a 2D vote plane, where the above solution applies. This way, each entry $\hat{v}_{nm}$ of $\widehat{V}^{(2)}$ reflects the support $v(e_i)$ of a candidate endpoint. The non-linear projection of $V^{(3)}$ is given by

$$\hat{v}_{nm} = \max_k \zeta_k v_{nmk} \quad , k = 1, \ldots, N, \tag{10}$$

where ζ_k is defined as above. In other words, each candidate endpoint of the DP matrix is associated with K nodes in the scale dimension and the mDP algorithm considers the maximum vote per node (see Fig.6 for a 3-scale example). As a consequence, the DP matrix is filled based on (10) and the final time mapping is established by back tracking the optimal path through (9).

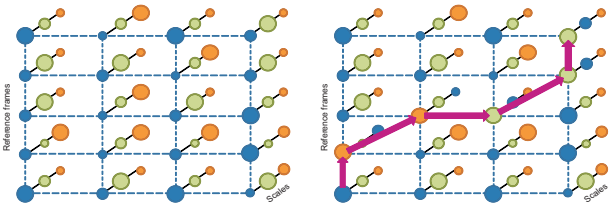


Fig. 6: (*Left*) A DP matrix with each node assigned to 3 votes (3-scale VS); the size of circles reflects the relative magnitude of votes in scale dimension. (*Right*) mDP enables the maximum vote assigned to each node towards the maximum-vote path.

D. Complexity - Efficiency

Recent works provide very efficient schemes for the Harris detector [35] and quads can be computed in linear time. Given N reference frames and L quads per frame on average, we can then show that the tree-based method requires $O(L^{\frac{7}{4}} N^{\frac{5}{4}})$ for retrieving a frame. This time amounts to $O(LW)$ and $O(LW^{\frac{3}{4}} + NW)$ for the VD-based and BoW-based method respectively, where W is the length of the visual dictionary. We can easily prove that for a database of $100K$ frames and typical values for L and W of, say, 500 and 1000 respectively, the VD-based method achieves a $100\times$ speedup compared to the standard BoW scheme that uses L SIFT features per frame and a dictionary of length W. Optionally, we can exploit the sequential form of the data, split the reference sequence into subsets of successive frames, and build a tree structure for each subset [15]. This way, we would obtain a forest structure and the query time would be further reduced.

VI. Subframe Video Alignment

Retrieval-based synchronization provides us with a sequence of pairs (m, n) that indicate a rough mapping between frame indices. Yet, our overall goal is to estimate a sequence of pairs $(m, T(m))$ where $T(m) \in \mathbb{R}$. Both, synchronization refinement with subframe accuracy and spatial alignment can be obtained simultaneously. To this end, we propose an extension of a recently presented image alignment algorithm [3] called ECC (Enhanced Correlation Coefficient). In its original form, ECC accomplishes only spatial alignments, here, we extend it to the space-time dimension.

ECC-based schemes are capable of compensating for illumination variations that are due to different recording times. Suppose that $\mathcal{A} = \{\mathbf{x}_s | s = 1, 2, \cdots S\}$ is the set of vixels in the input image or sequence. The video alignment task is to find the corresponding set $\hat{\mathcal{A}} = \{\hat{\mathbf{x}}_s | \mathbf{x}_s = \Phi(\hat{\mathbf{x}}_s; \mathbf{p}),\ s = 1, 2, \cdots S\}$ in the reference sequence. The mapping that establishes correspondences cannot be arbitrary, rather, we need to explicitly define a spatiotemporal model $\Phi()$. Although the fundamental matrix fits to our scenario, its use only describes the motion of each pixel up to an epipolar line and implies extra effort for a pixel-wise correspondence scheme [36]. Moreover, the computation of the fundamental matrix is susceptible to errors and, in our scenario, this uncertainty becomes more severe because of the camera motion. We therefore approximate the spatial motion using a 2D homography while the temporal model involves a pure time-offset. If required, an affine temporal model reduces to a pure temporal translation as the scale parameter can be determined from the ratio of frame rates [1].

In this context, our spatiotemporal mapping is written as

$$\mathbf{x}' = \begin{bmatrix} \mathrm{H} & \mathbf{0}_3 \\ \mathbf{0}_2^\top\ \tau & 1 \end{bmatrix} \hat{\mathbf{x}}', \tag{11}$$

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

Fig. 7: An example of the proposed method for the *Highway* dataset. (a) A query frame and (b) the retrieved frame with the best score. The space-time alignment after (c) 1 and (d) 10 iterations. Differences between query and reference frame are illustrated using lawn-green and hot-pink colors.

TABLE I: Experimental dataset (pairs of video sequences).

sequence	illumination	temporal overlap	camera	camera motion	resolution (pixels)	frame rate (fps)
Backroad	similar	no	same	yes	720×540	25
Campus	similar	no	same	yes	720×540	25
Highway	similar	no	same	yes	720×540	25
Suburb	different	no	same	yes	586×426	30
PedZone1	similar	no	same	yes	600×300	25
PedZone2	different	no	different	yes	720×576 - 600×300	25-25
Wind	similar	yes	similar	no	384×288	25
Water	similar	yes	similar	no	384×288	25
Inria	different*	yes	different	no	812×612 - 640×360	12-30

*very different sensors and exposure settings

where H is a 3×3 homography, τ is the temporal offset and $\mathbf{x}'$, $\hat{\mathbf{x}}'$ are vectors containing homogeneous spatial and temporal coordinates (see Appendix II for details). As a result, the unknown parameters are gathered in a nine dimensional vector $\mathbf{p} = [\underbrace{h_1, h_2, ..., h_8}_{\mathbf{p}_s}, \underbrace{\tau}_{\mathbf{p}_t}]^\top$ ($h_9 = 1$) where we tacitly assume that τ has been initialized via the previous synchronization.

The extended ECC scheme leads to the problem of estimating the optimal $\mathbf{p}$, so that the correlation coefficient between the query and the *spatiotemporally warped* reference frame is maximized. By stacking the intensities of the points in $\mathcal{A}$ and $\hat{\mathcal{A}}$ and normalize them to zero-mean, we form the query image vector $\mathbf{i}_q = [I_q(\mathbf{x}_1), I_q(\mathbf{x}_2), \cdots, I_q(\mathbf{x}_S)]^t$ and the warped reference image vector, parameterized by $\mathbf{p}$, $\mathbf{i_p} = [I_r(\hat{\mathbf{x}}_1), I_r(\hat{\mathbf{x}}_2), \cdots, I_r(\hat{\mathbf{x}}_S)]^t$ Then, the objective function to be maximized is the *enhanced correlation coefficient* which is given by

$$\rho(\mathbf{p}) = \frac{{\mathbf{i}_q}^\top \mathbf{i_p}}{\|\mathbf{i}_q\| \|\mathbf{i_p}\|} \tag{12}$$

and inherently involves the spatiotemporal model.

Although the maximization of $\rho(\mathbf{p})$ constitutes a highly non-linear problem, it has been shown in [3] that, through application of an iterative scheme, a closed form solution can be obtained in each iteration. As with the familiar Gauss-Newton optimization, we assume that a nominal parameter vector $\tilde{\mathbf{p}}$ is known, such that $\mathbf{p} = \tilde{\mathbf{p}} + \Delta\mathbf{p}$ and we linearize the warped vector so that $\rho(\mathbf{p})$ is approximated by the function

$$\rho(\Delta\mathbf{p}; \tilde{\mathbf{p}}) = \frac{\mathbf{i}_q^\top [\mathbf{i}_{\tilde{\mathbf{p}}} + \mathrm{J}_{\tilde{\mathbf{p}}} \Delta\mathbf{p}]}{\|\mathbf{i}_q\| \sqrt{\|\mathbf{i}_{\tilde{\mathbf{p}}}\|^2 + 2\mathbf{i}_{\tilde{\mathbf{p}}}^\top \mathrm{J}_{\tilde{\mathbf{p}}} \Delta\mathbf{p} + \Delta\mathbf{p}^\top \mathrm{J}_{\tilde{\mathbf{p}}}^\top \mathrm{J}_{\tilde{\mathbf{p}}} \Delta\mathbf{p}}} \ , \tag{13}$$

where $\mathrm{J}_{\tilde{\mathbf{p}}}$ is the Jacobian of $\mathbf{i_p}$ with respect to $\tilde{\mathbf{p}}$ (see Appendix II).

Unlike least-squares optimization, the function $\rho(\Delta\mathbf{p}; \tilde{\mathbf{p}})$ remains non-linear after linearizing the warped profile. Its maximization, however, results in an analytic formula and the correction vector is

$$\Delta\mathbf{p} = (\mathrm{J}_{\tilde{\mathbf{p}}}^\top \mathrm{J}_{\tilde{\mathbf{p}}})^{-1} \mathrm{J}_{\tilde{\mathbf{p}}}^\top \left\{ \frac{\mathbf{i}_{\tilde{\mathbf{p}}}^\top \mathrm{P}_\mathrm{J} \mathbf{i}_{\tilde{\mathbf{p}}}}{\mathbf{i}_q^\top \mathrm{P}_\mathrm{J} \mathbf{i}_{\tilde{\mathbf{p}}}} \mathbf{i}_q - \mathbf{i}_{\tilde{\mathbf{p}}} \right\} \ , \tag{14}$$

where $\mathrm{P}_\mathrm{J} = \mathrm{I} - \mathrm{J}_{\tilde{\mathbf{p}}} (\mathrm{J}_{\tilde{\mathbf{p}}}^\top \mathrm{J}_{\tilde{\mathbf{p}}})^{-1} \mathrm{J}_{\tilde{\mathbf{p}}}^\top$ is a projection matrix. [2]

As a result, $\rho(\mathbf{p})$ is maximized through a chain of secondary maximization problems whose solution obeys a closed form. Using successive parameter updates, we obtain the optimal $\mathbf{p}^o$ while $\Phi(\hat{\mathbf{x}}; \mathbf{p}^o)$ provides dense spatiotemporal correspondences with sub-vixel accuracy. The iterative procedure stops when a maximum number of iterations or a threshold for $\|\Delta\mathbf{p}\|$ is met. Further details on our extension are given in Appendix II.

The complexity of ECC has been analyzed in [3] and was shown to be $O(S\eta^2)$, where η is the number of parameters. Perspectives for lower complexity include a pyramid-based registration [1], a pixel selection scheme [37], and the inverse compositional alignment [3], [37]. Figure 7 shows an example of spatiotemporal alignment using the proposed scheme. Careful inspection of this figure confirms the refinement in synchronization (the adjustment in the dashed road line is not due to homography fitting only). In order to highlight changes with green and pink colors, we use a modified RGB representation [1], by replacing the G channel of the input frame with the G component of the reference frame, but warped in space and time based on the ECC outcome.

VII. Experiments

In order to evaluate the proposed framework for efficient video alignment and to compare it with several baseline methods from the literature, we consider nine pairs of real world video sequences. Table I lists the characteristics of these datasets. The sequences *Backroad*, *Campus*, *Highway* [8][3], *Suburb* have been captured by cameras attached to the windshield of moving cars.

[2] In [3], a two-case solution is presented; however, here we consider highly correlated image profiles and the conditions for the second case do not hold.

[3] These sequences are referred to as Backroad2, Campus2 and Highway2 in [8]

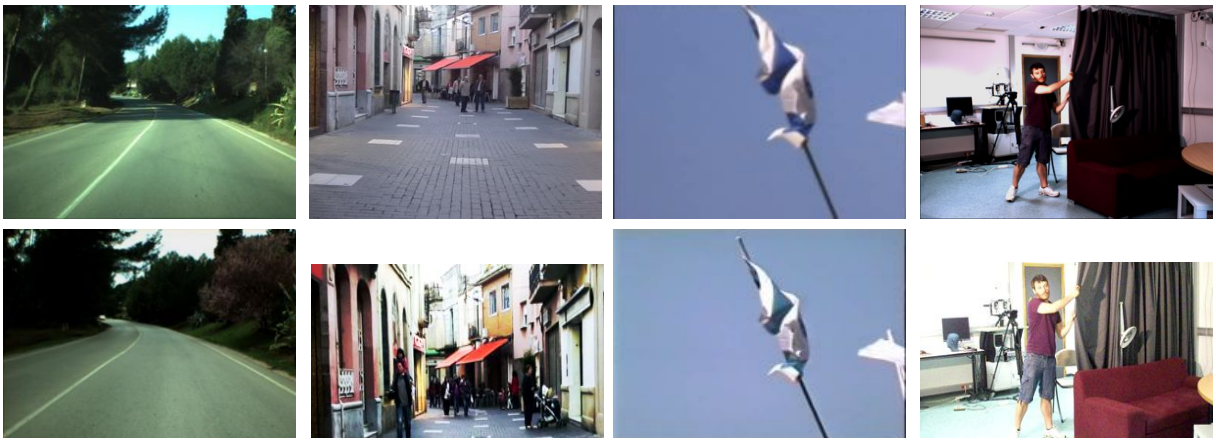


Fig. 8: *From left to right*: Temporally corresponding frames (*top-bottom*) of the sequences *Suburb*, *Pedzone2*, *Wind* and *Inria*.

During recordings, the drivers of these cars switched between driving styles and, in particular, executed *frequent sharp accelerations and decelerations* so that any temporal mapping between corresponding videos will be highly non-linear. In addition, the weather conditions and time of day in the *Suburb* sequences differ considerably. The two *PedZone1-2* sets were captured at different times by walking persons with hand-held cameras in a pedestrian zone. The former is captured by the same camera while the sequences of the latter are recorded by different cameras with various settings. The sequences *Wind*, *Water* [1] and *Inria* were taken by static cameras that capture a flag blowing in the wind, the water of a small fountain, and a person that moves a curtain respectively. Despite the cameras' stationarity and the absence of occlusions, these sequences are quite challenging because any alignment has to rely on the non-rigid motion in the scene. Moreover, the spatial texture in the *Wind* sequences is concentrated in a small area while the *Inria* sequences shows large variations in appearance due to different sensors. The length of each of these videos varies between 300 and 2000 frames. In Fig. 8, corresponding frames from the *Suburb*, *PedZone2*, *Wind* and *Inria* sequences are shown. Except for the car sequences, the spatial overlap is not complete due to the cameras' pose and settings. Finally, recall that we intentionally ignore the fixed spatial and rigid temporal mapping in the case of static cameras. It is obvious that the online alignment would reflect synchronized sequences up to frame accuracy.

For our data, synchronization ground-truth is available in the sense of lower and upper bounds. When the synchronization algorithm computes a reference frame index between the manually determined bounds we consider the match as true positive and no error occurs. Otherwise, we define the error to be the distance from the closest bound. The distance between the bounds is 3 frames on average, except for the *PedZone* sequences that contain more frames because of the slow motion. The performance of each method in our test is quantified using the synchronization error which is defined by the percentage of the false positives.

In what follows, we will refer to the three versions of the causal solution with respect to the VS building method, namely *Quad-Tree*, *Quad-VD*, and *Quad-BoW*. In addition, we shall use the suffix "*mDP*" to refer to the corresponding global variants of these methods. As for the baseline methods that are used for comparison, we consider the Caspi-Irani algorithm [1] which applies to the same initialization as the Quad-Tree scheme, the MAP-inference solution proposed by Diego et al. [8], a Dynamic Time Warping (DTW) algorithm which is a standard solution in sequence alignment [34], and a bag-of-words model based on the SIFT descriptor (SIFT-BoW) [5], [6], [22], [23].

We use a 6-scale framework with $\sigma_D = \sigma_k = 1.2\sqrt{1.8}^{k-1}, k = 1, ..., 6$. The distance tolerance ε in (2) is set to 0.1 and the radius R_c for the spatial constraint is 100 pixels. The size of the visual dictionary varies from 500 to 3000 with respect to the length of the sequences. In BoW modeling, a short-list of 50 reference frames is retrieved, where spatial consistency applies to. Regarding ECC and Caspi-Irani, we follow a coarse-to-fine framework with a 3-level Gaussian pyramid, permitting algorithm to execute 7 iterations per resolution level.

As discussed in Sec. I, Caspi-Irani algorithm is a sequence-to-sequence alignment algorithm. When the spatial transformation between corresponding frames is not fixed, sequences can be at most 3 frames long and the reference temporal axis has to be appropriately scaled to compensate for different frame rates. Therefore, the Caspi-Irani algorithm applies to 3-frame subsequences with the temporal mapping being initialized as in the Quad-Tree method. Below, a more detailed comparison will address issues due to various length of subsequences. As far as the SIFT-BoW method is concerned, we use the original SIFT algorithm, i.e. a DoG detector and SIFT descriptor, in conjunction with the method of [6]. We tested various distances between appropriately normalized vectors and –in agreement to [6]– we found the Bhattacharyya Coefficient to perform best. In contrast to [6], our experiments with Quad-BoW revealed a better performance if the L_1 distance is used. Similar findings regarding the BoW-model were earlier pointed out in [4]. Roughly speaking, our multi-level approach resembles in some sense the flexibility of [4], but the levels here are independent. The Map-inference method is implemented as described in [8] while the DTW implementation builds on a standard 3-step DP algorithm [34]. Both these methods use the above frame representation and distance in order to build the necessary similarity matrix (similarity scores are translated to distances for the DTW method as it needs a dissimilarity matrix).

The performance of the different algorithms is shown in Table II. The upper part of the table presents the causal (local) algorithms that base their decision on single frames whereas the Caspi-Irani method considers subsequences of 3 frames to perform refinement. The bottom part of the table presents results

TABLE II: Synchronization results

		Synchronization error (%)								
		Backroad	*Campus*	*Highway*	*Suburb*	*PedZone1*	*PedZone2*	*Wind*	*Water*	*Inria*
Local methods	Quad-Tree	8.52	10.8	**6.28**	**27.6**	**11.6**	**19.2**	**3.18**	**12.1**	**16.0**
	Quad-VD	**8.25**	**10.4**	7.82	28.2	13.6	20.9	5.45	24.2	32.9
	Quad-BoW	13.2	11.2	15.4	48.8	26.5	36.6	64.2	80.9	76.9
	Caspi-Irani	13.9	14.1	10.3	32.9	18.4	32.4	5.45	19.5	26.3
	SIFT-BoW	14.5	15.3	11.4	57.1	47.2	62.6	42.3	60.9	63.4
Global methods	Quad-Tree-mDP	**6.78**	8.53	**4.68**	23.7	**5.30**	15.4	**0.45**	**0.00**	**5.96**
	Quad-VD-mDP	7.61	**7.82**	5.51	**21.9**	6.82	**14.9**	**0.45**	**0.00**	14.9
	Quad-BoW-mDP	12.6	12.3	11.6	45.2	15.4	32.5	3.64	22.7	60.2
	DTW	26.0	18.3	29.2	60.6	41.9	47.1	20.2	77.3	56.2
	MAP-Inference	21.3	18.6	28.5	59.5	42.4	46.0	32.6	78.6	76.0

TABLE III: Performance of Quad-Tree algorithm assisted by a temporal window

	Synchronization error (%)								
Temporal window	*Backroad*	*Campus*	*Highway*	*Suburb*	*PedZone1*	*PedZone2*	*Wind*	*Water*	*Inria*
9 frames	8.16	8.06	5.69	23.9	8.84	19.9	0.37	3.14	4.92
15 frames	7.86	7.35	4.99	22.7	6.57	15.2	0.00	2.73	2.29
31 frames	6.54	6.71	2.74	21.4	6.45	13.1	0.00	0.00	1.71
all frames	-	-	-	-	-	-	0.00	0.00	0.00

for global algorithms that exploit the temporal continuity of whole video sequences. We observe that the proposed schemes outperform the competitors. The Quad-Tree algorithm is the best local algorithm on average and even outperforms global solutions. Except for the *Inria* and *Water* videos where a longer dictionary may be needed, the quantized version Quad-VD gives scores very close to the Quad-Tree version. Although the Caspi-Irani method considers more frames, it does not fare better than the proposed spatiotemporal alignment scheme. SIFT-BoW behaves similar to Quad-BoW on average, but textured frames favor Quad-Bow and weakly-textured content favors SIFT-BoW. It is important to note that the SIFT-BoW model considerably benefits from spatial consistency [6]. This is also evident in [8] where SIFT-BoW was reported to fail unless spatial constraints were enforced. In addition, the spatial consistency of SIFT-BoW seems to contribute more than the temporal consistency in DTW and MAP-inference. It is also worth pointing out that our multiscale framework appears to be so robust that the spatial consistency becomes sometimes redundant, e.g. the error of the Quad-Tree method without constraints is 12.7% on the *Backroad* sequence. The advantage of the multiscale approach is evident in the global schemes as well. All versions of the proposed global scheme achieve better scores than MAP-inference and DTW solutions. In general, global solutions are more robust than the local ones in the presence of sparsely distributed outliers. However, isolated inliers within a chunk of mostly mismatches are wrongly matched by the global methods, as opposed to the local ones.

As far as the descriptor is concerned, it is obvious that a group of keypoints presents lower repeatability than the keypoints per se and therefore quads are less repeatable than SIFT features. On the other hand, a quad is more discriminative in the sense that it encodes more points. Since our scenario does not involve very different viewing angles and strong scale changes, Harris detector presents high repeatability which makes the quads repeatable too. SIFT features would contribute more than quads, if there would be such strong variations. Overall, the multiscale extension improves the matching accuracy by resolving ambiguities of quad matches that occur in a single-scale framework [15].

All algorithms show declining performance when the lighting of scene drastically changes. Errors for the *Suburb* sequence are higher than for the other car sequences. The same applies to the *PedZone2* and *Inria* sequences which contrasts to the performance for *Pedzone1* and *Water* or *Wind*, respectively. Since handling severe illumination variations between two related videos resembles the task of dealing with different modalities, such videos should probably be synchronized using global offline methods that align space-time trajectories [1]. Yet, in our experiments, even the global methods yield considerable temporal misalignments for the *Suburb* sequence. Finally, the partial overlap in the *PedZone1-2* and *Inria* sequences also negatively affects the corresponding results.

A. Locality Vs Globality

Local algorithms may fail when a single frame does not carry much information and its corresponding frame is difficult to be established without using temporal data [1]. This affects the spatial alignment as well. For example, *Wind* contains frames with a large homogeneous area and *Inria* includes a periodic scene motion. These properties easily lead to mismatches when the decision is based on single frames. Global algorithms take into account the temporal continuity of videos and are more robust. However, this raises an obvious question: How "global" should an algorithm be in order to resolve uncertainty? Can a temporal window compensate for less informative single frames?

To investigate this question, we combine the Quad-Tree approach with a temporal window of several frames. The mapping within the window is assumed to be linear and frame-wise matches are used from a RANSAC-based scheme [38], thus solving for the line equation that finally matches the central frame only. This applies to all input frames in turn by shifting the temporal window. Consequently, isolated outliers are rejected and the local temporal coherence provides smoother results.

Table III summarizes the performance of the Quad-Tree algorithm assisted by a temporal window of various sizes. The contribution of the local temporal coherence depends on the distribution of false positives. As with the global solutions, sparse false positives are resolved while successive errors may remain. Overall, the proposed local scheme benefits from the temporal

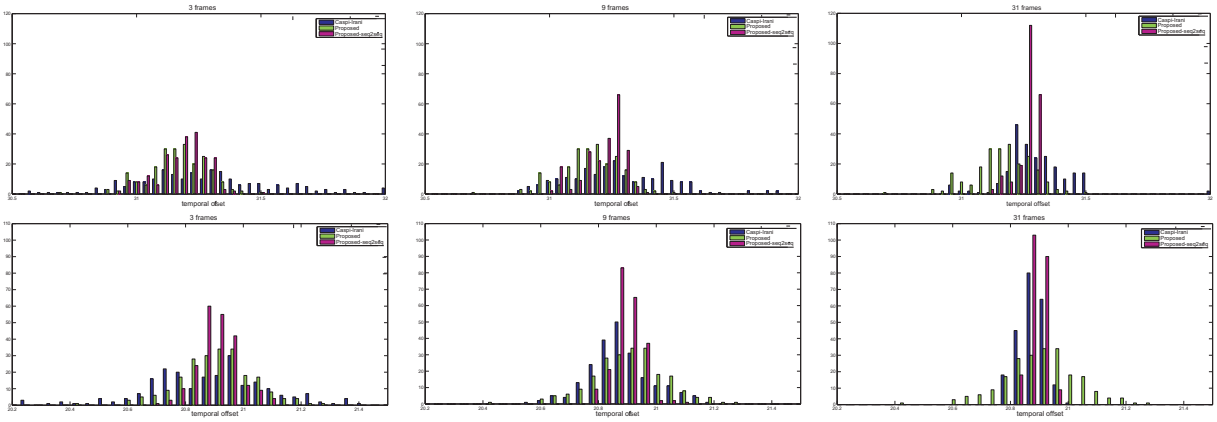


Fig. 9: The distribution of the subframe correction for (top) *Wind* and (bottom) *Water* sequences. "Proposed" stands for the pure frame–to–subframe alignment scheme while "Proposed-seq2seq" and Caspi-Irani are sequence–to–sequence alignment algorithms (the green distribution is the same for each row and is repeated for comparison).

window and competes with the proposed global solutions. In addition, when the cameras are static, a RANSAC-based line fitting that takes into account all frame matches can accurately align the sequences up to frame accuracy.

B. Proposed Vs Caspi-Irani

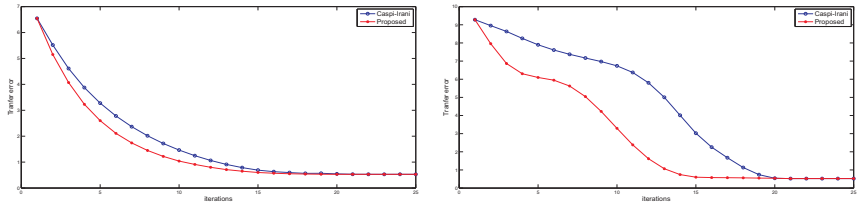
To the best of our knowledge, the sequence-to-sequence alignment as an extension of the image alignment scheme was first proposed by Caspi and Irani [1]. Our proposed method is a spatiotemporal alignment scheme as well, but it considers a frame-to-subframe alignment rather than the sequence-to-sequence problem in the Caspi-Irani algorithm. In order to compare these schemes, we consider the sequences *Wind* and *Water* provided by [1]. Since the cameras are static and of the same frame rate, the mapping is known up to frame accuracy, but even if it is unknown it can be obtained by a RANSAC-based line fitting. Using this initialization, we investigate here the ability of the algorithms to estimate the subframe correction by using subsequences or single frames in the spatiotemporal alignment framework. We also extend the proposed algorithm to a sequence-to-sequence mode by considering more frames (*Proposed-seq2seq*). The temporal offset distributions resulting from choosing subsequences of 3, 9, and 31 frames are shown in Fig. 9. Once again, we decide for the central frame of each subsequence while the next subsequence results from a one-frame shifting.

As long as the homography is not the ground transformation, the motion of the scene causes variations in the estimation of both spatial and temporal parameters. Apparently, the more frames we use in the subsequence, the less variance the distribution has. The comparison shows that the proposed scheme is more robust. The proposed frame-to-subframe scheme is more robust than even the Caspi-Irani algorithm with subsequences of 3 frames. The authors of [1] report that their implementation including all frames returns the offset 31.43 ± 0.1 for the *Wind* sequence. However, our Caspi-Irani implementation with 31-frame subsequences provides an offset of 31.29 on average (we used the compressed videos provided on the webpage referred to in [1]). The temporal offset obtained by the *Proposed-seq2seq* scheme with 31 frames is 31.27 on average while the frame-to-subframe scheme gives a translation of 31.17 frames. The superiority of the proposed algorithm is also shown by the results for the *Water* sequence. Our frame-to-subframe scheme is almost equivalent to the Caspi-Irani algorithm that uses subsequences with 9 frames. In addition, *Proposed-seq2seq* outperforms Caspi-Irani in all cases. Overall, the above comparison verifies the robustness of the spatiotemporal ECC algorithm in contrast to the least-squares approach of [1].

In order to quantitatively assess the spatial alignment, we apply a series of known spatiotemporal transformations in the input frames and investigate the ability of the algorithms to recover the warp. The deformation is a spatiotemporal model that smoothly varies with time. We consider the corners of a frame and find the homography between their coordinates and their noisy counterparts. The deviation of the point noise reflects the strength of the deformation. To differently warp each frame, we add to the 2×2 upper left part of the homography a scaled rotation matrix whose angle is a function of time, namely the cosine value is $\cos(0.04\pi m), m = 1, 2, ...$ and the scale is 0.1; the temporal offset varies as $0.25 + 0.25 \sin(0.04\pi m)$. To quantify the misalignment, we use the *space-time transfer error* of the four corners and we plot the error curve averaged over 5000 realizations on all frames as a function of the iteration. We compare our algorithm with the Caspi-Irani algorithm translated to the frame-to-subframe framework. Due to the full overlap, both algorithms run in a single resolution level executing 25 iterations. Before the alignment, we add zero-mean gaussian noise to the images with standard deviation equal to eight gray levels. Fig.10 shows the *learning ability* of the algorithms for all realizations that both algorithms have converged, i.e. the final transfer error is below 1 square vixel. Once a weak deformation occurs, both algorithms behave similarly but, for strong deformations, the learning rate of ECC is better. In addition, the proposed scheme converges more often and faster than its competitor, again, especially if strong deformations occur (Fig.10c). While these particular results were obtained for the *Backroad* footage, similar behavior was observed for the other sequences.

C. Partial overlap of images

In this subsection, we investigate the effects of lateral displacements between the viewpoints of the cameras. In our car sequences, the spatial overlap between corresponding frames is almost complete. We modify the sequences so that they are horizontally overlapped up to a certain percentage. The extent of the overlap is defined through the intersection-union area



Frequency of Convergence

	deformation	
	weak	strong
Caspi-Irani	89.6%	22.6%
Proposed	92.4%	72.2%

(c)

Fig. 10: Spatiotemporal transfer error as a function of iterations for (a) weak and (b) strong deformations; (c) the convergence frequency scores. The point noise is zero-mean gaussian with its deviation equal to 2 in (a) and 10 in (b) (see text).

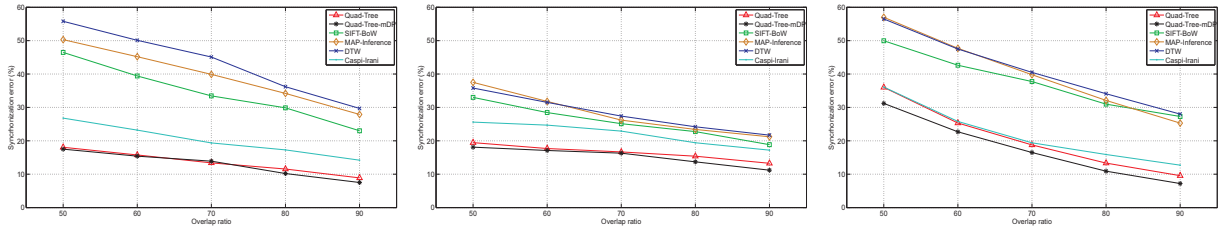


Fig. 11: Synchronization error curves ($\delta = 0$) for (a) *Backroad*, (b) *Campus* and (c) *Highway* sequences; the overlap-ratio varies from 50% to 90%.

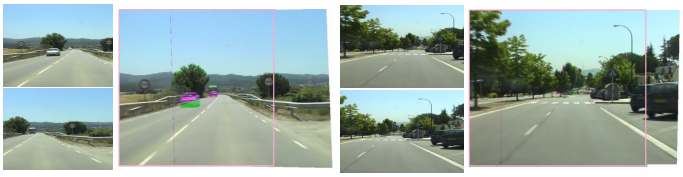


Fig. 12: Alignment of partially overlapped sequences. Input, retrieved and aligned frames by the Quad-Tree algorithm for the (*left*) *Backroad* and (*right*) *Campus* sequence with 50% and 70% overlap ratio respectively; the pink window marks the input frame.

ratio, called *overlap ratio*, with the smallest value being 50%. The common FOV does not regard near-camera objects and the current consideration resembles the real world situation (nearly parallel trajectories of cameras). A change in viewpoint can be seen as a camera panning causing scene points to move to the opposite direction in the image plane. Therefore, the space-time constraint is replaced by a weaker one, i.e. we simply use the sign consistency for horizontal coordinates while a reasonable threshold applies to vertical differences. The ground sign is easily extracted by the majority of matches.

Table IV summarizes observed synchronization errors when the overlap ratio is 70%. The synchro curves (in terms of the overlap ratio) of the competing algorithms are shown in Fig.11. As expected, the error decreases with growing overlap. We observe

TABLE IV: Synchronization results (Overlap Ratio: 70 Percent)

		Synchronization Error (%))		
		Backroad	*Campus*	*Highway*
Local methods	Quad-Tree	**13.4**(12.2*)	16.6 (17.6*)	**18.7** (15.6*)
	Quad-VD	14.0	**16.5**	23.1
	Quad-BoW	19.5	21.0	34.4
	Caspi-Irani	19.3	22.2	19.1
	SIFT-BoW	33.5	25.1	37.7
Global methods	Quad-Tree-mDP	**13.9**	16.3	**16.5**
	Quad-VD-mDP	14.9	**15.7**	17.0
	Quad-BoW-mDP	18.5	20.4	24.2
	DTW	45.1	27.4	40.5
	MAP-Inference	39.9	26.2	39.8

*with local temporal coherence (15-frame temporal window)

that the proposed framework is more robust than the SIFT-BoW model. Other than the benefit from the multiscale framework, we attribute this to the fact that occluded areas resemble overlapping ones, and, in that sense, the quad descriptor may yield higher distinctiveness than SIFT as it characterizes groups of interest points. Fig. 11 also shows that the loss in performance varies with the image content. For example, the small number of features (quads) in *Highway* footage seems to be advantageous when the frames are fully overlapped, while it is disadvantageous when a partial overlap occurs. Again, when the temporal mismatches are randomly distributed, local algorithms benefit from temporal windowing as introduced above.

Figure 12 displays examples of partially overlapping images and their alignment. The ability of the proposed algorithm to produce wide-field videos from narrow-field cameras becomes apparent from this experiment.

D. Qualitative comparison

Next, we visually compare the algorithms with several alignment instances. For actual video results, we refer the reader to the supplemental material. Apart from the ECC and Caspi-Irani algorithms, the comparison includes SIFT-flow which is combined with SIFT-BoW as presented in [5]. SIFT-flow accounts for alignment of different scenes, but it also allows for pixel-based image registration [5]. It returns the 2D flow that warps the reference image with respect to the input frame. Figure 13 shows alignment results obtained from the algorithms. It also hints at the ability of these algorithms to be tailored towards change detection when different scene objects cause occlusions. SIFT-flow creates artifacts leading to truncated objects when they are not visible in both FOV and the Caspi-Irani method, too, seems to be more affected by occlusions than the proposed scheme.

In Fig.14, we visually demonstrate the contribution of more frames in the estimation of the spatiotemporal transformation. We depict the spatiotemporal alignment of "integer" *temporally corresponding* frames for the challenging sequences *Wind* and *Inria*. It is evident that the Caspi-Irani approach requires more

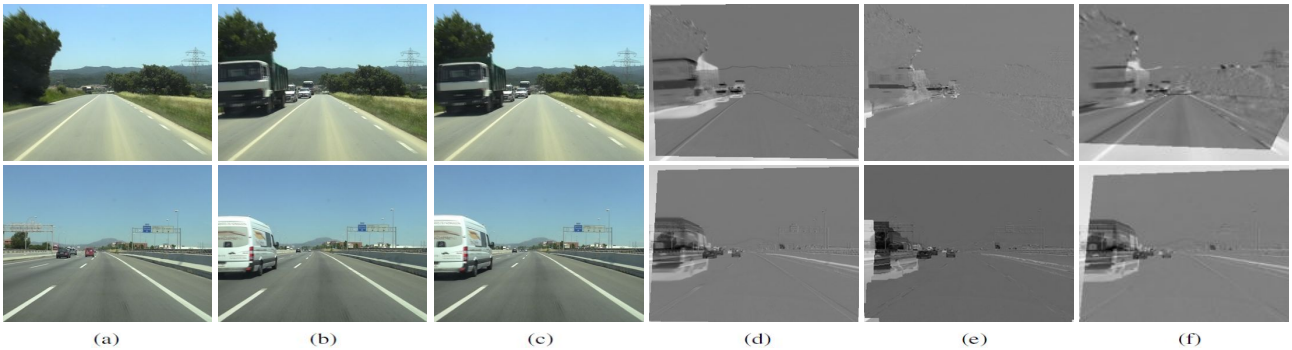


Fig. 13: Change detection instances for (*top*) *Backroad* and (*bottom*) *Highway* sequences. (a) Query and corresponding frames by (b) proposed and (c) SIFT-BoW. Pixel-wise differences after alignment by (d) proposed, (e) SIFT-flow and (f) Caspi-Irani algorithms.

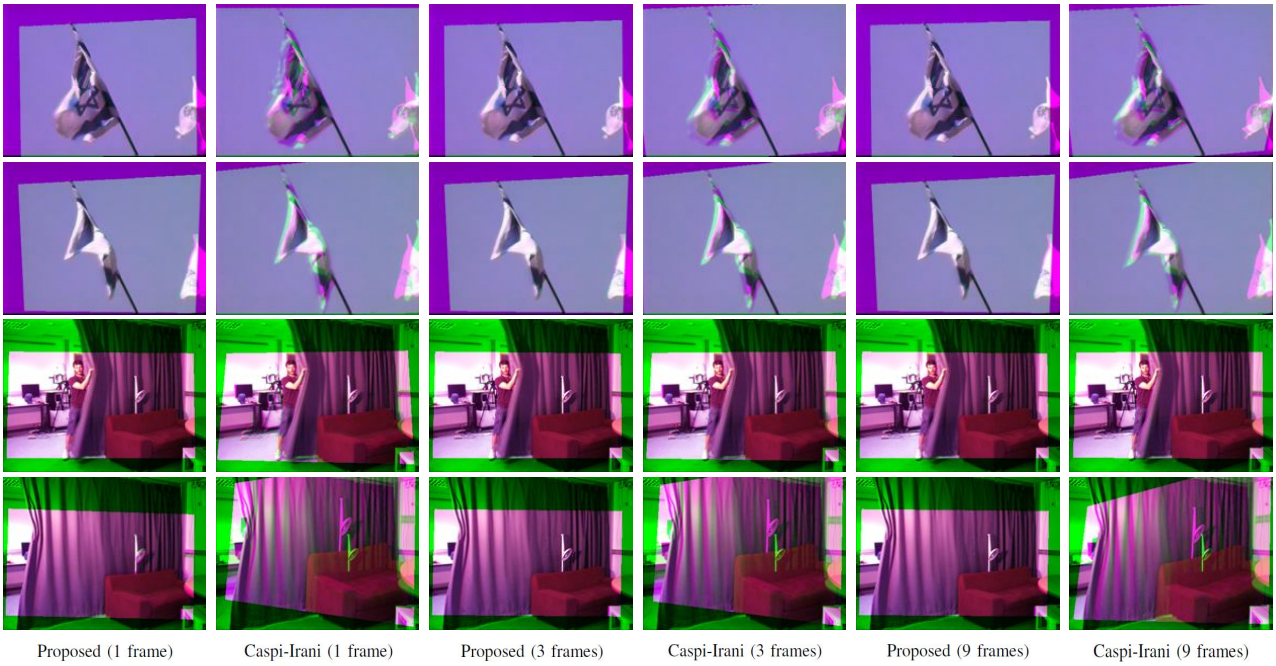


Fig. 14: Alignment between *temporally corresponding frames* using various sizes of temporal windows. Challenging frames of the *Wind* (top) and the *Inria* (bottom) sequences are shown. Caspi-Irani algorithm needs more frames than the proposed scheme to spatiotemporally register subsequences.

frames than ECC in order to correctly estimate the spatial transformation. Due to partial overlap, we found the initialization of both algorithms critical, i.e. 4 resolution levels were used. The piecewise subtraction of the average intensity for the proposed scheme was helpful as well. Again, we refer the reader to the supplemental material for more detailed visual comparisons. Among all sequences, we found more difficult the alignment of the *Suburb* sequence.

E. Synchronization and alignment times

All experiments were carried out using Matlab implementations on a 3GHz Pentium where code was not tuned to the machine. The following run time measurements assume that the descriptors (quads or SIFT features) have been pre-computed. When the resolution of the reference sequence is $720 \times 540 \times 2000$, the average retrieval time of the corresponding frame was found to be 0.82, 0.27 and 1.11 seconds for the Quad-Tree, Quad-VD, and Quad-BoW methods, respectively. On the other hand, the SIFT-BoW method required an average of 4.93 seconds for retrieving the synchronized frame because of the more involved bag-of-words representation. As far as the alignment time is concerned, ECC takes 1.43 seconds while the Caspi-Irani method was found to take 7.11 seconds for $3-$frame subsequences, when both methods register half-size images within a 3-level pyramid with 7 iterations per level. Computing SIFT-flow requires considerably more time since the approach represents each pixel as a $128-$element vector and estimates the motion in a flow basis. In our experiments, it took 32.1 seconds on half-size images. Consequently, a real-time implementation of the proposed causal solution appears achievable in a C-based environment.

VIII. Conclusions

A novel video alignment approach is presented in this paper. This approach allows for the spatiotemporal alignment of similar videos captured from independently moving cameras at different times. To achieve this, we adopted an efficient information retrieval scheme to the video synchronization problem. The method builds upon short descriptors for frame indexing and achieves remarkable synchronization results through a multi-querying (multiscale) scheme. We also presented a global solution

for offline temporal alignment by developing a multiscale dynamic programming method. As for the spatiotemporal alignment, we extended the ECC image-alignment algorithm to space-time dimensions for registering videos with subpixel and subframe accuracy. We tested our method on real video sequences captured by moving or static cameras and we compared the proposed methods with the state-of-the-art. The results verified both the efficiency and the effectiveness of the proposed method.

Although we mainly considered surveillance and driving assistance applications, the proposed scheme can be obviously adopted to other computer vision applications such as automated 3D map building, visual odometry and object identification. A reasonable extension of our work can be conceived w.r.t. incorporation in annotated databases. This way the system could operate on higher level problems by extending its capability from change detection to change (object) recognition.

Supplemental material: Please refer to `http://perception.inrialpes.fr/~evangelidis/video_alignment` for video results of the algorithms.

ACKNOWLEDGEMENTS

We thank the ADAS Group of the Computer Vision Center (CVC) in Barcelona (Spain) for data sharing and, especially, Ferran Diego for discussions.

REFERENCES

[1] Y. Caspi and M. Irani, "Spatiotemporal alignment of sequences," *IEEE Trans. on PAMI*, vol. 24, no. 11, pp. 1409–1424, 2002.

[2] ——, "Aligning non-overlapping sequences." *IJCV*, vol. 48, no. 1, pp. 39–51, 2002.

[3] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Trans. on PAMI*, vol. 30, no. 10, pp. 1858–1865, 2008.

[4] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. of CVPR*, 2006.

[5] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE TPAMI*, vol. 33, no. 5, 2011.

[6] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. on PAMI*, vol. 31, no. 4, pp. 591–606, 2009.

[7] D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis, "Astrometry.net: Blind astrometric calibration of arbitrary astronomical images," *The Astronomical Journal (AJ)*, vol. 37, pp. 1782–2800, 2010.

[8] F. Diego, D. Ponsa, J. Serrat, and A. M. Lopez, "Video alignment for change detection," *IEEE Trans. on Image Processing*, vol. Preprint, no. 99, 2010.

[9] P. Sand and S. Teller, "Video matching," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 592–599, 2004.

[10] D. Pundik and M. Y., "Video synchronization using temporal signals from epipolar lines," in *Proc. of ECCV*, 2010.

[11] Y. Ukrainitz and M. Irani, "Aligning sequences and actions by maximizing space–time correlations," in *Proc. of ECCV*, 2006.

[12] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "The video genome," *CoRR*, vol. abs/1003.5320, 2010.

[13] L. Wolf and A. Zomet, "Wide baseline matching between unsynchronized video sequences," *IJCV*, vol. 68, no. 1, pp. 43–52, 2006.

[14] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. von Stryk, S. Roth, and B. Schiele, "Vision based victim detection from unmanned aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[15] G. D. Evangelidis and C. Bauckhage, "Efficient and robust alignment of unsynchronized video sequences," in *DAGM*, 2011.

[16] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood, "View-invariant alignment and matching of video sequences," in *Proc. of ICCV*, 2003.

[17] A. Ravichandran and R. Vidal, "Video registration using dynamic textures," *IEEE Trans. on PAMI*, vol. 33, no. 1, 2011.

[18] F. Padua, R. Carceroni, G. Santos, and K. Kutulakos, "Linear sequence-to-sequence alignment," *IEEE Trans. on PAMI*, vol. 32, no. 2, pp. 304 –320, 2010.

[19] C. Lei and Y. Yang, "Trifocal tensor–based multiple video synchronization with subframe optimization," *IEEE Trans. on Image Processing*, vol. 15, no. 9, pp. 2473–2480, 2006.

[20] T. Tuytelaars and L. V. Gool, "Synchronizing video sequences," in *Proc. of CVPR*, 2004.

[21] H. Kong, J.-Y. Audibert, and J. Ponce, "Detecting abandoned objects with a moving camera," *IEEE Trans. on Image Processing*, vol. 19, no. 8, pp. 2201–2210, 2010.

[22] F. Fraundorfer, C. Engels, and D. Nister, "Topological mapping, localization and navigation using image collections," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[23] K. L. Ho and P. Newman, "Detecting loop closure with scene sequence," *IJCV*, vol. 74, no. 3, pp. 261–286, 2007.

[24] C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[25] C. Harris and M. Stephens, "A combined corner and edge detector." in *Proc. of 4th Alvey Vision Conf.*, 1988, pp. 147–151.

[26] J. Canny, "A computational approach to edge detection." *IEEE Trans. on PAMI*, vol. 8, no. 6, pp. 679–698, 1986.

[27] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *IJCV*, vol. 37, no. 2, pp. 151–172, 2000.

[28] T. Lindeberg, *Scale-space theory in computer vision*. Springer, ISBN: 0792394186, 1994.

[29] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors." *IJCV*, vol. 60, no. 1, pp. 63–86, 2004.

[30] S. R. Brown M. and S. Winder, "Mutli-image matching using multi-scale oriented patches." in *Proc. of CVPR*, 2005.

[31] D. Lowe, "Distinctive image features from scale invariant keypoints." *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[32] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *IJCV*, vol. 65, no. 1/2, pp. 43–72, 2005.

[33] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints." in *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[34] S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Algorithms*. McGraw-Hill, 2006.

[35] P. Mainali, Q. Yang, G. Lafruit, R. Lauwereins, and L. V. Gool, "Lococo: Low complexity corner detector," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.

[36] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[37] S. Baker, R. Gross, I. Matthews, and T. Ishikawa, "Lucas-kanade 20 years on: A unifying framework: Part 2," Robotics Institute - CMU, Tech. Rep. CMU-RI-TR-03-01, February 2003.

[38] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cryptography," *Comm. of ACM*, vol. 24, no. 6, pp. 381–395, 1981.

Georgios D. Evangelidis received his BSc, MSc and PhD degree in computer science in 2001, 2003 and 2008 respectively from the University of Patras, Greece. From 2007 to 2009 he was an adjunct lecturer in the Department of Informatics and Telecommunications at the Technological Institute of Larissa, Greece. During 2009-2010, he was an ERCIM (Alain Bensoussan) Fellow and joined the Visual and Social Media Group at the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS) in Sankt Augustin, Germany, as a postdoctoral researcher. Currently, he is a research engineer at the Perception Team of INRIA in Grenoble, France. His research interests are in the area of computer vision and include Stereo, 3D reconstruction, Image/Video alignment and Depth-Color Fusion.

Christian Bauckhage is professor of media informatics at the University of Bonn and lead scientists for multimedia pattern recognition at the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS). He obtained a PhD in computer Science from Bielefeld University, Germany, and was a postdoctoral researcher in the Centre for Vision Research in Toronto, Canada. Later, he worked as a senior research scientist at Deutsche Telekom Laboratories in Berlin, where he conducted and coordinated industrial ICT research. His expertise is in large scale data mining and pattern recognition and his current research interests focus on efficient approaches to high dimensional (hyperspectral) image and video analysis.