Video Alignment for Change Detection

Ferran Diego, Daniel Ponsa, Joan Serrat, and Antonio M. López

***Abstract*—In this work, we address the problem of aligning two video sequences. Such alignment refers to synchronization, i.e., the establishment of temporal correspondence between frames of the first and second video, followed by spatial registration of all the temporally corresponding frames. Video synchronization and alignment have been attempted before, but most often in the relatively simple cases of fixed or rigidly attached cameras and simultaneous acquisition. In addition, restrictive assumptions have been applied, including linear time correspondence or the knowledge of the complete trajectories of corresponding scene points; to some extent, these assumptions limit the practical applicability of any solutions developed. We intend to solve the more general problem of aligning video sequences recorded by independently moving cameras that follow similar trajectories, based only on the fusion of image intensity and GPS information. The novelty of our approach is to pose the synchronization as a MAP inference problem on a Bayesian network including the observations from these two sensor types, which have been proved complementary. Alignment results are presented in the context of videos recorded from vehicles driving along the same track at different times, for different road types. In addition, we explore two applications of the proposed video alignment method, both based on change detection between aligned videos. One is the detection of vehicles, which could be of use in ADAS. The other is online difference spotting videos of surveillance rounds.**

***Index Terms*—Bayesian network, change detection, GPS, image registration, Kalman filtering and smoothing, video alignment.**

I. Introduction

IMAGE matching or registration has received considerable attention for many years and is an active research subject due to its roles in segmentation, recognition, sensor fusion, construction of panoramic mosaics, motion estimation, and other tasks. Video matching or alignment, in contrast, has been much less explored despite it shares a number of potential applications with still image registration. It has been used for visible and infrared camera fusion and wide baseline matching [4], high dynamic range video and video mating [18], action recognition [23] and loop-closing detection in SLAM [9].

In general terms, video alignment is a more complex problem than image registration because it requires alignment in both the temporal and spatial dimensions. Temporal alignment, or synchronization, is used to map the time domain of the first sequence to that of the second one, such that each corresponding frame pair (one frame from each sequence) has the highest possible "similar content" (Fig. 1). Essentially, similar content is present when a warping can be found that spatially aligns one frame with the other, to the extent that the frames can be compared pixelwise.

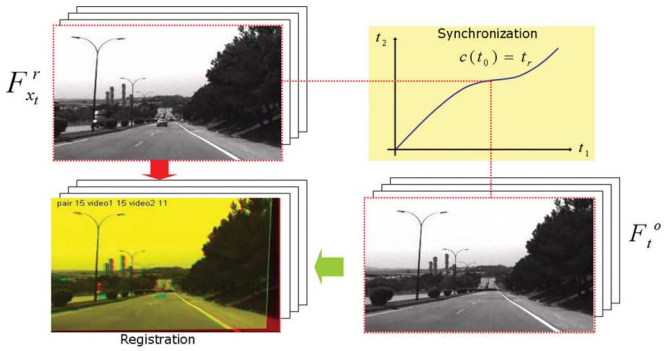


Fig. 1. Video alignment concept: temporal and spatial registration. Frame overlapping amounts to about 90% and yaw angle is 3°.

The problem of video alignment can be stated more specifically, yet without any assumption added to make it tractable, as follows. Let be two video sequences denoted as "reference" and "observed." The latter must be entirely contained in the former. Synchronization aims to estimate a discrete mapping $c(t_o) = t_r$ for all frames of the observed video, such that the t_r frame in a reference sequence maximises some measure of similarity with the t_o frame in a observed sequence. The discrete mapping $c(t)$ is many-to-one: one reference frame t_r is always assigned to each observed frame t_o, but t_r can be assigned to more than one observed frame t_o. The second part, registration, takes all corresponding pairs $(t_o, c(t_o))$ and warps the t_o frame in the observed sequence so that it matches the t_r frame in the reference sequence, according to some similarity measure and a spatial deformation model (Fig. 1).

A. Objective

The preceding general formulation needs to be completed by the addition of certain assumptions. Their choice determines, as we will note in the review of past works, the generality of the problem solution and difficulty. Our goal is to synchronise videos recorded at different times; such videos can thus differ in intensity and content, i.e., they show different objects under nonidentical lighting conditions. The videos are recorded by a pair of independently moving cameras, although their motion is not completely free. For video matching to be possible, there must be some overlap in the field of view of the two cameras when they are at identical or nearby positions. Furthermore,

Manuscript received April 26, 2010; revised September 15, 2010; accepted November 15, 2010. Date of publication November 29, 2010; date of current version June 17, 2011. This work was supported by the Spanish Ministry of Education and Science (MEC) under Project TRA2007-62526/AUT, TRA2010-21371-C03-01, Research Program Consolider Ingenio 2010: MIPRCV (CSD2007-00018). The work of F. Diego was supported by FPU MEC Grant AP2007-01558. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jesus Malo.

The authors are with the Computer Vision Center and Computer Science Department, Edifici O, Universitat Autònoma de Barcelona, 08193 Cerdanyola del Vallés, Spain (e-mail: fdiego@cvc.uab.es; daniel@cvc.uab.es; joans@cvc.uab.es; antonio@cvc.uab.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2095873

we require that the relative camera rotations between corresponding frames not be too large and, more importantly, that the cameras follow *approximately* coincident trajectories. In particular, we address the alignment of video sequences independently recorded from a vehicle by a forward facing camera attached to the windscreen shield. The vehicle must keep on the same lane, that is, the lateral displacement of the camera is bounded to about ± 2.5 m. Variations in the camera pose at one same place occur because of the different heading of the vehicle. They are modelled by a 3-D rotation in which the most important factor is the yaw angle. We have found our method to be robust even to a *constant* (along the whole sequence) maximum variation of 3°, what means a 90% of image overlapping.

Independent camera motion has the key implication that the correspondence $c(t)$ is of free form: for instance, either of the two cameras may stop at any time. Finally, we do not want to depend on error-free or complete point trajectories provided manually or by an ideal tracker, but rather to rely on the images themselves and possibly on further data collected by different sensors. In sum, and for the sake of a greater practical applicability, we are choosing a more general (and difficult) statement of the problem than previous works.

One possible application of video alignment is to spot differences between two videos. One scenario where this can be useful is in sequences captured by cameras mounted on vehicles, in the context of driving assistance systems. Suppose that the reference sequence has been recorded in the absence of traffic and that both reference and observed sequences were recorded under similar lighting conditions. Then, in principle, changes could be attributed to vehicles present in the observed sequence. A second application of difference spotting is mobile surveillance. Imagine a vehicle repeatedly driving round some facility, following the same track. The differences found subtracting two aligned video may be a sing of intrusion. Thus, change detection could serve to select regions of interest on which specialized classifiers could focus, instead of exploring the whole or a large part of the image. To our knowledge, this is a novel approach to these two applications.

B. Overview

Before going more deeply into the details of our method, we will review the past works in Section II. Then, we will focus on synchronization between two video sequences recorded from moving vehicles, which is the most difficult part of the problem. We formulate the video synchronization as a maximum *a posteriori* (MAP) inference problem on a dynamic Bayesian network (DBN) (Section III). This approach has the advantage that the problem assumptions can be expressed in probabilistic terms. More specifically, the Bayesian network utilized is the multiple-observation Hidden Markov model shown in Fig. 2. The hidden variables $\mathbf{x}_{1:n_o} = [x_1 \cdots x_t \cdots x_{n_o}]$ represent the number of frame in the reference sequence corresponding to the tth frame of the observed sequence. Each hidden node has two types of independent observations, $\mathbf{a}_t$ and $\mathbf{s}_t$, which are respectively an image descriptor of frame $\mathbf{F}_t^o$ and the georeferenced camera position at time t in the observed sequence. Hence, three conditional probabilities will be defined: $p(x_{t+1}|x_t)$, $p(\mathbf{a}_t|x_t)$ and $p(\mathbf{s}_t|x_t)$. The first will enforce the assumption that the camera/vehicle is either stopped or moves forward at some varying speed by setting $p(x_{t+1}|x_t) = 0$ if $x_{t+1} < x_t$. The latter two probabilities will express the necessary image similarity and the GPS receiver position proximity for each pair of corresponding frames, respectively.

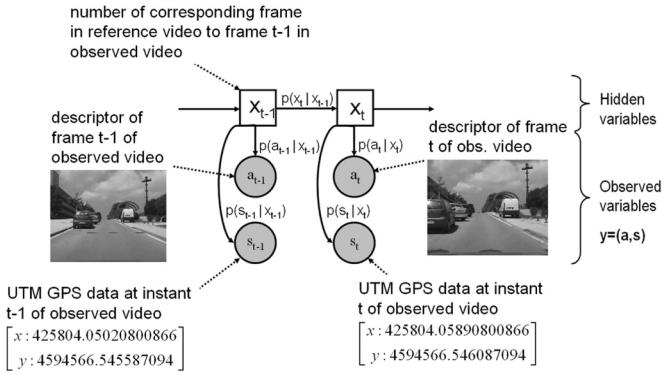


Fig. 2. Illustrative example of the Bayesian network.

Once the temporal correspondence has been computed, the next step is to perform the spatial registration of the frame pairs, which is described in Section IV. The key assumption in this case is that the two cameras are at the same position and only may differ in their pose. Hence, a conjugate rotation homography relates the frame coordinates of each pair. A version of the well-known Lucas-Kanade algorithm [1] is employed to estimate the warping from the reference to the observed frame. Fig. 3 shows an example of alignment of two video sequences. Section V presents the synchronization results for eight pairs of video sequences shot on different road types and evaluates them with respect to the manually obtained ground-truth. For these sequences, we also compare the contribution of the two types of observations; we calculate the synchronization error using appearance only, GPS only, and both together. In addition, an experiment on the use of video alignment for vehicle detection and a mobile surveillance are explained and its results presented. Finally, Section VI summarises this work and presents the main conclusions.

II. Previous Works

Several solutions to the problem of video synchronization have been proposed in the literature. Here, we briefly review those we consider the most significant, not only because they put our work into context, but also because, under the same generic label of synchronization, they try to solve different problems. The distinctions among methods are based on the input data and the assumptions made by each method. Table I compares these. Specifically, they are the type of time correspondence $c(t)$, the need or not for simultaneous video recording and cameras attached each other, the kind of input data on which to perform the synchronization (basically, image point trajectories, feature points or the whole image), and finally whether or not these methods need to estimate some fixed or varying image transform between potentially corresponding frames. Table I summarizes the characteristics of a number of previous works.

The first proposed methods assumed the temporal correspondence to be a simple constant time offset $c(t_o) = t_o + \beta$ [5], [11],

TABLE I
COMPARISON OF VIDEO SYNCHRONIZATION METHODS

		[26]	[4]	[11]	[15]	[22]	[23]	[18]	[9]	[3]	[14]	[21]	[24]	[27]	[5]	[16]	[25]	[19]	[7]	This work
Time correspondence	constant offset	•		•		•														
	linear		•				•				•	•	•	•	•	•	•			
	unconstrained				•			•	•	•								•	•	•
Simultaneous recording	yes	•	•	•		•					•	•	•	•	•	•	•			
	not necessary				•		•	•	•	•								•	•	•
Cameras rigidly attached	yes	•	•		•		•						•	•		•	•			
	no			•		•		•	•	•	•	•			•			•	•	•
Input Data	point trajectories	•	•		•	•												•		
	line trajectories			•				•	•	•	•	•							•	
	feature points													•		•				
	dynamic texture														•		•			
	space-time feature						•						•							
	images		•																	•
	images + GPS																			•
Need to estimate	fixed homography		•												•		•	•		
	fundamental matrix					•					•						•			
	trifocal tensor			•																
	deficient rank	•			•							•				•				
	fixed affine						•													
	variable motion field							•												
	frame similarity								•				•	•					•	•

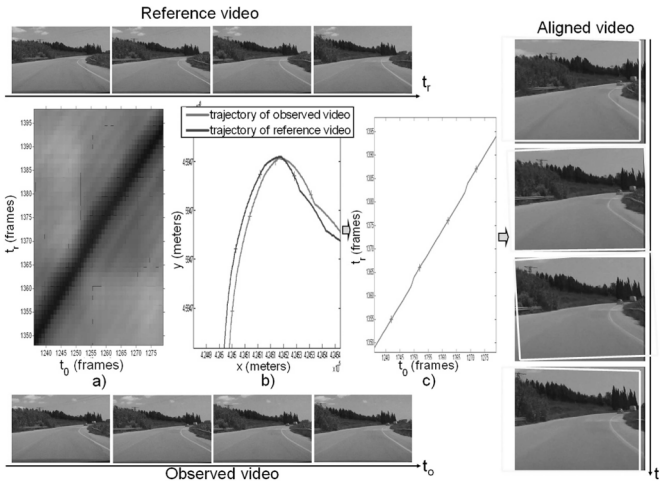


Fig. 3. Illustrative example of alignment of two video sequences recorded from moving vehicles. The estimation of the temporal mapping $c(t)$ shown in (c) considers the appearance likelihood $p(\mathbf{a}_t|x_t)$ in (a) and the GPS data in (b).

[22], [24], [26], [27] or a linear relationship $c(t_o) = \alpha t_o + \beta$ [4], [14], [16], [21], [23], [25] to account for different camera frame rates. More recent works [3], [7], [9], [15], [18], [19] let it be of free form. Clearly, the first case is simpler since only one or two parameters have to be estimated, in contrast to a nonparametric curve of unknown shape.

Most of these methods rely on the existence of an unknown geometric relationship between the coordinate systems of corresponding frames; these include an affine transform [23], a plane-induced homography [4], [5], [19], [25], the fundamental matrix [3], [14], [22], [25], the trifocal tensor [11], or a deficient rank matrix made of the coordinates of point trajectories tracked along the whole sequence [15], [16], [21], [26]. This assumption makes it possible either to formulate some minimization over the time correspondence parameters (e.g., α, β) or to perform an exhaustive search in the range of allowed values. The cases in which this geometric relationship is constant [4], [15], [16], [23]–[27], for instance because the two cameras are rigidly attached to each other, are easier to solve. Instead, works [3], [5], [7], [9], [11], [14], [18], [19], [21], [22] address, as we do, the more difficult case of independently moving cameras, where no geometric relationship is assumed beyond a more or less overlapping field of view.

Each method needs input data, which can be either more or less difficult to obtain. For instance, feature-based methods require tracking of one or more characteristic points along both whole sequences [15], [19], [22], [26], tracking points and lines in three sequences [11], or detecting interest points in space or space-time [3], [7], [9], [14], [18], [21], [25], [27]. In contrast, the so-called direct methods are based solely on the image intensity [4], [23], [24], Fourier transform of image intensity [5] or dynamic texture [16].

Perhaps the closest works to ours in that they do not require a parametric correspondence mapping, rigidly attached cameras, tracking points nor estimating a motion field or geometric transform, are [7] and [9]. Interestingly, they do not address the problem of video synchronization but view-based simultaneous localization and mapping (SLAM). View-based SLAM involves matching each novel frame of a sequence with previously shot frames, as opposed to landmark based SLAM, in which features extracted from the new frame are matched with 3-D landmarks in the map. If these shot frames are taken from another sequence acquired following roughly the same trajectory, then one may say that view-based SLAM produces video synchronization as a "by-product." And conversely, that video synchronization can be used for localization with respect to the reference sequence. These two works estimate the frame correspondence through local invariant SIFT features and approximate nearest-neighbor search in a high dimensional space. We will compare our method to that of [7] in Section V-B and show that this kind of frame matching does not solve the video synchronization problem, at least on outdoor driving sequences recorded by an onboard camera. A preliminary version of this work has appeared in [6].

III. VIDEO SYNCHRONIZATION AS AN INFERENCE PROBLEM

Let $\mathbf{F}^o$ and $\mathbf{F}^r$ be two video sequences n_o and n_r frames long, respectively, recorded from independent moving cameras following a similar trajectory. $\mathbf{F}^r$ denotes the reference sequence and $\mathbf{F}^o$ the "observed" sequence. The latter video is assumed to be entirely contained within the former. Synchronization aims to estimate a discrete mapping $c(t_o) = t_r$ for all frames $t_o = 1 \cdots n_o$ of the observed video. This mapping relates each frame of observed video $\mathbf{F}^o_{t_o}$ to one frame of the reference video $\mathbf{F}^r_{t_r}$ such that it maximises some similarity measure. Due to the independent motion of the cameras, we cannot rely on the existence

of a certain unknown but *constant* geometric entity relating the spatial coordinates of corresponding frames such as a homograpgy or fundamental matrix, to be estimated along with the temporal correspondence. In addition, the temporal correspondence does not adopt a parametric form, such as the constant offset or the linear dependence assumed by several past works. Therefore, we can not estimate the temporal correspondence function as a maximization of some overall similarity measure between corresponding frames, with regard to the function parameters. The alternative, trying to spatially register each possible pair of frames (one from each sequence) is clearly not feasible due to its computational cost for sequences longer than a few seconds at a frame rate of 25–30 frames per second. In addition, it is not possible if we want to perform video alignment online. We overcome these difficulties by formulating the video synchronization problem as a probabilistic labeling problem.

The labelling problem consists in estimating a list of n_o labels $\mathbf{x}_{1:n_o} = [x_1 \cdots x_t \cdots x_{n_o}]$. Each label $x_t \in \{1, \ldots, n_r\}$ denotes the number of the frame in the reference video corresponding to the tth frame of the observed sequence. To perform this task, we rely on the available observations $\mathbf{y}_{1:n_o}$, i.e., the frames of the observed sequence themselves and the GPS data associated with them. We pose this task as a maximum a posteriori Bayesian inference problem

$$\begin{aligned}\mathbf{x}_{1:n_o}^{MAP} &= \arg\max_{\mathbf{x}_{1:n_o}\in\mathcal{X}} p(\mathbf{x}_{1:n_o}|\mathbf{y}_{1:n_o}) \\ &\propto \arg\max_{\mathbf{x}_{1:n_o}\in\mathcal{X}} p(\mathbf{y}_{1:n_o}|\mathbf{x}_{1:n_o})P(\mathbf{x}_{1:n_o}) \qquad (1)\end{aligned}$$

where $\mathcal{X}$ is the set of all possible labellings. The prior $P(\mathbf{x}_{1:n_o})$ can be factored as

$$P(\mathbf{x}_{1:n_o}) = P(x_1)\prod_{t=1}^{n_o-1} P(x_{t+1}|x_t) \qquad (2)$$

under the assumption that the transition probabilities are conditionally independent given their previous label values. In addition, the constraint that the vehicle can stop but not reverse its motion direction in both the reference and observed sequences implies that the labels x_t increase monotonically. Therefore

$$P(x_{t+1}|x_t) = \begin{cases} v, & \text{if } x_{t+1} \geq x_t \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

where v is a constant that gives equal probability to any label greater than or equal to x_t. The prior for the first label of the sequence $P(x_1)$ gives the same probability to all labels in $\{1, \ldots, n_r\}$ because $\mathbf{F}^o$ can be any subsequence within $\mathbf{F}^r$.

If we also assume that the likelihoods of the observations $\mathbf{y}_{1:n_o}$ are independent given their corresponding label values, then $p(\mathbf{y}_{1:n_o}|\mathbf{x}_{1:n_o})$ factors as

$$p(\mathbf{y}_{1:n_o}|\mathbf{x}_{1:n_o}) = \prod_{t=1}^{n_o} p(\mathbf{y}_t|\mathbf{x}_t). \qquad (4)$$

Based on these dependencies between variables, it turns out that our problem is one of MAP inference on a first-order hidden Markov model. Hence, we can apply the well-known Viterbi algorithm [2], [17] to exactly infer $\mathbf{x}_{1:n_o}^{MAP}$. This algorithm is a dynamic programming algorithm that finds the single most likely explanation (sequence of hidden states) for a given observation sequence. However, alternative ways to solve the synchronization like max-product, loopy belief propagation, Gibbs Sampling and gradient-descent should have to be considered in the case of a Bayesian network different from a chain. For instance, if we had loops with the intention to express a different prior, the solution could be only approximated.

As we have mentioned, at each time t we will have two types of observations: an image and some GPS positioning data. We will now precisely define the nature of the observations $\mathbf{y}_{1:n_o}$ and the conditional probability $p(\mathbf{y}_t|\mathbf{x}_t)$. Fig. 2 illustrates the prior $P(\mathbf{x}_{1:n_o})$ and the conditional probabilities $p(\mathbf{a}_t|x_t)$ and $p(\mathbf{s}_t|x_t)$.

A. Appearance Likelihood

If two frames are corresponding then their content should be similar. Likewise, the camera positions at the time they were recorded should be identical or very close to each other and only their pose should be different. As we will see in Section IV, this means that one frame could be registered to its corresponding image by a simple parametric mapping. One possibility, then, is to perform the registration and employ some alignment error measure, such as, the mean square error, in order to define part of the likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$. This is clearly not feasible in practice, since sequences just a few minutes long would require the computation of a huge number $n_o n_r$ of image registrations, each one taking a non-negligible computational time. Instead, we will use an image description that will be simple to compute and allows a fast yet effective comparison. This description is a vector denoted by $\mathbf{a}$, after "appearance," and is computed as follows. The original image l (in our video sequences, 720×576 pixels) is smoothed with a Gaussian kernel with $\sigma = 2$ and then downsampled along each axis to 1/16th of the original resolution. The gradient (l_x, l_y) of this sampled image is computed with centered finite differences. Then, partial derivatives of locations where the gradient magnitude is less than 5% of the maximum are set to zero. Finally, $\mathbf{a}$ is built by stacking the rows of l_x, followed by those of l_y, and finally by rescaling the resulting vector to unit norm.

We propose a simple similarity measure between frames $\mathbf{F}_t^o$ and $\mathbf{F}_{x_t}^r$ based on the coincidence of the gradient orientation in their subsampled images through the scalar product $\langle \mathbf{a}_t, \mathbf{a}_{x_t} \rangle$, being $\mathbf{a}_t$ and $\mathbf{a}_{x_t}$ the descriptors for this pair of frames. This has proved to be a slightly better similarity measure in our sequences than other measures based on the intensity or the gradient magnitude. In addition, gradient orientation is less influenced by lighting changes.

Probability $p(\mathbf{a}_t|x_t)$ is the probability that the frame pair $\left(\mathbf{F}_t^o, \mathbf{F}_{x_t}^r\right)$ is corresponding given their frame appearance descriptors. Since the two vectors $\mathbf{a}_t$ and $\mathbf{a}_{x_t}$ are normalized, their scalar product is the cosine of the angle between them. From this, we define the appearance likelihood as

$$p(\mathbf{a}_t|x_t) = \Phi\left(\langle \mathbf{a}_t, \mathbf{a}_{x_t} \rangle; 1, \sigma_a^2\right) \qquad (5)$$

where $\Phi(v; \mu, \sigma^2)$ denotes the evaluation of the Gaussian pdf $\mathcal{N}(\mu, \sigma^2)$ at v. The closer $\langle a_t, a_{x_t} \rangle$ to 1, the higher the likelihood. The choice of the Gaussian distribution has been made

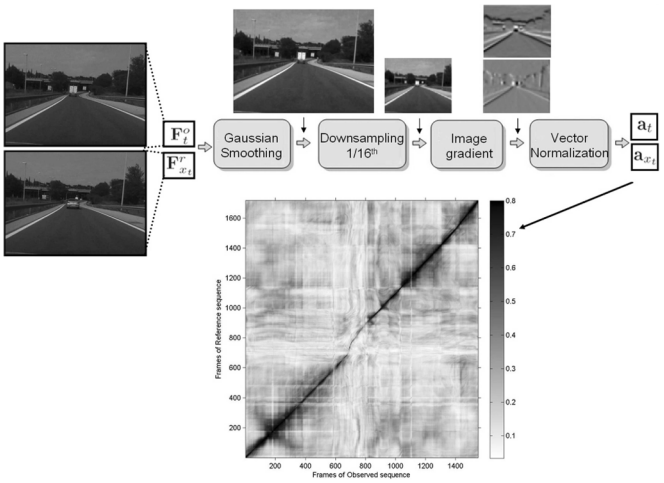


Fig. 4. Illustrative description of computing an image descriptor for and observed and reference frame. Below, the typical appearance likelihood $p(\mathbf{a}_t|x_t)$ for every t and x_t.

on the basis of the histogram of the similarity values of all the frames in all observed sequences and their corresponding reference frames, that is, the distribution of $\langle \mathbf{a}_t, \mathbf{a}_{c_{gt}(t)} \rangle$. Here c_{gt} stands for the manually set ground-truth of the synchronization mapping $c(t)$, which we will introduce in Section V. Its shape resembles the left half of a Gaussian centered at 1 with $\sigma_a = 0.5$. For an angle between descriptors $\mathbf{a}_t$ and $\mathbf{a}_{x_t}$ of 50°, the probability given by such density function is about 3/4 of the maximum probability $1/\sigma\sqrt{2\pi}$.

We need this likelihood to be high when frames are similar despite slight camera rotation and translation, and when they contain different, small-to medium-sized scene objects such as vehicles in road sequences. In order to deal with these factors, $\mathbf{a}_t$ is computed from horizontal and vertical translations of the low-resolution smoothed image up to ± 2 pixels. Then, the scalar product is taken between the appearance $\mathbf{a}_{x_t}$ and all 25 appearances computed this way. The maximum obtained value is then used in (5). Fig. 4 shows an example of the evaluation of $p(\mathbf{a}_t|x_t)$ for a pair of sequences and the computation of the frame appearance descriptors.

B. GPS Likelihood

The other observed variable of our DBN is the location of the acquisition data which is acquired using a Keomo 16–channel GPS receiver with Nemerix chipset. The GPS device localizes the camera in geospatial coordinates once per second, hence providing GPS data every 25 frames. This localization is estimated by the GPS receiver by first computing its relative distance (with some uncertainty) to different transmitters (satellites). This distance is derived by cross-correlating a pseudo-random noise code received from each satellite with a replica generated in the receiver. Since the location of each satellite at a given time instant can be obtained from a satellite almanac, it is possible to combine the computed distances to localize the receiver. In short, by defining an sphere centered at each satellite location, with radius equal to the relative distance between the satellite and the receiver, a 3-D volume is obtained from the intersection of at least four of these spheres. This volume delimits the uncertainty region of the receiver location. Depending on the geometry of the transmitters (i.e., the spatial relation of the different satellites), the uncertainty of the location will be bigger or smaller. Assuming an optimal configuration of satellites, from the accuracy of the GPS receiver in estimating the satellite relative distances (which depends on its capabilities to deal with different sources of noise), a quantity known as the total user equivalent range error can be established. This quantity corresponds to the standard deviation σ (in meters) of the Gaussian delimiting the extent of this uncertainty region. Depending on the characteristics of the receiver, σ can vary from few centimeters to tens of meters. For satellite configurations different from optimal, a multiplicative factor h can be computed to magnify properly σ and thus adjust the real location uncertainty [10]. This factor h is commonly denoted as *dilution of precision*.

We collect the GPS information from our receiver using the GPGGA message of the NMEA protocol. This message provides both the GPS geographic location and the value of h corresponding to the current satellite configuration. After converting the GPS geographic location into the corresponding 2-D coordinates $\mathbf{g} = [x \quad y]^T$ in the Universal Transverse Mercator system, we model the location uncertainty as the Gaussian distribution $\mathcal{N}\left(\mathbf{g}, \mathbf{R} = (h\sigma)^2\mathbf{I}_2\right)$, where $\mathbf{I}_2$ denotes a 2×2 identity matrix. Hence, the raw sensor information available at a given time t is the recorded frame $\mathbf{F}_t$ and, every 25 frames, the GPS fix with distribution $\mathcal{N}(\mathbf{g}_t, \mathbf{R}_t)$. As we will explain later Section III-C, we formalize the availability of the GPS information by defining an observed binary variable o_t that takes a value of 1 when frame $\mathbf{F}_t$ has a GPS fix associated.

The GPS information is available for only 4% of the sequence. However, for the rest of the frames there is still some knowledge that can be exploited, since the vehicle hosting the camera follows a regular trajectory. Thus, in order to estimate an observation for each frame, we apply a Kalman smoother to process the available GPS fixes $\mathcal{N}(\mathbf{g}_t, \mathbf{R}_t)$ and interpolate the missing information $\mathcal{N}(\mathbf{s}_t, \mathbf{\Sigma}_t)$ ($\mathbf{s}$ represents smoothed GPS information). To do so, we model the dynamical behavior of the vehicle to propagate the GPS information to the frames where it is not available. We have found that a model of constant acceleration gives a good approximation of the trajectory dynamics. We express this model as

$$\mathbf{g}_t = \mathbf{g}_{t-1} + \mathbf{v}_{t-1}\Delta_t + \frac{1}{2}\mathbf{a}_{t-1}\Delta_t^2 + \mathbf{w}_t \tag{6}$$

where $\mathbf{v}_{t-1}$ and $\mathbf{a}_{t-1}$ are respectively the average velocity and acceleration at the previous instant, Δ_t is the time interval between frames, and $\mathbf{w}_t$ is a stochastic disturbance term $\mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$ that accounts for model inaccuracies. In our experiments, we set $\mathbf{Q}_t = 2.25e^{-4}\mathbf{I}_2$, which means that, according to the two sigma rule, the model imprecision in one frame interval is below 3 cm with 0.95 probability. By expanding average velocity and acceleration respectively as $\mathbf{v}_t = (\mathbf{g}_t - \mathbf{g}_{t-1}/\Delta_t)$ and $\mathbf{a}_t = (\mathbf{v}_t - \mathbf{v}_{t-1}/\Delta_t) = (\mathbf{g}_t - 2\mathbf{g}_{t-1} + \mathbf{g}_{t-2}/\Delta_t^2)$ and grouping terms, we can express the constant acceleration model as the following third order autoregressive model

$$\mathbf{g}_t = 3\mathbf{g}_{t-1} - 3\mathbf{g}_{t-2} + \mathbf{g}_{t-3} + \mathbf{w}_t. \tag{7}$$

This expression, as well as the process of observing $\mathbf{g}_t$, is formalized in state-space notation as follows:

$$\begin{bmatrix} \mathbf{g}_t \\ \mathbf{g}_{t-1} \\ \mathbf{g}_{t-2} \end{bmatrix} = \begin{bmatrix} 3\mathbf{I}_2 & -3\mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 & \mathbf{0}_2 \end{bmatrix} \begin{bmatrix} \mathbf{g}_{t-1} \\ \mathbf{g}_{t-2} \\ \mathbf{g}_{t-3} \end{bmatrix} + \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{0}_2 \\ \mathbf{0}_2 \end{bmatrix} \mathbf{w}_t \quad (8)$$

$$\mathbf{y}_t = [\mathbf{I}_2 \quad \mathbf{0}_2 \quad \mathbf{0}_2] \begin{bmatrix} \mathbf{g}_t \\ \mathbf{g}_{t-1} \\ \mathbf{g}_{t-2} \end{bmatrix} + \mathbf{v}_t. \quad (9)$$

The term $\mathbf{0}_2$ is a 2×2 zero matrix, and $\mathbf{v}_t$ is the noise disturbing the observation process, with distribution $\mathcal{N}(\mathbf{0}, \mathbf{R}_t)$. This is the formal way to model systems in Kalman based algorithms, and from that we can apply them directly. In our case, we have instead applied the Rauch-Tung-Striebel Kalman smoother to combine the prior knowledge provided by the dynamical model with the GPS observations. It is an off-line algorithm that in a first step executes a Kalman filter that estimates, for each frame, a Gaussian distribution of its corresponding GPS location. In frames where no GPS fix is available, the dynamical model helps to predict their corresponding GPS distributions. Then, starting from the GPS estimation at the end of the sequence, Kalman filter estimations are propagated backward, readjusting preceding estimations according to the assumed motion model. In this way, the GPS estimation at each frame is conditioned on all GPS observations collected along the sequence. Details of this algorithm and the equations involved can be found in [8].

Concerning the definition of the GPS likelihood, note that defining $p(\mathbf{g}_t|x_t)$ or $p(\mathbf{s}_t|x_t)$ implies specifying them for any value of x_t, and this requires having GPS information for all frames of $\mathbf{F}^r$. Hence, both likelihood terms are defined using the smoothed GPS estimates $\mathcal{N}(\mathbf{s}_{x_t}, \mathbf{\Sigma}_{x_t})$ of the $\mathbf{F}^r$ frames. As in the case of the appearance likelihood, the likelihood of the observed GPS data (whether raw or smoothed) could be defined as the evaluation of $\Phi(\mathbf{v}; \mathbf{s}_{x_t}, \mathbf{\Sigma}_{x_t})$, where $\mathbf{v}$ would correspond respectively to $\mathbf{g}_t$ or $\mathbf{s}_t$ of the observed frame. However, the GPS data associated with the observed video sequence frames are not limited to a single location, but also include uncertainty in the form of a Gaussian distribution. Hence, the proper way to evaluate this likelihood is taking all the feasible GPS locations into account. The GPS likelihood is therefore the evaluation of $\mathbf{s}_t$ with respect to a Gaussian distribution $\mathcal{N}(\mathbf{s}_{x_t}, \mathbf{\Sigma}_{x_t} + \mathbf{\Sigma}_t)$ that takes both distributions into account

$$p(\mathbf{s}_t|x_t) = \Phi(\mathbf{s}_t; \mathbf{s}_{x_t}, \mathbf{\Sigma}_{x_t} + \mathbf{\Sigma}_t). \quad (10)$$

Notice that in case of the GPS data is not available for a long period of time, the uncertainty of the GPS, $\mathbf{\Sigma}_t$, will increase resulting in a very wide Gaussian. In this case, the information provided by GPS is diluted, and the extreme, the video alignment only considers the $p(\mathbf{a}_t|x_t)$ factor.

C. Dynamic Bayesian Network Synchronization Models

We have considered the four DBNs represented in Fig. 5. Dashed lines represent switching dependencies. The switching dependency is a special relation between nodes, meaning that the parents of a variable are allowed to change depending on the current value of some other parents. We use this notation in the method in Fig. 5(c) to represent the fact that the GPS receiver does not provide one raw GPS fix for all the frames in a sequence because of the lower sampling rate of the GPS receiver (1 fix per second) with respect to the camera (25 frames per second). Every 25 nodes, $o_t = 1$; therefore the node $\mathbf{g}_t$ is connected to its parents h_t and x_t, providing an additional sensor observation to node x_t. Elsewhere, the effective graphical model in Fig. 5(c) becomes that of Fig. 5(a); i.e., $o_t = 0$ and x_t has a single observation $\mathbf{a}_t$.

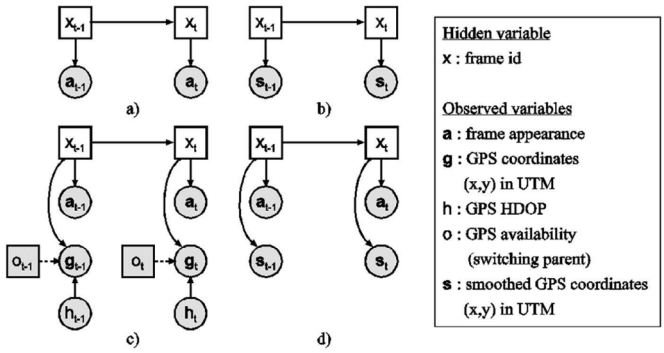


Fig. 5. Comparison of four DBNs. Square nodes represent discrete variables and circular nodes continuous variables. Shaded nodes denote observed variables; nonshaded nodes are hidden variables. The conditional dependencies between variables are represented by solid lines. Note that in all of these networks, we consider observations coming from different sensors to be independent. Dashed lines represent switching dependencies. The switching dependency is a special relation between nodes, meaning that the parents of a variable are allowed to change depending on the current value of some other parents. Fig. 5(a) to (d) show, respectively, the DBN for appearance only, smoothed GPS coordinates only, appearance combined with raw GPS coordinates and appearance combined with smoothed GPS coordinates. In the following figures, we will refer to them with four DBNs by "App," "GPS," "AppRawGPS," and "AppGPS," respectively.

The four DBNs have been proposed in order to assess the contribution of each type of observation to the final synchronization result. Thus, we want to ascertain whether using only the appearance or only the GPS data yields a worse result than combining the two observations, and eventually to quantify the improvement in the synchronization. In addition, we want to explain the need for GPS smoothing (having an estimate of the GPS data at each frame) instead of just taking the raw GPS fix (GPS data every 25 frames).

IV. Registration

The result of the synchronization is a list of pairs of corresponding frame numbers (t, x_t), $t = 1 \cdots n_o$. Ideally, for each such pair the camera is at the same position; hence, only the camera pose may be different. Let the rotation matrix R express the relative orientation of the camera for one such pair. It can then be seen that the coordinates of the two corresponding frames $\mathbf{F}_t^o$, $\mathbf{F}_{x_t}^r$ are related by the homography $H = KRK^{-1}$, where $K = \mathrm{diag}(f, f, 1)$, f being the camera focal length in pixels. Let the 3-D rotation R be parameterized by the Euler angles $\mathbf{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$ (pitch, yaw and roll respectively). Under the assumptions that these angles are small and the focal length is large enough, the motion vector field associated with this homography can be approximated by the following model [28],

which is quadratic in the x and y coordinates but linear in the parameters $\mathbf{\Omega}$

$$\mathbf{u}(\mathbf{x};\mathbf{\Omega}) = \begin{bmatrix} -\frac{xy}{f} & f+\frac{x^2}{f} & -y \\ -f-\frac{y^2}{f} & \frac{xy}{f} & x \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}. \quad (11)$$

R and consequently $\mathbf{\Omega}$ may be different for each pair, since the cameras have moved independently. Therefore, for each pair of frames we need to estimate the parameters $\mathbf{\Omega}$ that minimize some registration error. The chosen error measure is the sum of squared linearized differences (i.e., the linearized brightness constancy) that is used by the additive forward extension of the Lucas-Kanade algorithm [1]

$$\mathrm{err}(\mathbf{\Omega}) = \sum_{\mathbf{x}} \left[\mathbf{F}^r_{x_t}(\mathbf{x}+\mathbf{u}(\mathbf{x};\mathbf{\Omega})) - \mathbf{F}^o_t(\mathbf{x})\right]^2 \quad (12)$$

where $\mathbf{F}^o_t$ is the template image, and $\mathbf{F}^r_{x_t}$ is the image warped onto the coordinate frame of the template. The previous minimization is performed iteratively until convergence. In practice, we cannot directly solve for $\mathbf{\Omega}$ because a first order approximation of the error in (12) can be made only if the motion field $\mathbf{u}(\mathbf{x};\mathbf{\Omega})$ is small. Instead, $\mathbf{\Omega}$ is successively estimated in a coarse-to-fine manner. A Gaussian pyramid is built for both images, and, at each resolution level, $\mathbf{\Omega}$ is re-estimated based on the value of the previous level. For a detailed description we refer the reader to [1].

V. Results

A. Synchronization Results

We have aligned eight video sequence pairs with the four DBN models and evaluated the synchronization errors. These sequences were recorded with a SONY DCR-PC330E camcorder in different environments: a university campus during the day ("campus" pair) and at night ("night" pair), a suburban street, a back road and a highway. Fig. 6 shows a few sample frames of each. The back road and the night pair contain few distinct scene features as compared to the suburban street and the campus pairs, which are populated by a number of buildings, parked cars and lampposts on both sides of the image. Few features may mean that the appearance observation is less informative with regard to the video synchronization. In addition, the number of visible satellites and therefore the GPS data reliability is lower in the campus and the street sequences due to the proximity of tall buildings.

The eight sequence pairs can be divided into two groups. In the first one, containing one pair per scenario (labelled as "campus1," "backroad1," "highway1," "night," "street"), sequences were recorded while driving at "normal" velocity at each point of the track. We mean that we did not try intentionally to maintain a constant speed along the whole sequence, or even to drive with a similar speed in the reference and the observed sequence at a given point. Instead, we drove independently in all the reference and observed sequences, adjusting the speed at each moment to the road type and geometry, and to the traffic conditions. In consequence, in all of these pairs

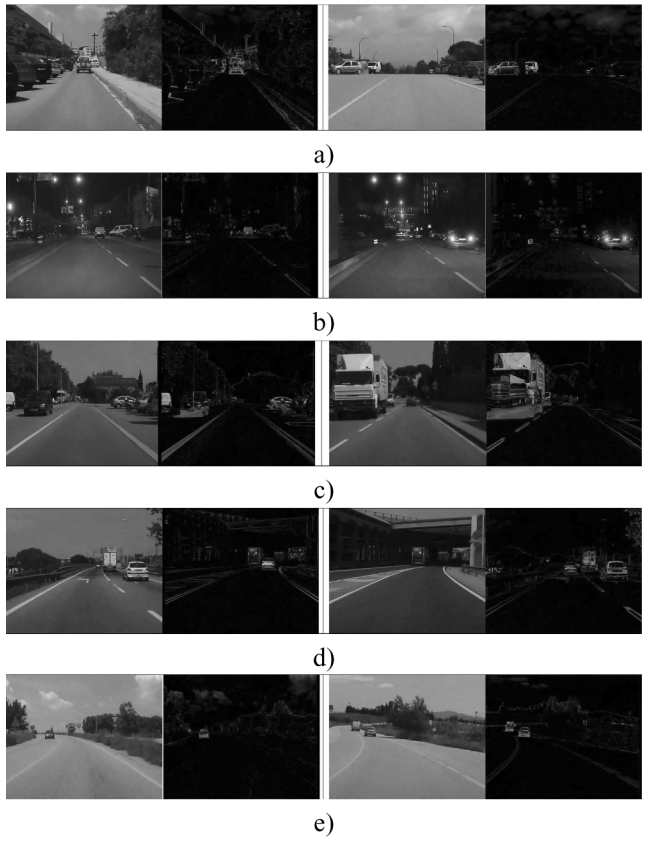


Fig. 6. Synchronization results on sample frames of five testing sequences: (a) "campus," (b) "night," (c) "street," (d) "highway," and (e) "back road." The first and third columns are the observed frames whereas the second and forth columns are the difference between the corresponding frame pair after spatial alignment. Differences are due to changes in content, mainly vehicles, in spite of ambient lighting differences.

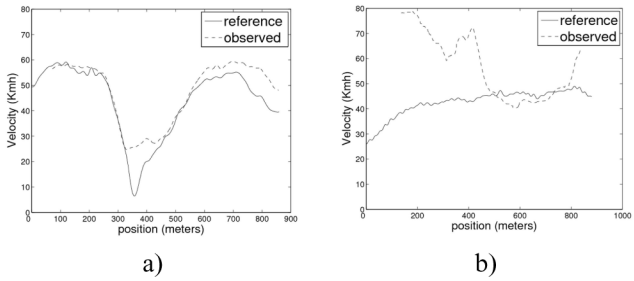


Fig. 7. Instantaneous speed of reference and observed sequence, for (a) the "backroad1" and (b) the "backroad2" pairs as approximated from GPS data.

the instantaneous velocities of the reference and observed sequences are not equal, though they tend to be close. However, medium to large speed differences may occasionally occur, as shown in Fig. 7(a), where a difference of 25 Km/h vs. 6.5 Km/h can be observed around Km 0.33.

The second group is formed by the three pairs "campus2," "backroad2," and "highway2." These were recorded with the intent of generating frequent and large velocity differences at the same location between the reference and the observed sequence. They contain points that do not correspond to a normal driving profile because of frequent and sharp acceleration and braking. Differences of $\pm 30\ \mathrm{Km/h}$ hold for relatively long intervals, as

TABLE II
CHARACTERISTICS OF SEQUENCE PAIRS. LENGTH IS THE NUMBER OF FRAMES, WITH A RECORDING RATE OF 25 FRAMES PER SECOND. THE INITIAL OFFSET IS THE FRAME NUMBER IN THE REFERENCE SEQUENCE CORRESPONDING TO THE FIRST FRAME OF THE OBSERVED SEQUENCE. THE RELATIVE SPEED DIFFERENCE IS THE RATIO $|v_{reference}(t) - v_{observed}(t)|/\max\{v_{reference}(t), v_{observed}(t)\}$, WHERE INSTANTANEOUS VELOCITIES ARE APPROXIMATED FROM THE GPS DATA ASSOCIATED WITH EACH SEQUENCE

Sequence pair	Length		Initial offset	Relative speed diff.	
	reference	observed		mean	max
backroad1	1719	1403	144	0.09	0.75
street	1002	601	62	0.13	0.26
night	1005	450	70	0.10	0.20
campus1	2000	901	99	0.29	0.62
highway1	1001	900	102	0.06	0.16
backroad2	2199	1294	366	0.20	0.51
campus2	2348	1560	322	0.24	0.43
highway2	2281	1433	574	0.20	0.39

Fig. 8. Difficulty in ground-truthing the video synchronization process. In most cases, a clear unique frame-to-frame correspondence cannot be established. From left to right: the observed frame and three reference frames.

illustrated by Fig. 7(b). Therefore, these pairs may be more challenging to synchronise. Table II lists, for each pair, the reference and observed sequence lengths, initial offset (reference frame number corresponding to the first observed frame), and mean and maximum relative instantaneous velocity difference.

In order to quantitatively assess the performance of the temporal alignment, we manually obtained the ground-truth for all eight video pairs. Every five frames of the observed video we determined the corresponding frame in the reference video. In between, we performed a linear interpolation. Mainly, we took into account the position and size of the closest static objects in the scene, i.e., lane markings, traffic signs, other cars. This decision, however, often proved difficult to make because the vehicle undergoes lateral and longitudinal relative displacements, to which camera pose variations are added. Fig. 8 illustrates the difficulty of making a single decision. Therefore, we eventually chose to select not a single frame number but rather an interval $[l_t, u_t]$ that would always contain the true corresponding frame. This can be appreciated in Fig. 12(a). The width of the ground truth intervals thus obtained is typically only 3–6 frames.

We define the synchronization error at time t, given the corresponding frame number x_t, as the distance of x_t to the closest ground-truth interval boundary,

$$\mathrm{err}(t) = \begin{cases} 0, & \text{if } l_t \le x_t \le u_t \\ l_t - x_t, & \text{if } x_t < l_t \\ x_t - u_t, & \text{if } x_t > u_t. \end{cases} \quad (13)$$

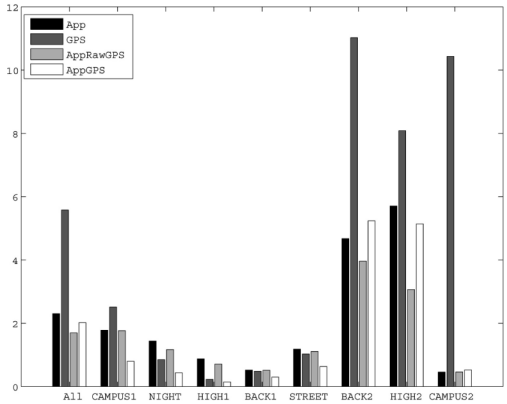


Fig. 9. Average error $\overline{e}$ for the four DBNs of Fig. 5 on each video sequence pair.

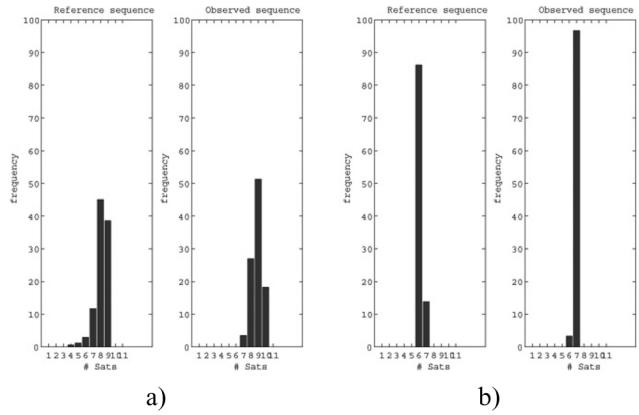


Fig. 10. Histogram of number of visible satellites viewed in reference and observed sequences for (a) the "campus1" and b) "campus2" sequence pairs. The histograms of the second pair (b) are less spread out because its sequences are shorter, due to the higher vehicle speed. This also happens in "highway2" and "backroad2."

The simplest way to evaluate the performance of the temporal alignment is to average all the individual errors $\overline{e} = (1/n_o)\sum_{t=1}^{n_o} err(t)$.

Fig. 9 shows the average error for each of the eight sequences synchronized by means of the four DBN models. We can first appreciate that overall, using the appearance as the unique observation produces a better synchronization than GPS data alone. In addition, in all but two pairs, the average error is less than one frame for the best DBN model. Second, the combination of the two types of observations, appearance and smoothed GPS, substantially decreases the average error in all of the tested sequences except "backroad2" and "campus2."

The low performance of the appearance plus smoothed GPS model in the "backroad2" and "highway2" pairs is due to the combination of two factors. First, their appearance is less informative because there are long stretches where images exhibit few content changes (much of the image shows a uniform road surface and a distant landscape that looks always the same). Second, the number of visible satellites is lower than in the first group of sequences (6 or 7 versus 8–10) making the GPS data less reliable and therefore the synchronization more prone to fail.

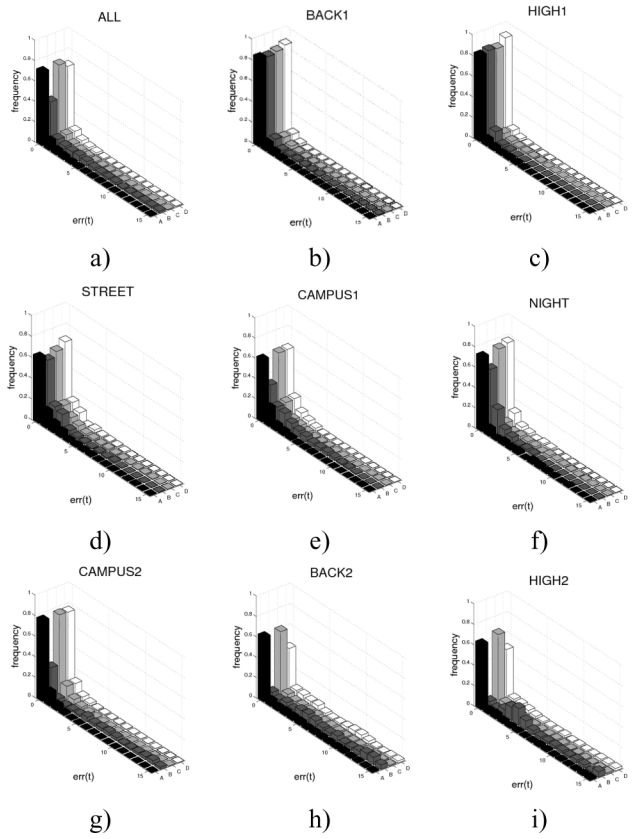


Fig. 11. Distribution of the synchronization error. The nomenclature of the x-axis is as follows: A is appearance, B is smoothed GPS, C is appearance plus raw GPS and D is appearance plus smoothed GPS.

The average error seems to be a sensible measure for calculating the performance because it is an overall measure, but it cannot distinguish a slight increase in outliers from a general reduction in accuracy. Synchronization outliers are time correspondence offsets that hamper the subsequent spatial registration and change detection. We need an error representation that explains the nature of the error. The distribution of the time correspondence error is more informative in this regard, because it tells us how many frames are at a given distance from the ground truth for all distances. Fig. 11 shows the error distribution for all the pairs separately. More than 70% of the frames of the "highway1" and "backroad1" pairs have no synchronization error. For the other six recorded pairs, the frames with no error plus those with an error of one frame exceed this score. Note that the parameters were set empirically to obtain good synchronization results on all the eight sequence pairs, which are quite diverse in content and vehicle speed. In addition, we checked that small variations did not change substantially the results, that is, the method is not too sensitive to them.

In the suburban scenario sequence pairs, some tall buildings close to the road degrade the GPS data, dragging the correspondence curve in the wrong direction at some locations. In addition, the reduction in the number of visible satellites decreases the performance for some pairs. The combination of appearance and smoothed GPS observations decreases the error in these cases because this other observation type "drags" the correspondence curve in the right direction, as shown in Fig. 12. The opposite situation also occurs, when the image similarity measure fails because the content is too different (big new objects appear in the close range) or does not change much along a certain road stretch (for instance, in highways and open landscapes void of distinct close objects). There, reliable GPS data attracts the correspondence curve to the right place.

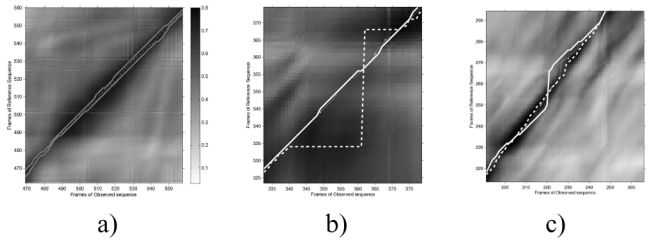


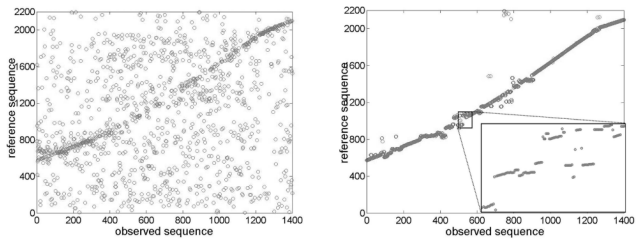
Fig. 12. Detail of ground-truth intervals (solid lines) and synchronization results (the dotted line) on a background inversely proportional to frame similarity of (5) are shown in (a) and two examples of how the two types of observations complement each other. We show how the GPS data drags the temporal correspondence, which is found only using the appearance likelihood, in the correct direction in (b). In (c), we show the inverse case where the appearance likelihood drags the temporal correspondence found only using GPS in the correct direction. The dashed line is the time correspondence using only appearance, solid line using only smoothed GPS data, and dotted line using both. The dotted line is the closest to the ground-truth band.

Fig. 6 shows the synchronization results in few sample frames of these tested scenarios. A figure containing just a few frames of the aligned videos would be a poor reflection of the results. The original, synchronized and fully aligned video sequences can be viewed at the web page www.cvc.uab.es/ADAS/projects/sincro/VA-changeDetection/. To visually assess the quality of the spatial registration we perform a simple image fusion assigning the reference frame to the red and blue channels of a color image and the registered observed frame to the green channel. This way mismatchings appear in rather unfrequent colors, mostly shiny green and violet.

B. Comparison

In this section, we compare our method with the frame matching scheme proposed by Fraundorfer *et al.* [7]. In spite of being addressed to visual-based SLAM, it can produce a video synchronization and makes the same general assumptions than us: nonparametric temporal correspondence, independently moving cameras and no need of point tracks. In addition, it develops a sophisticated frame similarity measure in order to perform a highly scalable matching suited to long image sequences, very similar to that of [9], another close work for the same reasons. Basically, it consists in considering each image as a document composed of "visual words," and then perform a fast retrieval on the set of images based on this description.

The computation of visual words proceeds as follows. First, a set of interest regions are detected using the maximally stable extremal regions detector [13]. Then, the interest regions are encoded using the SIFT feature descriptor [12]. The quantization of all the visual words in the reference sequence gives rise to a vocabulary. Now, given a certain image, to each quantized visual word is assigned a weight ω equal to the inverse of its frequency in the image [20]. An image is thus described by a feature vector q of which the ith component, q_i, is equal to $\omega_i \cdot n_i$, being n_i



a) b)

Fig. 13. Temporal correspondence based on retrieving the corresponding frame in the reference sequence with the highest similarity score for each frame in the observed sequence for in "highway2" pair. Each dot represents a corresponding frame pair. The similarity scores are (a) $\mathcal{S}(q^o, q^r)$ of (14) and (b) our appearance similarity $\langle \mathbf{a}_t, \mathbf{a}_{x_t} \rangle$.

the number of times the ith visual word in the vocabulary appears in the image. The similarity between an observed and a reference images represented by the feature vectors q^o and q^r is calculated as

$$\mathcal{S}(q^o, q^r) = \left| \frac{q^o}{\|q^o\|} - \frac{q^r}{\|q^r\|} \right|. \tag{14}$$

The estimation of the temporal correspondence is done by selecting, for each frame in the observed sequence, the frame in the reference sequence with the highest score of this similarity measure, as if it was an image retrieval problem.

In order to make a fair comparison of this method with that presented here, we restrict to the appearance-only version, that is, we discard the GPS observations. The quantitative assessment is based on a simple accuracy measure: the number of frames of the observed sequence for which the reference frame lies within its ground-truth interval, that is, the correspondence is correct. We have computed it for the "campus2" and "highway2" sequences due to their dissimilar image-content between scenarios and dissimilar velocities between sequence pairs. While we achieve 78% and 64%, the former temporal correspondence estimation accuracy is only 10% and 8%. If a maximum error of five frames is allowed the figures are not much better: we get 92% and 60% versus 19% and 13%. Why is this so? Fig. 13(a) shows the one-most similar reference frame for each observed frame according to the distance of (14). We can clearly see a dense concentration trail along most, but not all, the true correspondence function. However, a close-up view shows that there are large deviations from it. We have found two reasons for it. The first is that selecting the most similar frame is not a good strategy for precise temporal alignment unless the similarity measure is almost perfect, which is not the case. Our appearance similarity measure is not, either, but we can compensate it thanks to the prior term which imposes the simple but powerful constraint of monotonically increasing correspondence. We have checked that replacing the appearance similarity $\langle \mathbf{a}_t, \mathbf{a}_{x_t} \rangle$ by that of (14) in our method we get a better accuracy of 43%, 46% for the zero frames error and 71%, 72% for the five frames error. Second, our appearance similarity measure, being simpler and of less computational cost, is better for the driving sequences, as Fig. 13 shows.

C. Video Alignment for Change Detection

One possible application of video alignment is video change detection. It aims to spot differences between two videos recorded at different times on the same scenario. These differences could be foreground objects that only appear in one sequence and/or background changes. In order to spot differences, both video sequences must be first spatio-temporally aligned, so that they can be detected by pixel-wise subtraction. Specifically, once we have registered a corresponding frame pair, we subtract their respective R, G and B channels. Then, we threshold the absolute value of the differences and filter out the binary regions larger than a certain area, which we can make to depend on the row number to account for the changing size of objects due to perspective. Finally, the bounding boxes of the remaining regions are considered as the video differences.

This is an off-line process because it can take place only after having recorded the two videos. However, it may not make sense for applications like surveillance or vehicle detection for driving assistance which are online by nature. They require on-line and fast difference detection. It turns out that a relatively simple modification of the inference mechanism can be used to adapt our method to the online setting. Instead of calculating the MAP inference on a Bayesian network formed by observation and hidden nodes representing all of the observed video sequence, a dynamic Bayesian network is built on the fly and a different type of inference is carried out called fixed-lag smoothing [17]. Fixed lag smoothing solves the problem of online deferred inference

$$x_{t-l}^{MAP} = \arg \max_{x_{t-l} \in \mathcal{L}} p(x_{t-l} | \mathbf{y}_{t-L:t}) \tag{15}$$

where $\mathcal{L}$ is the set of labels at time $t - l$, $l > 0$ is the lag or delay, and $L > l$ is the total set of frames used to infer the label x_{t-l}. Fixed-lag smoothing infers the label x_{t-l} at time t, i.e., it gives an online answer but with a delay of l frames. The specific values assigned to $\mathcal{L}$ and l are 75 and 25 time units (frames, loosely speaking), respectively. In the following, we illustrate the potential of online differece spotting with two applications: onboard vehicle detection and mobile surveillance.

Suppose we have recorded the reference sequence along some road, in the absence of traffic. Later, we drive again along the same track and under not much different ambient lighting conditions. If we succeed in aligning online the reference and the newly recorded video, what would be the differences? Objects present in only one of the sequences, mainly vehicles. We could thus compute differences between videos as a mean of vehicle detection. Of course, we do not claim that this procedure is a vehicle detector competitive with the state of the art. Rather, video alignment may allow us to select a few image windows that could contain the objects of interest, to be analyzed by a specialized classifier. Nonetheless, we have performed a quantitative detection evaluation on the back road sequence. The chosen metric is the accuracy, defined as

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{16}$$

where TP, FP, TN, FN stand for true and false positives and negatives. We have obtained an accuracy of 0.76. We count a bounding box are really containing a vehicle if it encloses at

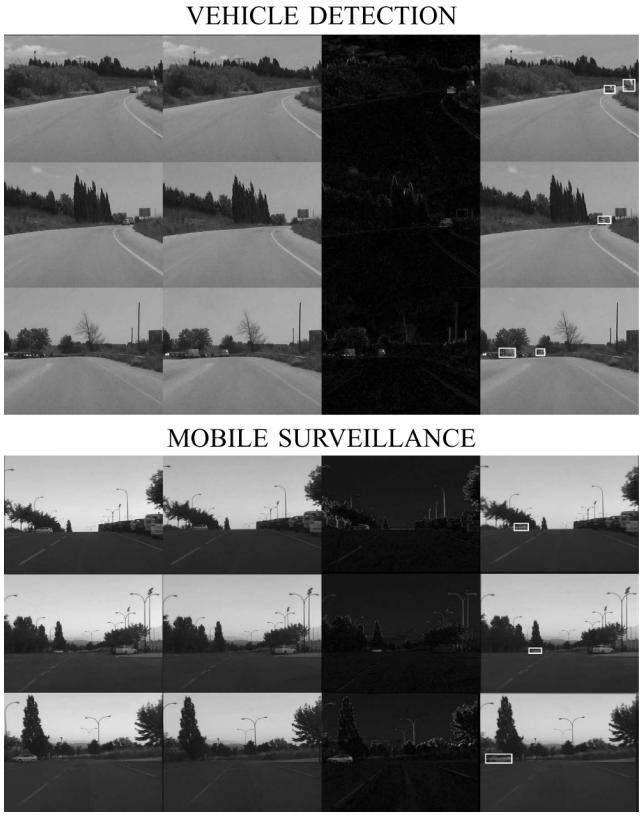


Fig. 14. Video alignment results for vehicle detection and mobile surveillance. From left to right:observed frame, corresponding frame in reference sequence, absolute difference, and change detection. Results in video form can be properly viewed at www.cvc.uab.es/ADAS/projects/sincro/VA-changeDetection/.

least 75% of its area and no more than 25% of the bounding box is background. Fig. 14 shows an example of vehicle detection but again still pictures are not the best way to present the results. Please view the web page www.cvc.uab.es/ADAS/projects/sincro/VA-changeDetection/ where the original, difference and detection videos can be played.

Another application of difference spotting is mobile surveillance. Consider the following scenario. A private guard vehicle patrols twice through a certain circuit, following approximately the same trajectory. Attached to the windshield screen, a forward facing camera records one video sequence for each of the two rides. Then, the differences between successive videos are assumed to be a sing of intrusion. In addition, the lighting conditions between video sequence are similar because they are recorded successively. We have performed a quantitative evaluation on a campus scenario recorded at sundown in order to evaluate the performance of detecting sings of intrusion. The sing of intrusion is the presence of parked vehicles in the second ride which does not appear in the first ride on this sequence pair. The chosen metric is the accuracy in (16). For the mentioned sequence, we have obtained an accuracy of 0.79. Fig. 14 shows an example of mobile surveillance.

VI. Conclusions

In this paper, we have introduced a novel approach to the problem of aligning video sequences recorded by independently moving cameras that follow a similar trajectory. We pose it as the MAP inference on a Bayesian network, where the values of the hidden variables represent the time correspondence between an observed and a reference sequence. The observations are features derived from the images, and optionally from associated GPS data. We have compared the performance of four Bayesian network models that differ in the type of observations they use. The best model, which combines smoothed GPS data and image features, achieves an average synchronization error of less than one frame in six out of eight sequence pairs. Errors in the two worst pairs are due to the combination of unreliable GPS data caused by the low number of visible satellites with the repetitive and slowly changing image content of those sequences. We have applied our method to align sequences recorded from moving vehicles driving twice along the same track. Thus, pixelwise subtraction can be used for difference spotting in the context of vehicle detection and mobile surveillance. Future work will address online video alignment, which we show in this paper to be possible through fixed-lag smoothing. In addition, we want to improve the appearance likelihood to make it more invariant to lighting changes and robust to a lower field of view overlapping. This later could be achieved by cropping the central part of the frames, which contain closer scene objects, and increasing the translation bound at the low resolution level, set now to just ± 2 pixels for the sake of efficiency.

References

[1] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.
[2] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
[3] X. Cao, L. Wu, J. Xiao, H. Foroosh, J. Zhu, and X. Li, "Video synchronization and its application to object transfer," *Image Vis. Comput.*, vol. 28, no. 1, pp. 92–100, 2010.
[4] Y. Caspi and M. Irani, "Spatio-temporal alignment of sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 11, pp. 1409–1424, Nov. 2002.
[5] C. Dai, Y. Zheng, and X. Li, "Accurate video alignment using phase correlation," *IEEE Signal Process. Lett.*, vol. 13, no. 12, pp. 737–740, Dec. 2006.
[6] F. Diego, D. Ponsa, J. Serrat, and A. López, "Video alignment for difference-spotting," in *Proc. Eur. Congr. Comput. Vis.*, 2008.
[7] F. Fraundorfer, C. Engels, and D. Nistér, "Topological mapping, localization and navigation using image collections," in *Proc. IEEE Conf. Intell. Robots Syst.*, 2007, vol. 1, pp. 3872–3877.
[8] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.
[9] K. L. Ho and P. Newman, "Detecting loop closure with scene sequences," *Int. J. Comput. Vis.*, vol. 74, no. 3, pp. 261–286, 2007.
[10] R. B. Langley, "Dilution of precision," *GPS World*, vol. 10, no. 5, pp. 52–59, May 1999.
[11] C. Lei and Y. Yang, "Trifocal tensor-based multiple video synchronization with subframe optimization," *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2473–2480, Sep. 2006.
[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[13] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. Brit. Mach. Vis. Conf.*, 2002, vol. 1, pp. 384–393.
[14] F. L. Padua, R. L. Carceroni, G. A. Santos, and K. N. Kutulakos, "Linear sequence-to-sequence alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 304–320, Feb. 2010.
[15] C. Rao *et al.*, "View-invariant alignment and matching of video sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 939–945.
[16] A. Ravichandran and R. Vidal, "Video registration using dynamic textures," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 514–526.
[17] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Pearson Education, 2003.
[18] P. Sand and S. Teller, "Video matching," *ACM Trans.*, vol. 22, no. 3, pp. 592–599, 2004.

[19] M. Singh, I. Cheng, M. Mandal, and A. Basu, "Optimization of symmetric transfer error for sub-frame video synchronization," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 554–567.
[20] J. Sivic and A. Zisserman, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds., "Video google: Efficient visual search of videos," in *Toward Category-Level Object Recognition*, 2006, vol. 4170, pp. 127–144.
[21] P. A. Tresadern and I. D. Reid, "Video synchronization from human motion using rank constraints," *Comput. Vis. Image Understand.*, vol. 113, no. 8, pp. 891–906, 2009.
[22] T. Tuytelaars and L. VanGool, "Synchronizing video sequences," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2004, vol. 1, pp. 762–768.
[23] Y. Ukrainitz and M. Irani, "Aligning sequences and actions by maximizing space-time correlations," in *Proc. Eur. Conf. Comput. Vis.*, 2006, vol. 3953, pp. 538–550.
[24] M. Ushizaki, T. Okatani, and K. Deguchi, "Video synchronization based on co-occurrence of appearance changes in video sequences," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2006, vol. 3, pp. 71–74.
[25] D. Wedge, D. Huynh, and P. Kovesi, "Using space-time interest points for video synchronization," in *Proc. IAPR Conf. Mach. Vis. Appl.*, 2007, pp. 190–194.
[26] L. Wolf and A. Zomet, "Wide baseline matching between unsynchronized video sequences," *Int. J. Comput. Vis.*, vol. 68, no. 1, pp. 43–52, 2006.
[27] J. Yan, J. Yan, and M. Pollefeys, "Video synchronization via space-time interest point distribution," in *Proc. Adv. Concepts Intell. Vis. Syst.*, 2004.
[28] L. Zelnik-Manor and M. Irani, "Multi-frame estimation of planar motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1105–1116, Oct. 2000.

Ferran Diego received the higher engineering degree in telecommunications from Politechnical University of Catalonia (UPC), Barcelona, Spain, in 2005, the M.Sc. degrees in language and speech from UPC and the University of Edinburgh, U.K., in 2005, and in computer vision and artificial intelligence from the Computer Vision Center and Universitat Autònoma de Barcelona, Spain, in 2007, and is currently working towards the Ph.D. degree at the Universitat Autònoma de Barcelona.

His research interests include video alignment, optical flow, sensor fusion, machine learning, and image retrieval.

Daniel Ponsa received the B.Sc. degree in computer science, the M.Sc. degree in computer vision, the Ph.D. degree, all from the Universitat Autònoma de Barcelona (UAB), Spain, in 1996, 1998, and 2007, respectively.

Since 1996, he has been giving lectures in the Computer Science Department of the UAB, where currently he is an Assistant Professor. He worked as full-time Researcher at the Computer Vision Center, UAB, from 2003 to 2010, in the research group on advanced driver assistance systems by computer vision. His research interests include tracking, motion estimation, pattern recognition, and machine learning.

Joan Serrat received the Ph.D. degree in computer science from the Universitat Autonoma de Barcelona (UAB), Spain, in 1990.

He is currently an Associate Professor in the Computer Science Department, UAB, and also member of the Computer Vision Center. His current research interest is the application of probabilistic graphical models to computer vision problems like feature matching, tracking and video alignment. He has also been head of several machine vision projects for local industries and member of the IAPR Spanish chapter board.

Antonio M. López received the B.Sc. degree in computer science from the Universitat Politècnica de Catalunya in 1992, the M.Sc. degree in image processing and artificial intelligence from the Universitat Autònoma de Barcelona (UAB) in 1994, and the Ph.D. degree in 2000.

Since 1992, he has been giving lectures in the Computer Science Department of the UAB, where he currently is an Associate Professor. In 1996, he participated in the foundation of the Computer Vision Center at the UAB, where he has held different institutional responsibilities, presently being the responsible for the research group on advanced driver assistance systems by computer vision. He has been responsible for public and private projects, and is a coauthor of more than 50 papers, all in the field of computer vision.