Video alignment for change detection

Ferran Diego, Daniel Ponsa, Joan Serrat and Antonio López

January 16, 2009

Abstract

We address the problem of aligning two video sequences. This means to synchronize them, that is, to establish the temporal correspondence between frames of the first and second video, and then to spatially register all the temporally corresponding frames. Video synchronization and alignment have been attempted before but most often in the simpler case of fixed or rigidly attached cameras and simultaneous acquisition. Also, restrictive assumptions have usually been assumed, like a linear time correspondence or the knowledge of complete trajectories of corresponding scene points, which to some extent limit its practical applicability. We intend to solve the more general problem of aligning video sequences recorded by independently moving cameras which follow a similar trajectory, 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 revealed complementary. Alignment results are presented in the context of videos recorded from vehicles driving along one same track, for different road types. In addition, we explore the feasibility, when one of the sequences is traffic free, to perform change detection as a means to detect moving vehicles.

1 Introduction

Image matching or registration has received a considerable attention for many years and is still an active research subject for its role in segmentation, recognition, sensor fusion, construction of panoramic mosaics, motion estimation, etc. Video matching or alignment, on the contrary, has been much less explored in spite of sharing with still image registration a number of potential applications. It has been used for visible and infrared camera fusion and wide baseline matching [1], high dynamic range video and video mating [2], action recognition [3] and loop closing detection in SLAM [4].
In general terms, it is a more complex problem than image registration because it requires alignment in both the temporal and spatial dimensions. Temporal alignment or synchronization means to find out a mapping from the time domain of the first sequence to the second one, such that each corresponding frame pair, one frame from each sequence, has the most 'similar content'. The simplest notion of similar content is that a warping can be found which spatially aligns one frame with the other, to the extent that they can be compared pixelwise.

The problem of video alignment can be stated more specifically, yet without any assumption to make it tractable, as follows. Let $F^o$ and $F^r$ be two video sequences $n_o$ and $n_r$ frames long, respectively. $F^r$ denotes the reference sequence and $F^o$ the 'observed' video, which we assume is entirely contained within $F^r$. Synchronization aims at estimating a discrete mapping $c(t_o) = t_r$ for all frames $t_o = 1 \ldots n_o$ of the observed video, such that frame $F^o_{t_o}$ maximizes some measure of similarity with frame $F^r_{t_r}$, among all frames of the reference sequence. The second part, registration, takes all corresponding pairs $\{(F^o_{t_o}, F^r_{t_r}), t_o = 1 \ldots n_o\}$ and warps frame $F^o_{t_o}$ so that it matches $F^r_{t_r}$, according to some similarity measure and a spatial deformation model.

1.1 Objective

The former general formulation does not actually state the problem we are going to address. It needs to be completed by the assumptions made which also determine, as we will comment in the past works review, the generality of the solution and its difficulty. Our goal is to synchronize videos recorded at different times, which can thus differ in intensity and content, i.e., show different objects under not exactly the same lighting conditions, up to an extent. They are recorded by a pair of independently moving cameras, although their motion is not completely free. For the video matching to be possible, there must be some overlapping in the field of view of the two cameras, when they are at the same or close position. Furthermore, we require that the relative camera rotations between corresponding frames are not too large and, more importantly, the cameras must follow approximately coincident trajectories. Independent camera motion has the key implication that the correspondence $c(t)$ is of free form: anyone of the two cameras may stop at any time. One sequence, called the reference, is assumed to contain the other, called 'observed', in the sense that there exist a corresponding reference frame for its first and last (and any other) observation frames. Finally, we do not want to depend on error–free or complete point trajectories, pro-
vided manually or by an ideal tracker, but to rely on the images themselves and possibly further data collected by different sensors. In sum, and for the sake of a greater practical applicability, we are choosing the most general (and difficult) settings of the problem, compared to previous works as we will see.

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 embarked in 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. Or, at least, changes 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. One of the difficulties of such classifiers have to face is the variability of vehicles in aspect, size and position within the image. This means a large number of windows (typically in the thousands) have to be processed looking for potential vehicles. To our knowledge, this is a novel approach to this application.

1.2 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. This is relevant to put into context our work, but also because, under the same generic label of synchronization, they try to solve different problems. The distinction is based on the input data and the assumptions made by each method. Accordingly, table 1 compares them.

The first proposed methods assumed the temporal correspondence to be a simple constant time offset \( c(t_o) = t_o + \beta \) \([5, 6, 7]\) or linear \( c(t_o) = \alpha t_o + \beta \) \([1, 3]\), to account for different camera frame rates. More recent works \([2, 8]\) let it be of free form. Clearly, the first case is simpler since just one or two parameters have to be estimated, in contrast to a curve of unknown shape.

Concerning the basis of these methods, most of them rely on the existence of an unknown geometric relationship between the coordinate systems of frames if they are corresponding: an affine transform \([3]\), a plane–induced homography \([1]\), the fundamental matrix \([7]\), the trifocal tensor \([5]\), and a deficient rank matrix made of the coordinates of point trajectories tracked along the whole sequence \([3, 6]\). This fact allows either to formulate some minimization over the time correspondence parameters (e.g. \( \alpha, \beta \)) or at least
to directly look for all pairs of corresponding frames. The cases in which this geometric relationship is constant \cite{1,3,8}, for instance because the two cameras are rigidly attached to each other, are easier to solve. To the best of our knowledge, only the works \cite{2,5,4} address like us the more difficult case of independently moving cameras, where no geometric relationship can be assumed beyond a more or less overlapping field of view.

Each method needs some input data which can be more or less difficult to obtain. For instance, feature-based methods require tracking one or more characteristic points along the two whole sequences \cite{1,8,6,7}, or points and lines in three sequences \cite{5}. In contrast, the so-called direct methods are based just on the image intensity or color \cite{1,2,3} which in our opinion is better from the point of view of practical applicability.

Our work is most closely related to \cite{2} in the sense of striving for a general solution to the video alignment problem, as table 1 shows. Beyond this, our approach is completely different. For instance, they do not adopt any explicit motion field model for corresponding frames, as we do. Also, their frame matching measure is based on feature point (Harris corners) correspondences, computed with an EM-like algorithm plus a Kanade–Lucas–Tomasi local motion optimization. We guess this makes their method highly dependent on having a number of such characteristic points evenly distributed on the images and along the whole sequences, as shown in their results. Finally, they align video sequences recorded by slowly panning the camera or moving it forward while being hand-held. In contrast, we try to align sequences recorded from vehicles driven on different road types, being the camera motion (and thus, the image content change) much faster.

1.3 Overview

Before deepening into the details of our method, we will expose its main ideas and the structure of the paper. We will focus first on synchronization, which is the most difficult part of the problem. Due to the independent motion of the cameras, we can not rely on the existence of a certain unknown but constant geometric entity relating the spatial coordinates of corresponding frames, like an homography or fundamental matrix, to be estimated along with the temporal correspondence. In addition, this later does not adopt a parametric form, so that we can not either to maximize some overall frame registration measure with regard to some parameters. The alternative, trying to spatially register each possible pair of frames, one from each sequence, is clearly infeasible due to its computational cost in sequences longer than a few seconds, at a frame rate of 25–30 frames per
second. We overcome this difficulty by formulating the synchronization as a maximum a posteriori (MAP) inference problem on a dynamic Bayesian network (DBN) (section 2). It has the advantage of allowing us to include our assumptions in probabilistic terms. More specifically, the Bayesian network is a multiple–observation Hidden Markov model of figure 3. The hidden variables $x_t, t = 1 \ldots n_o$ represent the number of frame in the reference sequence corresponding to the $t^{th}$ frame of the observed sequence. Each hidden node has two types of independent observations: $a_t$ and $s_t$, which are respectively an image descriptor of frame $F_t$ and the camera position estimation at time $t$ in the observed sequence. Hence, three conditional probabilities will be defined: $p(x_{t+1}|x_t), p(a_t|x_t)$ and $p(s_t|x_t)$. The first one will enforce the assumption that the camera/vehicle is either stopped or moves forward at some varying speed. The two later probabilities will express the necessary image similarity and GPS receiver position proximity for each pair of corresponding frames, respectively.

Section 3 briefly explains how do we perform the spatial registration of the frame pairs once the temporal correspondence has been computed. The key assumption is that in this case the two cameras are at the same position and only their pose may be different. Hence, a conjugate rotation homography relates the frame coordinates of each pair. A version of the well known Lucas–Kanade algorithm is employed to estimate the warping from the reference to the observed frame. Section 4 presents the synchronization results on five pairs of video sequences shot on different road types and evaluates them with regard the manually obtained ground–truth. On these sequences we also compare the contribution of the two types of observations, that is, we calculate the synchronization error using just appearance, just GPS and both of them. In addition, the experiment on the use of video alignment for vehicle detection is explained and its own results presented. Finally, section 5 summarizes this work and draws the main conclusions.

2 Video synchronization as an inference problem

We formulate the video synchronization problem as a labeling problem. A list of $n_o$ labels $x_{1:n_o} = [x_1 \ldots x_t \ldots x_{n_o}]$ has to be estimated. Each label $x_t \in \{1, ..., n_r\}$ is the number of the frame in the reference video corresponding to the $t^{th}$ frame of the observed sequence. To perform that, we rely on the available observations $y_{1:n_o}$, namely, the frames themselves of the observed sequence and the GPS data associated to them. We pose this task as a maximum a posteriori Bayesian inference problem,
\[
\mathbf{x}_{1:n_o}^{MAP} = \arg \max_{\mathbf{x}_{1:n_o} \in \mathcal{X}} p(\mathbf{x}_{1:n_o}|\mathbf{y}_{1:n_o})
\]

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

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

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 labels \(x_t\) increase monotonically. Therefore,

\[
P(x_{t+1}|x_t) = \begin{cases} v & \text{if } x_{t+1} \geq x_t \\ 0 & \text{otherwise} \end{cases}
\]

where \(v\) is a constant that gives equal probability to any label greater or equal than \(x_t\). 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 likelihood of the observations \(\mathbf{y}_{1:n_o}\) is independent given their corresponding label values, then \(p(\mathbf{y}_{1:n_o}|\mathbf{x}_{1:n_o})\) factorizes as

\[
p(\mathbf{y}_{1:n_o}|\mathbf{x}_{1:n_o}) = \prod_{t=1}^{n_o} p(\mathbf{y}_t|x_t)
\]

From these dependencies between variables, it turns out that our problem is one of MAP inference in a Hidden Markov model. Hence, we can apply the well-known Viterbi algorithm to exactly infer \(\mathbf{x}_{1:n_o}^{MAP}\).

As we have mentioned, at each time \(t\) we will have two types of observations: an image and some GPS positioning data. In this regard, we will turn now to precisely define the nature of the observations \(\mathbf{y}_{1:n_o}\) and the conditional probability \(p(\mathbf{y}_t|x_t)\).
2.1 Appearance likelihood

If two frames are corresponding then their content should be the similar. Likewise, the camera positions when they were recorded should be identical or very close to each other and only their pose could be different. As we will see in section 3, this means than one frame could be registered to its pair by a simple parametric mapping. One possibility then is to perform the registration and employ some alignment error measure, like the mean square error, in order to define part of the likelihood \( p(y_t|x_t) \). This is clearly infeasible in practice due to the huge number \( n_o n_r \) of image registrations to be calculated when sequences are just a few minutes long, and the computational cost of each one. Instead, we will resort to an image description which will be simple to compute and allows a fast yet effective comparison. It is a vector denoted by \( a \), after ‘appearance’, and is computed as follows.

The original image \( l \) (in our video sequences, \( 720 \times 576 \) pixels) is smoothed with a Gaussian kernel with \( \sigma = 2 \) and then downsampled to 1/16th of the original resolution. Of this sampled image the gradient \((l_x, l_y)\) 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, \( a \) is build by stacking the rows of \( l_x \) and next those of \( l_y \), and finally rescaling the resulting vector to unit norm.

The scalar product \( <a_t, a_{x_t}> \) can be seen as a simple similarity measure based on the coincidence of the gradient orientation in the subsampled images. This has proved to be a slightly better similarity measure in our sequences than others based on the intensity or the gradient magnitude. In addition, gradient orientation is less influenced by lighting changes.

\( p(a_t|x_t) \) is the probability of the frame pair \((F^o_t, F^r_{x_t})\) to be corresponding given that they are represented by \( a_t \) and \( a_{x_t} \), respectively. Since the two vectors \( a_t, a_{x_t} \) are normalized, their scalar product is the cosine of the angle between them. From it, we define the appearance likelihood as

\[
p(a_t|x_t) = \Phi(<a_t, a_{x_t}>, 1, \sigma_a^2)
\]

where \( \Phi(v; \mu, \sigma^2) \) denotes the evaluation of the Gaussian pdf \( \mathcal{N}(\mu, \sigma^2) \) at \( v \). The higher likelihood is hence the closer \( <a_t, a_{x_t}> \) to 1. We have set \( \sigma_a = 0.5 \) in order to give a significant likelihood only to frames whose appearance vectors form an angle less than 5 degrees, approximately.

We need this likelihood to be high when frames are similar in spite of slight camera rotation and translation, and when they contain different, small to medium sized scene objects like vehicles in road sequences.
In order to deal with camera rotations and translations, \( \mathbf{a}_t \) is computed from horizontal and vertical translations of the low resolution smoothed image up to \( \pm 2 \) pixels. Then, the scalar product is performed between the appearance \( \mathbf{a}_{x_t} \) and all 25 appearances computed this way. The maximum obtained value is then used in equation (5). Figure 2 shows an example of \( p(\mathbf{a}_t|\mathbf{x}_t) \) for a pair of complete sequences.

2.2 GPS likelihood

The other observation of our DBNs is the GPS data which are acquired from a vehicle equipped with a Keomo 16 channel GPS receiver with Nemerix chipset. The GPS device localizes the vehicle in geospatial coordinates once per second, hence providing GPS data every 25 frames. The GPS location is extracted from the GPGGA message of the NMEA protocol, since it provides the horizontal dilution of precision [9] (HDOP) of the given location. This HDOP value \( h \) is related to the location uncertainty. After converting the GPS location to corresponding 2D coordinates \( \mathbf{g} = [x \ y]^T \) in the Universal Transverse Mercator (UTM) system, we combine \( h \) with a user equivalent range error [9] sensible for our receiver (in our case, \( \sigma = 1.5 \) meters) to determine the Gaussian distribution \( \mathcal{N}(\mathbf{g}, \mathbf{R} = (h\sigma)^2 \mathbf{I}) \) that encodes the available knowledge in the vehicle location uncertainty (\( \mathbf{I} \) denotes the identity matrix). Hence, the raw sensor information available for a given frame \( \mathbf{F}_t^* \) of a sequence is the acquired image itself, a variable \( o_t \in \{0, 1\} \) whose value is 1 when it has an associated GPS fix, and the distribution \( \mathcal{N}(\mathbf{g}_t, \mathbf{R}_t) \) only if \( o_t = 1 \).

The GPS information is available only in 4% of the sequence. However, for the rest of frames there is still some knowledge that can be exploited, since a car follows a regular trajectory. Then, in order to estimate an observation to each frame, we apply a Kalman smoother to process the available GPS fixes \( \mathcal{N}(\mathbf{g}_t, \mathbf{R}_t) \) and interpolate the lacking information \( \mathcal{N}(\mathbf{s}_t, \mathbf{\Sigma}_t) \) (\( \mathbf{s} \) stands for smoothed GPS information). To do so, we model the dynamical behaviour of the vehicle to propagate the GPS information to the frames where it is not available. We find out that a model of constant acceleration gives a good approximation of the dynamics. This can be expressed by the third order autoregressive model

\[
\mathbf{g}_t = 3\mathbf{g}_{t-1} - 3\mathbf{g}_{t-2} + \mathbf{g}_{t-3} + \mathbf{w}_t
\]

where \( \mathbf{w}_t \) is a stochastic disturbance term \( \mathcal{N}(\mathbf{0}, \mathbf{Q}_t) \) that accounts for the model inaccuracies. In our experiments, we set \( \mathbf{Q}_t = 2.25e^{-4}\mathbf{I} \), which means
that the model imprecision after one second (i.e., 25 frames) is below 0.75 meters with 0.95 probability. We combine this model with the GPS observations by means of the Rauch-Tung-Striebel Kalman Smoother equations [10], using the prediction of the GPS location in frames where no GPS fix was available. As results, a Gaussian distribution constraining the GPS location at each frame is finally obtained.

For the case of GPS observations, notice that defining $p(g_t|x_t)$ or $p(s_t|x_t)$ implies specifying them for any value of $x_t$, and this requires having GPS information in all the frames of $F^r$. Hence, both likelihood terms are defined using the smoothed GPS estimations $N(s_{x_t}, \Sigma_{x_t})$ of the $F^r$ frames. Like 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 $\Phi(v; s_{x_t}, \Sigma_{x_t})$, where $v$ would correspond respectively to $g_t$ or $s_t$ of the observed frame. However, the GPS data associated to the observed video sequence frames does not limit to just a location, but includes also its uncertainty in the form of a Gaussian distribution. Hence, it is more proper to evaluate this likelihood taking all the feasible GPS locations into account. Hence, the GPS likelihood is the evaluation of $s_t$ with respect to a Gaussian distribution $N(s_{x_t}, \Sigma_{x_t} + \Sigma_t)$, which takes both distributions into account:

$$p(s_t|x_t) = \Phi(s_t; s_{x_t}, \Sigma_{x_t} + \Sigma_t) \quad (7)$$

2.3 Dynamic bayesian network synchronization models

We have considered the four DBNs represented in figure 3. Square nodes represent discrete variables and circular nodes continuous variables. Shaded nodes denote observed variables, non–shaded nodes are hidden variables. The conditional dependencies between variables are represented by solid lines. Notice 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. We use this notation to model the fact that the GPS receiver does not provide one raw GPS fix for all the frames in a sequence (figure 3c) because of the lower rate of the GPS receiver with respect to the camera. Only when a frame has a raw GPS fix associated (i.e., $o_t = 1$) the node $g_t$ is connected to its parent $h_t$, the HDOP. Elsewhere, the effective graphical model of figure 3c becomes to that of figure 3a.
The four DBNs have been proposed in order to assess the contribution of each type of observation to the final synchronization result. That is, 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. Also, we want to motivate the need for the GPS smoothing (having an estimation of the GPS data at each frame) instead of just taking the raw GPS fix (GPS data every 25 frames). Figures 3a to 3d show, respectively, the DBN for appearance only, smoothed GPS coordinates only, appearance combined with raw GPS coordinates and appearance combined with smoothed GPS coordinates.

3 Registration

The result of the synchronization is a list of pairs of corresponding frame numbers \((t, x_t), t = 1 \ldots n_o\). Ideally, for each such pair the camera was at the same position. In that case, 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 be seen then that the coordinates of the two corresponding frames \(F^o_t, F^r_t\) are related by the homography \(H = KRR^{-1}\), where \(K = \text{diag}(f, f, 1)\), \(f\) being the camera focal length in pixels. Let the 3D rotation \(R\) be parametrized by the Euler angles \(\Omega = (\Omega_x, \Omega_y, \Omega_z)\) (pitch, yaw and roll respectively). Under the assumptions of these angles being small and the focal length being large enough, the motion vector field associated to this homography can be approximated by the following model \([11]\), which is quadratic in the \(x\) and \(y\) coordinates but linear in the parameters \(\Omega\):

\[
\mathbf{u}(x; \Omega) = \begin{bmatrix}
-\frac{x y}{f} & f + \frac{y^2}{f} & -y \\
-f - \frac{x^2}{f} & \frac{x y}{f} & x
\end{bmatrix}
\begin{bmatrix}
\Omega_x \\
\Omega_y \\
\Omega_z
\end{bmatrix}
\] (8)

\(R\) and consequently \(\Omega\) may be different for each pair, since the cameras have moved independently. Therefore, for each pair of frames we need to estimate the parameters \(\Omega\) 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 \([12]\).

\[
err(\Omega) = \sum_x \left[ F^o(x) - F^o(x + \mathbf{u}(x; \Omega)) \right]^2
\] (9)
where $F^r_t$ is the template image, $F^r_{x_t}$ is the image warped onto the coordinate frame of the template. The previous minimization is performed iteratively until convergence. In practice, we can not directly solve for $\Omega$ because a first order approximation of the error in equation (9) can be made only if the motion field $u(x; \Omega)$ is small. Instead, $\Omega$ is successively estimated in a coarse–to–fine manner. A Gaussian pyramid is built for both images and at each resolution level $\Omega$ is re–estimated based on the value of the previous level. For a detailed description we refer the reader to [12].

4 Results

4.1 Synchronization results

We have aligned five video sequence pairs with the four DBN models and evaluated the synchronization error. They were recorded with a SONY DCR–PC330E camcorder on different environments: a university campus at day (‘campus’ pair) and night, a suburban street, a back road and a highway. Figure 4 shows a few sample frames of each one. The back road and the night pair contain few distinct scene features compared with the suburban street and the campus pairs, which are populated by a number of buildings, parked cars and lamp posts, on both sides of the image. In addition, the GPS reliability was also different in the campus and the street sequences due to the proximity of tall buildings and also to the variation of the number of visible satellites along time.

In order to quantitatively assess the performance of the temporal alignment, we manually obtained the ground–truth for all five 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 the position and size of the closest static scene objects were taken into account, like lane markings, traffic signs, other cars etc. This decision, however, often proved difficult to make because the vehicle undergoes lateral and longitudinal relative displacements, to which camera pose variations are added. Figure 5 illustrates the difficulty of making a single decision. Therefore, we ended up by selecting not a single frame number but an interval $[l_t, u_t]$ always containing the true corresponding frame. This can be appreciated in figure 6a. The width of the ground truth intervals thus obtained is typically just 3 to 6 frames.

The synchronization error at time $t$ given the corresponding frame number $x_t$ is the distance of $x_t$ to the closest ground–truth interval boundary,
\[
\text{err}(t) = \begin{cases} 
0 & \text{if } l_t \leq x_t \leq u_t \\
(l_t - x_t) & \text{if } x_t < l_t \\
(x_t - u_t) & \text{if } x_t > u_t 
\end{cases}
\]  

(10)

The simplest way to evaluate the performance of the temporal alignment is averaging all the individual errors, \( \overline{e} = \frac{1}{n_o} \sum_{t=1}^{n_o} \text{err}(t) \). Figure 8 shows the average error for each of the five sequences synchronized by means of the four DBNs. We can clearly appreciate how the combination of the two types of observations, appearance and smoothed GPS, decreases substantially the average error in all the tested sequences. Also, it pays to compute the smoothed GPS since we obtain always a much better result than with raw GPS when combined with the appearance observations. Finally, note that the average error in this best case is always under one frame in all the pairs.

The low performance of the only smoothed GPS DBN on the three suburban pairs ('campus', 'street' and to a lesser extent 'night') is due to the problems of multiple path reception and satellite outage. This is further illustrated by figure 7 which plots the standard deviation of the estimated GPS covariances \( \Sigma_t, \Sigma_{x_t} \), and the distance between the positions \( s_t \) and \( s_{x_t} \) necessary for the calculation of the GPS likelihood in equation 7. The proximity of tall buildings and a relatively large vehicle trajectory distance combine to reduce the GPS likelihood, which in the absence of other sensors observation induce a lower performance.

The average error seems to be a sensible measure to calculate the performance because it is an overall measure but it does not distinguish an slight increase of outliers from a general reduction in accuracy. Synchronization outliers are time correspondence offsets which hamper the subsequent spatial registration and change detection steps. We need an error representation which explains how the error is. The distribution of the time correspondence error is more informative in this regard because it tells us how much frames are at given distance of the ground truth, for all distances. Figures 10 to 14 show the error distribution for the five pairs. We can appreciate more than 70% of frames of the highway and back road pairs do not have any synchronization error. In the other three pairs recorded at suburban environments we need to add the frames with error equal to one frame to exceed this score. In these sequences, some tall buildings close to the road degrade the GPS data, dragging the correspondence curve in the wrong direction at some places. However, the combination of appearance and smoothed GPS observations decreases the error in this cases because this other observation type ‘drags’ the correspondende curve to the right direction, as shown in figure 15. The opposite situation happens, too: when the image similarity
measure fails because the content is too different (new objects appear or the relative camera rotation or translation is too large), some reliable GPS data can draw the correspondence curve to the right place.

A figure containing just a few frames of the aligned videos would be a poor reflex 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/CVIU. To assess the quality of the spatial registration we perform a simple image fusion assigning the reference frame to the red channel of a color image and the registered observed frame to the green channel.

4.2 Vehicle detection

Recall that our final goal is the pointwise subtraction of videos in order to spot differences of potential interest. Thus, we intend to perform a sort of change detection but in the case of a special dynamic background, that produced by the camera forward motion. As an illustration of this idea, we have addressed the problem of road vehicle detection for driving assistance. Suppose we record the reference sequence 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.

The video alignment method we have described would give us a pair of videos which we could subtract pixelwise, but of course, after they were recorded. However, an off-line method does not make much sense for this application since a fast and on-line detection is required. It turns out that a relatively simple modification of the inference mechanism can adapt our method to this on-line setting. Instead of calculating the MAP inference on a network formed by observation and hidden nodes representing all the observed video sequence, a truly dynamic Bayesian network is built on-line and a different type of inference is carried out called fixed-lag smoothing. It can be formulated as

\[
x_{t-l}^{MAP} = \arg \max_{x_{t-l} \in \mathcal{L}} p(x_{t-l} | y_{t-L:t})
\]

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 frames used to infer the label \( x_{t-l} \). Fixed-lag smoothing infers the label \( x_{t-l} \) at time \( t \), that is to say, gives an on-line answer but with a delay of \( l \) frames. The specific values assigned to \( \mathcal{L} \) and \( l \) are 75 and 25 time units, respectively (or frames, loosely speaking).
Vehicle detection proceeds as follows. Once we have registered the corresponding frames $F_{t-l}^o$ and $F_{x, t-l}^x$, we subtract their respective R, G and B channels. Then, we threshold the absolute value of the differences and take the logical Or at each pixel. Next, we filter out the binary regions larger than a certain area which depends linearly on the row number to account for the changing size of objects due to perspective. Also, we discard those regions with an eccentricity larger than a fixed threshold to remove elongated objects. Finally, the bounding boxes of the remaining regions are considered vehicles.

We do not claim this procedure is a competitive vehicle detector. Rather, video alignment has allowed us to perform a difficult task with very simple operations. We regard it as a means to select a few image regions which could contain the objects of interest to be fed to a specialized classifier. Nonetheless, we have performed a quantitative evaluation on part of the back road sequence. The chosen metric was the sequence frame detection accuracy (SFDA) with binary thresholding of 0.5 [13]. On a single frame, the frame detection accuracy with binary thresholding is defined as

$$\text{FDA}(t) = \frac{\text{Overlap}(t)}{(N_G(t) + N_D(t))/2}$$

$N_G(t)$ and $N_D(t)$ denote the number of ground truth and detected objects in frame $t$, respectively. The numerator is the number of detected objects for which the spatial overlap with one ground truth object is greater than 0.5 of their area. Previously, an optimal one-to-one correspondence among detected and ground truth objects has been computed. The SFDA measures the detection performance in the whole sequence. It is the sum of FDA($t$) for all the frames divided by the number of frames where at least on ground truth or one detected object exists. On the mentioned sequence, we have obtained an SFDA of 0.76.

Figure 16 shows some detection examples 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/CVIU](http://www.cvc.uab.es/ADAS/projects/sincro/CVIU) where the original, aligned, difference and detection videos can be played.

## 5 Conclusions

In this paper, we have introduced a novel approach to the video alignment. It relies on a MAP inference problem on a Bayesian network in order to deal with the more general problem of aligning video sequences recorded by
independent moving cameras which follow a similar trajectory, based only on the fusion of image intensity and GPS information. These observations have been formulated in probabilistic terms to introduce them in their respective Bayesian networks. We present a measure of temporal alignment to properly compare the performance of the four DBNs using the first ground truth for evaluating it. This measure asserts that the combination of frame appearance and GPS data increases the accuracy of synchronization with respect to both observations used separately. We have successfully applied it to align video sequences from moving vehicles recorded at different challenging scenarios in order to perform a change detection. In addition, we have successfully applied a simple modification of the inference mechanism to adapt our method on an on-line setting and perform the change detection as a means to detect moving vehicles, when one of the sequences is traffic free.

Acknowledgments

This work was supported by Spanish Ministry of Education and Science (MEC) under Project TRA2007-62526/AUT, Research Program Consolider Ingenio 2010: MIPRCV (CSD2007-00018) and MEC grant AP2007-01558 (first author).

References


Figure 1: Video alignment concept: temporal and spatial registration.
Figure 2: Typical appearance likelihood $p(a_t|x_t)$ for every $t$ and $x_t$. 
Figure 3: Four compared DBNS. See text for a description of the notation.
Figure 4: Sample frames of testing sequences: a) 'campus', b) 'night', c) 'street', d) 'highway' and e) 'back road'.
Figure 5: An example of the dilemma of making a single decision.
Figure 6: Detail of ground truth intervals (solid lines) and synchronization result (the dotted line), on a background inversely proportional to frame similarity.
Figure 7: Variables that affect the GPS likelihood $p(s_t|x_t)$ for the 'street' sequence pair. a) Standard deviation of observed reference (solid line) and ground truth aligned reference sequence (dashed line), b) GPS distance $\|s_t - s_{x_t}\|$ and c) GPS likelihood $p(s_t|x_t)$. 
Figure 8: Average error $\overline{e}$ with the four DBNs on each video sequence pair for each sequence alone and the union of them.
Figure 9: Distribution of the synchronization error of the union of all sequence pairs.
Figure 10: Distribution of the synchronization error of the 'back road' sequence pair.
Figure 11: Distribution of the synchronization error of 'street' sequence pair.
Figure 12: Distribution of the synchronization error of the campus sequence pair.
Figure 13: Distribution of the synchronization error of 'night' sequence pair.
Figure 14: Distribution of the synchronization error of 'highway' sequence pair.
Figure 15: Two examples of how the two types of observations complement each other. Dashed line is the time correspondence using only appearance, solid line using only smoothed GPS data and dotted line using both of them. The truth lies mostly under the dotted line.
Figure 16: Video alignment results: a) fusion, b) absolute difference, c) vehicle detection. Results in video form can be properly viewed at www.cvc.uab.es/ADAS/projects/sinco/CVIU/.
<table>
<thead>
<tr>
<th></th>
<th>Method 1</th>
<th>Method 2</th>
<th>Method 3</th>
<th>Method 4</th>
<th>Method 5</th>
<th>Method 6</th>
<th>Method 7</th>
<th>Method 8</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time correspondence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>constant offset</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>linear</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unconstrained</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simultaneous recording</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yes</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>not necessary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cameras rigidly attached</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>no</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>Input Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>point trajectories</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>line trajectories</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>point features</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>images</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>images + GPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Need to estimate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fixed homography</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fundamental matrix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trifocal tensor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>deficient rank matrix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fixed affine motion field</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>variable motion field</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>frame similarity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>