Video alignment for difference-spotting

Ferran Diego, Daniel Ponsa, Joan Serrat and Antonio López

Computer Vision Center & Computer Science Dept.
Edifici O, Universitat Autònoma de Barcelona,
08193 Cerdanyola, Spain
{ferran.diego,daniel,joan.serrat,antonio}@cvc.uab.es

Abstract. We address the synchronization of a pair of videos sequences captured from moving vehicles and the spatial registration of all the temporally corresponding frames. The final goal is to fuse the two videos pixel–wise and compute their pointwise differences. Video synchronization has been attempted before but often assuming restrictive constraints like fixed or rigidly attached cameras, simultaneous acquisition, known scene point trajectories etc. which to some extent limit its practical applicability. We intend to solve the more difficult problem of independently moving cameras which follow a similar trajectory, based only on the fusion of image intensity and GPS data information. The novelty of our approach is the probabilistic formulation and the combination of observations from these two sensors, which have revealed complementary. Results are presented in the context of vehicle pre–detection for driver assistance, on different road types and lighting conditions.

1 Introduction

Consider the following scenario. A vehicle is driven 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. Imagine that, somehow, we are able to substract pixelwise the two sequences. That is, for each frame of, say, the first sequence, we get to know which is the corresponding frame in the second sequence, in the sense of being the camera at the same location. In addition, suppose we succeed in spatially aligning every such pair of frames, so that they can be properly substracted to build the frames of the difference video. What would it display ? Moving objects and objects present in only one of the video sequences, like pedestrians and on road vehicles, provided ambient illumination was similar enough and these objects exhibit sufficient contrast with respect to their ”background” (what’s behind them) in the other sequence.

Let S^o, S^r be two video sequences n_o and n_r frames long, respectively. S^r denotes the reference sequence and S^o the ’observed’ video, which is contained within S^r. Video alignment or matching requires the simultaneous correspondence of two image sequences both in the time and space dimension. The first part, which we refer to as 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 $S^r(t_r)$

maximizes some measure of similarity with $S^o(t_o)$, among all frames of S^r. In the former scenario, we assume this will happen when the location where $S^r(t_r)$ was recorded is the closest to that of $S^o(t_o)$. The second part, registration, takes all corresponding pairs $S^o(t_o), S^r(c(t_o)), t_o = 1 \dots n_o$ and warps frame $S^o(t_o)$ so that it matches $S^r(c(t_o))$, according to some difference measure and a spatial deformation model.

1.1 Previous work

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. Table 1 compares them.

The first proposed methods assumed the temporal correspondence to be a simple constant time offset $c(t_o) = t_o + \beta$ [1,2,3] or linear $c(t_o) = \alpha t_o + \beta$ [4,5], to account for different camera frame rates. More recent works [6,7] 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 a geometric relationship between the coordinate systems of frames *if* they are corresponding: an affine transform [5], a plane–induce homography [4], the fundamental matrix [3], the trifocal tensor [1], and a deficient rank condition on a matrix made of the complete trajectories of tracked points along a whole sequence [7,2]. This fact allows either to formulate some minimization over the time correspondence parameters (e.g. α, β) or at least to directly look for *all* pairs of corresponding frames. Again, the cases in which this geometric relationship is constant [4,5,7], for instance because the two cameras are rigidly attached to each other, are easier to solve. Other works [6,1,8,9] address the more difficult case of independently moving cameras, where no geometric relationship can be assumed beyond a more or less overlaping 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 [4,7,2,3], or points and lines in three sequences [1]. In contrast, the so–called direct methods are based just on the image intensity or color [4,6,5] which in our opinion is better from the point of view of practical applicability.

Like still image registration, video alignment has a number of potential applications. It has been used for visible and infrared camera fusion and wide baseline matching [4], high dynamic range video, video mating and panoramic mosaicing [6], visual odometry [9], action recognition [5] and loop closing detection for SLAM [8].

		[2]	[4]	[1]	[7]	[3]	[5]	[6]	[9]	[8]	This work
Time correspondence	constant offset	•		•		•					
	linear		•				•				
	unconstrained				•			•	•	•	•
Simultaneous recording	yes	•	•	•		•					
	not necessary				•		•	•	•	•	•
Cameras rigidly attached	yes	•	•		•		•				
	no			•		•		•	•	•	•
Input Data	point trajectories	•	•		•	•					
	line trajectories			•							
	point features							•			
	images		•				•			•	•
	images + map/GPS								•		•
Need to estimate	fixed homography		•								
	fundamental matrix					•			•		
	trifocal tensor			•							
	deficient rank matrix	•			•						
	fixed affine motion field						•				
	variable motion field							•			
	frame similarity								•	•	•

Table 1. Comparison of video synchronization methods.

1.2 Objective

Our goal is to synchronize videos recorded at different times, which can thus differ in intensity and even in content, i.e., show different objects or actions, 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 they follow *approximately* coincident trajectories and, more importantly, that the relative camera rotations between corresponding frames are not too large. Independent camera motion has the important implication that the correspondence $c(t)$ is of free form: anyone 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. In sum and for the sake of a greater practical applicability, we are choosing the most difficult settings of the problem among those reviewed before.

Our motivation for addressing video alignment is to spot differences between two videos. The envisaged application is the vehicle and pedestrian recognition from onboard cameras, in the context of driver assistance systems. Specifically, we intend to perform a sort of pre–detection, that is, to select regions of interest on which supervised classifiers should be applied. One of the difficulties of such classifiers have overcome is the variability of such objects in size and position within the image. This means a large number of windows (typically in the thousands) have to be processed looking for potential objects. In addition, recent results with state–of–the art methods like boosting indicate that a fixed, known background greatly simplifies the complexity of the problem [10]. To our knowledge, this is a novel approach to the problem. At the moment, our method just performs an off–line pre–detection, which nevertheless speeds up the manual selection of positive and negative samples for a classifier. Additionally, and as a byproduct, video synchronization can be used for vehicle localization. If the reference sequence has associated positioning data like map coordinates, then we are already locating the frames of the other one.

Our work is most closely related to [9] (see table 1). However, we differ from them in the motivation and also in the method. For instance, this need to compute the fundamental matrix between any potential pair of corresponding frames (wich are spherical panoramas), as part of a frame similarity measure and to estimate the camera ego–motion. Like us, they solve the problem by inference on a Bayesian network, but theirs is a Markov Random Field (MRF) and inference is approximated whereas we perform exact inference on a Dynamic Bayesian Network (DBN). Finally, they need as input a map of the camera trajectory, which is compared with the estimated ego–motion. In contrast, we can obtain a solution from video data alone or combine them with GPS observations.

2 Video synchronization as an inference problem

We formulate the video synchronization problem as a labelling problem. A list of n_o labels $\mathbf{x}_{1:n_o} = [x_1 \dots x_{n_o}]$ with $x_t \in \{1, ..., n_r\}$ has to be estimated, each label x_t is the number of the corresponding frame in S^r to the t^th frame of S^o, that we denoted as t. To perform that, we rely on the observations available at each frame (appearance and GPS data). We pose this task as a Bayesian inference problem, being the desired sequence as the one maximizing $p(\mathbf{x}_{1:n_o}|\mathbf{y}_{1:n_o})$. That is,

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

where $\mathbf{y}_{1:n_o}$ are the observation of frames in S^o and $\mathcal{X}$ is the set of all possible labellings. Both sequences have been recorded by a vehicle following a trajectory, provided that the vehicle is just constrained to forward motion, that is, the vehicle can not reverse its motion direction. Therefore, the sequence of labels are necessarily to increase monotonically. The prior $P(\mathbf{x}_{1:n_o})$ can be factored as

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

where the constraining of increasing sequence is imposed by defining a label transition matrix of the form

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

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 S^o could be any subsequence inside S^r.

If we also assume that the likelihood of 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

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

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

We have considered the use of different observation types, leading to the four DBNs represented in Fig. 1. Square nodes represent discrete variables, while the rounded ones correspond to continuous variables. Shaded nodes denote observed variables. The conditional dependency between variables is represented by solid lines while dashed lines represent switching dependency. The switching dependency is a relation between nodes that express that a variable's parents are allowed to change depending on the current value of other parents. Notice that observations coming from different sensors can be assumed independent if they are not related physically. In our case, we use this mechanism to model that the GPS receiver does not provide raw GPS fix for all the frames in a sequence (Fig.1-c). Only when a videoframe has a raw GPS fix associated (i.e. $o_t = 1$) the node $\mathbf{g}_t$ is connected to its parent. Otherwise, the effective graphical model corresponds to the one in Fig. 1-a.

The four DBN have been proposed in order to assess the contribution of each observation separately (appearance and GPS data) and their combination. Note that the combination of GPS with the frame appearance can be only done every 25 frames, because of the GPS receiver rate, which have GPS information or estimate him for the entire sequence. The likelihood probabilities of the DBNs are explained below.

2.1 Appearance likelihood density

Let $\mathbf{F}_t^o$ and $\mathbf{F}_{x_t}^r$ denote the t^th and x_t^{th} frames of the observed and reference videos, respectively. The appearance $\mathbf{a}_t$ refers to the image description used in the definitions of the observation likelihood $p(\mathbf{a}_t|x_t)$, that is, the probability of (t, x_t) to be corresponding frames given $\mathbf{F}_t^o$ is represented by the feature vector

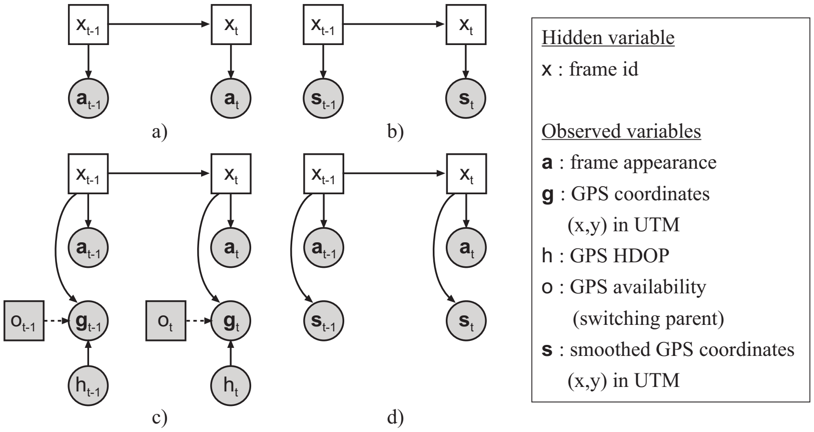


Fig. 1. Graphical model representation of the different approaches used to align videos. a) Using appearance. b) Using smoothed GPS coordinates. c) Combining appearance with GPS fixes when available. d) Combining appearance with smoothed GPS coordinates.

$\mathbf{a}_t$ and $\mathbf{F}^r_{x_t}$ by an analogous vector $\mathbf{a}_{x_t}$. The adopted description must be simple to compute and compare, since we will need to calculate $p(\mathbf{a}_t|x_t)$ for all possible frame pairs, ı.e., millions of times of sequences of just a few minutes long. At the same time, we want the probability to be high when frames are similar, in spite of slight camera rotations and translations, contrast or lighting changes and, of course, when they show different background objects like vehicles. To this end we propose to downsample the image at one fourth of resolution, compute the gradient orientation at each pixel and stack all the columns into a single vector $\mathbf{a}_t$. The scalar product $< \mathbf{a}_t, \mathbf{a}_{x_t} >$ can be seen as a simple similarity measure. The appearance likelihood is then defined as

$$p(\mathbf{a}_t|x_t) = \Phi(< \mathbf{a}_t, \mathbf{a}_{x_t} >; 1, \sigma_a^2) \ ,$$

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 $< \mathbf{a}_t, \mathbf{a}_{r_t} >$ 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.

2.2 GPS likelihood densities

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 [11] (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 [11] 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 $S(t)$ 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 [12], 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 $p(\mathbf{g}_t|x_t)$ or $p(\mathbf{s}_t|x_t)$ implies specifying them for any value of x_t, and this requires having GPS information in all the frames of S^r. Hence, both likelihood terms are defined using the smoothed GPS estimations $\mathcal{N}(\mathbf{s}^r, \mathbf{\Sigma}^r)$ of the S^r frames. Like in the case of $\mathbf{a}$, the likelihood of the observed GPS data (whether raw or smoothed) could be defined as the evaluation of $\Phi(\mathbf{v}; \mathbf{s}^r_t, \mathbf{\Sigma}^r_t)$, where $\mathbf{v}$ would correspond respectively to $\mathbf{g}_t$ of $\mathbf{s}_t$ of the observed frame. However, the GPS data in S^o frames does not limit to just a location, but a Gaussian distribution of this location. Hence, it is more proper to evaluate its likelihood taking all the feasible GPS locations into account. That is, compute the expected value of its likelihood according to the distribution of its associated GPS data. For instance, for the case of smoothed GPS coordinates, this corresponds to

$$E[p(\mathbf{s}_t = \mathbf{s}^o|x_t)] = \int \Phi(\mathbf{s}; \mathbf{s}^r_t, \mathbf{\Sigma}^r_t)\Phi(\mathbf{s}; \mathbf{s}^o, \mathbf{\Sigma}^o)d_{\mathbf{s}} \ \ ,$$

i.e., it is the integral of the product of two Gaussians. Since this product equals to an unnormalised Gaussian, computing its integral is just determining the inverse of its missing normalization constant, which is obtained from the parameters of the multiplied Gaussians from the following expression [13]

$$E[p(\mathbf{s}_t = \mathbf{s}^o | x_t)] = \frac{1}{2\pi\sqrt{|\mathbf{\Sigma}_t^r + \mathbf{\Sigma}^o|}} \exp\left(-\frac{1}{2}(\mathbf{s}_t^r - \mathbf{s}^o)^T (\mathbf{\Sigma}_t^r + \mathbf{\Sigma}^o)^{-1} (\mathbf{s}_t^r - \mathbf{s}^o)\right) \quad .$$

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 frames corresponding frames $\mathbf{F}_t^o, \mathbf{F}_{x_t}^r$ are related by the homography $H = KRK^{-1}$, 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 $\mathbf{\Omega} = (\mathit{\Omega}_x, \mathit{\Omega}_y, \mathit{\Omega}_z)$ (pitch, yaw and roll respectively). Under the assumptions of these angles being small and the focal length being large enough, the motion vector of this homography can be approximated by the following model [14], which is quadratic in the x and y coordinates but linear in the parameters $\mathbf{\Omega}$:

$$\mathbf{u}(\mathbf{x}; \mathbf{\Omega}) = \begin{bmatrix} -\frac{xy}{f} & f + \frac{x^2}{f} & -y \\ -f - \frac{y^2}{f} & \frac{xy}{f} & x \end{bmatrix} \begin{bmatrix} \mathit{\Omega}_x \\ \mathit{\Omega}_y \\ \mathit{\Omega}_z \end{bmatrix} \tag{1}$$

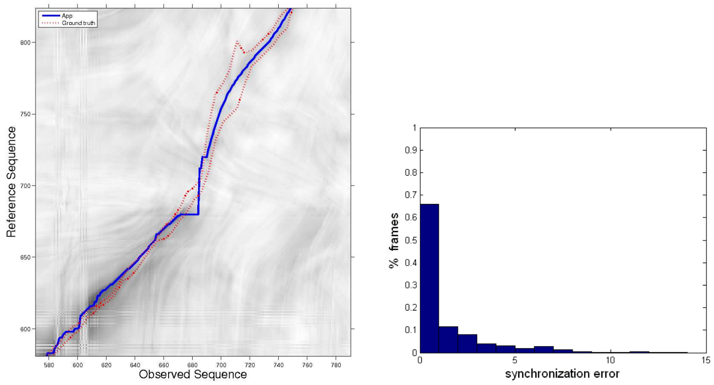
R and consequently $\mathbf{\Omega}$ may be different for each pair, since the cameras have moved independently. Therefore, for each pair of frames we need to estimate the parameters $\mathbf{\Omega}$ that minimize some registration error. The chosen error measure is the sum of squared *linearized* differences (i.e., the linearized brigthness constancy) that is used by the additive forward extension of the Lucas–Kanade algorithm [15],

$$\text{err}(\mathbf{\Omega}) = \sum_{\mathbf{x}} \left[\mathbf{F}_{x_t}^r(\mathbf{x} + \mathbf{u}(\mathbf{x}; \mathbf{\Omega})) - \mathbf{F}_t^o(\mathbf{x})\right]^2 \tag{2}$$

where $\mathbf{F}_t^o$ is the template image, $\mathbf{F}_{x_t}^r$ 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 $\mathbf{\Omega}$ because a first order approximation of the error in Eq. (2) 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 frames 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 [15].

4 Results

Four video pairs have been aligned and substracted. They were recorded on different road types: highway, rural and urban roads, so that they exhibited an increasing amount of 'structure'. Rural sequences contain less distinct landscape features, whereas urban sequences are populated by a number of buildings, parked cars and lamp posts on the left and right sides of the image. In addition, two urban sequence pairs were recorded at different places, one at day and another at night, to test very different lighting conditions. Not only the amount of content and lighting varied, but the GPS reliability was also different due to the proximity of tall buildings in the daytime urban sequence.



a) b)

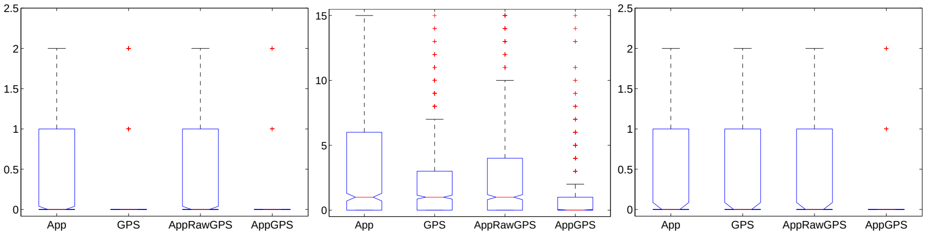
Fig. 2. Synchronization error for the rural sequence. a) detail of ground truth versus synchronization result, on a background proportional to frame similarity (the darkest, the higher) b) normalized histogram of the synchronization error.

In order to quantitatively assess the performance of the temporal alignment, we manually obtained the ground–truth for three of the video pairs: urban daytime, urban night and rural, which are 1550, 1020 and 325 frames long, respectively. For one out of every five frames t of the observed video we tried to select the corresponding frame x_t in the reference video and perform a linear interpolation inbetween. To do it, mainly the position and size of the closest scene objects were taken into account, like lane markings, traffic signs, other cars etc. This decision, however, often proved difficult to make because the vehicles undergo lateral and longitudinal relative displacements to which camera pose variations are added. 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 Fig. 2a. The width of the ground truth intervals obtained manually is typically 3 to 6 frames.

Accordingly, the synchronization error for a given pair (t, x_t) is

$$\mathrm{err}(t, x_t) = \begin{cases} 0 & \text{if } l_t \leq x_t \leq u_t \\ l_t - x_t & \text{if } x_t < l_t \\ x_t - u_t & \text{if } x_t > u_t \end{cases} \tag{3}$$

At first sight, the total synchronization error calculated by summing or averaging all the individual errors seems to be sensible measures of performance. However, their distribution is more informative because it tells us how much frames are at a given distance of their ground truth, for all distances. Fig. 2b shows a representative error distribution, in the sense that the error is usually low, although some outliers exists. The problem with this representation is that it is rather complex since we need then to compare histograms. A more compact graphical representation, the boxplot diagram, has been choosen. Fig. 3 shows the boxplot representation of the synchronization error for three sequence pairs. Within each plot, the result for the four DBNs are represented (from left to right: only appearance, only smoothed GPS, appearance plus raw GPS and appearance plus smoothed GPS observations). Thus, we can compare the performance of the synchronization step for the different sequences and graphical models, and assess the contribution of each type of observation.



a) b) c)

Fig. 3. Boxplot representation of the synchronization error. Outliers are only partially represented. From left to right: rural, urban at daytime and urban at night.

Method Scenario	Frame Appeance	Smoothed GPS	AppRawGPS	AppGPS
Rural (1550)	65% (1022)	84% (1302)	66% (1032)	85% (1320)
Urban daytime (1020)	44% (449)	33% (341)	46% (474)	58% (601)
Urban nightime (325)	73% (238)	58% (189)	73% (238)	79% (258)

Table 2. Percentage of frames inside the ground truth interval. In brackets, the number of frames.

In plots 3a and 3c we see that for the rural and nighttime urban pairs, the median is equal to zero and the third quartile is at most 1 for the four types of observations. In other words, there is no error at half of the frames and it is one frame or less at 75% of the sequence. Results are slightly worse on the daytime urban pair. There are two reasons for that. First, in this sequence some tall buildings close to the road degrade the GPS data, dragging the correspondence curve in the wrong direction at some places. Second, these same buildings give rise to a repetitive pattern in the video sequences which complicates the image comparison. However, the combination of image intensity (appearance) and smoothed GPS observations still achieves a small error, outperforming the three other types of observations. This happens also in the two other video pairs, suggesting that they are somehow complementary. Table 2 confirms the former observations, now with regard the number of frames correclty synchronized. This is a more restrictive condition which gives an idea (close to a lower bound) of how much frames will be correctly registered and substracted. Recall that our final goal is not the synchronization of video pairs but their pointwise substraction in order to spot differences of potential interest. As an illustration of the utility of this idea, we have addressed the problem of vehicle detection. Once a video pair has been synchronized and the corresponding frames registered, they are substracted and a video of the absolute value of the difference is built. On each frame, we perform a simple thresholding. Then, only regions larger than a certain area and with an eccentricity less than a fixed theshold are kept. Their bounding boxes are the regions of vehicle pre–detection, like in fig. 4. Still pictures are a poor representation of the results. Please visit `www.cvc.uab.es/ADAS/projects/sincro/ECCV08Workshop/` where the original, fusion, difference and pre–detection videos can be visualized.

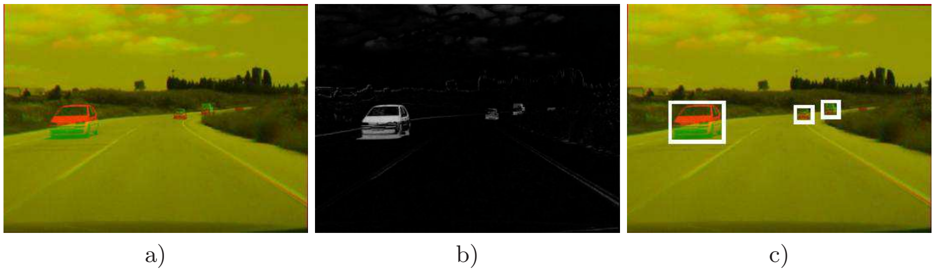


Fig. 4. Video alignment results: a) fusion, b) absolute difference, c) pre–detection of vehicles. Results in video form can be properly viewed at `www.cvc.uab.es/ADAS/projects/sincro/ECCV08Workshop/`.

5 Conclusions

In this paper, we have introduced a novel approach to the video alignment problem. Our approach relies on a graphical model formulation in order to deal with any temporal correspondence between sequences, and it combines observations from different sources. These observations, frame appearance and GPS data, have been formulated in a probabilistic framework to introduce them in the graphical model properly, i.e. the raw GPS observations. We also have proposed a novel measure of alignment and the generation of ground truth to compare properly the methods proposed. This measure indicates that the combination of frame appearance and smoothed GPS data gives the best results in the video alignment problem instead of using smoothed GPS alone. We have successfully applied it to align video sequences from moving vehicles, in order to detect the differences between these videos. The differences indicate zones where we can predetect possible objects, like vehicles.

Acknowledgments

This work has been partially funded by grants TRA2007–62526/AUT of the Spanish Education and Science Ministry, Consolider Ingenio 2010: MIPRCV (CSD2007–00018)

References

1. Lei, C., Yang, Y.: Trifocal tensor–based multiple video synchronization with subframe optimization. IEEE Trans. Image Processing **15** (2006) 2473–2480
2. Wolf, L., Zomet, A.: Wide baseline matching between unsynchronized video sequences. Int. Journal of Computer Vision **68** (2006) 43–52
3. Tuytelaars, T., Gool, L.V.: Synchronizing video sequences. cvpr **01** (2004) 762–768
4. Caspi, Y., Irani, M.: Spatio–temporal alignment of sequences. IEEE Trans. Pattern Analisys and Machine Intelligence **24** (2002) 1409–1424
5. Ukrainitz, Y., Irani, M.: Aligning sequences and actions by maximizing space–time correlations. In: European Conf. on Computer Vision. Volume 3953 of Lecture Notes in Computer Science., Springer (2006) 538–550
6. Sand, P., Teller, S.: Video matching. ACM Transactions on Graphics (Proc. SIGGRAPH) **22** (2004) 592–599
7. Rao, C., Gritai, A., Shah, M.: View-invariant alignment and matching of video sequences. In: In ICCV. (2003) 939–945
8. Ho, K.L., Newman, P.: Detecting loop closure with scene sequences. Int. J. Computer Vision **74** (2007) 261–286
9. Levin, A., Szeliski, R.: Visual odometry and map correlation. In: Proc. Computer Vision and Pattern Recognition, IEEE Computer Society (2004) 611–618
10. Grabner, H., Roth, P., Bischof, H.: Is pedestrian detection really a hard task ? In: IEEE Workshop on Performance Evaluation of Tracking and Surveillance, IEEE Computer Society (2007) to appear.
11. Langley, R.B.: Dilution of precision. GPS World **10** (1999) 52–59
12. Gelb, A., ed.: Applied Optimal Estimation. MIT Press, Cambridge, MA (1974)
13. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press (2006)
14. Zelnik-Manor, L., Irani, M.: Multi–frame estimation of planar motion. IEEE Trans. Pattern Analisys and Machine Intelligence **22** (2000) 1105–1116
15. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision **56** (2004) 221–255