Video Synchronization Using Temporal Signals from Epipolar Lines

Dmitry Pundik and Yael Moses

The Interdisciplinary Center Herzliya , Israel

Abstract. Time synchronization of video sequences in a multi-camera system is necessary for successfully analyzing the acquired visual information. Even if synchronization is established, its quality may deteriorate over time due to a variety of reasons, most notably frame dropping. Consequently, synchronization must be actively maintained. This paper presents a method for online synchronization that relies only on the video sequences. We introduce a novel definition of low level temporal signals computed from epipolar lines. The spatial matching of two such temporal signals is given by the fundamental matrix. Thus, no pixel correspondence is required, bypassing the problem of correspondence changes in the presence of motion. The synchronization is determined from registration of the temporal signals. We consider general video data with substantial movement in the scene, for which high level information may be hard to extract from each individual camera (e.g., computing trajectories in crowded scenes). Furthermore, a trivial correspondence between the sequences is not assumed to exist. The method is online and can be used to resynchronize video sequences every few seconds, with only a small delay. Experiments on indoor and outdoor sequences demonstrate the effectiveness of the method.

1 Introduction

Applications of multiple camera systems range from video surveillance of large areas such as airports or shopping centers, to videography and filmmaking. As more and more of these applications utilize the information obtained in the overlapping fields of view of the cameras, precise camera synchronization and its constant maintenance are indispensable. Given enough video time, however, synchronization will be violated because of technical imperfections that cause frame dropping or incorrect timing between sequences. The tendency to use mostly inexpensive components makes such violations a certainty in many video systems. Manual synchronization is out of the question, as it is labor-intensive and cannot be performed constantly; thus, it cannot handle arbitrary frame-dropping. Precise time synchronization via satellite, as in GPS systems, may be too expensive or limited in indoor environments. Using distributed protocols for clock synchronization methods depends on the properties of the communication network and is sensitive to communication failures. Obvious alternative sources of time information are the video streams themselves, which often provide sufficient and reliable information for automatic synchronization. In this work we address the problem of computing and maintaining the temporal synchronization between a pair of video streams with the same frame rate, relying only on the video data.

Previous work

Synchronization can be achieved using visual information by correlating spatio-temporal features or events viewed by two or more cameras. Several synchronization methods considered moving cameras viewing a static scene [10, 12] or a scene with relatively little motion [8, 6]. Our method considers static cameras acquiring a moving scene. Previous attempts to synchronize such sequences can be classified by the choice of features used for matching. The most straightforward approach is finding both spatial and temporal correspondence between point features at frames taken in all possible time shifts between the two video streams. Such approaches are vulnerable to correspondence ambiguities and require a large search space. A method for reducing the complexity of the search was suggested in [1]. Higher level features that contain temporal information also assist to reduce the matching ambiguity and the search complexity. Motion trajectories of features [9, 14, 12, 2, 6, 11] or objects [13, 3] could be used to this end. The computation of the trajectories and its quality strongly depend on the scene and can often be hard to compute as in the video considered in this paper. Since the motion of the objects may be 3D, matching the observed 2D trajectories in each sequence is ill posed. Several directions were considered for overcoming this problem, for instance, assuming a homography between two trajectories [2], or using a three-or-more camera system and 3D tensors [13, 6]. Another direction assumed an affine projection and used a linear combination approach in order to avoid exact point correspondence [14, 11]. Highly discriminative action recognition features were also proposed for synchronization [4]. Naturally, such high-level features are limited to scenes for which these actions appear and can be detected.

In an effort to avoid complex computations such as tracking and action recognition, an approach based on brightness variation over the entire image was suggested in [2]. However, this method requires spatial alignment of the sequences, and a homography transformation between the views must also be assumed. Another approach suggested using statistics over low level space-time interest points in each of the sequences [15]. This concept steers clear of computing point-to-point, trajectory, or action correspondence. However, since the statistics are computed over the entire image, the approach is strongly sensitive to the overlapping regions of the two views, relative viewing angle, and the complexity of the motion appearing in the scene. The limitations of these two approaches motivate the solution suggested in this paper.

Proposed approach

We present a method for obtaining online time synchronization of a pair of video sequences acquired by two static cameras, possibly in a wide-baseline setup. The fundamental matrix between each pair of sequences, which provides epipolar line-to-line correspondence, is assumed to be known. (For example, it can be computed directly from static corresponding features of the videos when there is no motion in the scene.) This is the only spatial correspondence required by our method. We consider sequences of general 3D scenes which contain a large number of moving objects, focusing on sequences for which features or object trajectories may be hard to compute due to occlusions and substantial movement (see Fig. 2). Furthermore, trivial correspondence (e.g., homography) between the sequences is not assumed. The temporal misalignment is considered to be only a translation, i.e., the sequences have the same frame rate. Therefore, we do not detect sub-frame time shifts, as we are correcting synchronization errors as frame-drops.

Our method is based on matching temporal signals defined on epipolar lines of each of the sequences. Hence, the spatial matching is given by the fundamental matrix. The temporal matching is performed using a probabilistic optimization framework; independent simultaneous motion occurring on different epipolar lines improve our synchronization. Failure to find such a matching (despite the observed motion in the scene) indicates that the epipolar geometry is incorrect. The temporal signal is defined as an integration of the information along an epipolar line, during a sufficient interval of time (at least 2 seconds). A simple background subtraction algorithm is used as an input to the integration. Integrating the information along epipolar lines rather than considering signals at the pixel level not only avoids the search for correspondence but allows the handling of general moving scenes. In a general scene, the correspondence between pixels at different time steps changes due to 3D motion of objects in space. Therefore, the synchronization cannot rely on corresponding pixels.

The main contribution of this paper is the use of low level temporal events along corresponding epipolar lines for video synchronization. Our method does not require high level computation such as tracking, which may be hard to compute in crowded scenes as the ones considered in our experiments. Furthermore, we bypass the need to compute point-to-point correspondences between pixels [5]. Finally, our method can be used in an *online* framework, because it detects the synchronization errors (e.g., frame drops) in a matter of seconds, as they occur in the video.

2 Method

Given a pair of color (or gray-level) sequences and a fundamental matrix, we achieve synchronization by time registration of the temporal signals from the two sequences. We first present our novel definition of temporal signals of a sequence, followed by a probabilistic approach for registering two of them. The summary of the algorithm flow is presented in Algorithms 1 and 2. The set of epipolar lines in the two images together with their correspondence are computed from the given fundamental matrix.

2.1 A temporal signal

To define the temporal signals, we make unconventional use of epipolar geometry of a pair of images. Given the fundamental matrix F for a pair of images, a set of epipolar lines $\mathcal{L} = \{\ell_r\}$ and $\mathcal{L}' = \{\ell'_r\}$ and their correspondence, $\ell_r \leftrightarrow \ell'_r$ are computed [5]. The correspondence of a given point $\hat{p} \in \ell_r$ is constrained to lie on the epipolar line $\hat{\ell}'_r = F\hat{p}$ in its synchronized frame (the points and the lines are given in homogeneous coordinates). Traditionally, this property is used for constraining the correspondence search in stereo or motion algorithms. Pixel correspondence is not guaranteed to remain the same over time due to 3D motion. However, two corresponding epipolar lines in both sequences will continue to correspond. (The only possible exception is a major occlusion on one of the views.) Using this observation, we define the signals on the entire epipolar line, avoiding not only the problem caused by the change of pixels correspondence over time but also the general challenge of computing spatial correspondence between frames.

A background subtraction algorithm is used for defining the temporal signal of each sequence. The base of the motion signal is the Euclidean distance between the data frame and the selected background frame for each pixel. For each epipolar line, a motion indicator is taken to be the sum of these distances of the line's pixels. The temporal signal of an epipolar line, the *line signal*, is defined to be the set of motion indicators on an epipolar line as a function of time. Formally, let $I(p,t)$ and $B(p,t)$ be the intensity values of a pixel $p \in \ell_r$, in some video frame and corresponding background frame[1], respectively. The *line signal* of that epipolar line, $\mathcal{S}_r(t)$, is defined to be the distance between the two vectors:

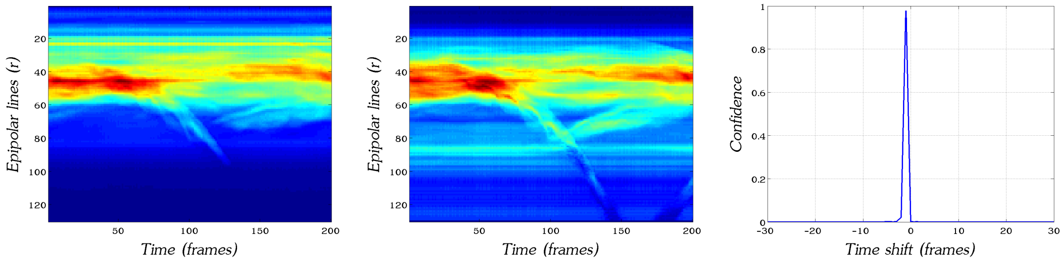
$$\mathcal{S}_r(t) = \mathit{\Sigma}_{p \in \ell_r} \|I(t,p) - B(t,p)\|. \tag{1}$$

The collection of *line signals* for all the epipolar lines in a video, is the temporal signal of the video sequence. The temporal signals of two considered sequences are represented by matrices $\mathbb{S}$ and $\mathbb{S}'$ (Fig. 1), where each row r of this matrix consists of a *line signal*, $\mathcal{S}_r$. That is, $\mathbb{S}_{r,t}$ is the motion indicator of an epipolar line ℓr at a time step t. Only a few dozen epipolar lines from each frame, a few pixels apart, are considered.

2.2 Signal Registration

In this section we present the time registration of a given pair of temporal signals of the video sequences. For robust results, and in order to combine information from different *line signals*, the matching is determined using a probabilistic framework, utilizing a maximum a posteriori estimation. The time shift is detected by finding a maximum likelihood value for the two signals, with different time shifts applied to the second signal. A sliding window in a predefined range is used to determine $\mathit{\Delta}t$.

[1] $B_r(p,t)$ is a function of t, because in the general case an adaptive background subtraction can be used.



(a) (b) (c)

Fig. 1. (a),(b) are examples of temporal signals $\mathbb{S}$ and $\mathbb{S}'$ of two sequences, containing 130 epipolar lines for a time period of 8 seconds (200 frames). Each pixel in the signal is the motion indicator of an epipolar line at a time point. (c) is the matching result for those signals with a high-confidence peak at the correct time shift of $\Delta t = -1$ frames.

Let $\mathcal{S}$ and $\mathcal{S}'$ be a pair of *line signals*, extracted from corresponding epipolar lines in two video sequences. At this stage, assume a single consistent time shift between the two sequences and no frame drops in any of them. We begin with considering the probability distribution of a time shift Δt of $\mathcal{S}'$ to match $\mathcal{S}$. Applying Bayes' law we obtain:

$$P(\Delta t \mid \mathcal{S}, \mathcal{S}') = \frac{P(\mathcal{S}, \mathcal{S}' \mid \Delta t) P(\Delta t)}{P(\mathcal{S}, \mathcal{S}')}. \tag{2}$$

The denominator term, $P(\mathcal{S}, \mathcal{S}')$, is an a priori joint probability distribution of $\mathcal{S}$ and $\mathcal{S}'$. A uniform distribution of this probability is assumed, hence it is taken to be a normalization constant factor. In general, prior knowledge about the time and space distribution of objects in the scene, or the overlapping regions of the two sequences can be used for computing this prior. Extracting such knowledge is out of the scope of this paper. The term $P(\Delta t)$ is another prior, in this case on the probability distribution of Δt. Use of this prior is discussed in the experimental part.

For estimating the likelihood term, $P(\mathcal{S}, \mathcal{S}' \mid \Delta t)$, we apply a simple stochastic model on the temporal signals. A commonly used assumption of additive white Gaussian noise is applied to the two *line signals*:

$$\mathcal{S}(t) = \mathcal{S}'(t + \Delta t) + \mathcal{N}(\mu, \sigma^2), \tag{3}$$

where Δt is the correct time shift between the two signals, and μ is the difference between the averages of both. For simplicity, in the rest of the paper we omit the value μ: from here on, each line signal $\mathcal{S}$ participates in the computation after its average was subtracted. Using this model assumption, the likelihood of two *line signals*, given Δt, is obtained by:

$$L(\mathcal{S},\mathcal{S}',\Delta t)=\frac{1}{\sigma\sqrt{2\pi}}\;e^{-\sum_t\dfrac{(\mathcal{S}(t)-\mathcal{S}'(t+\Delta t))^2}{2\sigma^2}}\;. \tag{4}$$

In reality, the relation between the signals is more complicated than this simple presentation. Differences in the cameras' photometric parameters and the foreshortening effects caused by perspective projection may result in some gain effect between these signals. Another simplification is a hidden assumption of independence between the motion indicators in a single *line signal*. Adjacent indicators are expected to be correlated to some degree, because the objects captured in the video have finite speed, relatively to the sampling frame rate. Despite these simplifications, the results are satisfying, as demonstrated in our experiments.

The maximal value of $P(\Delta t\;|\mathcal{S},\mathcal{S}')$ and the maximal value of $P(\mathcal{S},\mathcal{S}'\;|\Delta t)P(\Delta t)$ will be obtained for the same value of the desired time shift Δt:

$$\arg\max_{\Delta t}P(\Delta t\;|\mathcal{S},\mathcal{S}')=\arg\max_{\Delta t}P(\Delta t)\frac{1}{\sigma\sqrt{2\pi}}\;e^{-\sum_t\dfrac{(\mathcal{S}(t)-\mathcal{S}'(t+\Delta t))^2}{2\sigma^2}}\;. \tag{5}$$

As defined above, each row in $\mathbb{S}$ and $\mathbb{S}'$ represents a *line signal* for an epipolar line $\ell_r\in\mathcal{L}$. We consider those signals to be independent, due to the spatial distance between the selected epipolar lines. Therefore, computing the likelihood can be extended to *sequence signals* $\mathbb{S}$ and $\mathbb{S}'$ by taking the product of the likelihoods of all the *line signals*.

Up to this point, this method assumed a single consistent time shift between $\mathbb{S}$ and $\mathbb{S}'$. In order to incorporate it into an online framework, the algorithm must work on a finite time interval at each iteration. Thus, the synchronization at a given time step, t_0, is determined only from a k interval of the *sequence signal*, taken from t_0-k up to t_0 . Furthermore, the sought for Δt is bounded by some finite range $-c\leq\Delta t\leq c$. (In our experiments, k corresponds to roughly 4 to 8 seconds and c corresponds to 1 to 3 seconds). Inserting all of the above into the equations Eq. 2 and Eq. 5, we obtain:

$$\begin{aligned}\arg\max_{\Delta t}P(\Delta t\;|\mathbb{S},\mathbb{S}')&=\arg\max_{\Delta t}P(\Delta t)\prod_{\ell_r\in\hat{\mathcal{L}}(t)}P(\Delta t\;|\mathcal{S}_r,\mathcal{S}'_r) \qquad (6)\\ &=\arg\max_{-c\leq\Delta t\leq c}P(\Delta t)\;e^{-\sum\limits_{r\in\mathcal{L}(t)}\sum\limits_{t=t_0-k}^{t_0}\dfrac{\left(\mathbb{S}_{r,t}-\mathbb{S}'_{r,t+\Delta t}\right)^2}{2\sigma^2}}\end{aligned}$$

where $\hat{\mathcal{L}}\subseteq\mathcal{L}$ is the subset of epipolar lines participating in the computation (defined in 2.3), and S_r and S'_r are signals of corresponding epipolar lines ℓ_r.

The time shift Δt that yields maximal likelihood according to Eq. 6 is the correct time shift for the two given video sequences (Fig. 1(c). The actual value of the likelihood is used as a confidence level of the resulting Δt. This value is taken after a normalization step, which ensures that the probability distribution of Δt in the range $-c \leq \Delta t \leq c$ sums up to 1. The higher the probability is, the more robust the answer is. In the online synchronization framework, only the high-confidence results will be taken into account.

2.3 Epipolar line filtering

Registration of only a subset of the *line signals* is sufficient for synchronization. Moreover, line signals that contain negligible motion information may insert noise into the registration process, and are therefore removed from the computation. We next define the subset of epipolar lines $\hat{\mathcal{L}} \subseteq \mathcal{L}$, that participate in the computation for a given time step t. The signals are removed on the basis of both sequences considered. We test for motion information only at a single time step. We do so by computing the temporal gradient along an epipolar line, taking into consideration some noise estimation of such a gradient. The noise at each image pixel is assumed to be additive white Gaussian noise with some variance σ_m^2. Hence, we determine significant motion on the epipolar line r only if the residual information on the time gradient along the epipolar line goes beyond the estimated noise threshold. In case of no real motion, this time gradient yields only noise. Formally, the motion probability at a given time t, for an epipolar line ℓ_r is given by:

$$P_{motion}(\ell_r, t) = \frac{1}{\sigma_m\sqrt{2\pi}} e^{-\sum_{p\in\ell_r} \frac{(I(t,p) - I(t-1,p))^2}{2\sigma_m^2}} . \tag{7}$$

The subset $\hat{\mathcal{L}}$ consists only of epipolar lines with motion probability over some threshold. This simple filtering process compensates for the background subtraction algorithms, which are not ideal, and eliminates any wrongly detected residual motion caused by them.

Algorithm 1 Temporal signal update

```
The algorithm is triggered for every new frame acquired.
Input: two new frames from the video sequences

1. Perform background subtraction
2. For each epipolar line $\ell_r$: calculate the motion indicators (Eq. 1).
3. Update the matrices $\mathbb{S}$ and $\mathbb{S}'$.
```

Algorithm 2 Synchronization iteration

```
The algorithm is triggered every 0.8 seconds.
Input: two temporal signals $\mathbb{S}$ and $\mathbb{S}'$.

  1. Extract the data corresponding to the time interval $k$ from $\mathbb{S}$ and $\mathbb{S}'$.
  2. Compute $\hat{\mathcal{L}}$ by filtering $\ell_r \in \mathcal{L}$ for the current time step (Sec. 2.3).
  3. For each $\ell_r \in \hat{\mathcal{L}}$ subtract its average $\mu_r$.
  4. Compute the likelihood for each $-c \leq \Delta t \leq c$ using Eq. 6.
  5. Apply the prior for $P(\Delta t)$.
  6. Normalize the distribution of resulting probability such that it sums up to 1.
  7. Find the maximal value of the probability.
```

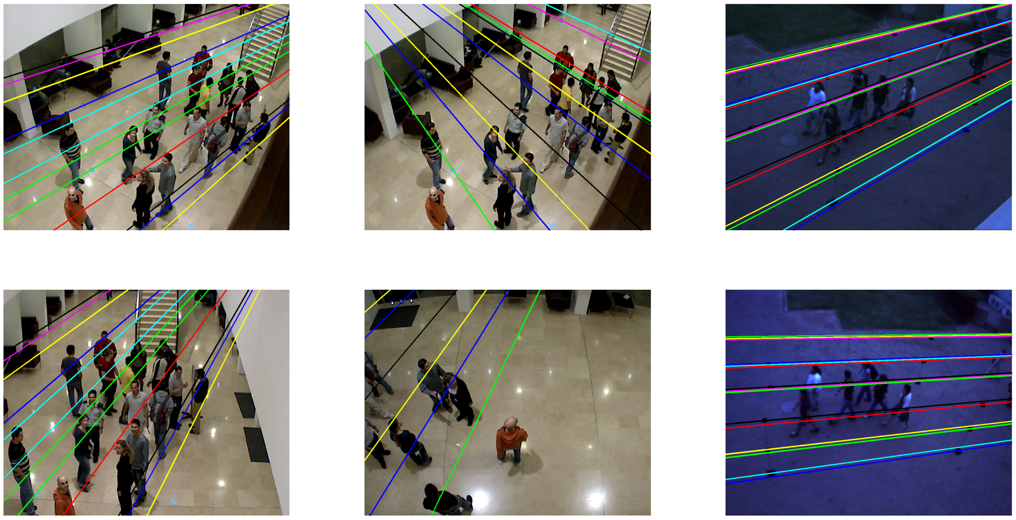
3 Experiments

We conducted a number of experiments to test the effectiveness of our method. The input for each is a pair of video sequences taken with the same frame rate. In addition, a fundamental matrix (computed manually) and a rough synchronization (up to an error of 50 frames) are assumed to be given. The method was implemented in Matlab. The corresponding epipolar lines of each pair of sequences were computed using a standard rectification method. A naive background subtraction was used where the background consists of an empty frame, subtracted from all the other frames in the video stream.

Three sequences were taken, as shown in Fig. 2. In *Set 1* an indoor scenario was acquired, in which a dense crowd – around 30 people – walk about. The cameras were placed at an elevation of approximately 6 meters. The cameras' fields of view have a relatively large overlap. The videos were recorded at 25 fps, with a frame size of 640 × 480. *Set 2* is similar to *Set 1* but the cameras' fields of view only partially overlap. This set represents a difficult case in the sense of viewing angles, since there is a big difference in the view points of the two cameras. Total runtime of both video sequences is 5000 frames (3.33 minutes). *Set 3* is of a relatively dark outdoor scene with only few people walking around. This case represents another difficult scenario, with a small amount of motion in dark conditions. A pair of cameras were located at an elevation of about 6 meters: the videos were recorded at 15 fps, with frame size of 640 × 512 pixels. In the indoor video sequences a flicker effect is evident, caused by fluorescent lighting in the scene. In order to avoid distractions to the synchronization algorithm, the flicker was removed by temporal low-pass filtering of the video. The framework triggers the synchronization computation every 0.8 seconds of the video.

3.1 Basic results

The presented tests were performed on the three sets. The interval size was taken to be $k = 140$, no prior on $P(\Delta t)$ was used (i.e., uniform distribution is assumed on $P(\Delta t)$). The value of σ for *Set 1* and *Set 2* was set to 1300, and for *Set 3* to 600. (Setting the values of σ is discussed bellow.) The results consist of a set



Set 1 *Set 2* *Set 3*

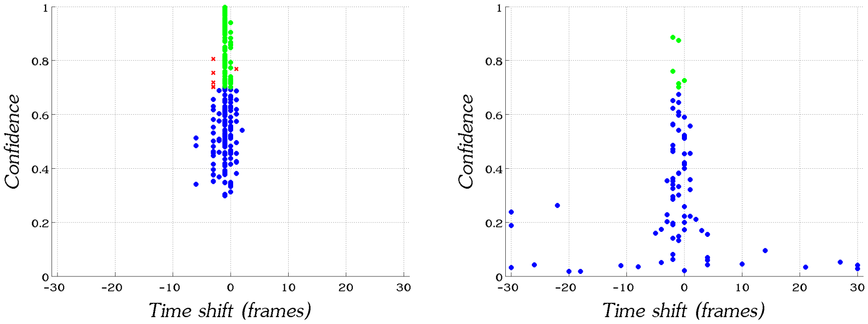
Fig. 2. Example of two frames from each video pairs. The two rows show frames from the first and second view of each pair, respectively; the images contain an exemplary subset of the used epipolar lines, each set of lines for each video pair.

of time shifts between two video streams with a probability (confidence) value for each shift. Each of the time-shifts for *Set 1*, *Set 2*, and *Set 3* are represented by a single dot in Fig. 3(a), Fig. 3(b), and Fig. 4(a), respectively. The x-axis is the computed time shift and the y-axis is the confidence in the computed result. Ideally, we would like the dots to align along the correct time shift, and to have high confidence. The correct time shift, computed by hand, is $\Delta t = -1$ frames for all sets.

To evaluate the percentage of correct results, it is necessary to set a threshold on the confidence value. The threshold 0.7 is considered in the analysis of the three data sets. A result is considered to be correct if it is in the range of ±1 frames from the correct synchronization.

Using this threshold on *Set 1*, approximately 50% of the obtained results have high levels of confidence, and 95% pecent of them are correct. That is, the system yields, on average, a high-confidence result each 1.6 seconds.

The percentage of the correct high-confidence results obtained for *Set 2* is 100%. However, only 12% of the obtained results had high confidence(> 0.7). It is mostly due to the relatively small overlapping field of view of the two cameras, resulting in a small number of epipolar lines that can participate in the registration. As the working area is small, the algorithm analyses long time periods without motion, which yield low-confidence results.



(a) (b)

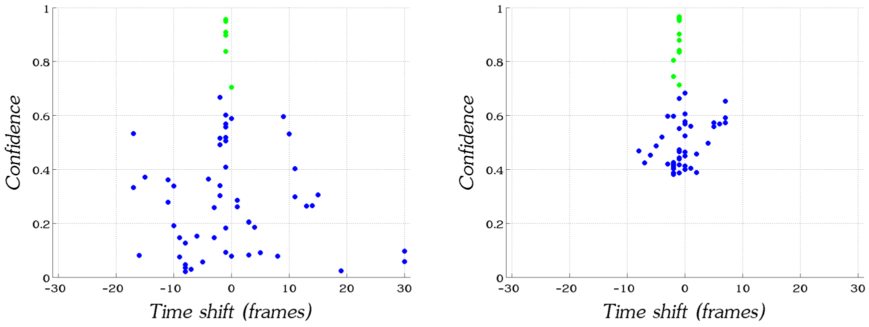
Fig. 3. Each of the 250 computed time-shifts for (a) *Set 1* and (b) *Set 2*, one for each 0.8 seconds, are represented by a single dot. Each dot in the graph represents the computed time shift for a single time step. Low confidence results are marked in blue, correct and incorrect high confidence results are marked by green and red, respectively. The x-axis is the computed time shift and the y-axis is the confidence in the computed result.

For *Set 3*, the percentage of correct results is 100% with only 13% of the results having high confidence. In addition, the low confidence results consist of a relatively large amount of errors. This is due to the small number of moving objects in the scene and objects moving along the direction of epipolar lines. Note that a movement along an epipolar line is not expected to produce good synchronization, since it induces ambiguities, as discussed in Sec. 4. The effect of a non-uniform prior on $P(\Delta t)$ when incorporated into this set is discussed in Sec. 3.3.

To summarize, our method constantly and reliably maintains the time synchronization between the two sequences. It is important to note that tracking objects or features in the crowded scene of *Set 1* and *Set 2* from a single camera is considered to be an extremely difficult task due to substantial movement and a large number of occlusions. Hence, synchronization studies that rely on trajectories detected by each of the cameras (e.g., [13, 3]) are not adequate in this case. Furthermore, the scene consists of a genuine 3D structure and the distance between the cameras is non-negligible. Hence, a homography transformation of the pair of sequences cannot be used to match pixels or trajectories (as in [2]).

3.2 Frame dropping

Frame dropping is expected in a simple commercial system when it operates over a long period of time. The need to detect frame dropping and resynchronize is one of the main motivations for an online synchronization algorithm. To test the robustness of our method in the presence of frame dropping, we applied our algorithm to *Set 1* where 3 frame drops occurred during the video. That is, the correct time shift changed from -1 to 16, then to -8 and finally, back to -1.



(a) (b)

Fig. 4. Each of the 60 computed time-shifts, one for each 0.8 seconds, are represented by a single dot computed fot *Set 3*. (a) without prior and (b) with a prior. The axes description and color codes interpretation are as in Fig. 3

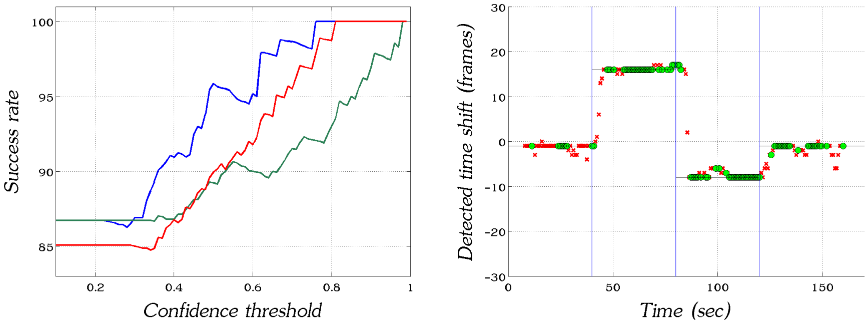
The rest of the experiment setup was identical to the basic one. The results are presented in Fig. 5(b), where the detected time shift is plotted as a function of time. The result demonstrates that the correct time shift is detected, and the reaction time to the drop is approximately 7-8 seconds. This reaction time is due to the interval of 140 frames, which, in addition to the search range $c = 30$, corresponds to 8 seconds. During this time period the two registered temporal signals contain inconsistent information with a frame drop in it. Hence, the results are incorrect and have low confidence.

3.3 Using a prior on $P(\Delta t)$

In an online framework, a non-uniform probability distribution on Δt can be applied, using the result of the previous synchronization iteration. It is assumed that the time synchronization rarely changes during the video, and the changes are of a few frames only (due to frame dropping). We tested our method using a Gaussian distribution of $P(\Delta t)$ with $\sigma = 2$ and a mean set to the previously detected high-confidence time shift (starting with 0) . Comparing the results with (Fig. 4(a)) and without (Fig. 4(b)) use of the prior, shows that the prior reduces the instability of the low-confidence results. We tested the effect of using a prior on *Set 1* (with and without frame dropping) and on *Set 2*. In all these tests the results remain the same. Hence we can conclude that on the one hand the prior can reduce errors for unstable results, and on the other hand it does not impair other results.

3.4 Setting the parameters

In addition to the confidence threshold, there are two more parameters that have to be set. The time interval k controls the number of frames that participate



(a) (b)

Fig. 5. (a) A graph showing the success rate as a function of the confidence threshold for three different values of σ (see Eq. 6). The blue, red and green lines represent $\sigma = 1600$, 1300, and 1000, respectively. (b) Frame dropping example, drop reaction time = 7 seconds. The green and the red dots represent high and low confidence, respectively. The vertical blue lines indicate the time at which the frame drop occurred. The black line is the correct time shift.

in the signal registration procedure. Longer intervals will lead to more robust results, especially for areas and times with limited motion. According to our tests, in a video pair with a lot of motion, an interval of $k = 20$ frames (0.8 seconds) is sufficient for robust synchronization results. However, for limited and sporadic motion, such an interval yields a somewhat noisy output, therefore $k = 140$ frames was used in all our experiments. The downside of large intervals is the increase in computation time and the slower reaction time in the presence of frame drops. The reaction time to such changes can, in the worst case, be as long as the interval time, as discussed in Sec. 3.2.

The other parameter is the σ in Eq. 3-6. This value serves as a normalization factor in the probability calculations. In general, it depends on photometric parameters of the used cameras, as well as on their joint epipolar geometry. In the experiments, the value of σ was set empirically. This factor affects the numerical outcome of the confidence for each time shift, as demonstrated in Fig. 5(a). High values of σ suppress the confidence, hence flatten the probability distribution of $P(\Delta t \mid \mathcal{S}, \mathcal{S}')$, causing indecisiveness and noisy output. However, lower values of σ increase the confidence of all the measurements, and as a result, the confidence of incorrect time shifts increases as well. Thus, in order the preserve the correct output of the framework, the final confidence threshold must be selected in accordance to the value of σ.

3.5 Verification of Calibration

The main goal of our method was to compute synchronization between a pair of sequences, while the camera calibration (i.e., the epipolar geometry) is assumed to be given to the system. Incorrect epipolar geometry causes motion indicators on corresponding epipolar lines to be uncorrelated. In particular, the confidence of all the possible synchronization results is expected to be low. An experiment for demonstrating this observation was conducted, simulating a scenario of a small tilt in one of the cameras. The tilt causes calibration failure, as it breaks the correspondence of the epipolar lines. This leads to a total synchronization failure. Consequently, it is impossible to use our method when the system is out of calibration. Yet, this property of our method can be used to verify calibration, i.e., to distinguish between correct and incorrect calibration of the cameras. Although it cannot be used in a straightforward manner for camera calibration, because the search space for a fundamental matrix is too large, it does serve as an essential first step towards recalibration, following calibration failure.

3.6 Additional tests

We discussed in the introduction and the method sections why we choose to use epipolar lines signals rather than point signals. Here we challenge our choice to use epipolar line signals rather than a similar signal defined by a motion indicator based on the entire frame (similar to the approach taken by [15]). When the temporal signal is defined on the entire frame, any spatial correspondence between motion indicators is neglected. We modified our method to sum the motion indicators on the entire frame in order to obtain the motion signal. As expected, the obtained result cannot be used for sequence synchronization. Such an approach fails in the presence of complex motion in the scene.

To verify that our method works properly on other video sequences used in literature, we have performed the synchronization of a pair of short videos used in [2]. The sequences contain a single car moving in a parking lot, and are taken from http://www.wisdom.weizmann.ac.il/~vision/VideoAnalysis/Demos/Seq2Seq/Seq2Seq.html. The success rate of our method on this sequence is 100% with the parameters: confidence threshold of 0.6, $\sigma = 400$ and $k = 80$.

4 Conclusion

We presented a novel method for synchronizing a pair of sequences using only motion signals of corresponding epipolar lines. Our method is suitable for detecting and correcting frame dropping. Its simplicity is in bypassing the computation of spatial correspondence between features, tracked trajectories or image points, which may be hard to compute in the scenes considered in our experiments. The only spatial correspondence required is between epipolar lines, which are computed directly from the given fundamental matrix of the image pairs. The relatively low computational effort will enable our algorithm to be incorporated into real-time systems, after a short optimization cycle. Furthermore, it can detect the synchronization errors (e.g., frame drops) in a matter of seconds, as they occur in the video. Thus, it can be used in an online framework. Finally, the method can be used for detecting calibration failures, as a first step in recalibration.

It is worth noting that our method may fail in rare cases such as an object moving strictly along an epipolar line and no other information is available. In this case the temporal matching is expected to yield the same probability for all time shifts. In addition, if the object also moves across a non-overlapping regions of the cameras, an incorrect synchronization is expected. In order to overcome such problems, a method for detecting overlapping regions of cameras can be utilized, e.g. [7]. Additionally, this problem can be resolved when working in a system with more than two cameras, by using other pairs of sequences with different sets of epipolar lines. We intend to study the extension of the proposed approach to handle more than two sequences. This extension should be natural due to the probabilistic properties of the algorithm.

References

1. R. Carceroni, F. Padua, G. Santos, and K. Kutulakos. Linear sequence-to-sequence alignment. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, volume 1, 2004.
2. Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *IEEE Trans. Patt. Anal. Mach. Intell.*, pages 1409–1424, 2002.
3. Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *Int. J. of Comp. Vision*, 68(1):53–64, 2006.
4. E. Dexter, P. Pe'rez, and I. Laptev. Multi-view synchronization of human actions and dynamic scenes. In *Proc. British Machine Vision Conference*, 2009.
5. R. Hartley and A. Zisserman. *Multiple view geometry.* Cambridge university press, 2000.
6. C. Lei and Y. Yang. Tri-focal tensor-based multiple video synchronization with subframe optimization. *IEEE Transactions on Image Processing*, 15(9), 2006.
7. Z. Mandel, I. Shimshoni, and D. Keren. Multi-camera topology recovery from coherent motion. In *ACM/IEEE International Conference on Distributed Smart Cameras*, pages 243–250, 2007.
8. J. Serrat, F. Diego, F. Lumbreras, and J.M. Álvarez. Synchronization of Video Sequences from Free-Moving Cameras. In *Proc. Iberian Conf. on Pattern Recognition and Image Analysis*, page 627, 2007.
9. M. Singh, A. Basu, and M. Mandal. Event dynamics based temporal registration. *IEEE Transactions on Multimedia*, 9(5), 2007.
10. L. Spencer and M. Shah. Temporal synchronization from camera motion. In *Proc. Asian Conf. Comp. Vision*, 2004.
11. P. Tresadern and I. Reid. Synchronizing image sequences of non-rigid objects. In *Proc. British Machine Vision Conference*, volume 2, pages 629–638, 2003.
12. T. Tuytelaars and L. Van Gool. Synchronizing video sequences. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, volume 1, 2004.
13. A. Whitehead, R. Laganiere, and P. Bose. Temporal synchronization of video sequences in theory and in practice. In *IEEE Workshop on Motion and Video Computing*, 2005.
14. L. Wolf and A. Zomet. Correspondence-free synchronization and reconstruction in a non-rigid scene. In *Proc. Workshop Vision and Modeling of Dynamic Scenes*, 2002.
15. J. Yan and M. Pollefeys. Video synchronization via space-time interest point distribution. In *Proc. Advanced Concepts for Intelligent Vision Systems*, 2004.