Synchronizing Video Cameras with Non-Overlapping Fields of View

Darlan N. Brito, Flávio L. C. Pádua
Mestrado em Modelagem Matemática e Computacional
Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675, Belo Horizonte, MG, Brasil
{darlan,cardeal}@lsi.cefetmg.br

Rodrigo L. Carceroni
Depart. de Ciência da Computação
Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, BH, MG, Brasil
carceron@dcc.ufmg.br

Guilherme A. S. Pereira
Depart. de Engenharia Elétrica
Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, BH, MG, Brasil
gpereira@cpdee.ufmg.br

Abstract

This paper describes a method to estimate the temporal alignment between N unsynchronized video sequences captured by cameras with non-overlapping fields of view. The sequences are recorded by stationary video cameras, with fixed intrinsic and extrinsic parameters. The proposed approach reduces the problem of synchronizing N non-overlapping sequences to the robust estimation of a single line in $\mathbb{R}^{N+1}$. This line captures all temporal relations between the sequences and a moving sensor in the scene, whose locations in the world coordinate system may be estimated at a constant sampling rate. Experimental results with real-world sequences show that our method can accurately align the videos.

1. Introduction

In this work we consider the problem of temporal synchronization (temporal alignment) of multiple video sequences, captured from distinct viewpoints by cameras with non-overlapping fields of view. Normally, the temporal misalignment between video sequences occurs when the input sequences have different frame rates, or when there is a time shift between the sequences (e.g. when the cameras are not activated simultaneously). Examples of applications where video synchronization is essential include three-dimensional photogrammetric analysis [16], periodic motion detection and segmentation [12], multi-sensor alignment for image fusion [4] and video metrology from television broadcasts [18].

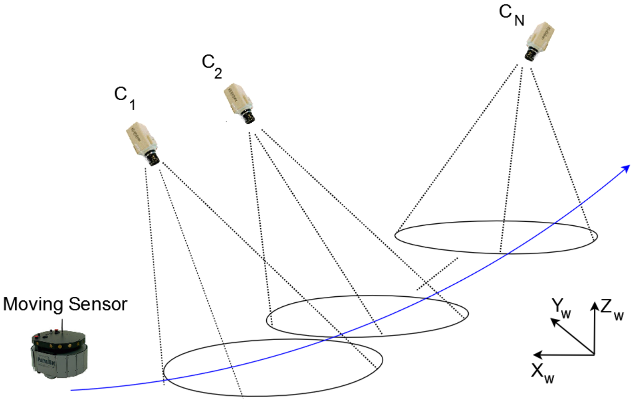


Figure 1. A 3D scene is viewed simultaneously by N stationary cameras located at distinct viewpoints, whose fields of view do not necessarily overlap. A moving sensor crosses the fields of view of all cameras.

Unfortunately, even though synchronization can be performed in hardware, for example, by embedding a timestamp in the video stream or sending a synchronization signal to cameras [11], this can be costly and must be set up prior to recording. Alternatively, software algorithms can be used to recover synchronization from visual cues and this strategy is used in the present work. A reliable algorithm for the solution of the asynchronism problem between multiple video sequences should be able to handle cases like [2]:

- unknown frame rates of the cameras;
- arbitrary time shift between the sequences;

- arbitrary object motion and speed;
- presence of tracking failures, that is, individual scene points cannot be tracked reliably over many frames;
- computational efficiency should degrade gracefully with increasing number of video sequences;
- unknown user-defined camera set-up;

These requirements have directed us during the development of our approach, which operates under all of the above conditions except the last one. In particular, we assume that the camera set-up is composed by stationary cameras, whose intrinsic and extrinsic parameters are known *a-priori*.

Our method is derived from the method presented in [2] and is based on the concept of a *timeline*. Consider the scenario illustrated in Figure 1. Given N non-overlapping sequences, the timeline is a line in $\Re^{N+1}$ that completely describes all temporal relations between the sequences and a moving sensor in the viewed scene. A key property of the timeline is that even though knowledge of the timeline implies knowledge of the sequences' temporal alignment, we can compute points on the timeline without knowing this alignment [2].

Using this property as a starting point, the temporal alignment problem for N sequences is reduced to the problem of estimating a single line of $N + 1$ dimensions from a set of appropriately-generated points in $\Re^{N+1}$.

Video alignment algorithms can be divided in two main groups: the *feature–based methods* and the *direct methods*. Feature–based methods [1, 5, 12–17, 23–27] extract all information needed to perform temporal alignment from detected features, for example, frame–to–frame object motion, or object trajectories throughout an entire sequence. On the other hand, direct methods [3, 4, 6, 7, 19, 21, 22] extract that information from the intensities and intensity gradients of all pixels that belong to overlapping regions.

Therefore, direct methods tend to align sequences more accurately if their appearances are similar, while feature–based methods are widely prescribed for sequences with dissimilar appearance such as those acquired with wide baselines, different magnifications, or by cameras with distinct spectral sensitivities. Our approach belongs to group of feature–based methods.

Most existing feature-based techniques [5, 12, 14, 17, 23–28] were developed for overlapping video sequences. Moreover, most of these works conduct an explicit search in the space of all possible alignments and are aware of use constraints based on correspondences between points of object trajectories. The combinatorial nature on that search requires several additional assumptions to make it manageable [2]. These include assuming known frame rates; restricting N to be two; assuming that the temporal misalignment is an integer; and assuming that this misalignment falls within a small user-specified range (typically less than fifty frames). Hence, efficiency considerations greatly limit the applicability of these solutions [2].

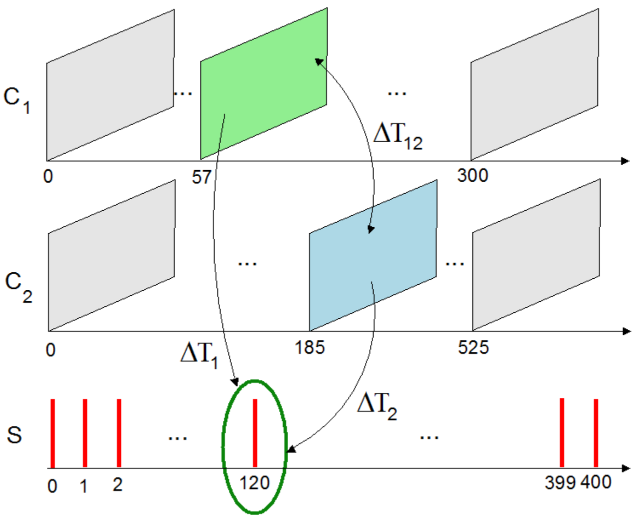


Figure 2. Illustration of the temporal misalignment between a moving sensor s and two cameras c_1 and c_2. The location sample 120 of the moving sensor s correspond in time to the frames 57 and 185 of cameras c_1 and c_2, respectively. In this case, we have $\Delta T_1 = 63$, $\Delta T_2 = 65$ and $\Delta T_{12} = 128$.

Unlike these techniques, the proposed approach aligns N non-overlapping video sequences, can handle arbitrarily-large misalignments, and does not require any *a priori* information about their temporal relations.

A few works have also proposed feature-based methods for temporally aligning a general number of video sequences [1, 16]. Raguse and Heipke [16] propose a method where the temporal misalignment is modelled by a 2^{nd} order polynomial and is converted to an interpolation factor in image space. Through the use of the interpolation factor, temporal correction terms for the image coordinates are calculated and introduced in the functional model of a bundle adjustment. Unlike the method proposed in this paper, the technique developed by Raguse and Heipke works with overlapping sequences, requires a reliable tracker and if the acquisition network consists only of two cameras, it is necessary that the object motion does not occur in an epipolar plane, because otherwise the temporal misalignment results in a systematic point shift in that plane since the two image rays still intersect.

Anthony et al. [1] present a method that uses a two stage approach that first approximates the synchronization by tracking moving objects and identifying inflection points.

Their method is closely related to the technique proposed by Carceroni et al. [2] and proceeds to refine the estimate using a consensus based matching heuristic to find moving features that best agree with the pre-computed camera geometries from stationary image features. However, unlike our approach, it was developed to work with overlapping sequences, requires the presence of at least three cameras monitoring the scene and the use of a reliable tracker.

Finally, there are only a few works based on direct methods to align sequences without any overlap [4,19]. The most relevant work was developed by Caspi and Irani [4], and, unlike our approach, it does not work with stationary cameras. Specifically, it only works with sequences acquired by pairs of cameras that remain rigidly attached to each other while moving relative to a mostly rigid scene.

2. Problem Formulation

Suppose that a dynamic scene is viewed simultaneously by N stationary cameras located at distinct viewpoints, whose fields of view do not necessarily overlap. Moreover, consider the presence of a moving sensor in the 3D scene, whose locations in the world coordinate system may be estimated with a constant sampling rate. Suppose also that this sensor crosses the fields of view of all cameras, as illustrated in Figure 1.

We assume that each camera captures frames with a constant, unknown frame rate and that the cameras as well as the moving sensor are unsynchronized, i.e., they began capturing frames and location samples at a different time with possibly-distinct sampling rates. In Figure 2, for example, we illustrate the temporal misalignment between a moving sensor and two cameras. In that example, the location sample 120 of the moving sensor s correspond in time to the frames 57 and 185 of cameras c_1 and c_2, respectively. Therefore, the temporal misalignment between camera c_1 and sensor s is $\Delta T_1 = 63$, while the temporal misalignment between camera c_2 and sensor s is $\Delta T_2 = 65$. Analogously, the temporal misalignment between the cameras is $\Delta T_{12} = 128$.

The constant sampling rate assumption for the video cameras and the moving sensor implies that the temporal coordinates (time stamps) of the sensor samples and the temporal coordinates (frame numbers) of all video sequences are related by a onedimensional affine transformation [2]:

$$t_i = \alpha_i \, t_s \, + \, \beta_i, \qquad (1)$$

where t_i and t_s denote the temporal coordinates of the i-th video sequence and the temporal coordinates of the moving sensor, respectively. The parameters α_i, β_i are unkown constants describing the temporal dilation and temporal shift, respectively, between the sensor and the i-th sequence. In general, these constants will not be integers [2].

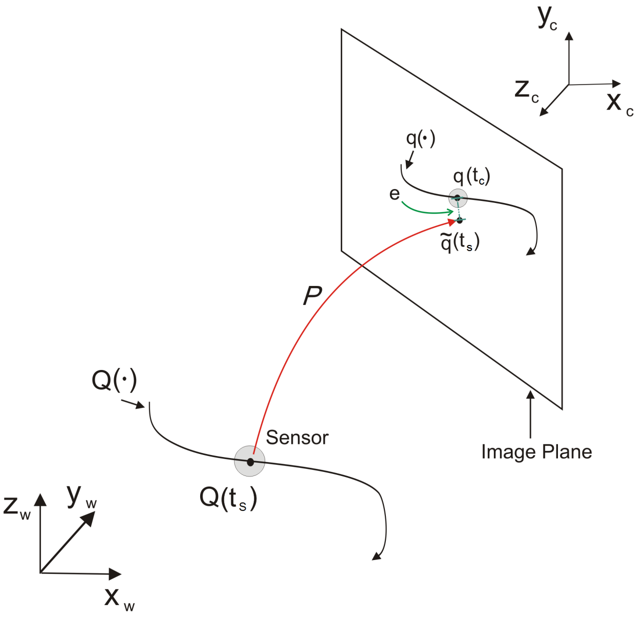


Figure 3. A sensor moves along a trajectory $\mathbf{Q}(\cdot)$ in a 3D scene, viewed by a camera. Let $\mathbf{q}(\cdot)$ be the trajectory traced by the sensor's projection in the image plane, computed by a tracking algorithm. Consider that $\mathbf{q}(t_c)$ represents the sensor's instantaneous position in the image plane at frame t_c and $\mathbf{Q}(t_s)$ represents the 3D sensor's instantaneous position at the temporal coordinate t_s, whose projection in the image plane, computed by using the projection matrix P, is given by $\tilde{\mathbf{q}}(t_s)$. If $\mathbf{q}(t_c)$ and $\mathbf{Q}(t_s)$ correspond in time, the vector $[t_c \;\; t_s]$ retrieves the temporal alignment between the sensor and the camera.

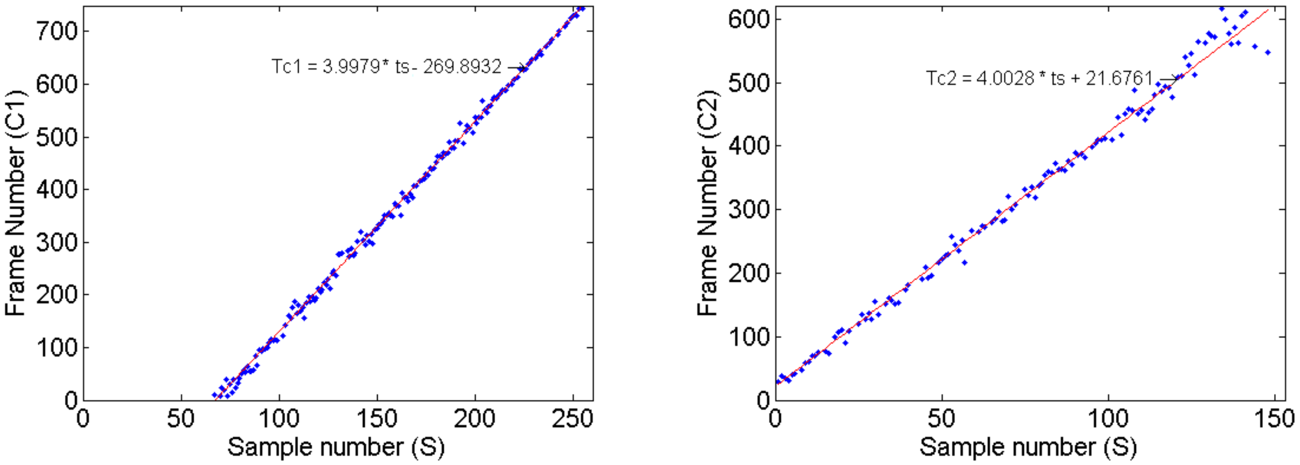
The pairwise temporal relations captured by Equation (1) induce a global relationship between the frame numbers of the sequences and the sample numbers of the moving sensor. We represent this relationship by a line $\mathcal{L}$ of $N+1$ dimensions, that we call the *timeline*:

$$\mathcal{L} = \left\{ \left[\alpha_1 \;\; \dots \;\; \alpha_{n+1}\right]^\top t + \left[\beta_1 \;\; \dots \;\; \beta_{n+1}\right]^\top \;\middle|\; t \in \Re \right\}. \qquad (2)$$

Observe that the timeline captures all temporal relations between the video sequences. Therefore, the problem addressed in this work consists in to obtain an accurate estimate for such a line.

3. Temporal Synchronization Algorithm

Even though knowledge of $\mathcal{L}$ implies knowledge of the temporal alignment of the sequences, we can com-



(a)

(b)

Figure 4. (a) *Voting Space* for the camera c_1 and the moving sensor s. (b) *Voting Space* for the camera c_2 and the moving sensor s. Each point in (a) and (b) represents candidate temporal alignments. The reconstructed lines $t_{c_1} = 3.9979 t_s - 269.8932$ and $t_{c_2} = 4.0028 t_s + 21.6761$ describe the temporal alignments between the sensor and cameras c_1 and c_2, respectively. From those equations we obtain the new equation $t_{c_2} = 1.0011 t_{c_1} + 291.8797$ that retrieves the temporal alignment between the two video sequences.

pute points on the timeline without knowing the sequences' alignment [2]. This observation leads to a simple algorithm for reconstructing the timeline by using a moving sensor in the scene that crosses the fields of view of all cameras.

Specifically, consider Figure 3 where a sensor moves along a trajectory $\mathbf{Q}(\cdot)$ in a 3D scene, viewed by a camera. Suppose that the 3D sensor's trajectory in the world coordinate system may be estimated by using a localization system as the one proposed by Garcia et al. [9]. Let $\mathbf{q}(\cdot)$ be the trajectory traced by the sensor's projection in the image plane, computed by an object tracking algorithm [10].

Assuming that the camera is calibrated, we may estimate for each 3D sensor's position its corresponding projection in the image plane. In Figure 3, for example, $\mathbf{Q}(t_s)$ represents the 3D sensor's instantaneous position at the temporal coordinate t_s and its projection $\tilde{\mathbf{q}}(t_s)$ in the image plane was computed by using the projection matrix P, obtained during the calibration of the camera.

Our key observation is that, by determining correspondences between 2D sensor positions in the image plane, computed by the tracking algorithm and by the projection matrix P, we may also determine correspondences between the temporal coordinates of the frame numbers of the video sequence and the sample numbers of the moving sensor.

Consider, for example, that $\mathbf{q}(t_c)$ in Figure 3 represents the sensor's instantaneous position in the image plane at frame t_c, computed by the tracker. Assuming that $\mathbf{q}(t_c)$ and $\mathbf{Q}(t_s)$ correspond in time, the projection $\tilde{\mathbf{q}}(t_s)$ of $\mathbf{Q}(t_s)$ should coincide with $\mathbf{q}(t_c)$ or stay at a distance of e pixels, due to errors in the camera calibration and tracking algorithms. From this observation, we may also establish correspondence between the temporal coordinates t_c and t_s of $\mathbf{q}(t_c)$ and $\tilde{\mathbf{q}}(t_s)$, respectively, since they represent the same 3D instantaneous position $\mathbf{Q}(t_s)$ of the sensor. In fact, we may estimate for each camera c and the moving sensor s a set $\mathcal{V}$ of 2D points with coordinates $[t_c \;\; t_s]$ that represent "candidate" temporal alignments for the camera and the sensor. Specifically, the set $\mathcal{V}$ defines a *voting space* that is built as follows:

$$\mathcal{V} = \left\{ [t_c \;\; t_s]^\top \;\; | \;\; D\left(\mathbf{q}(t_c), \tilde{\mathbf{q}}(t_s)\right) \leq \varepsilon, \right\}, \tag{3}$$

where $D(\cdot)$ denotes the euclidean distance between the points $\mathbf{q}(t_c)$ and $\tilde{\mathbf{q}}(t_s)$, and ε denotes a tolerance in pixels, whose value is given by the average of the errors in the camera calibration and tracking algorithms.

In Figures 4(a)-(b), we illustrate two examples of *voting spaces* obtained in the real experiment described in the next section. In general, the set $\mathcal{V}$ described in Equation (3) will contain outliers. To reconstruct the timeline in the presence of outliers, we use the RANSAC algorithm [8]. RANSAC can be regarded as an algorithm for robust fitting of models in the presence of many data outliers. Since it gives us the opportunity to evaluate any estimate of a set of parameters no matter how accurate the method that generated this solution might be, the RANSAC method represents an interesting approach to the solution of many computer vision problems [2].

The algorithm randomly chooses a pair of candidate temporal alignments to define the timeline, and then computes the total number of candidates that fall within an δ-distance of this line. These two steps are repeated for a number of iterations. Provided sufficient repetitions are performed, RANSAC is expected to identify solutions computed from outlier-free data. Therefore, the two critical parameters of the algorithm are the number k of RANSAC iterations and the distance δ. To determine k, we use the formula

$$k = \left\lceil \frac{\log(1-p)}{\log(1-r^2)} \right\rceil, \qquad (4)$$

where p is the probability that at least one of our random selections is an error-free set of candidates and r is the probability that a randomly-selected candidate is an inlier.

Equation (4) expresses the fact that k should be large enough to ensure that, with probability p, at least one randomly-selected pair of candidates is an inlier. We used $p = 0.99$ and $r = 0.05$ ($k = 1840$ iterations) for our experiments, which are conservative values that lead to accurate results in our experiments. To compute the distance δ, we observe that δ can be thought of as a bound on the distance between tracked sensor locations in the input cameras and their associated projections.

After the use of RANSAC, the last step consists in to apply the least-squares method over the data set estimated to compute the timeline parameters. By combining the computed equations $t_i = \alpha_i t_s + \beta_i$ with parameters α_i and β_i, $i = 1, ..., N$, we may obtain new equations that capture the temporal relation between any two arbitrary sequences i and j, as well as the line $\mathcal{L}$ that captures the global relationship between the sequences.

4. Experimental Results

To demonstrate the applicability of our algorithm, we present experimental results with real-world sequences. Specifically, we tested our approach on a two-view dataset of an indoor scene. Image dimensions in both datasets were about 720×480 pixels. The data were acquired by two cameras Sony DCR-SR62 without significant overlap between their fields of view and that worked with identical frame rate of 30fps, implying a unit ground-truth temporal dilation ($\alpha = 1$). The ground-truth temporal shift between the video sequences was $\beta = 292 \pm 0.5$ frames. The values of the main parameters used in our temporal alignment algorithm are listed in Table 4.

The moving sensor that crossed the fields of view of both cameras was a robot Pioneer 2 AT, produced by *Active Media*. The 3D localization data of the sensor were estimated at a rate of 7.5 samples per second by using the visual localization system proposed by Garcia et al. [9]. The frames in the resulting video sequences contain a single rigid object (sensor) moving over a static background, along a fairly smooth trajectory, as illustrated in Figures 5(a)-(b). We used the WSL tracker [10] to track the sensor (blue trajectories in Figures 5(a)-(b)). WSL was initialized manually in the first frame of each sequence.

Parameters	Meaning	Values
ε	Tolerance used during the construction of the voting space	10
p	RANSAC parameter: probability that at least one of our random selections is an error-free set of candidates	$0,99$
r	RANSAC parameter: probability that a randomly-selected candidate is an inlier	$0,05$
δ	RANSAC parameter: tolerance for the distance between a candidate temporal alignment and the timeline	$0,5$

Table 1. Values of the main parameters of our temporal alignment algorithm.

The cameras were calibrated according to the algorithm implemented by Strobl et al. [20]. In Figures 5(a)-(b) we show the projections of the 3D sensor locations in the image planes of both cameras (red trajectories). Observe that those projections defined very noisy trajectories in the image planes.

We use the average temporal alignment error as our basic measurement for evaluating the accuracy of our approach. Specifically, its value is given by the average of the absolute values of the differences between the temporal coordinates computed by the estimated line and the temporal coordinates computed by the “ground-truth” affine transformation in Equation (5):

$$t^g_{c_2} = t_{c_1} + 292, \qquad (5)$$

where t_{c_1} represents the temporal coordinate of the sequence acquired by camera c_1 and $t^g_{c_2}$ represents its corresponding temporal coordinate in the sequence acquired by camera c_2, computed by the "ground-truth" affine transformation.

Therefore, if $t^e_{c_2}$ represents the corresponding temporal coordinate of t_{c_1}, computed by using the line estimated by our method, the average temporal alignment error ε_t is given by:

$$\varepsilon_t = \frac{1}{M} \sum_{t_{c_1}=0}^{M-1} \left| t^e_{c_2}(t_{c_1}) - t^g_{c_2}(t_{c_1}) \right| . \quad (6)$$

where M is the number of frames in the video sequence acquired by camera c_1 (in this case, $M = 756$).

In Figures 4(a)-(b), we show the estimated *voting spaces* for the moving sensor s and the two cameras c_1 and c_2 used in our experiment. The reconstructed lines $t_{c_1} = 3.9979t_s - 269.8932$ and $t_{c_2} = 4.0028t_s + 21.6761$ describe the temporal alignments between the sensor and cameras c_1 and c_2, respectively. From those equations we obtain the new equation $t^e_{c_2} = 1.0011t_{c_1} + 291.8797$ that retrieves the temporal alignment between the two video sequences. According to Equation (6), the reconstructed line gives an average temporal alignment error of 0.9764 frames or 32.5 miliseconds.

Therefore, our results show that our method may work successfully even when the video sequences have large temporal misalignments (in this example, 292 frames). This scenario may be critical for most of the current temporal alignment methodologies. Figures 5(c)-(d) confirm that the computed temporal alignment between the video sequences was effectively retrieved. In Figure 5(c), the *before alignment image* was created by superimposing the green band of a frame t_{c_2} with the red and blue bands of frame $t_{c_1} = (t_{c_2} - \beta^g)/\alpha^g$, using ground truth timeline coefficients α^g and β^g. Observe the temporal misalignment between the video sequences. In Figure 5(d), the *after alignment image* was created by replacing the green band of frame t_{c_2} with that of frame $t_{c_1} = (t_{c_2} - \beta^e)/\alpha^e$, with α^e, β^e computed by our algorithm. Note that the sequences were aligned quite well and the "double exposure" artifacts disappeared.

5. Conclusions

This paper presents an approach to estimate the temporal alignment between N unsynchronized video sequences captured by cameras with non-overlapping fields of view. The results suggest that timeline reconstruction provides a simple and effective method for temporally aligning multiple video sequences that do not have spatial overlap.

Additional theoretical investigations need to be considered for future work. Firstly, the methodology proposed assumes that all cameras acquire frames at constant (albeit not necessarily identical) temporal sampling rates. Based on that assumption, the approach model the temporal misalignment between a pair of video sequences as an one-dimensional affine transformation. The pairwise temporal relations modelled by that transformation induce a global relationship between the frame numbers of the input sequences and the sample numbers of the moving sensor. However, such a kind of mathematical modelling is not appropriate when some sequences work with variable frame rates. Therefore, the development of an alternative mathematical model, which can couple with this problem represents an important topic for future research.

Another important direction for future work is 3D scene reconstruction. By combining the temporal alignment approach with multi-view stereo techniques, important advances could be achieved in the development of robust systems for reconstructing 3D dynamic scenes.

We are currently investigating the problem of estimating the temporal synchronization in wireless sensor networks, by adapting the methodology proposed by Carceroni et al [2]. Unlike existing methods, which are frequently based on adaptations of techniques originally designed for wired networks with static topologies, or even based on solutions specially designed for *ad hoc* wireless sensor networks, but that have a high energy consumption and a low scalability regarding the number of sensors, we are developing an approach that reduces the problem of synchronizing a general number *N* of sensors to the robust estimation of a single line in R^{N+1}. In this new scenario, we consider that the moving sensors are distributed in an environment that is viewed by one or more cameras.

Acknowledgments We would like to thank Pedro Shiroma and Vilar da Camara Neto for their support with the experiments. Flávio Pádua and Darlan Brito thank the support of FAPEMIG-Brazil under Proc. EDT- 162/07.

References

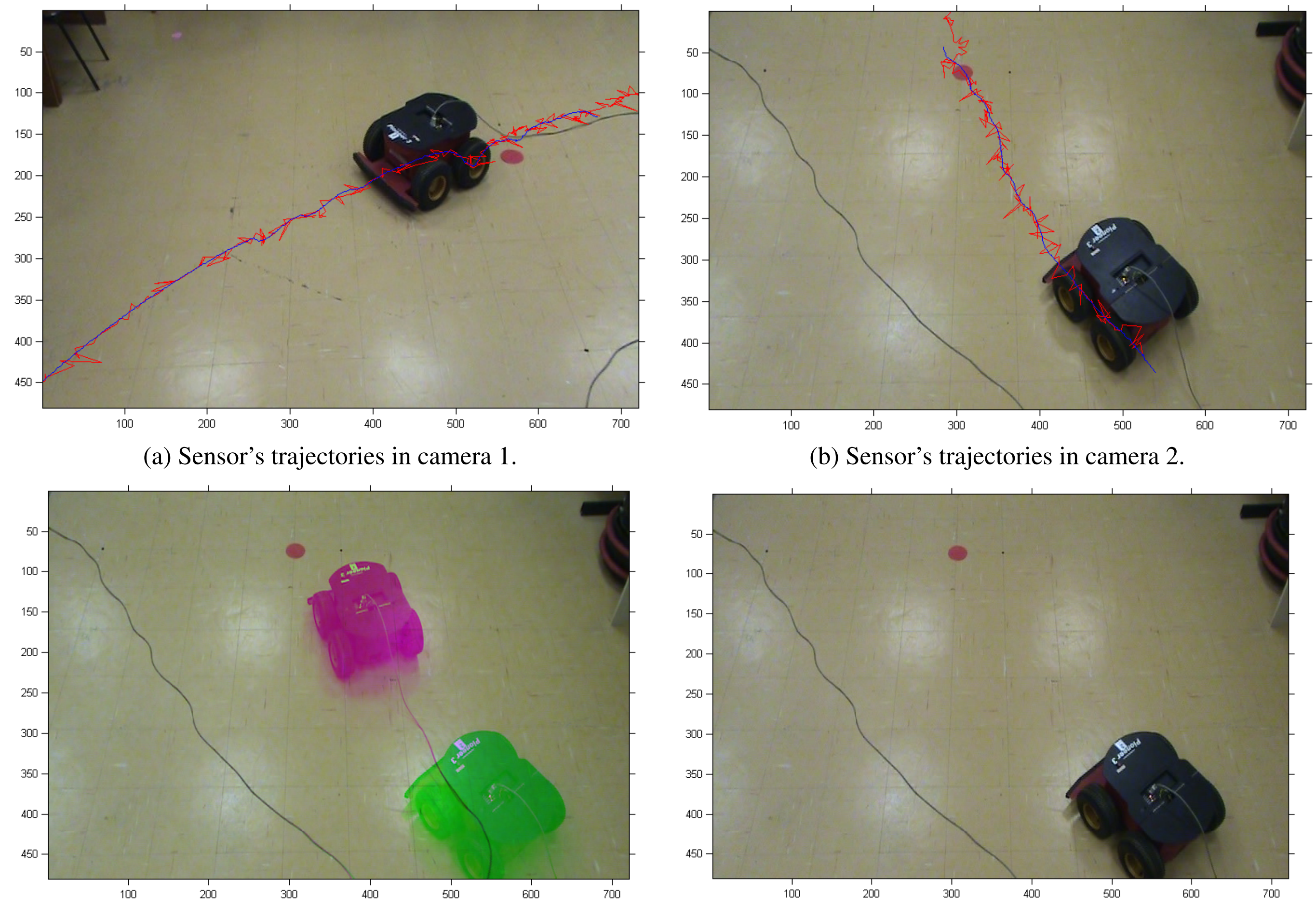
[1] W. Anthony, L. Robert, and B. Prosenjit. Temporal synchronization of video sequences in theory and in practice. In *Workshop on Motion and Video Computing*, volume 2, 2005.

[2] R. Carceroni, F. Pádua, G. Santos, and K. Kutulakos. Linear Sequence-to-Sequence Alignment. In *Proc. of IEEE Computer Vision and Pattern Recognition Conference*, volume 1, pages 746–753, June 2004.

[3] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 682–689, Hilton Head Island, South Carolina, June, 13-15 2000.

[4] Y. Caspi and M. Irani. Alignment of non-overlapping sequences. In *Proc. of International Conference on Computer Vision*, volume 2, pages 76–83, Vancouver, Canada, July, 9-12 2001.

[5] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. In *VAMODS (Vision and Modelling of*



(a) Sensor's trajectories in camera 1.

(b) Sensor's trajectories in camera 2.

(c) Before temporal alignment.

(d) After temporal alignment.

Figure 5. (a) and (b) Trajectories of the sensor's centroid in cameras 1 and 2, respectively. The blue trajectories were estimated by the WSL tracker [10], while the red ones were obtained by the projection of the 3D sensor's trajectory in the image planes, using the projection matrices computed during the calibration of the cameras. (c) *Before alignment image* was created by superimposing the green band of a frame t_2 with the red and blue bands of frame $t_1 = (t_2 - \beta^g)/\alpha^g$ using ground truth timeline coefficients α^g and β^g. (d) *After alignment image* was created by replacing the green band of the image with that of frame $t_1 = (t_2 - \beta^e)/\alpha^e$, with α^e, β^e computed by our algorithm. Deviations from the ground-truth alignment cause "double exposure" artifacts.

Dynamic Scenes) workshop with ECCV, Copenhagen, Denmark, May, 28-31 2002.

[6] C. Dai, Y. Zheng, and X. Li. Accurate video alignment using phase correlation. *IEEE Signal Processing Letters*, 13(12):737–740, 2006.

[7] C. Dai, Y. Zheng, and X. Li. Subframe video synchronization via 3d phase correlation. In *Proc. IEEE International Conference on Image Processing*, pages 501–504, 2006.

[8] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[9] R. F. Garcia, P. Shiroma, L. Chaimowicz, and M. F. M. Campos. Um Arcabouço para a Localização de Enxames de Robôs. In *Proc. of VIII SBAI*, 2007.

[10] A. Jepson, D. Fleet, and T. El-Maraghi. Robust on-line appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003.

[11] I. Kitahara, H. Saito, S. Akimichi, T. Onno, Y. Ohta, and T. Kanade. Large-scale virtualized reality. In *Conference on Computer Vision and Pattern Recognition - Technical Sketches*, 2001.

[12] I. Laptev, S. J. Belongie, P. Perez, and J. Wills. Periodic motion detection and segmentation via approximate sequence alignment. In *Proc. of International Conference on Computer Vision*, volume 1, pages 816–823, 2005.

[13] K. Lee and R. D. Green. Temporally synchronising image sequences using motion kinematics. In *Proc. of Image and Vision Computing New Zeland*, 2005.

[14] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22:758–767, August 2000.

[15] D. W. Pooley, M. J. Brooks, A. J. van den Hengel, and W. Chojnacki. A voting scheme for estimating the synchrony of moving-camera videos. In *Proc. IEEE International Conference on Image Processing*, volume 1, pages 413–416, 2003.

[16] K. Raguse and C. Heipke. Photogrammetric synchronization of image sequences. In *Proc. ISPRS Commission V Symposium on Image Engineering and Vision Metrology*, pages 254–259, 2006.

[17] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood. View-invariant alignment and matching of video sequences. In *Proc. of International Conference on Computer Vision*, volume 2, pages 939–945, Nice,France, October, 13-16 2003.

[18] I. Reid and A. Zisserman. Goal directed video metrology. In *Proc. of the European Conference on Computer Vision*, pages 647–658, 1996.

[19] O. Shakil. An efficient video alignment approach for non-overlapping sequences with free camera movement. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 257–260, 2006.

[20] K. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. Camera Calibration Toolbox for Matlab. `http://www.vision.caltech.edu/bouguetj`, 2008.

[21] Y. Ukrainitz and M. Irani. Aligning sequences and actions by maximizing space-time correlations. In *Proc. European Conference on Computer Vision*, pages 538–550, 2006.

[22] M. Ushizaki, T. Okatani, and K. Deguchi. Video synchronization based on co-occurrence of appearance changes in video sequences. In *Proc. IEEE International Conference on Pattern Recognition*, pages 71–74, 2006.

[23] D. Wedge, D. Huynh, and P. Kovesi. Using space-time interest points for video sequence synchronization. In *Proc. IAPR Conference on Machine Vision Applications*, 2007.

[24] D. Wedge, P. Kovesi, and D. Huynh. Trajectory based video sequence synchronization. In *Proc. of Digital Image Computing: Techniques and Applications*, pages 79–86, 2005.

[25] L. Wolf and A. Zomet. Correspondence-free synchronization and reconstruction in a non-rigid scene. In *Workshop on Vision and Modelling of Dynamic Scenes*, Copenhagen, Denmark, May 2002.

[26] L. Wolf and A. Zomet. Sequence to sequence self calibration. In *Proc. of the European Conference on Computer Vision*, volume 2, pages 370–382, May 2002.

[27] L. Wolf and A. Zomet. Wide baseline matching between unsynchronized video sequences. *International Journal of Computer Vision*, 68(1):43–52, 2006.

[28] J. Yan and M. Pollefeys. Video synchronization via space-time interest point distribution. In *Proc. of Advanced Concepts for Intelligent Vision Systems*, 2004.