Temporal Synchronization of Non-Overlapping Videos Using Known Object Motion

Darlan N. Brito[a], Flávio L. C. Pádua[a], Guilherme A. S. Pereira[b], Rodrigo L. Carceroni[c]

[a] *Department of Computing, Centro Federal de Educação Tecnológica de Minas Gerais, Av. Amazonas, 7675, 30510-000, BH, MG, Brazil*
[b] *Department of Electrical Engineering, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, 31270-010, BH, MG, Brazil*
[c] *Department of Computer Science, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, 31270-010, BH, MG, Brazil*

Abstract

This paper presents a robust technique for temporally aligning multiple video sequences that have no spatial overlap between their fields of view. It is assumed that (i) a moving target with known trajectory is viewed by all cameras at non-overlapping periods in time, (ii) the target trajectory is estimated with a limited error at a constant sampling rate, and (iii) the sequences are recorded by stationary cameras with constant frame rates and fixed intrinsic and extrinsic parameters. The proposed approach reduces the problem of synchronizing N non-overlapping sequences to the problem of robustly estimating a single line from a set of appropriately-generated points in $\mathbb{R}^{N+1}$. This line describes all temporal relations between the N sequences and the moving target. Our technique can handle arbitrarily-large misalignments between the sequences and does not require any a priori information about their temporal relations. Experimental results with real-world and synthetic sequences demonstrate that our method can accurately align the videos.

Keywords: Video Synchronization, Temporal Alignment, Non-Overlapping Fields of View

1. Introduction

The use of multiple video cameras to visualize and reconstruct large-scale scenes has become popular worldwide. As a result, many novel applications have been developed using multiple video recordings, such as, video post-production (Cao et al., 2009), three-dimensional photogrammetric analysis (Raguse and Heipke, 2006), video mosaicing (Gong and Yang, 2005), sporting analysis (Saito et al., 2004) and interactive reconstruction of virtual environments (Gibson et al., 2003). In order to retrieve accurate semantic and geometric information from the monitored scene, all those applications demand on synchronized video sequences. Typically, the temporal misalignment between video sequences occurs when they have different frame rates, or when there is a time shift between them (e.g. the cameras are not activated simultaneously).

Although synchronization can be manually performed, this approach is prone to human error, especially when there are several video sequences. Alternatively, the temporal alignment may be estimated using camera synchronization hardwares or network connections (Kitahara et al., 2001). Unfortunately, special hardwares are not a practical solution for remote and wireless applications. Moreover, it is very complex to specify special hardwares for synchronizing cameras of different technologies and vendors. On the other hand, the use of a network connection to synchronize cameras requires the application of special methods to deal with the non-determinism in the network dynamics, such as propagation time or physical channel access time, which makes the synchronization task a very challenging problem in many multi-camera systems (Sivrikaya and Yener, 2004).

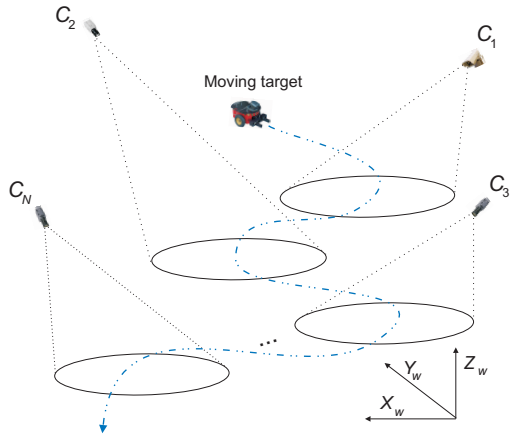


Figure 1: A 3D scene is viewed simultaneously by N stationary cameras at distinct viewpoints, whose fields of view do not necessarily overlap. A moving target crosses the fields of view of all cameras.

In this context, the use of software-based methods has demonstrated to be an interesting alternative to recover synchronization from visual cues. This work proposes a method that belongs to this last group of software based solutions. Specifically, we propose an algorithm for temporally aligning multiple video sequences that have no spatial overlap between their fields of view. Our method is derived from the method presented by Pádua et al. (2008)

Email addresses: darlan@lsi.cefetmg.br (Darlan N. Brito), cardeal@decom.cefetmg.br (Flávio L. C. Pádua), gpereira@ufmg.br (Guilherme A. S. Pereira), carceron@dcc.ufmg.br (Rodrigo L. Carceroni)

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 $\mathbb{R}^{N+1}$ that completely describes all temporal relations between the sequences and a moving target in the viewed scene. Note that the space considered has an additional dimension, indistinguishable from the other N, which refers to the moving target. The trajectory of the moving target is assumed to be known *a priori*, being related to a fixed reference frame and estimated with a limited error at a constant sampling rate.

An interesting characteristic 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. 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 $\Re^{N+1}$.

Importantly, a reliable algorithm for the solution of the asynchronism problem between multiple video sequences should be able to handle cases like (Pádua et al., 2008):

- Unknown cameras frame rates;
- Arbitrary time shift between the sequences;
- Unknown object motion;
- Presence of tracking failures;
- Computational efficiency should degrade gracefully with increasing number of video sequences;
- Unknown user-defined camera set-up;

Our approach operates under all 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*. This scenario is typical in several applications, such as, automatic video-based surveillance of large-scale scenes and video-based modeling and rendering of three-dimensional scenes.

1.1. Related Work

Some authors classify the existing video synchronization methods in two groups: the feature-based methods and the direct methods.

Most part of recent related work on video synchronization is formed by feature-based methods (Cao et al., 2009; Tresadern and Reid, 2009; Pádua et al., 2008; Wedge et al., 2007; Wolf and Zomet, 2006; Raguse and Heipke, 2006; Lei and Hang, 2006; Caspi et al., 2006), to list just a few. Those methods 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 (Ushizaki et al., 2006; Ukrainitz and Irani, 2006; Shakil, 2006; Dai et al., 2006a,b; Sand and Teller, 2004; Caspi and Irani, 2001, 2000) 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. The method proposed in this work belongs to the group of feature-based methods.

Many existing feature-based techniques (Wolf and Zomet, 2006; Caspi et al., 2006; Wedge et al., 2005; Rao et al., 2003; Wolf and Zomet, 2002a,b; Lee et al., 2000; Stein, 1998) perform 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. 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. Hence, efficiency considerations greatly limit the applicability of these solutions.

Some other feature–based methods are based on the establishment of putative frame correspondences and use robust line-fitting techniques such as RANSAC (Pádua et al., 2008) or the Hough Transform (Tresadern and Reid, 2009; Pooley et al., 2003) to reduce the effect of gross outliers when estimating synchronization parameters. Tresadern and Reid (2009) present a method for synchronizing two video sequences, which recovers potential frame correspondences, estimates the synchronization parameters via the Hough transform and refines these parameters using non-linear optimization methods. The main limitation of that method is that it relies on two matched sets of points moving non-rigidly in the scene. Pádua et al. (2008) propose an approach to align N overlapping sequences directly. That work reduces the problem of synchronizing N sequences to the problem of robustly estimating a single line in $\mathbb{R}^N$. Moreover, it is based on the assumption of known epipolar geometry, which is often computed using static background points that are common to the views. For pairs of cameras with wide baselines, this should be done using feature descriptors that are appropriate for wide-baseline stereo matching (Tola et al., 2008).

The method proposed in this work is inspired by the work of Pádua et al. (2008). Differently from their approach, we propose a technique to align N non-overlapping sequences that are acquired by stationary cameras, whose intrinsic and extrinsic parameters are known a priori. In this scenario, instead of using the assumption of known epipolar geometry and estimating trajectories of objects that are observed by all the cameras, our approach uses a single moving target in a preliminary step, whose 3D localization with respect to a fixed reference frame may be estimated with a constant sampling rate. This moving target crosses all the fields of view (see Figure 1) and our technique establishes correspondences between the temporal coordinates of the frames of the sequences and the sample numbers of the moving target. This procedure leads to

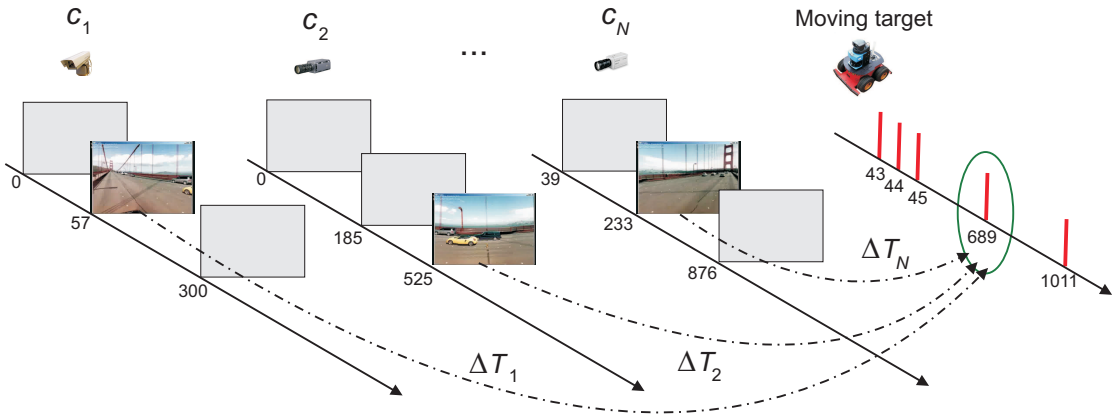


Figure 2: Temporal misalignment between a moving target and N cameras. The location sample 689 of the moving target corresponds in time to the frames 57, 525 and 233 of cameras c_1, c_2 and c_N, respectively. Our goal is to determine a global timeline that recovers the temporal alignment between the cameras by using the synchronization offsets ΔT_1, ΔT_2, ..., ΔT_N, between the target and the cameras.

a simple algorithm for reconstructing the timeline that recovers the temporal alignment between the sequences. Our approach is useful, not only in the case of non-overlapping sequences, but also when there is very little common appearance information between images, and are therefore difficult for standard video alignment techniques.

Feature-based techniques relying on space-time interest points have also been proposed for pairwise video alignment (Wedge et al., 2007; Laptev et al., 2005; Yan and Pollefeys, 2004). Usually, these techniques are not robust when the sequences contain objects moving in front of a cluttered background and tend to fail on sequences from widely-separated viewpoints.

Regarding feature-based methods for simultaneously aligning more than two sequences, only a few works have been proposed (Cao et al., 2009; Pádua et al., 2008; Lei and Hang, 2006; Raguse and Heipke, 2006; Whitehead et al., 2005). Cao et al. (2009) present a method to synchronize multiple non-overlapping video sequences that are captured by cameras undergoing similar ego-motions. The proposed algorithm makes use of fundamental ratios, which are the ratios of the elements of homogeneous four-dimensional feature vectors characterizing the camera ego-motion. As our method, that approach assumes that the camera internal parameters are fixed throughout the video.

Raguse and Heipke (2006) propose a method where the temporal misalignment is modeled 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 (2006) 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. Whitehead et al. (2005) present a two-stage approach for aligning three sequences. The method relies on 2D shape heuristics in order to bring feature trajectories into a rough temporal alignment. This alignment is then refined by enforcing trifocal constraints. Differently from their approach, our method is applied to a general number of video sequences and does not demand on a reliable tracker across many frames.

Finally, there are only a few works based on direct methods to align sequences without any overlap (Shakil, 2006; Caspi and Irani, 2001). The most relevant work was developed by Caspi and Irani (2001), 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.

1.2. Paper outline

This paper builds on our previous work (Brito et al., 2008) with (1) an updated discussion of related work, (2) a new set of experiments on real-world and synthetic sequences, (3) a detailed analysis of our method's reliability with respect to errors in the 3D target localization, in the tracking system and in the camera calibration and (4) an improved explanation of our method.

The remainder of this paper is organized as follows. Section 2 presents the problem formulation. Section 3 covers our temporal synchronization algorithm. Experimental results are presented in Section 4, followed by the conclusions and discussion in Section 5.

2. Problem Formulation

Suppose that a dynamic scene is viewed simultaneously by N stationary calibrated cameras, whose fields of view do not necessarily overlap. Moreover, consider the presence of a moving target in the 3D scene, whose trajectory in the world coordinate system may be estimated with a constant sampling rate. Suppose also that this target crosses the fields of view of all cameras (see Figure 1).

We assume that each camera captures frames with a constant, unknown frame rate and that the cameras as well as the moving target 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 target and N cameras. In that example, the location sample 689 of the moving target corresponds in time to the frames 57, 525 and 233 of cameras c_1, c_2 and c_N, respectively. Therefore, the temporal misalignments between the target and those cameras are $\Delta T_1 = 632$, $\Delta T_2 = 164$ and $\Delta T_N = 456$, respectively. Analogously, the temporal misalignments between those cameras are $\Delta T_{12} = 468$, $\Delta T_{1N} = 176$ and $\Delta T_{2N} = 292$. Our goal is to determine a global timeline that recovers the temporal alignment between the cameras, by using the synchronization offsets between the moving target and these cameras.

The constant sampling rate assumption for the video cameras and the moving target implies that the temporal coordinates (time stamps) of the target samples and the temporal coordinates (frame numbers) of all video sequences are related by a one dimensional affine transformation (Pádua et al., 2008):

$$t_i = \alpha_i \, t_r \; + \; \beta_i, \tag{1}$$

where t_i and t_r denote the temporal coordinates of the i-th video sequence and the temporal coordinates of the moving target, respectively. The parameters α_i, β_i are unknown constants describing the temporal dilation and temporal shift, respectively, between the target and the i-th sequence. In general, these constants will not be integers (Pádua et al., 2008).

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

$$\mathcal{L} = \left\{ \begin{bmatrix} \alpha_1 & \dots & \alpha_{n+1} \end{bmatrix}^\top t_r + \begin{bmatrix} \beta_1 & \dots & \beta_{n+1} \end{bmatrix}^\top \mid \; t \in \Re \right\}. \tag{2}$$

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.

Finally, we assume that the target moves along smooth 3D trajectories, which can be captured in the image planes of all cameras by using the projection matrices obtained during their calibration, as well as by using standard trackers (Jepson et al., 2003; Isard and MacCormick, 2001; Shi and Tomasi, 1994) that output trajectory segments as parametric curves.

In this work, a moving target is any visual entity moving in the space observed by the cameras with the ability to localize itself relatively to the world reference frame. Examples of targets include mobile robots that combine several sensors for localization and a human being carrying out a GPS (Global Positioning System) receiver in an outdoor environment. Although the accurate localization of mobile robots and moving entities is still an open problem, which has been a major research topic in the past few years (Thrun et al., 2005; Se et al., 2005; Pereira et al., 2003), our experiments in Section 4 show that the proposed methodology is robust to relatively large localization errors, thus requiring fairly simple localization techniques. In that section we present very controlled experiments performed with synthetic data, which allows us to analyze how the accuracy of the proposed methodology is affected by the errors in the 3D target localization, and real world experiments, which validate the methodology for realistic scenarios and actual localization approaches.

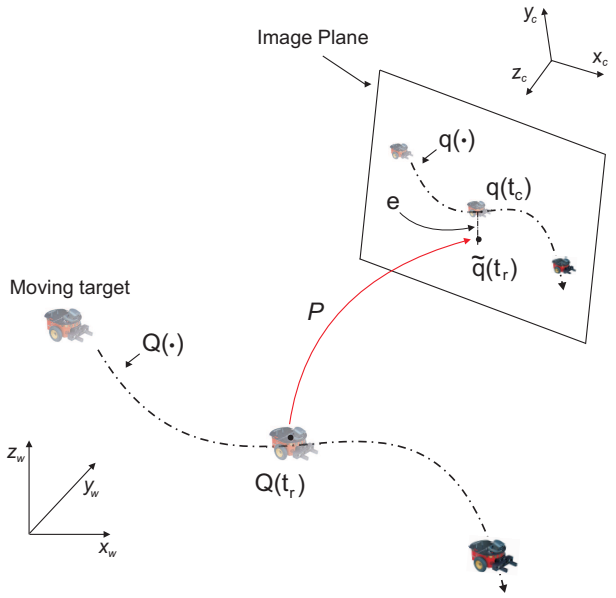


Figure 3: A target moves along a trajectory $\mathbf{Q}(\cdot)$ in a 3D scene, viewed by a camera. Let $\mathbf{q}(\cdot)$ be the trajectory traced by the target in the image plane, computed by a tracking algorithm. Consider that $\mathbf{q}(t_c)$ represents the target's instantaneous position in the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

3. Temporal Synchronization Algorithm

The proposed synchronization algorithm operates in three steps. Consider Figure 3 where a target moves along a trajectory $\mathbf{Q}(\cdot)$ in a 3D scene, viewed by a camera. Suppose that the 3D target's trajectory in the world coordinate system may be estimated by combining localization sensors or using a GPS receiver. Let $\mathbf{q}(\cdot)$ be the trajectory traced by the target's projection in the image plane, computed by an object tracking algorithm (Jepson et al., 2003; Isard and MacCormick, 2001; Shi and Tomasi, 1994).

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

In this scenario, the first step of our algorithm is based on the key observation that, by determining correspondences between 2D target 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 frames of the video sequence and the sample numbers of the moving target.

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

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

where $D(\cdot)$ denotes the euclidean distance between the points $\mathbf{q}(t_c)$ and $\tilde{\mathbf{q}}(t_r)$, and ε denotes a tolerance in pixels. In Figure 4, we illustrate four examples of *voting spaces* that were obtained in the real-world experiments described in the next section. In general, the set $\mathcal{V}$ described in Equation (3) will contain outliers.

The second step of our algorithm consists in to determine the most appropriate subset of candidate temporal alignments in $\mathcal{V}$ that will be used to determine the timeline that recovers the temporal alignment between the camera and the moving target. To estimate this subset in the presence of outliers, we use the RANSAC algorithm (Fischler and Bolles, 1981). 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.

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 target 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 subset estimated to compute the timeline parameters. By combining the computed equations $t_i = \alpha_i t_r + \beta_i$ 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

In this section, we present and discuss experimental results with real-world and synthetic sequences. Firstly, we illustrate the applicability of our approach by testing it on two-view datasets of real-world dynamic scenes. After that, we perform a careful analysis of the accuracy of our approach, by using synthetic sequences of an artificial scene. Specifically, we computed quantitative measurements of the quality of the estimated temporal alignments as a function of three key factors: (1) the accuracy in the target 3D localization (2) the accuracy of the tracking system and (3) the accuracy in the camera calibration. The values of the main parameters used by our temporal alignment algorithm in our experiments are listed in Table 1.

Importantly, we use the average absolute temporal alignment error ε_t as our basic measurement for evaluating the accuracy of our approach. Consider, for instance, a two-view dataset. In this case, we have:

$$\varepsilon_t = \frac{1}{M} \sum_{t_{c_1}=0}^{M-1} \left| t^e_{c_2}(t_{c_1}) - t^g_{c_2}(t_{c_1}) \right|. \qquad (5)$$

where t_{c_1} and M are, respectively, the temporal coordinate and the number of frames of the sequence acquired by camera c_1, while $t^g_{c_2}$ and $t^e_{c_2}$ represent the corresponding temporal coordinate of t_{c_1} in the sequence acquired by camera c_2, which were computed by the "ground-truth" timeline and the timeline estimated by our method.

4.1. Real-World Sequences

The timeline reconstruction algorithm was tested on two challenging real-world two-view datasets. In both datasets, the video sequences were acquired by two cameras with identical frame rates (30fps). Image dimensions were about 720×480 pixels in all cases. The cameras

Parameters	Meaning	Value
ε	Tolerance for obtaining the voting space	10
p	RANSAC parameter: probability that at least one sample set is error-free	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 vote and the timeline	0.5

Table 1: Parameter values used in the experiments.

were calibrated according to the algorithm proposed by Zhengyou Zhang (Zhang, 2000).

4.1.1. Indoor Scene

As a first test, we applied our method to a two-view dataset of an indoor scene. The moving target that crossed the fields of view of both cameras was a robot Pioneer 3AT, by *Active Media*. The 3D localization data of the target were estimated at a rate of 7.5 samples per second by using the visual localization system proposed by Garcia et al. (2007). The average localization error of that system was about 1% of the area covered by the smallest field of view. We used the WSL tracker (Jepson et al., 2003) to track the robot in each sequence.

The 3D localization system estimated the positions of the robot's center of gravity in the world reference frame. On the other hand, the WSL tracker estimated a blob centroid in the image coordinate system of each camera. Although the blob centroid did not necessarily correspond to the projection of the robot's center of gravity, we noted that, in practice, this fact did not affect the accuracy of the timeline, since the offset between both estimations remained approximately invariant for all points.

The lengths of the sequences acquired by both cameras c_1 and c_2 were 1692 and 914 frames, respectively. These sequences contained a single rigid object (robot) moving over a static background, along a fairly smooth trajectory, as illustrated in Figures 4(a)-(b). The blue trajectories were estimated by the WSL tracker, while the red ones were obtained by projecting the robot's 3D trajectory in the image planes, using the projection matrices computed during the calibration of the cameras. The robot appears in 750 frames of the sequence acquired by camera c_1 and in 308 frames of the sequence acquired by camera c_2.

In this experiment, the cameras had a very small overlap between their fields of view, which was intentionally created to verify the accuracy of alignment. However, this overlapping region was not used in the estimation process, to imitate the case of truly non-overlapping sequences. In fact, that region was used only for display and verification purposes. Moreover, in order to test the method with cameras of different frame rates, we modified the sequence acquired by camera c_2, simulating a frame-rate of 15fps (half its original value). In this case, the ground-truth temporal dilation and temporal shift between the sequences were $\alpha = 0.5$ and $\beta = 146 \pm 0.5$ frames, respectively.

In Figures 4(c) and 4(f), we show the estimated *voting spaces* for the moving target r and the two cameras c_1 and c_2 used in our experiment. The reconstructed lines $t_{c_1} = 3.9979t_r - 269.8932$ and $t_{c_2} = 1.9947t_r + 11.3588$ describe the temporal alignments between the target and cameras c_1 and c_2, respectively. From those equations we obtain the new equation $t^e_{c_2} = 0.4989t_{c_1} + 146.0185$ that retrieves the temporal alignment between the two video sequences. According to Equation (5), the reconstructed line gives an average absolute temporal alignment error ε_t of 0.3939 frames or 26.2731 miliseconds.

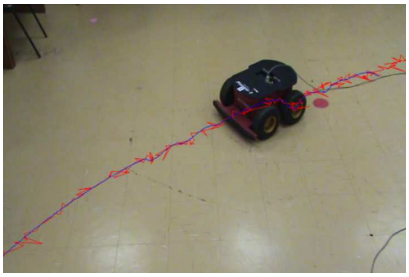
Therefore, our results show that our method may work successfully even when the sequences have large temporal misalignments (in this example, 146 frames). This scenario may be critical for several current temporal alignment techniques. Figures 4(d)-(e) confirm that the temporal alignment between the video sequences was effectively retrieved. In Figure 4(d), 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 sequences. In Figure 4(e), 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.

4.1.2. Outdoor Scene

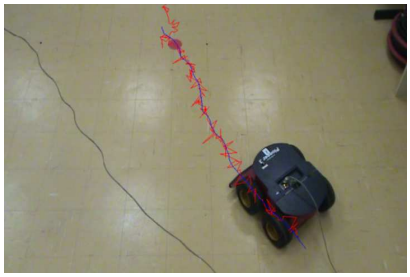
In a second experiment, we applied our method to a wide-baseline setup of cameras in an outdoor scene. In this case, the moving target crossing the fields of view of both cameras was a pedestrian carrying out a GPS (Global Positioning System) receiver.

Differently from the first experiment, the 3D localization data of the target were estimated at a rate of 1 sample per second and the average localization error was much more severe, representing about 10% of the area covered by the smallest field of view. Moreover, the WSL tracker could not be used in both sequences, since one of them suffered from occlusions. For this case, we have manually estimated the 2D trajectory of the pedestrian.

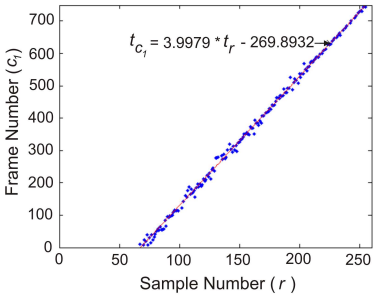
The lengths of the video sequences acquired by both cameras c_1 and c_2 were 2455 and 2230 frames, respectively. These sequences contained a pedestrian moving in a dynamic scene, specifically, an avenue that meets heavy crosstown traffic. The pedestrian followed a smooth trajectory, as illustrated in Figures 4(g)-(h). The blue trajectory in Figure 4(g) was estimated by using the WSL tracker,



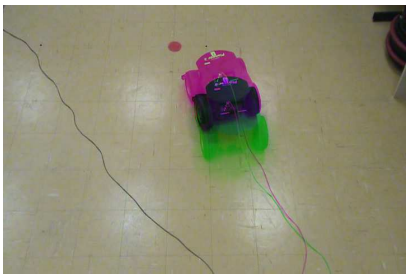
(a) Target's trajectories in c_1.



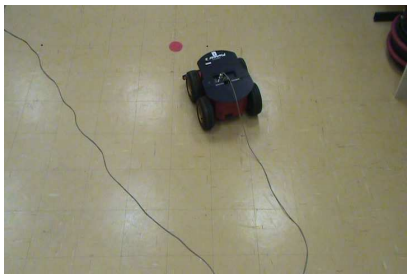
(b) Target's trajectories in c_2.



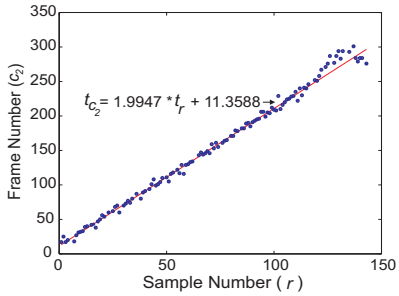
(c) *Voting Space* for c_1 and r.



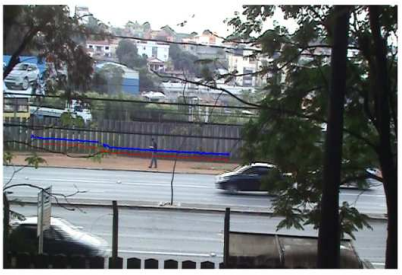
(d) Before temporal alignment.



(e) After temporal alignment.

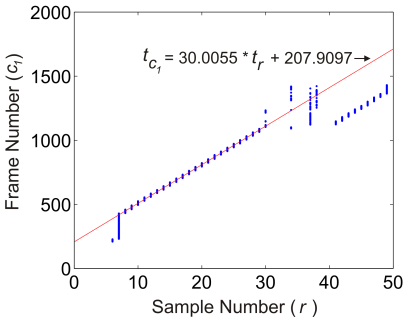


(f) *Voting Space* for c_2 and r.



(g) Target's trajectories in c_1.

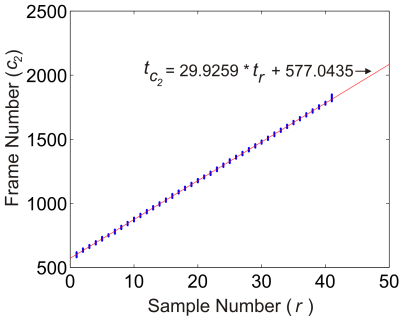
(h) Target's trajectories in c_2.



(i) *Voting Space* for c_1 and r.

(j) Before temporal alignment.

(k) After temporal alignment.



(l) *Voting Space* for c_2 and r.

Figure 4: (a)-(b) and (g)-(h) Trajectories of the moving targets in cameras 1 and 2, respectively. The blue trajectories were estimated by the WSL tracker (except the trajectory in (h) that was manually estimated), while the red ones were obtained by projecting the 3D target's trajectory in the image planes. (d) and (j) *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. (e) and (k) *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. (c) and (f) *Voting Spaces* of the indoor scene for the cameras c_1, c_2 and the moving target r. (i) and (l) *Voting Spaces* of the outdoor scene for the cameras c_1, c_2 and the moving target r.

while the blue trajectory in Figure 4(h) was manually estimated. The red trajectories were acquired by projecting the pedestrian's 3D trajectory, obtained from GPS data, in the image planes. The pedestrian appears in 1683 frames of the sequence acquired by camera c_1 and in 1568 frames of the sequence acquired by camera c_2.

Similarly to the robot dataset, the difference between the tracker trajectory and the projected trajectory was fairly invariant during the experiment and did not affect the estimation of the timeline. This difference occured because the GPS estimates corresponded to the receiver position in the 3D world, while the tracker estimates corresponded to the blob centroid in the image planes.

In this experiment, the cameras had no spatial overlap between their fields of view. The ground-truth temporal dilation and temporal shift between the video sequences were $\alpha = 1$ and $\beta = 370 \pm 0.5$ frames, respectively. In Figures 4(i) and 4(l), we show the estimated *voting spaces* for the moving target r and the two cameras c_1 and c_2 used in our experiment. The reconstructed lines $t_{c_1} = 30.0055t_r + 207.9097$ and $t_{c_2} = 29.9259t_r + 577.0435$ describe the temporal alignments between the target and cameras c_1 and c_2, respectively. From those equations we obtain the new equation $t^e_{c_2} = 0.9974t_{c_1} + 369.6848$ that retrieves the temporal alignment between the two video sequences. The reconstructed line gives an average absolute temporal alignment error ε_t of 2.9665 frames or 98.8 miliseconds. As expected, the severe 3D localization error observed in this case resulted in a temporal alignment that was more inaccurate than that one computed in our previous experiment. In fact, from Figures 4(j)-(k) we note that the computed alignment was not sufficient to completely cancel the "double exposure" artifacts.

4.2. Synthetic Sequences

Through the use of synthetic data, we evaluated how the method's reliability is affected by errors (i) in the 3D target localization, (ii) in the tracking system and (iii) in the camera calibration. We considered an artificial scene monitored by two synthetic calibrated cameras that had no spatial overlap between their fields of view.

The 3D target trajectory was randomly generated in the world reference frame of the artificial scene, following a very simple dynamics model and having a lifespan of 256 samples. In particular, the target's 3D instantaneous position, $\mathbf{Q}(t)$, was computed according to a randomly-drawn vector, $\vec{\mathbf{A}}(t)$:

$$\mathbf{Q}(1) = \mathbf{Q}(0) + \vec{\mathbf{A}}(1), \tag{6}$$

$$\mathbf{Q}(t) - \mathbf{Q}(t-1) = \mathbf{Q}(t-1) - \mathbf{Q}(t-2) + \vec{\mathbf{A}}(t), \quad 2 \leq t \leq 255. \tag{7}$$

The orientation and length of vector $\vec{\mathbf{A}}(t)$ were drawn from normal distributions with mean zero and standard deviations of 5 degrees and 0.5 meters, respectively. This choice produces an average projected velocity of two pixels per frame for both cameras, which is approximately equal to that observed in some of our real sequences.

We have simulated 100 distinct 3D target trajectories in the artificial scene and computed the voting spaces relating those 3D trajectories with their corresponding projections on the cameras' image planes. Controlled levels of noise were added both to the 3D and 2D trajectories computed, as well as to the projection matrices. The goal was to simulate the accuracy limitations of actual 3D localization systems, as well as of common camera calibration and tracking techniques. To illustrate the applicability of our approach, we have considered the percentage of simulation runs that produced highly-accurate timelines ($\varepsilon_t \leq 1$ frame) and less challenging situations ($\varepsilon_t \leq 5$ frames), as illustrated in Figures 5(a)-(f).

All synthetic video sequences had the same length (512 frames) and were obtained by cameras with the same frame rate. For all runs we defined the following ground-truth affine transformations for modeling the temporal misalignment between cameras c_1, c_2 and the moving target r:

$$t_{c_1} = 2\,t_r + 4, \tag{8}$$

$$t_{c_2} = 2\,t_r + 45, \tag{9}$$

that is, the temporal offset between the cameras was 41 frames ($t_{c_2} = t_{c_1} + 41$).

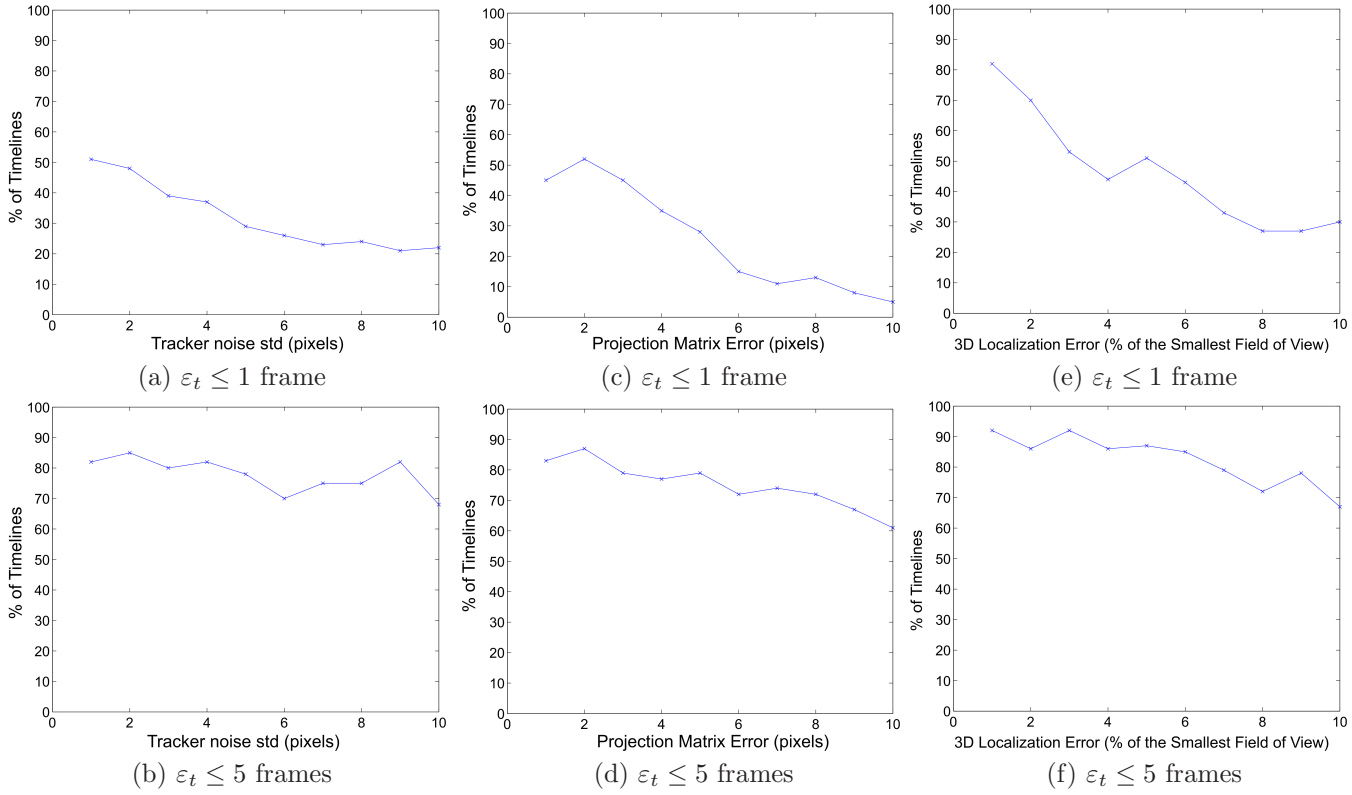
Initially, we simulated errors in the tracking algorithm by adding a random displacement to the projection of the moving target. This displacement had a uniformly distributed orientation, and a magnitude drawn from a normal distribution with mean zero and standard deviation of S_t pixels, for $S_t \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

To make the alignment problem even more challenging, both projection matrices used in this experiment had an average error of about 2 pixels, while the 3D target trajectory was corrupted with a random displacement with a magnitude drawn from a normal distribution with mean zero and standard deviation of about 5% of the area covered by the smallest field of view. Figures 5(a)-(b) show the impact of the tracking error on alignment accuracy.

In a second step, we simulated errors in the camera calibration technique. In order to simulate the fact that the projection matrices obtained during calibration may be inaccurate, we perturbed the ground-truth projection matrices before each run to achieve a predefined average error of S_c pixels, for $S_c \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. To generate a projection matrix with a given error S_c, we began with the ground-truth matrix, add the constant 10^{-5} to each element, measure the projection error, and iterate until the average error becomes equal to S_c.

Again, to make the alignment problem more challenging, we considered the 2D trajectories in both image planes corrupted by a random displacement with a uniformly distributed orientation, and a magnitude drawn from a normal distribution with mean zero and standard deviation of 2 pixels. Moreover, the 3D target trajectory was corrupted with a random displacement that had a magnitude drawn from a normal distribution with mean zero and standard deviation of about 5% of the area covered by the smallest field of view. Figures 5(c)-(d) show the impact of the projection matrices errors on alignment accuracy.

As a final step, we applied our technique in scenarios where the error in the 3D localization of the moving target



(a) $\varepsilon_t \leq 1$ frame (c) $\varepsilon_t \leq 1$ frame (e) $\varepsilon_t \leq 1$ frame

(b) $\varepsilon_t \leq 5$ frames (d) $\varepsilon_t \leq 5$ frames (f) $\varepsilon_t \leq 5$ frames

Figure 5: Percentage of runs for which the reconstructed timeline was below a specified bound on alignment error ($\varepsilon_t \leq 1$ frame or $\varepsilon_t \leq 5$ frames), as a function of the: (a)-(b) tracking error, (c)-(d) projection matrix error and (e)-(f) 3D target localization error.

varied according to a random displacement with magnitude drawn from a normal distribution with mean zero and standard deviation equal to $S_l\%$ of the area covered by the smallest field of view, for $S_l \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Both projection matrices used in this experiment had an average error of about 2 pixels, while the 2D trajectories in both image planes were corrupted by a random displacement with a magnitude drawn from a normal distribution with mean zero and standard deviation of 2 pixels. Figures 5(e)-(f) show the impact of the 3D target localization errors on alignment accuracy.

By observing Figures 5(a)-(f), we note that the ability of the method to achieve accurate alignments diminishes with increased noise levels. This degradation is especially pronounced when we consider the computation of highly-accurate timelines (Figures 5(a), 5(c) and 5(e)).

Two reasons explain this degradation in accuracy for all cases. First, as the noise levels increase, potential inliers are shifted from their "true" positions in the voting space, and the magnitude of these shifts is proportional to the noise level. This affects the line-fitting process in RANSAC and produces timelines with inaccurate parameters. Second, the presence of noise causes a significant increase in outlier votes, which also affects negatively the accuracy of RANSAC estimation.

Finally, from Figure 5(e), we note that for achieving highly-accurate alignments in more than 80% of the trials for $S_t = S_c = 2$ pixels (common noise levels in many applications), the 3D localization error should be smaller than or equal to 1% of the area covered by the smallest field of view. On the other hand, when the temporal alignment error may be up to 5 frames, our approach may succeed in more than 70% of the trials even when severe noise levels are considered (Figures 5(b), 5(d) and 5(f)).

5. Conclusions

This work 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 algorithm provides a simple and effective method that is able to handle arbitrary temporal dilations and large time shifts. By reducing the alignment problem to a RANSAC-based procedure, our method is able to tolerate large proportions of outliers in the data due to errors in the 3D target localization and in the tracking and camera calibration techniques.

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 target. 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 direction for future work consists in to extend our approach to be used with unknown user-defined camera setups. If rough information about the relative camera positions and orientations is available, it may be possible to extend the algorithm to do both synchronization and refinement of the camera calibration parameters, assuming that several objects cross the fields of view.

Acknowledgments We would like to thank Pedro Shiroma, Vilar da Camara Neto, Frederico Lima and Michelle Santos for their support with the experiments. Flávio Pádua and Darlan Brito thank the support of FAPEMIG-Brazil under Proc. EDT- 162/07. Guilherme Pereira thank CNPq-Brazil for the financial support.

References

Brito, D., Pádua, F., Pereira, G., Carceroni, R., 2008. Synchronizing Video Cameras with Non-overlapping Fields of View. In: Brazilian Symposium on Comp. Graphics and Image Processing. pp. 37–44.

Cao, X., Wu, L., Xiao, J., Foroosh, H., J., Z., Li, X., 2009. Video Synchronization and its Application to Object Transfer. Image and Vision Computing (in press).

Caspi, Y., Irani, M., 2000. A step towards sequence-to-sequence alignment. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 682–689.

Caspi, Y., Irani, M., 2001. Alignment of non-overlapping sequences. In: IEEE International Conf. on Computer Vision. pp. 76–83.

Caspi, Y., Simakov, D., Irani, M., 2006. Feature-based Sequence-to-Sequence Matching. International Journal of Computer Vision 68 (1), 53–64.

Dai, C., Zheng, Y., Li, X., 2006a. Accurate video alignment using phase correlation. IEEE Signal Proc. Letters 13 (12), 737–740.

Dai, C., Zheng, Y., Li, X., 2006b. Subframe video synchronization via 3d phase correlation. In: IEEE International Conference on Image Processing. pp. 501–504.

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

Garcia, R. F., Shiroma, P., Chaimowicz, L., Campos, M. F. M., 2007. Um Arcabouço para a Localização de Enxames de Robôs. In: Brazilian Symposium on Intelligent Automation.

Gibson, S., Hubbold, R., Cook, J., Howard, T., 2003. Interactive Reconstruction of Virtual Environments from Video Sequences. Computers & Graphics 27 (2), 293–301.

Gong, M., Yang, Y., 2005. Camera Field Rendering for Static and Dynamic Scenes. Graphical Models 67 (2), 73–99.

Isard, M., MacCormick, J., 2001. Bramble: A bayesian multiple-blob tracker. In: IEEE Int. Conf. on Computer Vision. pp. 34–41.

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

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

Laptev, I., Belongie, S. J., Perez, P., Wills., J., 2005. Periodic motion detection and segmentation via approximate sequence alignment. In: IEEE International Conf. on Computer Vision. pp. 816–823.

Lee, L., Romano, R., Stein, G., 2000. Monitoring activities from multiple video streams: Establishing a common coordinate frame. IEEE Trans. on Pattern Analysis and Mach. Intel. 22, 758–767.

Lei, C., Hang, Y., 2006. Tri-Focal Tensor-Based Multiple Video Synchronization With Subframe Optimization. IEEE Transactions on Image Processing 15 (9), 2473–2480.

Pádua, F., Carceroni, R., Santos, G., Kutulakos, K., 2008. Linear Sequence-to-Sequence Alignment. IEEE Transactions on Pattern Analysis and Machine Intelligence (in press).

Pereira, G., Kumar, V., Campos, M., 2003. Localization and Tracking in Robot Networks. In: IEEE International Conference on Advanced Robotics. pp. 465–470.

Pooley, D. W., Brooks, M. J., van den Hengel, A. J., Chojnacki, W., 2003. A voting scheme for estimating the synchrony of moving-camera videos. In: IEEE International Conference on Image Processing. Vol. 1. pp. 413–416.

Raguse, K., Heipke, C., 2006. Photogrammetric synchronization of image sequences. In: ISPRS Commission V Symposium on Image Engineering and Vision Metrology. pp. 254–259.

Rao, C., Gritai, A., Shah, M., Syeda-Mahmood, T., 2003. View-invariant alignment and matching of video sequences. In: IEEE International Conference on Computer Vision. pp. 939–945.

Saito, H., Inamoto, N., Iwase, S., 2004. Sports scene analysis and visualization from multiple-view video. In: IEEE International Conference on Multimedia and Expo. pp. 1395–1398.

Sand, P., Teller, S., 2004. Video Matching. ACM Transactions on Graphics 23 (3), 592–599.

Se, S., Lowe, D., Little, J., 2005. Vision-based Global Localization and Mapping for Mobile Robots. IEEE Transactions on Robotics 21 (3), 364–375.

Shakil, O., 2006. An efficient video alignment approach for non-overlapping sequences with free camera movement. In: IEEE Int. Conf. on Acoustics, Speech and Signal Proc. pp. 257–260.

Shi, J., Tomasi, C., 1994. Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 593–600.

Sivrikaya, F., Yener, B., 2004. Time Synchronization in Sensor Networks: A Survey. IEEE Network 18 (4), 45–50.

Stein, G., 1998. Tracking from multiple view points: Self-calibration of space and time. In: DARPA Image Understanding Workshop. pp. 521–527.

Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic Robotics. The MIT Press.

Tola, E., Lepetit, V., Fua, P., Lausanne, S., 2008. A Fast Local Descriptor for Dense Matching. In: IEEE Conference on Computer Vision and Pattern Recognition.

Tresadern, P., Reid, I., 2009. Video Synchronization from Human Motion Using Rank Constraints. Computer Vision and Image Understanding (in press).

Ukrainitz, Y., Irani, M., 2006. Aligning sequences and actions by maximizing space-time correlations. In: European Conference on Computer Vision. pp. 538–550.

Ushizaki, M., Okatani, T., Deguchi, K., 2006. Video synchronization based on co-occurrence of appearance changes in video sequences. In: IEEE International Conf. on Pattern Recognition. pp. 71–74.

Wedge, D., Huynh, D., Kovesi, P., 2007. Using space-time interest points for video sequence synchronization. In: IAPR Conference on Machine Vision Applications. pp. 190–194.

Wedge, D., Kovesi, P., Huynh, D., 2005. Trajectory based video sequence synchronization. In: Digital Image Computing: Techniques and Applications. pp. 79–86.

Whitehead, A., Laganiere, R., Bose, P., 2005. Temporal synchronization of video sequences in theory and in practice. In: Workshop on Motion and Video Computing. pp. 132–137.

Wolf, L., Zomet, A., 2002a. Correspondence-free synchronization and reconstruction in a non-rigid scene. In: Workshop on Vision and Modelling of Dynamic Scenes.

Wolf, L., Zomet, A., 2002b. Sequence to sequence self calibration. In: European Conference on Computer Vision. pp. 370–382.

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

Yan, J., Pollefeys, M., 2004. Video synchronization via space-time interest point distribution. In: Advanced Concepts for Intelligent Vision Systems. pp. 501–504.

Zhang, Z., 2000. A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (11), 1330–1334.