Synchronization of Two Independently Moving Cameras without Feature Correspondences

Tiago Gaspar[1], Paulo Oliveira[1], and Paolo Favaro[2]

[1] Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
[2] University of Bern, Bern, Switzerland

Abstract. In this work, a method that synchronizes two video sequences is proposed. Unlike previous methods, which require the existence of correspondences between features tracked in the two sequences, and/or that the cameras are static or jointly moving, the proposed approach does not impose any of these constraints. It works when the cameras move independently, even if different features are tracked in the two sequences. The assumptions underlying the proposed strategy are that the intrinsic parameters of the cameras are known and that two rigid objects, with independent motions on the scene, are visible in both sequences. The relative motion between these objects is used as clue for the synchronization. The extrinsic parameters of the cameras are assumed to be unknown. A new synchronization algorithm for static or jointly moving cameras that see (possibly) different parts of a common rigidly moving object is also proposed. Proof-of-concept experiments that illustrate the performance of these methods are presented, as well as a comparison with a state-of-the-art approach.

1 Introduction

In the last few years, the proliferation of digital cameras transformed the acquisition and manipulation of videos into common tasks. Having several videos of a given event, recorded by different people from different viewpoints, is thus more and more common. Synchronizing these videos is essential to merge all the available information, which can then be used in a wide range of areas, such as 3D reconstruction, human action recognition, calibration of multiple cameras, or dynamic depth estimation, see examples in [22], [24], [16], and [26], respectively.

In professional applications, it is possible to synchronize two cameras using proper hardware. However, such hardware is expensive and is usually not available to the common user. Moreover, in many situations, synchronizing the videos turns out to be important only after their acquisition. Therefore, since accurate manual synchronization is both tedious and difficult, the problem of synchronizing two videos, usually acquired by cameras with unknown relative inter-camera extrinsic parameters, has received a lot of attention in the last decade.

1.1 Previous Work

The video synchronization problem that has received more attention from the scientific community considers that the cameras are static and that there exist

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 189–204, 2014.

correspondences between the features observed in the two videos. Two of the directions of work that have been pursued to solve this problem are presented by Tresadern and Reid in [20] and by Caspi and Irani in [2], in what they called the "feature-based sequence alignment" approach. In the first case, the time offset is found by searching for the minimum of the relative magnitude of the fourth singular value of the "measurement matrix" introduced by Tomasi and Kanade in [19]. The second strategy aligns video sequences, in time and space, when the two sequences are related by a homography or by the projective epipolar geometry. To overcome the requirement that the cameras are static or jointly moving, i.e., that the relative inter-camera extrinsic parameters do not change, and that correspondences between the features observed in the two videos exist, some research has been done on algorithms that drop one of these assumptions.

In [2], Caspi and Irani present a second method, the "direct intensity-based sequence alignment", which exploits spatio-temporal brightness variations within each sequence. This approach can handle complex scenes and drops the need for having feature correspondences across the two sequences, but still requires that the cameras are static or jointly moving and that they see the same scene. In [22], Wolf and Zomet propose a strategy that builds on the idea that every 3D point tracked in one sequence results from a linear combination of the 3D points tracked in the other sequence. This approach copes with articulated objects, but still requires that the cameras are static or moving jointly.

There are also some works that can deal with independently moving cameras, but at the cost of requiring the existence of correspondences between features tracked in the two video sequences. Tuytelaars and Van Gool were the first to address the problem of automatic video synchronization for independently moving cameras and general 3D scenes, see [21]. This is done by reformulating the video synchronization problem in terms of checking the rigidity of at least 5 non-rigidly moving points, matched and tracked throughout the two sequences. Another example is the work by Meyer et al. [11], which consists of a two-step algorithm that leads to subframe-accurate synchronization results. First, an algorithm that estimates a frame-accurate offset by analysing the motion trajectories observed in the images and by matching their characteristic time patterns is used. After this step, subframe-accurate results are obtained by estimating a fundamental matrix between the two cameras, using a correspondence of 9 non-rigidly moving points in the scene. Both the motion of the cameras and the motion of the tracked object are assumed to be linear between consecutive time instants.

Video synchronization has been addressed from different perspectives. However, to the best of our knowledge, the most general and complex case, which arises when the cameras move independently and the parts of the moving object in the field of view of each camera do not intersect, is yet to be solved. None of the previous strategies would work in this situation, as there is no correspondence between the features observed in the two cameras. In [23], Yan and Pollefeys suggest a novel algorithm that uses the correlation between the distributions of space-time interest points, which represent special events in the videos, to synchronize them. This method does not explicitly require feature correspondences

and static or jointly moving cameras, but their fields of view must intersect and its performance degrades as the baseline between the cameras gets wider.

1.2 Contributions

The main contribution of our work is a method that synchronizes two video sequences acquired by independently moving cameras that see (possibly) different parts of a common rigidly moving object, see Fig. 1. The scene recorded by the cameras must also include a second common object (typically a static background), whose motion must be independent of the one of the first object. From now on, this second object is referred to as the background. The fields of view of the two cameras may not intersect and no knowledge about the correspondence between the two video sequences, in terms of which trajectories belong to which objects, is required. This is one of the most common video synchronization problems, as it occurs every time two people use handheld cameras to record a rigid object moving on a static background (e.g., a car moving on the street). The relative inter-camera extrinsic parameters are unknown and the intrinsic parameters of the cameras (which can be calibrated *a priori* using the typical approaches, see [25] and [1]) are assumed to be known. The idea is to track two sets of features in each video sequence: one on the moving object and other on the background. These sets are used to retrieve the motion of the two objects with respect to each camera, using state-of-the-art structure and motion methods, see [8] and [18]. These results can be used to obtain information about the motion of one object with respect to the other, which is used as clue for the synchronization process. When the correspondences between the features and the two trajectories are not known, subspace clustering algorithms, such as the ones presented in [4] and [9], can be used to segment the two motions.

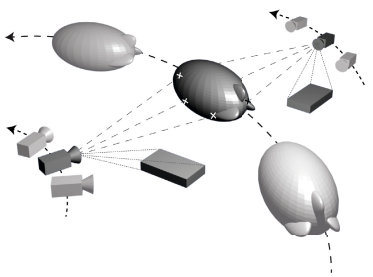


Fig. 1. Example of setup for the synchronization problem with independently moving cameras. The blimp and the parallelepipeds represent, respectively, a moving object and a static background.

In addition to the previous contribution, a new method that synchronizes two video sequences acquired by static or jointly moving cameras that see (possibly) different parts of a common rigidly moving object is also presented. This

method is closely related to the one mentioned in the previous paragraph, and is introduced first in the paper as it serves as a starting point for the more general case of independently moving cameras. The assumptions about the intrinsic parameters, extrinsic parameters, and fields of view of the cameras are the same as before. In this case, the motion of the object with respect to each camera can be used directly as clue for the synchronization, due to the constraints imposed on the motion of the cameras. A study of the uniqueness of the solutions obtained with this method is also presented.

The two video sequences, which are the only available data, are assumed to be acquired by cameras with the same frame rate, thus a single temporal offset between them is considered. This is without loss of generality since the multirate problem can be tackled by interpolating the measurements of the features in the video acquired with the lowest frame rate (note that typical object motions are smooth). The strategies proposed in this paper can be used after this resampling.

1.3 Notation and Paper Organization

In this document, the identity matrix with dimensions $k \times k$ is denoted I_k, and $0_{k\times n}$ is used to represent a matrix of zeros with k lines and n columns. The notation $||v||$ denotes the Euclidean norm of the vector v and $[v]_\times$ is used to represent the skew-symmetric matrix obtained from a given vector $v \in \mathbb{R}^3$. This matrix is such that $[v]_\times s = v \times s$, for any vector $s \in \mathbb{R}^3$, where $\times$ represents the cross-product. For a generic rotation matrix $R \in SO(3)$, the corresponding unit quaternion is given by $q = [\sin(\theta/2)v^T \cos(\theta/2)]^T$, where θ and v denote, respectively, the associated non-negative angle of rotation and the unit Euler axis. These quantities are such that $R = e^{[v]_\times \theta}$.

The remaining of this paper is organized as follows. A new algorithm that synchronizes video sequences acquired by static or jointly moving cameras is presented in section 2, as well as a study of the object trajectories that lead to a unique identification of the correct temporal offset. This algorithm is generalized for independently moving cameras in section 3. In section 4, a strategy that recovers the motion of an object from the time evolution of the images of its features is described, and in section 5 experimental results illustrating the performance of the proposed synchronization algorithms are presented. Finally, concluding remarks are provided in section 6.

2 Static and Jointly Moving Cameras

In this section, the synchronization of static or jointly moving cameras, when no correspondences between the features tracked in the two videos exist, is addressed. Instead of explicitly using the features to solve the synchronization problem, the rigid body transformations that explain their motion in the reference frame of each one of the cameras are used.

Let $X_i(k) \in \mathbb{R}^3$, for $k \in [k_0, k_0 + F]$, where k_0 and $k_0 + F$ correspond to the times of acquisition of the first and final frames of the videos, denote the

3D coordinates of an object feature expressed in the reference frame of camera i, $i = 1, 2$, at the time of the acquisition of the k-th frame. These coordinates can be obtained from the coordinates of the same feature at k_0 as $X_i(k) = R_i(k)X_i(k_0) + T_i(k)$, $i = 1, 2$, for all $k \in [k_0, k_0 + F]$, where $R_i(k) \in SO(3)$ and $T_i(k) \in \mathbb{R}^3$ denote, respectively, rotation matrices and translation vectors that describe the evolution in time of the coordinates of object features expressed in the reference frame of camera i.

Since the cameras are static or jointly moving, the relative inter-camera extrinsic parameters are constant, i.e., there exist a constant rotation matrix and a constant translation vector that transform coordinates expressed in the reference frame of camera 1 into the one of camera 2, similarly to what happens in the hand-eye calibration problem, see [7]. If these rotation and translation are denoted $R \in SO(3)$ and $T \in \mathbb{R}^3$, respectively, then it is possible to show that $X_2(k) = RR_1(k)R^T X_2(k_0) + RT_1(k) + (I_3 - R_2(k))T$, and consequently

$$R_2(k) = RR_1(k)R^T \quad \text{and} \quad T_2(k) = RT_1(k) + (I_3 - R_2(k))T, \tag{1}$$

for all $k \in [k_0, k_0 + F]$, when the two videos are synchronized. These expressions are not valid for unsynchronized videos, except in the cases detailed in section 2.2.

2.1 Video Synchronization

There are several methods that can be used to track a set of features belonging to an object moving on a video, being one of the most used the KLT feature tracker [15]. By combining such strategies with algorithms that recover structure and motion from image sequences, see [8] and [18], it is possible to retrieve the motion of the object, apart from a non-negative scaling factor in the magnitude of its translational component. More details about this procedure are provided in section 4. By applying this strategy to the two video sequences, the quantities $R_1(k)$, $\alpha_1 T_1(k)$, $R_2(k)$, and $\alpha_2 T_2(k)$, are obtained for all $k \in [k_0, k_0 + F]$. The constants α_1 and α_2 are non-negative scalars that account for the scaling ambiguity in the magnitude of the translation of the moving object.

According to the discussion above, for unsynchronized videos the expressions in (1) have the form

$$R_2(k') = RR_1(k)R^T \tag{2}$$

$$\alpha_2 T_2(k') = RT_1(k) + (I_3 - R_2(k'))T, \tag{3}$$

for all $k \in [k_0, k_0 + F]$, with $k' = k + \delta$, where δ denotes the temporal offset between the two sequences. This offset is considered to belong to a given interval, $\delta \in [-\Delta, \Delta]$, with $\Delta \leq F$ positive and known. Even though the two videos are unsynchronized, they are assumed to have at least $F + 1$ frames acquired at the same time instants. In the expressions, α_1 is considered to be the unit. This is without loss of generality, as there is an overall ambiguity in the magnitude of the two terms of equation (3).

If quaternions are used to parameterize the attitude associated with the rotation matrices in (2), this expression takes the form $q_2(k').q = q.q_1(k)$, where

"." denotes quaternion multiplication, and $q_2(k')$, $q_1(k)$, and q, are the unit quaternions (quaternions with unit norm) associated with $R_2(k')$, $R_1(k)$, and R, respectively (see [12] for details about the use of quaternions to represent rotations and section 1.3 for details about the notation used to represent quaternions). This expression can be written as

$$M(q_1(k), q_2(k'))q = 0_{4\times 1}, \tag{4}$$

with $M(q_1(k), q_2(k')) = \begin{bmatrix}\Psi(q_2(k')) - \Xi(q_1(k)) & q_2(k') - q1(k)\end{bmatrix}$. For a given quaternion $g = [w^T \; g_4]^T$, the matrices $\Xi(g)$ and $\Psi(g)$ have the form

$$\Xi(g) = \begin{bmatrix} g_4 I_3 + [w]_\times \\ -w^T \end{bmatrix} \quad \text{and} \quad \Psi(g) = \begin{bmatrix} g_4 I_3 - [w]_\times \\ -w^T \end{bmatrix}, \tag{5}$$

where $w \in \mathbb{R}^3$ is a vector and $g_4 \in \mathbb{R}$ a scalar, see [3].

If (3) is also written as a function of the unit quaternion q, we have that

$$\alpha_2 T_2(k') = \Xi^T(q)\Psi(q)T_1(k) + (I_3 - R_2(k'))T, \tag{6}$$

as $R = \Xi^T(q)\Psi(q)$, see [3]. The use of quaternions in this paper allows avoiding singularities in the representation of rotations, see [12].

By combining (4) and (6), the synchronization problem for static or jointly moving cameras can be cast into the form of the minimization problem

$$\hat{\delta} = \arg\min_{\delta} \; E_s(\delta), \tag{7}$$

where $\hat{\delta}$ denotes the estimated temporal offset and $E_s(\delta)$ is the error function

$$E_s(\delta) = \min_{(q,\,T,\,\beta_2)} \mu_R E_R(\delta, q) + \mu_T E_T(\delta, q, T, \beta_2) + \mu_q (q^T q - 1)^2, \tag{8}$$

with μ_R, μ_T, and μ_q, positive weighting coefficients. The last term in the expression forces $||q||$ to be the unit and the other two are obtained from

$$E_R(\delta, q) = \sum_{k=k_0+\Delta}^{k_0+F-\Delta} ||M(q_1(k), q_2(k+\delta))q||^2 \qquad \text{and}$$

$$E_T(\delta, q, T, \beta_2) = \sum_{k=k_0+\Delta}^{k_0+F-\Delta} ||\beta_2^2 T_2(k+\delta) - \Xi^T(q)\Psi(q)T_1(k) - (I_3 - R_2(k+\delta))T||^2.$$

The scalar β_2 is used to guarantee that α_2, with $\alpha_2 = \beta_2^2$, is not negative.

The optimization problem in (8) is a nonlinear least-squares problem due to the nonlinear dependence of $E_T(\delta, q, T, \beta_2)$ on q and β_2, thus it can be solved using the Levenberg-Marquardt method [10]. This problem transforms into a linear least-squares problem if α_2 is used and if R, in (3), is considered to be a generic constant matrix $P \in \mathbb{R}^{3\times 3}$. Linear constraints on the trace of P and on the l_1 and l_∞ norms of its line and column vectors are imposed to guarantee

that P is close to a rotation matrix, see [5] for details about these norms. If this relaxation is used and if the estimate found for P, by solving the linear version of the problem, is approximated by a rotation matrix, an initial guess for q, T, and β_2, is easily found. This approximation can be obtained using the algorithm proposed in [14], which can be used to approximate a given matrix by the closest rotation matrix in the least-squares sense.

The temporal offset between the two videos is the one that solves (7), and is found by evaluating the error function in (8) for all the offsets in a given range.

2.2 Uniqueness of Solution

There are situations in which the motion of the object does not have enough information for the synchronization process (imagine for instance that the object is stopped or moves with constant velocity). In these cases, the solution of the minimization problem introduced in the previous section is not unique, i.e., there are several temporal offsets that minimize the error function in (8).

According to (2) and (3), it is possible to conclude that the solution of the optimization problem in (7) is unique in terms of the temporal offset δ (meaning that $E_s(\delta)$ is null only for the correct offset), if and only if there not exist a non-negative constant scaling factor α, a constant rotation matrix R, and a constant translation vector T, that verify such equations for some temporal offset, different from the real δ. The trajectories of the objects that violate this condition are summarized in Lemma 1, where $\theta_i(k) \in \mathbb{R}$ and $v_i(k) \in \mathbb{R}^3$ are used to denote, respectively, the non-negative rotation angle and the corresponding Euler axis associated with $R_i(k)$, for all $k \in [k_0 - \Delta, k_0 + F + \Delta]$. The conditions presented on the lemma depend on T_i and R_i, but they do not need to be tested for both $i = 1$ and $i = 2$. It is enough to choose one of the cameras, for instance camera 1, and test if T_1 and R_1 verify any of such conditions.

Lemma 1. *The solution of the optimization problem presented in* (7) *is unique if and only if none of the following three conditions are met for some non-null* $\overline{\delta}$ *verifying* $|\overline{\delta}| \leq \Delta$*:*

1. $\theta_i(k) = 0$ *for all* $k \in [k_0 + \overline{\delta}_1, k_0 + F + \overline{\delta}_2]$ *and there exist a non-negative constant scalar* $\overline{\alpha}$ *and a constant rotation matrix* $\overline{R}$ *such that* $\overline{\alpha}\, T_i(k + \overline{\delta}) = \overline{R}\, T_i(k)$*, for all* $k \in [k_0, k_0 + F]$.
2. $\theta_i(k)$ *is periodic with period* $|\overline{\delta}|$ *for* $k \in [k_0 + \overline{\delta}_1, k_0 + F + \overline{\delta}_2]$*, the direction of* $\theta_i(k)v_i(k)$ *is constant in the same interval, and there exist a non-negative constant scalar* $\overline{\alpha}$ *and a constant rotation matrix* $\overline{R}$ *such that* $\theta_i(k+\overline{\delta})v_i(k+\overline{\delta}) = \overline{R}\,\theta_i(k)v_i(k)$*, for all* $k \in [k_0, k_0 + F]$*, and* $\overline{\alpha}\,[T_i(k + 2\overline{\delta}) - T_i(k + \overline{\delta})] = \overline{R}\,[T_i(k + \overline{\delta}) - T_i(k)]$*, for all* $k \in [k_0 - \overline{\delta}_1, k_0 + F - \overline{\delta}_2]$.
3. $\theta_i(k)$ *is periodic with period* $|\overline{\delta}|$ *for* $k \in [k_0 + \overline{\delta}_1, k_0 + F + \overline{\delta}_2]$*, the direction of* $\theta_i(k)v_i(k)$ *is not constant in the same interval, and there exist a non-negative constant scalar* $\overline{\alpha}$*, a constant vector* $\overline{T}$*, and a constant rotation matrix* $\overline{R}$ *such that*

$$\begin{aligned} \theta_i(k + \overline{\delta})v_i(k + \overline{\delta}) &= \overline{R}\,\theta_i(k)v_i(k) \\ \overline{\alpha}\, T_i(k + \overline{\delta}) &= \overline{R}\, T_i(k) + (I_3 - R_i(k + \overline{\delta}))\overline{T} \end{aligned}$$ *, for all* $k \in [k_0, k_0 + F]$.

In the previous expressions, $\overline{\delta}_1 = \min[\overline{\delta}, 0]$ *and* $\overline{\delta}_2 = \max[0, \overline{\delta}]$.

The conditions presented on the lemma can be easily tested for a given trajectory of the moving object. The procedure used to test them and the proof of the lemma are omitted here due to space constraints.

3 Independently Moving Cameras

When videos are acquired with independently moving cameras, tracking features on a single rigidly moving object is not enough for the synchronization. This is because the projection of such 3D features into acquired images results both from the motion of the object, which includes information for the synchronization, and from the motion of the camera, which does not. In this situation, features on a second rigidly moving object, for instance a static background, must be used. If the motion of this object is independent from the one of the first object, the relative motion between the two objects has information for the synchronization.

Let ${}^{c_0}M_i^j(k) \in \mathbb{R}^4$, for $k \in [k_0, k_0 + F]$, denote the homogeneous coordinates of a point of the j-th object, $j = 1, 2$, at the time of acquisition of the k-th frame. The superscript c_0 and the subscript i indicate that these coordinates are expressed in the reference frame $\{c_0\}$ of camera i, $i = 1, 2$, at the time of acquisition of the first frame k_0 of the video sequence. The evolution in time of the coordinates of this point is given by ${}^{c_0}M_i^j(k) = g_i^j(k)\,{}^{c_0}M_i^j(k_0)$, where $g_i^j(k)$ denotes a homogeneous transformation. Moreover, let ${}^{c_k}_{c_0}g_i(k)$ denote another homogeneous transformation, which converts coordinates of points expressed in $\{c_0\}$, into the coordinates of the same points expressed in $\{c_k\}$. Here, $\{c_k\}$ is used to identify the reference frame of camera i at the time of acquisition of frame k. This transformation represents the motion of camera i. If these two transformations are combined, a new transformation ${g_T}_i^j(k)$ that includes both the motion of the j-th object and the motion of the i-th camera, results

$$ {}^{c_k}M_i^j(k) = \underbrace{{}^{c_k}_{c_0}g_i(k)\; g_i^j(k)}_{{g_T}_i^j(k)}\; {}^{c_0}M_i^j(k_0). $$

This transformation relates ${}^{c_0}M_i^j(k_0)$, the homogeneous coordinates in the initial time instant of points of object j expressed in $\{c_0\}$, with ${}^{c_k}M_i^j(k) \in \mathbb{R}^4$, their homogeneous coordinates at the time of acquisition of frame k expressed in $\{c_k\}$.

From the three aforementioned transformations, only ${g_T}_i^j(k)$ can be obtained from the available features (apart from a non-negative scaling factor, as discussed in section 3.1), for all $k \in [k_0, k_0 + F]$. Thus, any relation used for the synchronization process has to be based on such transformation. Consider, for instance, the homogeneous transformation ${g_T}_i^1(k) = {}^{c_k}_{c_0}g_i(k)\; g_i^1(k)$, associated with the motion of object 1 with respect to camera i, which can be written as ${g_T}_i^1(k) = {}^{c_k}_{c_0}g_i(k)\; g_i^2(k)\; [g_i^2(k)]^{-1}\, g_i^1(k)$, since $g_i^2(k)[g_i^2(k)]^{-1} = I_4$. If $[g_i^2(k)]^{-1}\, g_i^1(k)$, which does not depend on the motion of the cameras, is denoted by $g_i(k)$, the previous expression can be rearranged in the form

$$ g_i(k) = [{g_T}_i^2(k)]^{-1}\; {g_T}_i^1(k). \tag{9} $$

If the homogeneous transformation from the reference frame of camera 1, at the initial instant, to the reference frame of camera 2, at the same instant, is denoted g, it is easy to show that $g_2^j(k) = g\, g_1^j(k)\, g^{-1}$, $j = 1, 2$, and consequently

$$g_2(k) = g\, g_1(k)\, g^{-1}, \tag{10}$$

as $g_i(k) = [g_i^2(k)]^{-1}\, g_i^1(k)$. When the two video sequences are synchronized, this expression is valid for all $k \in [k_0, k_0 + F]$.

If the rotations and translations associated with g and $g_i(k)$, $i = 1, 2$, are denoted by $R \in SO(3)$ and $T \in \mathbb{R}^3$, and by $R_i(k) \in SO(3)$ and $T_i(k) \in \mathbb{R}^3$, respectively, then (10) can be cast into the form of (1). The difference is that in section 2, $T_i(k)$ was determined using structure and motion strategies, which is not possible in this case. Here, the translational components of $g_{T_i}^j(k)$ can also be determined using structure and motion strategies, thus they are known up to a scaling factor, but $T_i(k)$ cannot. For independently moving cameras, $T_i(k)$ is obtained using (9). This procedure induces some structure on $T_i(k)$, which cannot be modelled with a single scaling factor. This is why the strategy proposed in section 2.1 cannot be used for independently moving cameras.

3.1 Video Synchronization

By using the strategy described in the beginning of section 2.1, it is possible to retrieve the values of $R_{T_i}^j(k)$ and $\alpha_i^j T_{T_i}^j(k)$, with $i = 1, 2$, and $j = 1, 2$, for all $k \in [k_0, k_0 + F]$. The rotation $R_{T_i}^j(k)$ and translation $T_{T_i}^j(k)$ are the ones associated with the homogeneous transformation $g_{T_i}^j(k)$, and α_i^j is a non-negative constant that accounts for the ambiguity in the magnitude of the translation of the j-th object, when it is estimated using the features observed in camera i.

According to the discussion above and to (9), we have that

$$\begin{aligned} R_i(k) &= [R_{T_i}^2(k)]^T R_{T_i}^1(k) \\ T_i(k) &= \alpha_i^1 \underbrace{[R_{T_i}^2(k)]^T T_{T_i}^1(k)}_{h_i^1(k)} - \alpha_i^2 \underbrace{[R_{T_i}^2(k)]^T T_{T_i}^2(k)}_{h_i^2(k)}, \end{aligned}$$

for all $k \in [k_0, k_0 + F]$ and for $i = 1, 2$. Note that the use of a single scaling factor is not enough to model the structure of the ambiguity in the determination of $T_i(k)$. The vectors $h_i^1(k) \in \mathbb{R}^3$ and $h_i^2(k) \in \mathbb{R}^3$, introduced in the expression, are used in this section with the single purpose of becoming the notation clearer.

If the expression in (10) is separated into its rotational and translational parts, it takes the following form for unsynchronized video sequences

$$R_2(k') = R R_1(k) R^T \tag{11}$$

$$\alpha_2^1 h_2^1(k') - \alpha_2^2 h_2^2(k') = R\left[h_1^1(k) - \alpha_1^2 h_1^2(k)\right] + (I_3 - R_2(k'))T, \tag{12}$$

for all $k \in [k_0, k_0 + F]$, and with $k' = k + \delta$, where δ is as defined in section 2.1. Note that α_1^1 was omitted from (12) as it is assumed to be the unit. This is

without loss of generality since there is an overall ambiguity in the magnitude of the two terms of equation (12).

If quaternions are used, the equation in (11) reduces to the form of (4), see details in section 2.1, where $q_1(k)$, $q_2(k')$, and q, are the unit quaternions associated, respectively, with the rotation matrices $R_1(k)$, $R_2(k')$, and R, redefined in this section for the case of independently moving cameras.

The expression in (12) can also be written as a function of the quaternion q, associated with the rotation R that relates the reference frames of the two cameras in the initial time instant. In this case, this expression takes the form $\alpha_2^1 h_2^1(k') - \alpha_2^2 h_2^2(k') = \Xi^T(q)\Psi(q)\left[h_1^1(k) - \alpha_1^2 h_1^2(k)\right] + (I_3 - R_2(k'))T$, where the matrices $\Xi(q)$ and $\Psi(q)$ are as defined in (5).

By combining the previous expression with the one relating the rotations perceived from both sequences, the synchronization problem for independently moving cameras can be cast into the form of the minimization problem

$$\hat{\delta} = \arg\min_{\delta} \; E_m(\delta), \tag{13}$$

where $\hat{\delta}$ denotes the estimated temporal offset and $E_m(\delta)$ is the error function

$$E_m(\delta) = \min_{(q,\, T,\, \beta_2^1,\, \beta_2^2,\, \beta_1^2)} \mu_R E_R(\delta, q) + \mu_T E_T(\delta, q, T, \beta_2^1, \beta_2^2, \beta_1^2) + \mu_q (q^T q - 1)^2, \tag{14}$$

with μ_R, μ_T, and μ_q, positive weighting coefficients and

$$E_R(\delta, q) = \sum_{k=k_0+\Delta}^{k_0+F-\Delta} ||M(q_1(k), q_2(k+\delta))q||^2$$

$$\begin{aligned} E_T(\delta, q, T, \beta_2^1, \beta_2^2, \beta_1^2) = \sum_{k=k_0+\Delta}^{k_0+F-\Delta} ||(\beta_2^1)^2 h_2^1(k+\delta) - (\beta_2^2)^2 h_2^2(k+\delta) - \\ - \Xi^T(q)\Psi(q)\left[h_1^1(k) - (\beta_1^2)^2 h_1^2(k)\right] - (I_3 - R_2(k+\delta))T||^2. \end{aligned}$$

The scalars β_2^1, β_2^2, and β_1^2 are used in these expressions to guarantee that α_2^1, α_2^2, and α_1^2 (with $\alpha_2^1 = (\beta_2^1)^2$, $\alpha_2^2 = (\beta_2^2)^2$, and $\alpha_1^2 = (\beta_1^2)^2$) are not negative.

The optimization problem in (14) is a nonlinear least-squares problem due to the nonlinear dependence of $E_T(\delta, q, T, \beta_2^1, \beta_2^2, \beta_1^2)$ on q, β_2^1, β_2^2, and β_1^2, thus it can be solved using the Levenberg-Marquardt method [10]. An initial guess for the unknowns q, T, β_2^1, β_2^2, and β_1^2, can be obtained by relaxing the problem, similarly to what was done in the end of section 2.1.

The temporal offset between the two videos is the one that solves (13), and is found by evaluating the error function in (14) for all the offsets in a given range. Moreover, note that with the proposed strategy it is possible to estimate the relative scales between the two objects. This is only possible because two cameras are used. In the monocular multi-body structure-from-motion problem, for instance, each reconstructed object has a different unknown scale, thus objects are distorted with respect to each other, see [13].

In this work, the correspondence between the two video sequences, in terms of which trajectories belong to which objects, is assumed to be unknown. A set

of features in one camera may correspond to any of the two sets in the other camera, thus two possible combinations between the sets are possible (once an association is assumed, the other is implicitly defined). The correct combination can be found by solving the previous optimization problem for the two cases, and choosing the one that leads to the minimum value for $E_m(\delta)$.

4 Object Motion Recovery

There are several methods to retrieve structure and motion from a sequence of images, see [8]. In this work, a strategy based on the concept of epipolar geometry is used to estimate the rotation matrices and translation vectors that define the motion of a set of 3D rigidly moving features, see [6].

Given the projection of a set of 3D features into two images, the essential matrix (the intrinsic parameters of the cameras are known) that relates the two views can be obtained using different strategies, depending on the number of available features, see examples in [6] and [17]. By using such methods, the essential matrices that relate the coordinates of features at a given time instant with their coordinates in the initial instant result. Moreover, if the standard algorithms described in [6] are used, these matrices can be converted into rotations and normalized translations of the object with respect to that instant. These rigid body transformations do not enforce a globally consistent geometry, as only the directions of the translations are retrieved, rather than the full 3D translation vectors. Strategies that enforce such global consistency are described in [8] and [18]. In this work, a modified version of such approaches, not described here in detail due to space constraints, is used. It is based on the alignment of 3D point clouds, and leads to translation vectors that are defined up to an overall ambiguity (in this case, a non-negative scaling factor) in their magnitude. This ambiguity cannot be removed unless some metric information about the scene is considered to be available, which is not the case.

5 Experimental Results

In this section, experimental results illustrating the performance of the proposed methods are presented, as well as a comparison with a state-of-the-art approach.

The videos were acquired with cameras of regular mobile phones, at 29 fps, and images with the spatial resolution 960×540 pixel were used. The intrinsic parameters of the cameras were calibrated using the toolbox in [1].

The features used in the synchronization were selected manually in the first frame of each sequence, and then tracked along the videos with the KLT feature tracker [15]. No strategy to deal with occlusions or outliers was implemented, as this is not the focus of this work, thus good features must be selected to guarantee that the motion recovery algorithm performs properly.

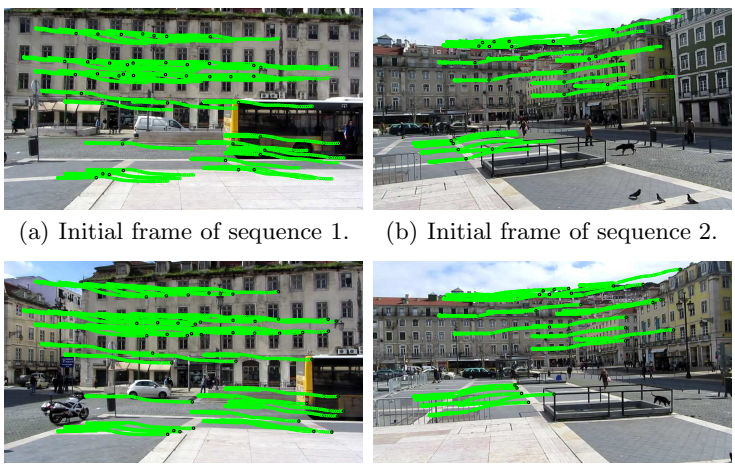
For the proof-of-concept experiments presented in this section, in which the video sequences are small, $\Delta = 10$ frames was considered. Larger values for Δ

can be used, specially for long sequences. The ground truth information was obtained using a photo-flash to mark some of the frames, as suggested in [21].

Two experiments are presented. In the first, two cameras were mounted on the same rigid platform, in such a way that their fields of view do not intersect (they were facing opposite directions). The inter-camera extrinsic parameters between the two remain constant over time, and features on the static background are used. This problem is the same as the more common situation where the cameras record an object that is moving between the two. In the second experiment, a tram was recorded with independently moving cameras, and the static background is used as a second object. The strategies proposed in this work were implemented using $\mu_T = 1$, $\mu_R = 10$, and $\mu_q = F$.

The results obtained with our algorithms are compared to the ones obtained with the method proposed in [22]. This method was developed to synchronize videos acquired with static or jointly moving cameras, when no correspondence between the features tracked in both videos exists. It consists in using an heuristic to examine the effective rank of a matrix constructed from the measurements. The heuristic proposed in the paper and the suggested threshold were used in the implementation of this algorithm. The comparisons with [22] serve two purposes: i) understand how our algorithm for static or jointly moving cameras compares to a state-of-the-art approach, and ii) confirm that such approach cannot be used to synchronize videos acquired with cameras that move independently.

The videos used in the first experiment have 121 frames and were obtained with two cameras moving jointly in the center of a public square. The first and final frames of the two sequences are depicted in Fig. 2, with the motion of the

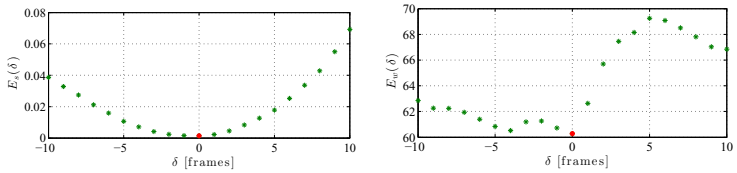


(a) Initial frame of sequence 1. (b) Initial frame of sequence 2.

(c) Final frame of sequence 1. (d) Final frame of sequence 2.

Fig. 2. Initial and final frames of the two videos in the experiment with jointly moving cameras. Green dots represent the evolution over time of features on the background, and black dots identify their position at the time of acquisition of the presented frames.

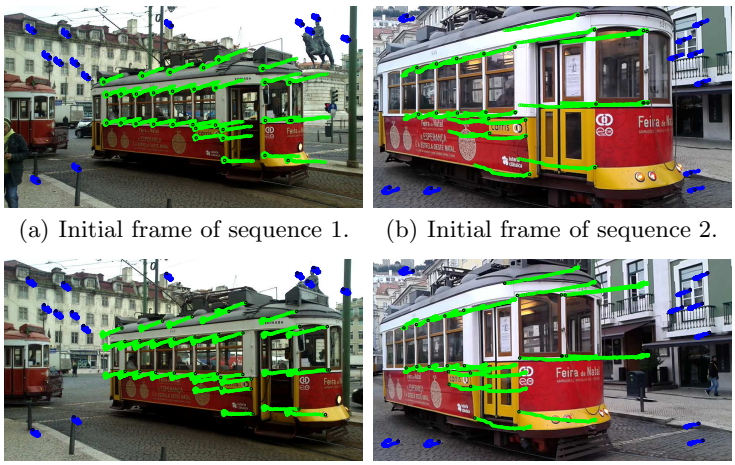
features used in the synchronization (that results from the motion of the platform in which the cameras were installed) superimposed on them. No correspondence between the features tracked in the two sequences exists, as the fields of view of the cameras do not intersect at any point.



(a) Our method. (b) Method proposed in [22].

Fig. 3. Error functions for the experiment with jointly moving cameras (videos previously synchronized using the ground truth obtained by marking some of the frames with a photo-flash). The dots in red identify the minima of the functions.

A comparison between the approach presented in section 2 and the one presented in [22] can be found in Fig. 3. In particular, the values of the error functions $E_s(\delta)$ and $E_w(\delta)$, proposed respectively in section 2.1 of this document and in [22], are presented for each one of the considered temporal offsets. Both methods correctly identify the temporal offset between the two video sequences ($\delta = 0$ as the videos were previously synchronized using the ground truth).



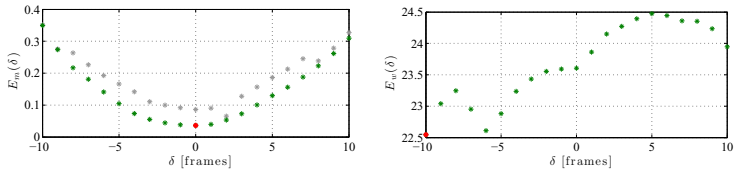
(a) Initial frame of sequence 1. (b) Initial frame of sequence 2.

(c) Final frame of sequence 1. (d) Final frame of sequence 2.

Fig. 4. Initial and final frames of the two sequences in the experiment with independently moving cameras. The evolution along time of features in the tram and features in the background are represented in green and blue, respectively. The black dots identify the position of the features at the time of acquisition of the presented frames.

The sequences used in the experiment with independently moving cameras have 96 frames. As before, there is no time offset between the two as they were previously synchronized using the ground truth. The first and final frames of the two videos are depicted in Fig. 4, with the time evolution of the features used in the synchronization superimposed on them. The motion of the features in blue result from the motion of the users that were holding the cameras. No correspondence between the features tracked in both sequences exists, as the cameras were in opposite sides of the tram (note for instance the open/closed door or the differences in the background). The values of the error functions $E_m(\delta)$ and $E_w(\delta)$, proposed respectively in section 3 of this document and in [22], are presented in Fig. 5, for each one of the considered time offsets. Our method identifies the temporal offset ($\delta = 0$) between the two sequences successfully, whereas the methods proposed in [22] does not. This was expected, since this algorithm was proposed for the case of static or jointly moving cameras.

The two curves in Fig. 5(a) correspond to the two combinations between the sets of features acquired by the cameras. The combination associated with the green curve is the correct, as it minimizes the minimum of the error function.



(a) Our method. (b) Method proposed in [22].

Fig. 5. Error functions for the experiment with independently moving cameras (videos previously synchronized using the ground truth obtained by marking some of the frames with a photo-flash). The dots in red identify the minima of the functions. The two curves in (a) result from evaluating $E_m(\delta)$ for the two possible combinations between the set of features associated with the moving object and with the static background.

6 Conclusions

In this paper, the video synchronization problem for cameras with fields of view that may not intersect was addressed. Our approach differs from previous methods as it can deal with independently moving cameras. Features on two rigidly moving objects with independent motions are tracked in both sequences, and used to retrieve the relative motion between the objects, which is used as clue for the synchronization. A similar approach is used to solve this problem for the particular case of static or jointly moving cameras. Both methods were tested and validated with real data, and the strategy proposed for static or jointly moving cameras was shown to perform similarly to a state-of-the-art approach.

References

1. Bouguet, J.: Camera Calibration Toolbox for Matlab, `http://www.vision.caltech.edu/bouguetj/calib_doc/`
2. Caspi, Y., Irani, M.: Spatio-temporal alignment of sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(11), 1409–1424 (2002)
3. Crassidis, J., Markley, F., Cheng, Y.: Survey of nonlinear attitude estimation methods. Journal of Guidance, Control, and Dynamics 30(1), 12–28 (2007)
4. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2790–2797 (2009)
5. Golub, G.H., Loan, C.F.V.: Matrix Computations, 3rd edn., vol. 1. JHU Press (1996)
6. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press (2004)
7. Horaud, R., Dornaika, F.: Hand-eye calibration. The International Journal of Robotics Research 14(3), 195–210 (1995)
8. Kanade, T., Morris, D.: Factorization methods for structure from motion. Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 356, 1153–1173 (1998)
9. Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y.: Robust recovery of subspace structures by low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(1), 171–184 (2013)
10. Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial & Applied Mathematics 11(2), 431–441 (1963)
11. Meyer, B., Stich, T., Magnor, M., Pollefeys, M.: Subframe temporal alignment of non-stationary cameras. In: British Machine Vision Conference (2008)
12. Murray, R., Li, Z., Sastry, S.: A Mathematical Introduction to Robotic Manipulation, 1st edn. CRC Press, Inc. (1994)
13. Ozden, K.E., Schindler, K., Van Gool, L.: Multibody structure-from-motion in practice. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(6), 1134–1141 (2010)
14. Schönemann, P.: A generalized solution of the orthogonal procrustes problem. Psychometrika 31(1), 1–10 (1966)
15. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (June 1994)
16. Sinha, S.N., Pollefeys, M.: Synchronization and calibration of camera networks from silhouettes. In: International Conference on Pattern Recognition, vol. 1, pp. 116–119 (August 2004)
17. Stewénius, H., Engels, C., Nistér, D.: Recent developments on direct relative orientation. Journal of Photogrammetry and Remote Sensing 60(4), 284–294 (2006)
18. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Buxton, B., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 709–720. Springer, Heidelberg (1996)
19. Tomasi, C.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision 9, 137–154 (1992)
20. Tresadern, P., Reid, I.: Synchronizing image sequences of non-rigid objects. In: British Machine Vision Conference, pp. 629–638 (2003)
21. Tuytelaars, T., Van Gool, L.: Synchronizing video sequences. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 762–768 (June 2004)

22. Wolf, L., Zomet, A.: Correspondence-free synchronization and reconstruction in a non-rigid scene. In: Workshop on Vision and Modelling of Dynamic Scenes (2002)
23. Yan, J., Pollefeys, M.: Video synchronization via space-time interest point distribution. Advanced Concepts for Intelligent Vision Systems (2004)
24. Yilma, A., Shah, M.: Recognizing human actions in videos acquired by uncalibrated moving cameras. In: IEEE International Conference on Computer Vision, vol. 1, pp. 150–157 (October 2005)
25. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11), 1330–1334 (2000)
26. Zhou, C., Tao, H.: Dynamic depth recovery from unsynchronized video streams. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 351–358 (June 2003)