On the Two-View Geometry of Unsynchronized Cameras

Cenek Albl[1] Zuzana Kukelova[1] Andrew Fitzgibbon[2] Jan Heller [3] Matej Smid[1] Tomas Pajdla[1]

[1]Czech Technical University in Prague
Prague
Czechia
{alblcene,kukelova}@cmp.felk.cvut.cz
smidm@cmp.felk.cvut.cz,pajdla@cvut.cz

[2] HoloLens, Microsoft
Cambridge
UK
awf@microsoft.com

[3]Magik Eye Inc.
New York
US
jan@magik-eye.com

Abstract

We present new methods for simultaneously estimating camera geometry and time shift from video sequences from multiple unsynchronized cameras. Algorithms for simultaneous computation of a fundamental matrix or a homography with unknown time shift between images are developed. Our methods use minimal correspondence sets (eight for fundamental matrix and four and a half for homography) and therefore are suitable for robust estimation using RANSAC. Furthermore, we present an iterative algorithm that extends the applicability on sequences which are significantly unsynchronized, finding the correct time shift up to several seconds. We evaluated the methods on synthetic and wide range of real world datasets and the results show a broad applicability to the problem of camera synchronization.

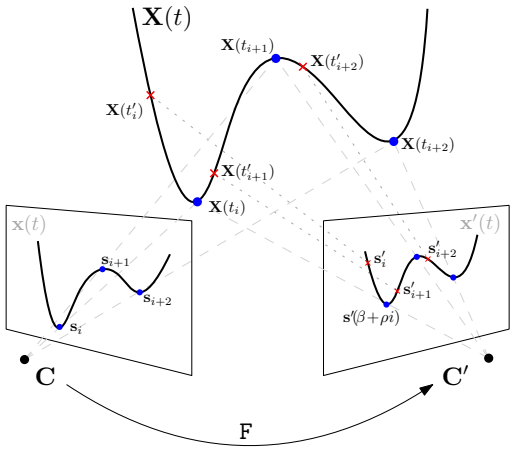

Figure 1. Two cameras capture a moving point at different times, the projection rays of the two cameras meet nowhere.

1. Introduction

Many computer vision applications, e.g., human body modelling [30, 5], person tracking [8, 36], pose estimation [11], robot navigation [1, 12], and 3D object scanning [26], benefit from using multiple-camera systems. In tightly-controlled laboratory setups, it is possible to have all cameras temporally synchronized. However, applicability of multi-camera systems could be greatly enlarged when cameras might run without synchronization [15]. Synchronization is sometimes not possible, e.g. in automotive industry, but even if it was possible, using asynchronous cameras may produce other benefits, e.g., reducing bandwidth requirements and improving temporal resolution of event detection and motion recovery [6].

In this paper, we (1) introduce *practical solvers* that simultaneously compute either a fundamental matrix or a homography and time shift between image sequences, and (2) we propose a fast *iterative algorithm* that uses RANSAC [10] with our solvers in the inner loop to synchronize large time offsets. *Our approach can accurately calibrate large time shifts, which was not possible before.*

1.1. Related work

Many video and/or image sequence synchronization methods are based on image content analysis [23, 2, 35, 4, 3, 7, 22, 24, 32], or on synchronizing video by audio tracks [29] and therefore their applicability is limited. Other approaches employed compressed video bitrate profiles [25] and still camera flashes [28].The methods differ in temporal transformation models. Often, time shift [23, 32, 35, 3], or time shift combined with variable frame rate [7, 22, 2], are used.

Many methods share a similar basis. A set of trajectories is detected in every video sequence using an interest point detector and an association rule or a 2D tracker. The trajectories are matched across sequences. A RANSAC based algorithm is often used to estimate jointly or in an iterative manner the parameters of temporal and spatial transfor-

mations [7, 22, 2]. In [7], RANSAC is used to search for matching trajectory pairs in filtered set of all combinations of trajectories in a sequence pair. The epipolar geometry has to be provided. The method [22] enables joint synchronization of N sequences by fitting a single N-dimensional line called *timeline* in a RANSAC framework. The algorithm [2] estimates temporal and spatial transformation based on tentative trajectory matches.

Methods using exhaustive search to find the homography [32] and either fundamental matrix or homography [33] along with the time offset were presented. These are searching over the entire space of possible time shifts.

The two most closely related works to ours are [21, 20] that jointly estimate two-view geometry together with time shift from approximated image point trajectories. In [21] estimated epipolar geometry or homography along with time shift using non-linear least squares, approximating the image trajectory by a straight line. The algorithm is initialized by the 7pt algorithm [14] and a zero time shift. Work [20] extended this approach by estimating difference in frame rate and using splines instead of lines. Both the above works achieve good results only when given a good initialization, e.g., on sequences less than 0.5 seconds time shift and with no gross matching errors.

1.2. Contribution

In this paper we present two new contributions.

First, we present a new method for *simultaneous computation of two-view camera geometry and temporal offset parameters from minimal sets of point correspondences*. We solve for fundamental matrix or homography together with temporal offset of image sequences. Our methods need only moving image point trajectories, which are easy to track. Unlike in [21, 20], we use a small (minimal) numbers of correspondences and we therefore are robust to outliers when combined with RANSAC robust estimation.

Secondly, we present an *iterative scheme using the minimal solvers to efficiently estimate large time offsets*. Our approach is based on RANSAC loop running our minimal solvers. This approach efficiently searches in the space of possible time offsets, which is much more efficient than exhaustive search methods [32, 33] developed before.

We evaluated our approach on a wide range of scenes and demonstrated its capability of synchronizing various kinds of real camera setups, such as driving cars, surveillance cameras, or sports match recordings with no other information than image data.

We demonstrate that our solvers are able to synchronize small time shifts of fractions of a second as well as large time shifts of tens of seconds. Our iterative algorithm is capable of synchronizing medium time shifts (i.e. tens of frames) with less than 5 RANSAC iterations and large time offsets (i.e. tens to hundreds of frames) using tens of RANSAC iterations. Overall, our approach is much more efficient than other methods utilizing RANSAC [22].

By solving two-camera synchronization problem, we also solve the multi-camera synchronization problem since temporal offsets of multiple cameras can be determined pairwise to serve as the initialization point for a global iterative solutions based on bundle adjustment [34].

2. Problem formulation

Let us consider two unsynchronized cameras with a fixed relative pose [14] producing a stereo video sequence by observing a dynamic scene. Motions of objects in the video sequence are indistinguishable from camera rig motions, and therefore, we will present the problem for static cameras and moving objects.

2.1. Geometry of two unsynchronized cameras

The coordinates of a 3D point moving along a smooth trajectory in space can be described by function

$$\mathbf{X}(t) = [X_1(t), X_2(t), X_3(t), 1]^\top, \tag{1}$$

where t denotes time, see Figure 1. Projecting $\mathbf{X}(t)$ into the image planes of the two distinct cameras produces two 2D trajectories $\mathbf{x}(t)$ and $\mathbf{x}'(t)$. Now, let's assume that the first camera captures frames with frequency f (period $p = 1/f$) starting at time t_0. This leads to a sequence of samples

$$\mathbf{s}_i = [u_i, v_i, 1]^\top = \mathbf{x}(t_i) = \pi(\mathbf{X}(t_i)),\ i = 1, \dots, n. \tag{2}$$

of the trajectory $\mathbf{x}(t)$ at times $t_i = t_0 + ip$.

Analogously, assuming a sampling frequency f' (period $p' = 1/f'$), at times $t'_j = t'_0 + jp'$, the second camera produces a sequence of samples

$$\mathbf{s}'_j = [u'_j, v'_j, 1]^\top = \mathbf{x}'(t'_j) = \pi'(\mathbf{X}(t'_j)),\ j = 1, \dots, n'. \tag{3}$$

In general, there is no correspondence between the s_i and s'_j samples, *i.e.*, for $i = j$, s_i and s'_j do not represent the projections of the same 3D point. There are two main sources of desynchronization in video streams. The first one is the different recording starts or camera shutters triggering independently leading to a constant time shift. The second source are different frame rates or imprecise clocks leading to different time scales. Assuming these two sources, we can map the time t to t' for frame i using $\jmath(i) : \mathbb{N} \to \mathbb{R}$ as

$$\jmath(i) = \frac{t_i - t'_0}{p'} = \frac{t_0 + ip - t'_0}{p'} = \frac{t_0 - t'_0}{p'} + \frac{p}{p'} i = \beta + \rho i, \tag{4}$$

where $\beta \in \mathbb{R}$ is captures the time shift and $\rho \in \mathbb{R}$ the time scaling. Note that $\jmath(i)$ is an integer-to-real linear mapping with an analogous inverse mapping $\imath(j)$. Given the model in (4) and a sequence of image samples $\mathbf{s}'_j, j = 1, \dots, n'$, we can interpolate a continuous curve $\mathbf{s}'(\jmath)$, for example

using a spline, so that the 2D point corresponding to $\mathbf{s}_i$ is approximately given as

$$\mathbf{s}_i \longleftrightarrow \mathbf{s}'(\beta + \rho i). \tag{5}$$

Notice that the interpolated image curve $\mathbf{s}'(\cdot)$ is not equivalent to the true image trajectory $\mathbf{x}'(\cdot)$, but may be expected to be a good approximation under certain conditions. Even though it might appear reasonable to assume time shift to be known within a fraction of a second, it is often the case in practice that the timestamps are based on CPU clocks which together with startup delays can lead to time shift β being in the order of seconds. On the other hand, the time scaling ρ is more often known or can be calculated accurately.

2.2. Epipolar geometry

At any given time t, the epipolar constraint of the two cameras is determined by the following equation:

$$\mathbf{x}'(t)^\top \mathtt{F} \mathbf{x}(t) = 0. \tag{6}$$

For a sample $\mathbf{s}_i$ in the first camera, we can rewrite (6) using the corresponding point $\mathbf{x}'(t_i)$ in the second camera as

$$\mathbf{x}'(t_i)^\top \mathtt{F} \mathbf{s}_i = 0, \tag{7}$$

Using the approximation of the trajectory $\mathbf{x}'$ by $\mathbf{s}'$, we can express the approximate epipolar constraint as

$$\mathbf{s}'(\beta + \rho i)^\top \mathtt{F} \mathbf{s}_i = 0. \tag{8}$$

In principle, we can solve for the unknowns β, ρ, and $\mathtt{F}$ given 9 correspondences s_i, s'_j. However, such a solution would be necessarily iterative and too slow to be used as a RANSAC kernel. In the following, another subsequent approximation is used to expresses the problem as a system of polynomials, which can be solved efficiently [18]. In §6 we show an iterative solution built on this kernel, which can recover offsets of up to hundreds of frames.

2.3. Linearization of $\mathbf{s}'$ for known ρ

Let us assume that the relative framerate ρ is known. In practice, the image curve $\mathbf{s}'$ is a complicated object. To arrive to our polynomial solution we approximate $\mathbf{s}'$ by the first order Taylor polynomial at $\beta_0 + \rho i$

$$\mathbf{s}'(\beta + \rho i) \approx \mathbf{s}'(\beta_0 + \rho i) + (\beta - \beta_0)\mathbf{v} = \mathbf{s}''(\beta + \rho i) \tag{9}$$

where $\mathbf{v}$ is the tangent vector $\dot{\mathbf{s}}'(\beta_0 + \rho i)$, and β_0 is an initial time shift estimate. We denote this approximation as $\mathbf{s}''$.

Further, we choose $\mathbf{v}$ to approximate the tangent over the next d samples. Let $j_0 = \lfloor \beta_0 + \rho i \rfloor$ be the approximate discrete correspondence, and then

$$\mathbf{v} = \mathbf{s}'_{j_0+d} - \mathbf{s}'_{j_0}. \tag{10}$$

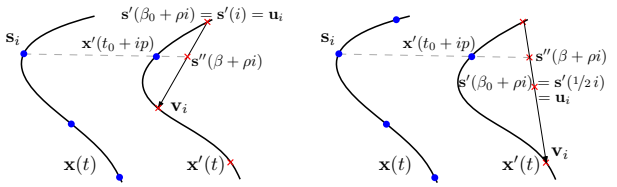

Figure 2. Illustration of the proposed trajectory linearization. (Left) Situation for $\rho = 1$, β_0 and $d = 1$ (Right) Situation for $\rho = 1/2$, $\beta_0 = 0$ and $d = 1$.

Note, that now $\mathbf{v}$ depends on i. For compactness, we write $\mathbf{u}_i = \mathbf{s}'(\beta_0 + \rho i) - \beta_0 \mathbf{v}_i$, and (8) becomes

$$(\mathbf{u}_i + \beta \mathbf{v}_i)^\top \mathtt{F} \mathbf{s}_i = 0 \tag{11}$$

In the rest of the paper, we will assume that $f = f'$ and the initial estimate $\beta_0 = 0$. This situation is illustrated in Figure 2 (Left). However the key results hold for general known ρ, Figure 2 (Right), and $\beta_0 \neq 0$.

2.4. Homography

Using the same approach, we can write the equation for homography between two unsynchronized cameras. In synchronized case, the homography between two cameras can be expressed as

$$\mathtt{H} \mathbf{s}_i = \lambda_i \mathbf{s}'_i. \tag{12}$$

Approximating the image motion locally by a straight line gives for two unsynchronized cameras

$$\mathtt{H} \mathbf{s}_i = \lambda_i \left(\mathbf{u}_i + \beta \mathbf{v}_i \right). \tag{13}$$

3. Solving the equations

3.1. Minimal solution to epipolar geometry

The minimal solution to the simultaneous estimation of the epipolar geometry and the unknown time shift β starts with the epipolar constraint (11). The fundamental matrix $\mathtt{F} = [f_{ij}]_{i,j=1}^{3}$ is a 3×3 singular matrix, *i.e.* it satisfies

$$\det(\mathtt{F}) = 0. \tag{14}$$

Therefore, the minimal number of samples $\mathbf{s}_i$ and $\mathbf{s}'_i$ necessary to solve this problem is eight.

For eight samples in general position in two cameras, the epipolar constraint (11) can be rewritten as

$$\mathtt{M} \mathbf{w} = \mathbf{0}, \tag{15}$$

where $\mathtt{M}$ is a 8×15 coefficient matrix of rank 8 and $\mathbf{w}$ is a vector of monomials $\mathbf{w} = [f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}, \beta f_{11}, \beta f_{12}, \beta f_{13}, \beta f_{21}, \beta f_{22}, \beta f_{23}]$. Since the fundamental matrix is only given up to scale, the monomial vector $\mathbf{w}$ can be parametrized using the 7-dimensional nullspace of the matrix $\mathtt{M}$ as

$$\mathbf{w} = \mathbf{n}_0 + \textstyle\sum_{i=1}^{6} \alpha_i \mathbf{n}_i, \tag{16}$$

where α_i, $i = 1, \ldots, 6$ are new unknowns and $\mathbf{n}_i$, $i = 0, \ldots, 6$ are the null space vectors of the coefficient matrix $\mathtt{M}$. The elements of the monomial vector $\mathbf{w}$ satisfy

$$\beta w_j = w_k, \tag{17}$$

where w_j is the j^{th} element of the monomial vector $\mathbf{w}$ and $j \in \{1, \ldots, 6\}$ and $k \in \{10, \ldots, 15\}$.

The parametrization (16) used in the rank constraint (14) and in the quadratic constraints (17) results in a quite complicated system of 7 polynomial equations in 7 unknowns $\alpha_1, \ldots, \alpha_6, \beta$. Therefore, we first simplify these equations by eliminating the unknown time shift β from these equations using the elimination ideal method presented in [19]. This results in a system of 18 equations in 6 unknowns $\alpha_1, \ldots, \alpha_6$. Even though this system contains more equations than the original system, its structure is less complicated. We solve this system using the automatic generator of Gröbner basis solvers [18]. The final Gröbner basis solver performs Gauss-Jordan elimination of a 194×210 matrix and the eigenvalue computations of a 16×16 matrix, since the problem has 16 solutions. Note that by simply applying [18] to the original system of 7 equations in 7 unknowns a huge and numerically unstable solver of size 633×649 is obtained.

3.2. Generalized eigenvalue solution to epipolar geometry

Using the non-minimal number of nine point correspondences, the epipolar constraint (11) can be rewritten as

$$(\mathtt{M}_1 + \beta \mathtt{M}_2)\mathbf{f} = \mathbf{0}, \tag{18}$$

where $\mathtt{M}_1$ and $\mathtt{M}_2$ are 9×9 coefficient matrices and $\mathbf{f}$ is a vector containing nine elements of the fundamental matrix $\mathtt{F}$

The formulation (18) is a generalized eigenvalue problem (GEP) for which efficient numerical algorithms are readily available. The eigenvalues of (18) give us solutions to β and the eigenvectors to fundamental matrix $\mathtt{F}$.

For this problem the rank of the matrix $\mathtt{M}_2$ is only six and three from nine eigenvalues of (18) are always zero. Therefore, instead of 9×9 we can solve only 6×6 GEP.

This generalized eigenvalue solution is more efficient than the minimal solution presented in section 3, however note that the GEP solution uses non-minimal number of nine point correspondences and the resulting fundamental matrix does not necessarily satisfy $\det(\mathtt{F}) = 0$.

3.3. Minimal solution to homography estimation

The minimal solution to the simultaneous estimation of the homography and the unknown time shift β starts with the equations of the form (13).

First, the solver eliminates the scalar values λ_i from (13). This is done by multiplying (13) by the skew symmetric matrix $[\mathbf{u}_i + \beta \mathbf{v}_i]_\times$. This leads to the matrix equation

$$[\mathbf{u}_i + \beta \mathbf{v}_i]_\times \mathtt{H}\mathbf{s}_i = \mathbf{0}. \tag{19}$$

The matrix equation (19) contains three polynomial equations from which only two are linearly independent, because the skew symmetric matrix has rank two. This means that we need at least 4.5 (5) samples in two images to estimate the unknown homography $\mathtt{H}$ as well as the time shift β.

Now let us use the equations corresponding to the first and second row of the matrix equation (19). In these equations β multiplies only the 3^{rd} row of the unknown homography matrix. This lead to nine homogeneous equations in 12 monomials $\mathbf{w} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}, \beta\, h_{31}, \beta\, h_{32}, \beta\, h_{33}]^\top$ for 4.5 samples in two images (i.e. we use only one equation from the three equations (19) for the 5th sample).

We can stack these nine equations into a matrix form $\mathtt{M}\,\mathbf{w} = \mathbf{0}$, where $\mathtt{M}$ is a 9×12 coefficient matrix. Assuming that $\mathtt{M}$ has full rank equal to nine, *i.e.*, we have non-degenerate samples, the dimension of $\mathrm{null}(\mathtt{M})$ is 3. This means that the monomial vector $\mathbf{w}$ can in general be rewritten as a linear combination of three null space basis vectors $\mathbf{n}_i$ of the matrix $\mathtt{M}$ as

$$\mathbf{w} = \textstyle\sum_{i=1}^{3} \gamma_i\, \mathbf{n}_i, \tag{20}$$

where γ_i are new unknowns. Without the loss of generality, we can set $\gamma_3 = 1$ to fix the scale of the homography and to bring down the number of unknowns. For 5 or more samples, instead of null space vectors $\mathbf{n}_i$, we use in (20) three right singular vectors corresponding to three smallest singular values of $\mathtt{M}$.

The elements of the monomial vector $\mathbf{w}$ are not independent. We can see that $w_{10} = \beta\, w_7, w_{11} = \beta\, w_8$, and $w_{12} = \beta\, w_9$, where w_i is the i^{th} element of the vector $\mathbf{w}$. These three constraints, together with the parametrization from equation (20) form a system of three quadratic equations in three unknowns γ_1, γ_2, and β and only 6 monomials. This system of three equations has a very simple structure and can be directly solved by performing G-J elimination of the 3×6 coefficient matrix $\mathtt{M}_1$ representing these tree polynomials, and then by computing eigenvalues of the 3×3 matrix obtained from this eliminated matrix $\mathtt{M}_1$. This problem results in up to three real solutions.

Note, that the problem of estimating homography and β can also be formulated as a generalized eigenvalue problem, similarly as the problem of estimating epipolar geometry (Section 3.2). However, due to the lack of space and the fact that the presented minimal solution is extremely efficient, we do not describe the GEP homography solution here.

4. Using RANSAC

In this section we would like to emphasize the role of RANSAC for our solvers. RANSAC is generally used for

robustness since the minimal solvers are sensitive to noise and outliers. Outliers in the data will usually come from two sources. One are the mismatches and misdetections and the other is the non-linearity of the point trajectory. Even without gross outliers due to false detections, there will always be outliers with respect to the model in places where the trajectory is not straight on the interpolating interval. Therefore, it is usually beneficial to use RANSAC even if we are sure the correspondences are precise.

By using RANSAC, we avoid those parts of the trajectory and pick the parts that are approximately straight and linear in velocity. Basically we only need to sample 8(F) or 5(H) parts of the trajectory where this assumption holds to obtain a good model, even if the rest of the trajectory is highly non-linear.

5. Performance of the solvers on synthetic data

First, we investigated the performance of estimating the time shift β using the proposed `F` and `H` minimal solvers. We simulated a random movement of a 3D point in front of two cameras. The simulated 3D trajectory was then sampled at different times in each camera, the difference being the ground truth time shift β_{gt}. Image noise was added from a normal distribution with $\sigma = 0.5$ px. We tested the minimal solvers with various interpolation distances d and compared them also to the standard seven point fundamental matrix (7pt-F) and four point homography (4pt-H) solvers [14]. Each algorithm was tested on 100 randomly generated scenes for each β_{gt}, resulting in tens of thousands of experiments.

There are multiple observations we can make from the results. The main one is that both `F` and `H` solvers perform well in terms of estimating β_{gt}, even for the minimal interpolation distance $d = 1$. Figure 3 shows that almost all inliers are correctly classified using $d = 1, d = 2, d = 4$ up to shift of 5 frames forward. Furthermore, even though the inlier ratio begins to decrease with larger shifts, time shift β is still correctly estimated, up till frame shifts of 20. Overall, for a given d, each algorithm was able to estimate correct β at least up to d. This is a nice property, suggesting that for larger time shifts we should be able to estimate them simply by increasing d.

For $d = 8, d = 16, d = 32$, the situation is slightly different with respect to inliers. Notice that there are two peaks in the number of inliers, one at $\beta_{gt} = 0$ and the other at $\beta_{gt} = d$. This is expected, because at $\beta_{gt} = d$, the interpolating vector $\mathbf{v}$ passes through the sample $\mathbf{s}'_{i+\beta_{gt}}$ which is in temporal correspondence with $\mathbf{s}_i$. When $\beta_{gt} \neq 0$ our solvers are for any d well above the number of inliers provided by standard `F` and `H` algorithms.

Another thing to notice is the non-symmetricity of the results. Obviously, when $\beta_{gt} < 0$ (backward) and we are interpolating with d-th (forward) sample, the peaks in inliers

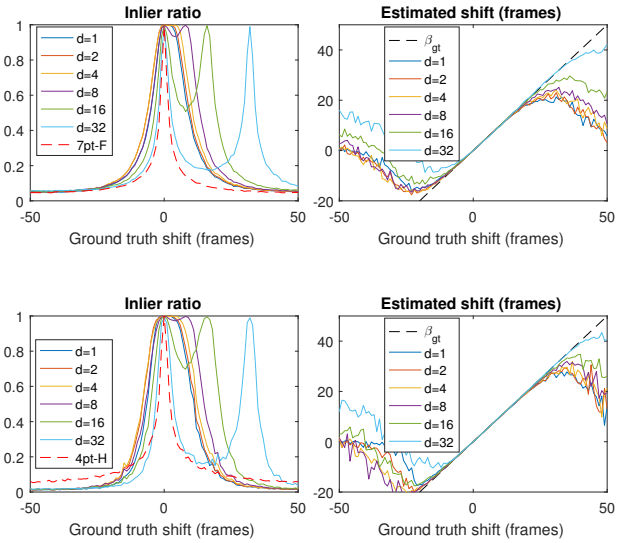

Figure 3. Results on randomly generated scene with various time shift β between cameras and several different interpolation distances d. Temporal distance of one frame equals to approximately 8 pixels distance in 1000x1000px image. Top two figures are results for epipolar matrix and bottom two for homography.

are not present, since we will never hit the sample which is in correspondence. Also, the performance in terms of inliers is reduced when interpolating in the wrong direction, although still above the algorithms not modelling the time shift. Estimation of β deteriorates significantly sooner for negative β_{gt}, at around -10 frames. We will show how to overcome this non-symmetricity by searching over d in both directions using an iterative algorithm.

6. Iterative algorithm

As we observed in the synthetic experiments, the performance of the minimal solvers will depend on the distance from the optimum, i.e. the distance between the initial estimate β_0 and the true time shift β_{gt}, and on the distance d of the samples used for interpolation. The results from synthetic experiments (Figure 3) provide useful hints on how to construct an iterative algorithm to improve the performance and applicability of the minimal solvers. In particular, there are three key observations to consider.

First, the number of inliers obtained from RANSAC seems to be a reasonable function to optimize. Generally it will have two strong local maxima, one at $t_i = t'_i$ and one at $(t_i - t'_i) = d$. At $t_i = t'_i$ the sequences are synchronized and at $(t_i - t'_i) = d$, Fig. 3, we obtain the correct β. Both situations give us synchronized sequences. Second, the β computed even far from the optimum, although not precise, provides often a good indicator of the direction towards $t_i = t'_i$. Finally, it can be observed that increasing d improves the estimates when we are far from the optimum. Moreover, as seen from the peaks in Fig. 3, selecting larger

d yields increasingly better estimates of β, which are lower or equal than the actual $(t_i - t'_i)$, but never higher. This suggests that we could safely increase d until a better estimate is found.

The observations mentioned above lead us to algorithm 1. The basic principle of the algorithm is the following. In the beginning, assume $i = j$. At each iteration k, estimate β and $\mathtt{F}$. If this model gives more inliers than previous estimate, change j to the nearest integer to $j + \beta$ and repeat. If the new estimate gives less inliers than the last one, extend the search direction by increasing d by powers of 2 until more inliers are found. If $d^{p_{max}}$ is reached, p is reset to 0, so interpolation distances keep circling between d^0 and $d^{p_{max}}$. This is essentially a line search over the parameter d. Algorithm is stopped when the number of inliers did not increase p_{max} times. This ensures, that at each t'_j, all interpolation distances are tested at maximum once. The resulting estimate of β is then $j - i + \beta$, which is the difference in frames the algorithm traveled plus the last estimate of time shift at this point (subframe synchronization).

Estimating of β and $\mathtt{F}$ is done using RANSAC and interpolating from both the next and previous d^{th} sample, searching the space of β in both directions. Whichever direction returns more inliers is taken as current estimate. By changing the values p_{min} and p_{max} we have the option to adjust the range of search. Having an initial guess about the amount of time shift, e.g. not more than 100 frames, but definitely more than 10 frames, we could start the algorithm with values $p_{min} = 3$ and $p_{max} = 7$ so the search in d would start with $d = 8$ and not go further than $d = 128$.

The symbol $\mathtt{T}$ represents a geometric relation, in our case either a fundamental matrix or a homography.

Algorithm 1 Iterative sync

Input: $\mathbf{s}_0, \ldots, \mathbf{s}_n$,$\mathbf{s}'_0, \ldots, \mathbf{s}'_{n'}$,$k_{max}$,$p_{max}$,$p_{min}$
Output: β,$\mathtt{T}$
  $\beta_0 \leftarrow 0$,$i = j$,*skipped* $\leftarrow 0$, $d \leftarrow 2^{p_{min}}$,*inliers*$_0 \leftarrow 0$,$p \leftarrow p_{min}$
  while $k = 1 < k_{max}$ **do**
    $\mathtt{T}_1$,β_1 and *inliers*$_1 \leftarrow$ RANSAC($\mathbf{s}_i, \mathbf{s}'_j, d$)
    $\mathtt{T}_2$,β_2 and *inliers*$_2 \leftarrow$ RANSAC($\mathbf{s}_i, \mathbf{s}'_j, -d$)
    if *inliers*$_1 >$ *inliers*$_2$ **then**
      inliers$_k \leftarrow$ *inliers*$_1$, $\beta_k \leftarrow \beta_1$, $\mathtt{T}_k \leftarrow \mathtt{T}_1$
    else
      inliers$_k \leftarrow$ *inliers*$_2$, $\beta_k \leftarrow \beta_2$, $\mathtt{T}_k \leftarrow \mathtt{T}_2$
    end if
    if *skipped* $> p_{max}$ **then**
      return $\mathtt{T}_{k-1}$,$\beta \leftarrow j - i + \beta_{k-1}$
    else if *inliers*$_k <$ *inliers*$_{k-1}$ **then**
      if $p < p_{max}$ **then**
        $p \leftarrow p + 1$
      else
        $p \leftarrow 0$
      end if
      $d \leftarrow 2^p$
      skipped $\leftarrow$ *skipped* $+ 1$
    else
      $j \leftarrow j + \lceil \beta_k \rfloor$
      skipped $\leftarrow 0$
      $k \leftarrow k + 1$
    end if
  end while

7. Real data experiments

Our real datasets contain two private datasets and three publicly available multi-camera datasets. We aimed at collecting various types of scenes to cover wide range of applications. The public data were always synchronized and we manually shifted the frame to frame correspondences to simulate the ground truth time shift. We experimented with shift of -50 to 50 frames on each dataset, which produced time shifts ranging from 2s to 5s based on the camera framerate.

7.1. Datasets

Dataset Marker was obtained by moving Aruco marker in front of a two webcams running at 10fps. A digital clock running was captured for each frame and processed by OCR to provide ground truth timestamps. Further, we used three public datasets and one private: UvA [17],KITTI [12], Hockey and PETS [8]. The UvA dataset consists of video sequences taken by three static cameras with manual annotations of humans. The KITTI dataset contains stereo video sequences taken from a moving car. In our experiments, we used raw unsynchronized data provided by the authors. The Hockey dataset was synchronized by [31] and the trajectories are manually curated tracks of [16]. The PETS dataset is a standard multi-target tracking dataset. Trajectories were detected by [13, 9, 27] and manually joined.

7.2. Algorithms

We compared seven different approaches to simultaneously solving two-camera geometry and time shift. Depending on the data, either fundamental matrix or homography was estimated. We denote both geometric relations by T, where T means $\mathtt{H}$ or $\mathtt{F}$ was estimated using standard 4 or 7 point algorithms [14] and T_β means that $\mathtt{H}$ or $\mathtt{F}$ was estimated together with β. The rightmost column of figure 4 shows which model, i.e. homography or fundamental matrix, was estimated on a particular data set.

The closest alternatives to our approach are the least-squares based algorithms presented in [21] and [20]. Both optimize $\mathtt{F}$ or $\mathtt{H}$ and β starting from an initial estimate of $\beta = 0$ and T. Method [21] uses linear interpolation from the next sample, whereas method [20] uses spline interpolation of the image trajectory and we will refer to these methods as T_β-lin and T_β-spl respectively. In our implementation of those methods, we used Matlab's lsqnonlin function with Levenberg-Marquardt algorithm, all stopping criteria set to epsilon and maximum number of 100 iterations.

We tested the solvers presented in section 3 with $d = 1$ as algorithm T_β-new-d1. The proposed iterative algorithm 1

β_{gt}	0-10	10-20	20-30	30-40	40-50
T_β-new-iter-pmax0	4.7	4.3	3.5	4.1	3.8
T_β-new-iter-pmax6	23	22	21.2	21.6	21.2
T_β-new-iter-pmaxvar	18	19	17.5	16.7	16.5

Table 1. Average number of RANSACs executed before termination. Evaluated on Marker dataset.

that uses the solvers was tested using several different settings. The user can control the algorithm using parameters p_{max} and p_{min}, which determine the distances d that will be used for interpolation. As we observed in section 5, there is a good chance of computing a correct β if $d > \beta_{gt}$. First, we ran the algorithm with $p_{min} = 0$ and $p_{max} = 6$, which gives maximum $d = 64$ as algorithm T_β-new-iter-pmax6. This version of the algorithm is guaranteed to try $d = 1, 2, 4, 8, 16, 32, 64$ at each β_k before it stops or it finds more inliers. This covers the time shifts we tested, but can lead to unnecessary iterations for smaller shifts. Therefore, we also tested $p_{max} = 0$ as T_β-new-iter-pmax0 which only tried $d = 1$ at each iteration to see the capabilities of the most efficient version of the algorithm.

The last version of our algorithm, T_β-new-iter-pmaxvar, adapted both p_{max} and p_{min} to β_{gt} such that $2^{p_{min}} \leq \beta_{gt} < 2^{p_{max}}$. This represents a case when user has a rough estimate about the expected time shift and sets the algorithm accordingly. We remind that setting p_{min} only affects the initial interpolating distance, after reaching $d = 2^{p_{max}}$ the algorithm starts again with $d = 2^0$.

Finally, algorithm T-lin [21] also takes the next samples for interpolation, making it comparable to our T_β-new-d1. We used T-lin in the same iterative scheme as T_β-new-iter-pmax6 and tested it as T-new-lin-iter, where instead of using the number of inliers as a criteria for accepting a step, we used the value of the residual.

7.3. Results and discussion

The results on real datasets demonstrate a wide practical usefulness of the proposed methods. For most datasets, T_β-new-d1 itself performed at least as good as the least squares algorithms T_β-lin and $T_\beta - spl$. A single RANSAC was enough to synchronize time shifts of 2-5 frames across all datasets. The iterative algorithm T_β-new-iter-pmax6 build upon our solvers performed the absolute best across all datasets, converging successfully from as far as 5s time difference on Marker and Hockey datasets, 2s difference on UvA dataset and 2,5 seconds on Kitti dataset as seen in the success rate column of figure 4.

On the Kitti dataset, T_β-new-iter-pmax6 was outperformed by the T_β-new-lin-iter, which uses the iterative algorithm proposed by us, but with a solution from [21] inside. T_β-new-lin-iter was able to estimate time differences larger than 2,5s but only in roughly half of the cases, where T_β-new-iter-pmax6 was 100% successful up to 2,5s when it sharply fell off. We account this to the high non-linearity of the 2D velocity of the image points, where as the objects got closer to the car, they moved faster. The tracks of length 25 frames and more were very sparse here and the longer they were the more non-linear in the velocity.

On the contrary, the hockey dataset posed a big challenge for the least squares algorithms, which struggled even with the smallest time offsets. We account this to the poor estimate of F by the seven point algorithm which causes the LM algorithm to get stuck in local minima. We also tested the homography version of all algorithms on this dataset, since the trajectories are approximately planar, which resulted in the least squares algorithms performing slightly better whereas the algorithms with minimal solvers performing slightly worse.

PETS dataset was probably the most challenging, because of the low framerate (7FPS), coarse detections and abrupt change of motion. Still, our methods managed to synchronize the sequences in majority of cases.

Table 1 shows the average number of RANSACs used before termination of different variants of iterative algorithm 1 for the dataset Marker. We can see that using 8pt-iter-pmax0 greatly reduces the computations needed, still allowing this method to reliably estimate time shifts of 0.5s-2s depending on the scene, rendering it useful if we are certain that the sequences are off by only a several frames. Knowing the time shift approximately and setting p_{max} and p_{min} can also reduce the computations as shown by 8pt-iter-pmaxvar, which provided identical performance to 8pt-iter-pmax6, sometimes even outperforming it.

8. Conclusion

We have presented solvers for simultaneously estimating epipolar geometry or homography and time shift between image sequences from unsynchronized cameras. These are the first minimal solutions to these problems, making them suitable for robust estimation using RANSAC. Our methods need only trajectories of moving points in images, which are easily provided by state-of-the-art methods, e.g. SIFT matching, human pose detectors, or pedestrian trackers. We were able to synchronize wide range of real world datasets shifted by several frames using a single RANSAC with our solvers. For larger time shifts, we proposed an iterative algorithm using these solvers in succession. The iterative algorithm proved to be reliable enough for synchronizing real world camera setups ranging from autonomous cars, surveillance videos, and sport game recordings, which were de-synchronized by several seconds.

Acknowledgement

This work was partly done during an internship of C. Albl and a postdoc position of Z. Kukelova at Microsoft Re-

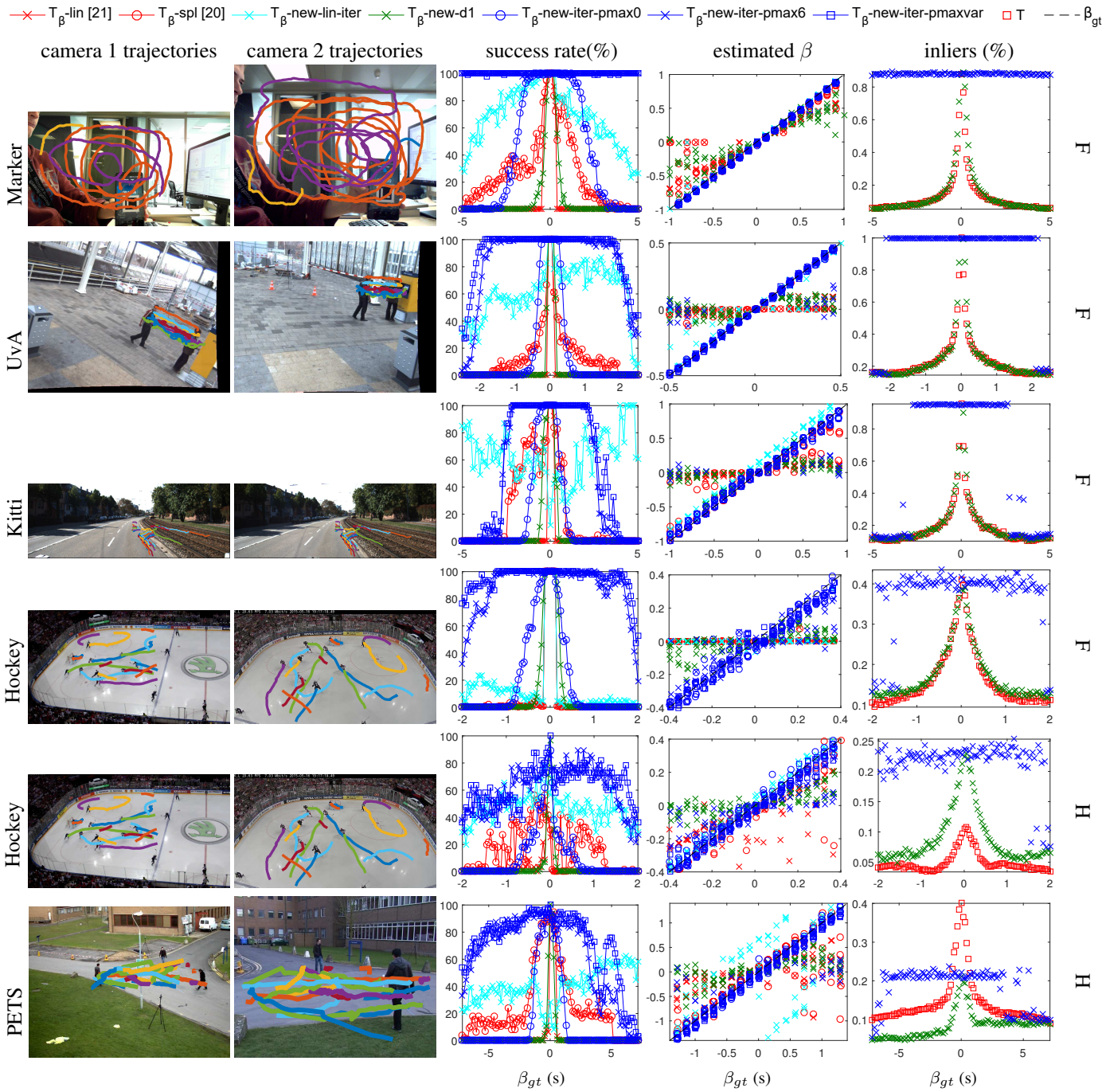

Figure 4. Results on real data. In the two leftmost columns, trajectories used for the computations are depicted in coloured lines over a sample images from the dataset. Third column shows the rates with which different algorithms succeeded to synchronize the sequence to single frame precision for various ground truth time shifts. Fourth column shows a closer look at the individual results for β for smaller ground truth time shifts and five runs for each algorithm, each data point corresponds to one run of the algorithm at corresponding β_{gt}. Letters H and F on the right signalize whether homography or fundamental matrix was computed.

search Cambridge and was supported by the EU-H2020 project LADIO (number 731970), The Czech Science Foundation Project GACR P103/12/G084, Grant Agency of the CTU Prague projects SGS16/230/OHK3/3T/13, SGS17/185/OHK3/3T/13 and by SCCH GmbH under Project 830/8301544C000/13162.

References

[1] G. Carrera, A. Angeli, and A. J. Davison. Lightweight slam and navigation with a multi-camera rig. In *ECMR*, pages 77–82, 2011. 1

[2] Y. Caspi, M. Irani, and M. I. Yaron Caspi. Spatio-temporal alignment of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, 2002. 1, 2

[3] Cheng Lei and Yee-Hong Yang. Tri-focal tensor-based multiple video synchronization with subframe optimization. *IEEE Transactions on Image Processing*, 15(9):2473–2480, sep 2006. 1

[4] C. Dai, Y. Zheng, and X. Li. Subframe video synchronization via 3D phase correlation. In *International Conference on Image Processing*, pages 501–504, 2006. 1

[5] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, 2005. 1

[6] A. Elhayek, C. Stoll, N. Hasler, K. I. Kim, H.-P. Seidel, and C. Theobalt. Spatio-temporal motion tracking with unsynchronized cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1870–1877. 1

[7] A. Elhayek, C. Stoll, K. I. Kim, H. P. Seidel, and C. Theobalt. Feature-based multi-video synchronization with subframe accuracy. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7476 LNCS:266–275, 2012. 1, 2

[8] A. Ellis, A. Shahrokni, and J. Ferryman. PETS 2009 Benchmark Data, 2009. 1, 6

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, sep 2010. 6

[10] M. a. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with. *Communications of the ACM*, 24:381–395, 1981. 1

[11] J.-M. Frahm, K. Köser, and R. Koch. Pose estimation for multi-camera systems. In *Joint Pattern Recognition Symposium*, pages 286–293. Springer, 2004. 1

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 1, 6

[13] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively Trained Deformable Part Models, Release 5. http://people.cs.uchicago.edu/∼rbg/latent-release5/. 6

[14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press. 2, 5, 6

[15] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 224–231. IEEE, 2009. 1

[16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7575 LNCS(PART 4):702–715, 2012. 6

[17] M. Hofmann and D. M. Gavrila. Multi-view 3D human pose estimation combining single-frame recovery, temporal integration and model adaptation. In *CVPR*, pages 2214–2221, 2009. 6

[18] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *ECCV, Part III*, volume 5304 of *Lecture Notes in Computer Science*, 2008. 3, 4

[19] Z. Kukelova, J. Kileel, B. Sturmfels, and T. Pajdla. A clever elimination strategy for efficient minimal solvers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. http://arxiv.org/abs/1703.05289. 4

[20] M. Nischt and R. Swaminathan. Self-calibration of asynchronized camera networks. In *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2164–2171, Sept. 2009. 2, 6

[21] M. Noguchi and T. Kato. Geometric and Timing Calibration for Unsynchronized Cameras Using Trajectories of a Moving Marker. In *IEEE Workshop on Applications of Computer VisionWACV*, pages 20–20, 2007. 2, 6, 7

[22] F. L. C. Padua, R. L. Carceroni, G. Santos, and K. N. Kutulakos. Linear Sequence-to-Sequence Alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):304–320, 2010. 1, 2

[23] D. Pundik and Y. Moses. Video synchronization using temporal signals from epipolar lines. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6313 LNCS(PART 3):15–28, 2010. 1

[24] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood. View-invariant alignment and matching of video sequences. In *IEEE International Conference on Computer Vision*, pages 939–945 vol.2, 2003. 1

[25] G. Schroth, F. Schweiger, M. Eichhorn, E. Steinbach, M. Fahrmair, and W. Kellerer. Video synchronization using bit rate profiles. In *International Conference on Image Processing*, pages 1549–1552, 2010. 1

[26] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006. 1

[27] X. Shi, Z. Yang, and J. Chen. Modified Joint Probabilistic Data Association. In *IEEE International Conference on Computer Vision*, pages 6615–6620, 2015. 6

[28] P. Shrestha, H. Weda, M. Barbieri, and D. Sekulovski. Synchronization of multiple video recordings based on still camera flashes. In *International Conference on Multimedia*, volume 2, page 137, 2006. 1

[29] P. Shrstha, M. Barbieri, and H. Weda. Synchronization of multi-camera video recordings based on audio. In *International Conference on Multimedia*, page 545, New York, New York, USA, 2007. ACM Press. 1

[30] S. Singh, S. A. Velastin, and H. Ragheb. MuHAVi: A multicamera human action video dataset for the evaluation of action recognition methods. In *2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 48–55. 1

[31] M. Šmíd and J. Matas. Rolling Shutter Camera Synchronization with Sub-millisecond Accuracy. In *Proc. 12th Int. Conf. Comput. Vis. Theory Appl.*, page 8, 2017. 6

[32] G. Stein. Tracking from multiple view points: Self-calibration of space and time. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 521–527. IEEE Computer Society, 1999. 1, 2

[33] P. A. Tresadern and I. D. Reid. Video synchronization from human motion using rank constraints. *Computer Vision and Image Understanding*, 113(8):891–906, 2009. 2

[34] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, number 1883 in Lecture Notes in Computer Science, pages 298–372. Springer Berlin Heidelberg. DOI: 10.1007/3-540-44480-7_21 bibtex: triggs_bundle_1999. 2

[35] T. Tuytelaars and L. Van Gool. Synchronizing video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 762–768, 2004. 1

[36] J. Zhou and D. Wang. Solving the perspective-three-point problem using comprehensive grbner systems. pages 1–27. 1