Subframe Temporal Alignment of Non-Stationary Cameras

B. Meyer, T. Stich, M. Magnor and M. Pollefeys
Computer Graphics Lab at TU Braunschweig, Germany
Department of Computer Science, ETH Zurich, Switzerland

Abstract

This paper studies the problem of estimating the sub-frame temporal offset between unsynchronised, non-stationary cameras. Based on motion trajectory correspondences, the estimation is done in two steps. First, we propose an algorithm to robustly estimate the frame accurate offset by analysing the trajectories and matching their characteristic time patterns. Using this result, we then show how the estimation of the fundamental matrix between two cameras can be reformulated to yield the sub-frame accurate offset from nine correspondences. We verify the robustness and performance of our approach on synthetic data as well as on real video sequences.

1 Introduction

In this work we present a method to find the subframe-accurate time offset between two or more recorded video sequences recorded by unsynchronised, non-stationary cameras. We address the problem of identifying the time relation between recorded sequences without the need to invade the scene to use special cameras. Main application of the method is multi-view video acquisition. Most multi-view processing algorithms rely on the assumption that the video sequences are temporally synchronised e. g. stereo vision, visual hull estimation and viewpoint interpolation algorithms. Synchronicity can be achieved by hardware synchronisation of the recording cameras. While this is feasible for laboratory or studio situations, it reduces the applicability of these methods in outdoor environments. Computing the subframe-accurate time offset between unsynchronised non-stationary cameras is necessary to apply multi-video algorithms to a wider range of scenes.

Our method is based on tracking feature points and the resulting trajectories. It is divided into two steps. First, we find the time offset up to per-frame accuracy by extracting salient points of trajectories and matching their time patterns. This is possible already with only one single trajectory. Camera viewing angle differences of up to 90 degrees can be handled, as long as the tracked feature points are visible in both sequences. Using this coarse alignment, we can reformulate the estimation of the fundamental matrix to directly find the time offset of the non-stationary cameras to subframe accuracy. The remainder of the paper is organised as follows: The next section summarises previous work. Section 3 formalises the problem, in Section 4 our approach to compute the per-frame accurate time shift is presented. In Section 5 we describe how sub-frame accuracy is achieved. Experiments and results on synthetic data with ground truth and real world sequences are presented in Section 6.

BMVC 2008 doi:10.5244/C.22.11

2 Related Work

Over the last years, the problem of finding the temporal offset between multiple video sequences of one scene recorded with unsynchronised video cameras has been addressed by many researchers. The proposed approaches can be roughly classified in two categories depending on the goal of achieving frame or sub-frame accuracy.

[1] and [2] are examples for algorithms limited to the detection of the integer frame offset between unsynchronised cameras. In [1] points with spatio-temporal variations are detected in the video sequences and are described as a temporal distribution. As the dynamics of the scene are caught in both cameras, their distributions are similar and the temporal difference between the sequences can be calculated through an alignment. In [2] the movement of the objects is analysed and compared, which allows to synchronise camera streams of different scenes as long as they have the same dynamic.

In contrast, in [3] and [4] sub-frame accuracy is achieved by calculating the fundamental matrix of the cameras and by evaluating it afterwards on the basis of correspondences between trajectories of moving objects. While in [3] it is proposed to use three cameras and to calculate the trifocal tensor, in [4] the dynamic properties of trajectories from two video sequences are exploited. In [5] RANSAC-based approaches are extended to recover either a homography or fundamental matrix from putative matched spatial features in two images. The matches are then used to compute the frame offset. These approaches are however limited to stationary or jointly moving cameras. In [6] an iterative algorithm is presented using 3D phase correlation based on a projective geometry constraint. For this purpose, the simplified assumption is made that the centres of the cameras are comparative close to each other and, therefore, parallax can be neglected. In [7] an approach of calculating the epipolar geometry of dynamic silhouettes for temporally calibrating camera networks is discussed. In [8] a self-calibration method for two affine cameras is presented using a rank constraint framework for estimating the temporal shift between the video sequences.

For dealing with independently moving cameras, in [9] an iterative approach is presented based on a cost function. In [10] the rigidity of a set of points is evaluated by backprojecting the points as lines into the 3D world and looking for intersecting lines. This backprojection is also calculated in [11] for a set of three cameras. The problem of video synchronisation is then reformulated as a geometric alignment problem and a subframe accurate solution is obtained applying a Levenberg-Marquardt-based optimisation.

3 Problem Formulation

Let $\vec{p}_t = (x_t, y_t,)$ and $\vec{p}'_\tau = (x'_\tau, y'_\tau)$ be two spatio-temporal points in the input video sequences S and S' respectively, where S denotes the reference sequence and S' the second sequence. Let further (x_t/y_t) and (x'_τ/y'_τ) be pixel coordinates of the images taken at the reference time t of the reference camera and the intern time τ of the second camera. Assuming that $\vec{p}_t$ and $\vec{p}'_\tau$ are a corresponding point pair, the temporal misalignment is given as Δt with

$$t + \Delta t = \tau. \qquad (1)$$

The relative camera positions are unknown, and changing over time as the cameras are allowed to move separately, but can be described at the time t using the fundamental matrix F_t . Applying (1) to the definition of the fundamental matrix yields

$$(\vec{p}'_\tau)^T \ F_t \ \vec{p}_t = 0 \tag{2}$$

where $\vec{p}$ and $\vec{p}'$ are a corresponding point pair and p'^T is the transposition of p'.

While single image matching contain no information of the temporal shift, tracking points over multiple frames and reconstructing their moving trajectories allows to compare point correspondences on a temporal aspect. Hence let $T_{\vec{p}} = \{\vec{p}_t, \vec{p}_{t+1}, ..., \vec{p}_{t+k}\}$ be the trajectory resulting in tracking $\vec{p}_t$ over k frames. In this paper we assume these trajectories as well as their correspondences to be known. They can be obtained using standard algorithms as the Kanade-Lucas-Tomasi feature tracker ([12] and [13]). Further we expect the camera motion to be smooth and slow compared to the object movement and the recorded scene is supposed to contain linear and non-linear object motion.

4 Frame-accurate Temporal Alignment

Our first step in achieving an exact estimation of the temporal offset between two video sequences is to estimate the integer time shift from correspondences between motion trajectories. Since we are recording the same dynamic scene with both cameras, we expect to catch the same movement in both video sequences. But rather than matching the trajectories directly, which is hard to achieve if the cameras are moving independently, we focus only on interest points extracted from the trajectories. Our goal is to find the interest points that represent best the temporal information from these trajectories while achieving maximal view-independence. The frame-accurate temporal misalignment can then be robustly estimated by finding the best alignment between these representations.

The extraction of the interest points needs to be robust and largely view-independent, as we want in both camera sequences the same points to be extracted. Hence a naive approach, e. g. choosing the points depending on time derivatives, is problematic as such values of corresponding trajectories will grow very dissimilar with increasing view angle differences and different camera motion. From our analysis of these trajectories we concluded that the points located at extremal points on the trajectories represent motion cues that can be found in the other sequences as well. Especially, points that build the basis for a linear approximation of a motion trajectory fulfil this criterion. Interestingly, we can resort to an algorithm from a different domain to find these points. The recursive Douglas-Peucker-Algorithm [14] provides a robust simplification of vector lines which is used to scale down coastlines in geographic maps. Applied to trajectories Algorithm 1 results in the points we are interested in as can be seen in Figure 1. Here the motion trajectory of a bouncing ball is reduced to a linear approximation, yielding the interest points at the extremal positions. Depending on a scale parameter ε different degrees of simplification can be achieved.

Since we are interested in the temporal offset we do not need the spatial position of the interest points and even want to drop this information to increase view independence. Thus, we represent the trajectories as binary codes allocating every found motion interest point on the trajectory with a *one* and the remaining with a *zero* (cf. Figure 2). These representations of the trajectories can then easily be matched using common robust string

Algorithm 1 Douglas-Peucker-Algorithm for extracting motion interest points

```
Require: Trajectory $T$, scale parameter $\varepsilon$.
1: Connect the first and the last point ($\vec{p}_{start}$ and $\vec{p}_{end}$) of $T$ with a straight line $l$.
2: Determine the point $\vec{p}_{max}$ of $T$ with the highest distance $d$ to $l$.
3: If $d > \varepsilon$ recursive start the algorithm with the partial trajectories $\{\vec{p}_{start}...\vec{p}_{max}\}$ and
   $\{\vec{p}_{max}...\vec{p}_{end}\}$. Else $\vec{p}_{start}$ and $\vec{p}_{end}$ are motion interest points.
```

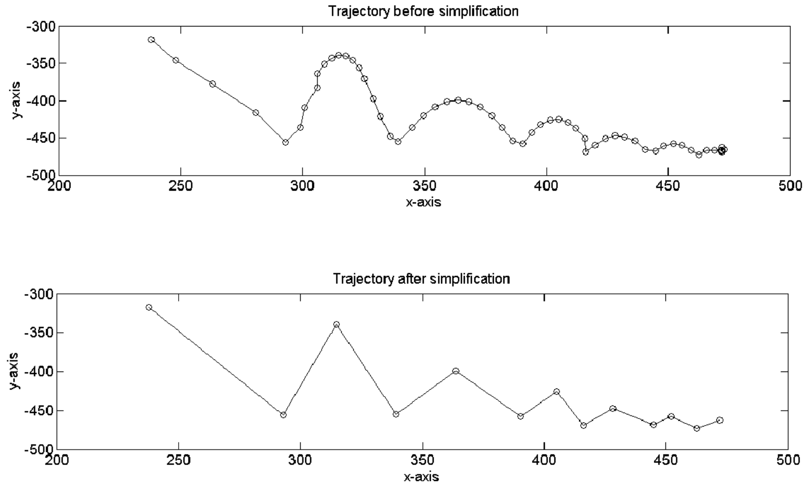


Figure 1: 2D motion trajectory of a bouncing ball from a single camera. The extraction of interest points on the trajectory can be computed using the Douglas-Peucker-Algorithm.

alignment algorithms. We implemented a string alignment algorithm based on a simplified Needleman-Wunsch-Algorithm [15] to compute this match (cf. Algorithm 2).

Already a single trajectory contains enough information to compute the frame accurate solution. If in practise the tracking results are not reliable, one can use more trajectories and apply a RANSAC [16] approach to eliminate outliers.

5 Achieving subframe accuracy

Using the previously computed frame-accurate temporal offset, corresponding trajectories can already be roughly synchronised. But as they are recorded with uncalibrated cameras, there is in general also a subframe offset. Let T and T' be such a pair of unsynchronised corresponding trajectories with no integer time shift, then we assume $\vec{p}_t$ and $\vec{p}_{t+1}$ of T to be the neighbouring points in time to $\vec{p}'_t$ on T' and so T to be slightly ahead to T'. Further, we assume that a point $\vec{p}_{t+a}$ for $0 \leq a < 1$ can be sufficiently accurately approximated by

$$\vec{p}_{t+a} = (1-a)\,\vec{p}_t + a\,\vec{p}_{t+1}. \tag{3}$$

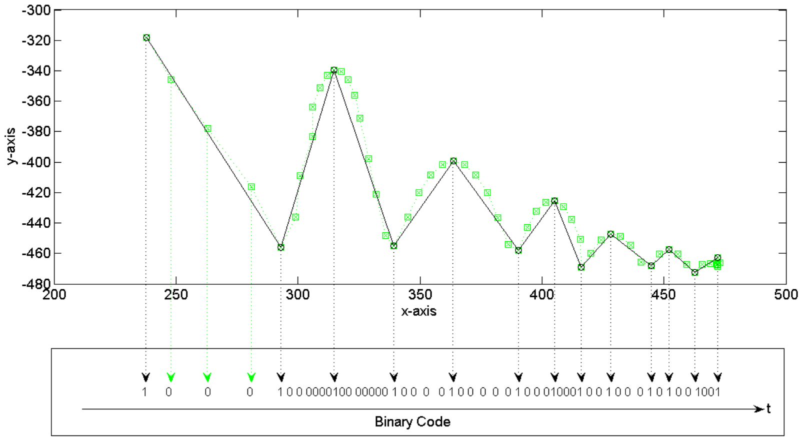


Figure 2: Trajectory representation as a binary code, with a *one* representing an extracted interest point and a *zero* for the remaining points.

Algorithm 2 String Alignment Algorithm

```
Require: 2 binary codes $B,C$ with $|B| = m$ and $|C| = n$.
1: Construct a $m \cdot n$ table T as follows: $T(i,j) = B(i)\ xor\ C(j)$ (with $xor$ the binary exclusive or).
2: Assign to every diagonal its sum divided by its length as an error measure.
3: Let $(i_0, j_0)$ be the first element of the diagonal with the lowest error value, the time shift between the cameras is given as $i_0 - j_0$ (depending on which camera is ahead, this can be a positive or negative value).
4: As for increasing $i_0$ (or $j_0$, one of them has always to be zero) the length of the diagonals is decreasing avoiding accurate results, it is required to assure a minimal length of the diagonals (which means a minimal temporal overlap of the two sequences).
```

Note, that the motion vector $\vec{p}_{t+1} - \vec{p}_t$ can be a combination of linear object and linear camera motion. Substituting $\vec{p}_{t+a}$ for $\vec{p}$ in (2) yields

$$\vec{p}_t^{\prime T}\ F_t\ ((1-a)\vec{p}_t + a\vec{p}_{t+1}) = 0 \tag{4}$$

with $0 \leq a < 1$ and F_t an unknown fundamental matrix.

Thus, one pair of corresponding trajectories yields one equation at the time t. Extracting nine independent motion trajectories from each camera sequence allows to generate a system of equations and to estimate the temporal shift directly. At the time t, (4) can then be reformulated using f_t, the corresponding vector containing the nine unknown entries of F_t in descending order, to

$$M(a)f_t = 0 \tag{5}$$

with $M(a)$ a 9×9 matrix in only a single free variable a. As for the correct temporal offset there must exist a solution to the above equation, the following constraint needs to be satisfied

$$det(M(a)) = 0 \tag{6}$$

which corresponds to a degree six polynomial in a (as only the first two coordinates of $\vec{p}_t$ are a function of a). For the case of no motion or only camera motion, there is no unique solution for a since the equation holds for all $a \in [0 \ldots 1)$. The solution for this will be unique if the trajectories result of at least two independent motion. This is for example the case for a static background and a moving foreground, two different objects with independent motion or a non-linear deforming object. Hence solving (5) for a provides a direct estimation of Δt in one single step under the assumption that object and camera motion can be linearly approximated between two consecutive frames. As the time shift Δt is assumed to be constant over the whole video sequences, this estimation can be repeated for different t.

In practise due to noise and tracking errors, the solutions are more robust the more independent the motions are. To measure this independence we calculate the fundamental matrix *within* the considered two steps of each camera. The larger the residual error the better this independence assumption will be fulfilled. Repeating this approach for these time points results in a distribution over $[0...1)$ of possible time shifts where the real time shift is expected to be the mean value. For the case that more than one solution for a lies in the requested interval $[0 \ldots 1)$, or none at all, the iteration will be dropped. If most of the iterations yield no result in the given interval, it indicates that our first assumption of T being temporally ahead is wrong. Repeating the calculation for $\vec{p}'_t$ and $\vec{p}'_{t+1}$ as the neighbouring points in time of $\vec{p}_t$ will then result in the correct time shift.

6 Results

We have applied our synchronisation method to various video sequences, both synthetic and real camera data to demonstrate the robustness and applicability of this approach.

For generating synthetic data we used 3-dimensional trajectories of tracked markers on moving people provided by the Carnegie Mellon University [17] and then calculated the perspective projection for different camera views. This enabled us to freely specify camera positions and orientations as well as the exact time shift and thus to compare the maintained results with ground truth. As these 3-dimensional trajectories were also measured from real world scenes, the received data is not free of noise and, for a nonintegral temporal offset between the generated views, also the projected 2-dimensional trajectories are not noise free.

As expected, the robustness of the results depends on the length of the considered trajectory as well as on the baseline of the generated views: a longer trajectory provides more information simplifying the computation whereas a wider baseline hinders finding the correct alignment. Figure 3 illustrates this relation using only one trajectory for evaluation: in the green area our algorithm for finding the frame-accurate offset provides a correct estimation and fails in the red one. In these test cases we chose ε in such a way that no more than fifty and not less that five percent of the points in the trajectory were considered as points of interest. Summarising the results, one can expect to find the correct solution for image sequences with a length of 100 or more frames for baseline differences up to 45 degrees and even up to 90 degrees for longer trajectories.

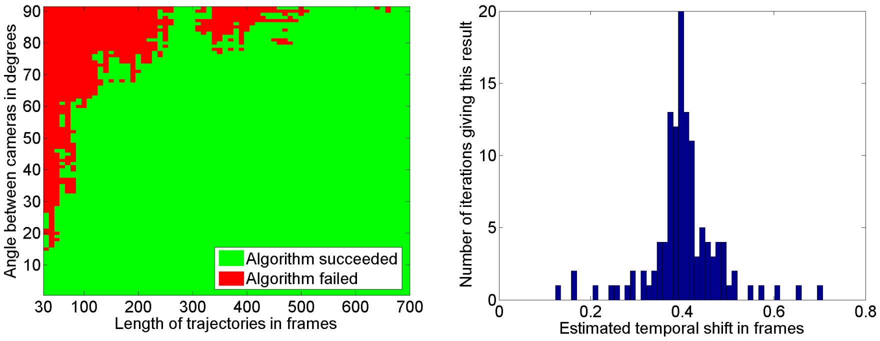


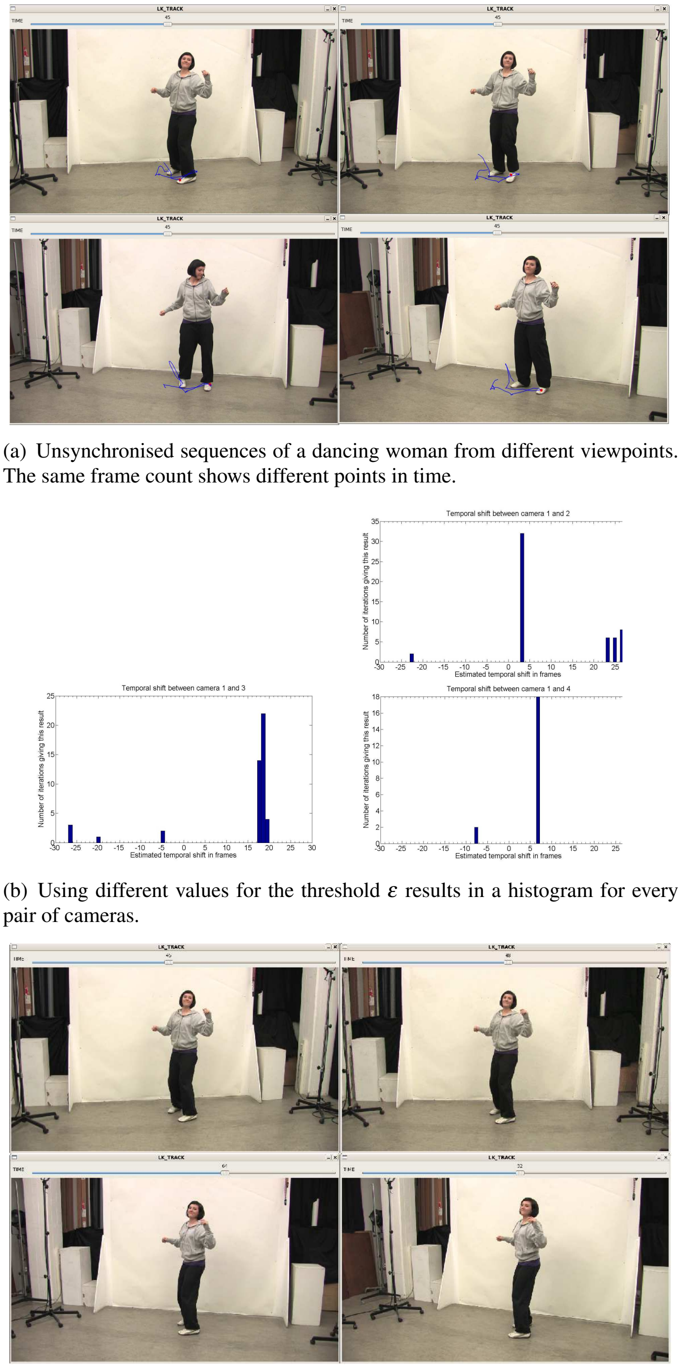
Figure 3: Left: the coherence between trajectory length and camera baselines. Right: histogram of the estimated possible subframe-accurate time shifts with a distinct maximum at 0.40, in accordance with ground truth.

For an evaluation of the subframe accurate temporal difference we chose an angle of 20 degrees between the cameras and a time shift of 0.4 frames. The results can be seen in Figure 3. The median of the distribution of possible time shifts was 0.3999 and the mean value was 0.4031 with a variance of 0.0062, both according to ground truth.

In our first experiment with real data we recorded a dancing woman with a set of four stationary, but uncalibrated, cameras. The cameras were placed around the scene next to each other with an angle of 15 degrees to their respective neighbours (so the angle between the first and the last camera were 45 degrees) as can be observed in Figure 4(a). The utilised trajectories were obtained by using a pyramid Lucas-Kanade feature tracker [12] and had a length of 100 frames each. We only evaluated a pair of two cameras at once but for different values for ε changing the amount of points being extracted. The results are shown in the histograms in Figure 4(b). In Figure 4(c) four frames with the correct temporal shift are shown. Computing the offsets for all pairs of cameras, we found that the results are consistent as the sum of all offsets is zero as can be seen in Figure 5.

Our second experiment is an outdoor scene with two moving, hand held cameras recording a trial biker. The sequences we used for synchronisation were of 100 frames length. We used only one pair of corresponding trajectories. The obtained time shift of 6 frames is according to ground truth, as can be seen in Figure 6, and shows that our algorithm is unaffected of the camera movement.

For testing the algorithm on subframe accuracy we recorded with one moving hand held and one stationary camera, a single throw of a ball as here the ground truth can at least roughly be estimated. The manual estimated time difference was 0.8 frames. The motion of the ball provided one trajectory. The remaining eight trajectories were generated by tracking background points. The obtained derivation of possible time shifts had its peak at 0.7681, conforming to the estimated ground truth, and a variance of only 0.0019 frames and can be therefore regarded as stable. Notice that the variance was even lower than the variance using our synthetic data as the synthetic data contained less linear motion. The trajectories used for this estimation were of 89 frames length but only 10 frames fulfilled the additional robustness constraint as discussed in Section 5.



(a) Unsynchronised sequences of a dancing woman from different viewpoints. The same frame count shows different points in time.

(b) Using different values for the threshold ε results in a histogram for every pair of cameras.

(c) Using the estimated offsets the videos are correctly synchronised.

Figure 4: Frame-accurate temporal alignment of four camera sequences of a dancing woman.

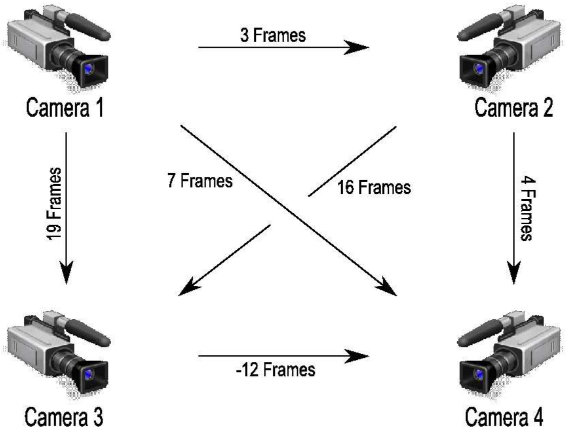


Figure 5: Computing the temporal shift for every pair of cameras yields a graph showing the relationships between the cameras. Note that the sum of every loop in this graph must be zero.

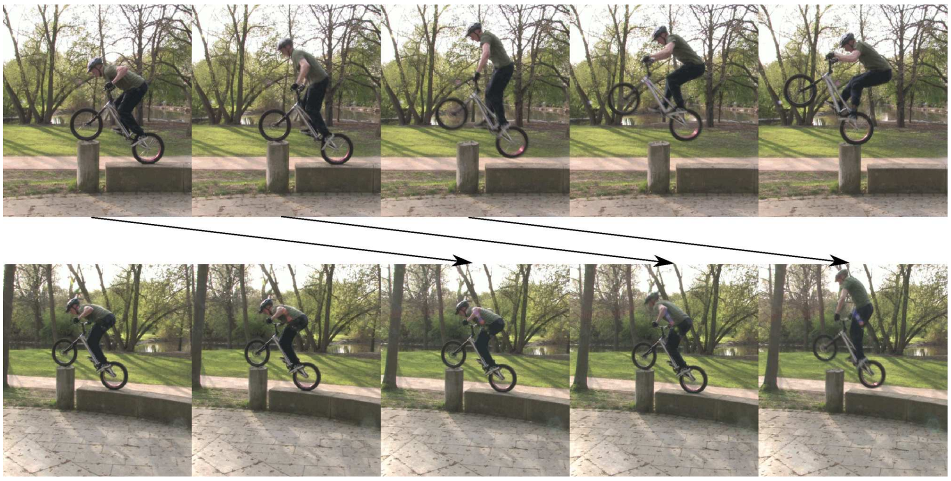


Figure 6: Beginning at frame 78, every third image of both video sequences is displayed. The sequences show a time delay of 6 frames.

References

[1] Jingyu Yan and Marc Pollefeys. Video synchronization via space-time interest point distribution. In *Advanced Concepts for Intelligent Vision Systems*, pages 501–504, 2004.

[2] Lisa Spencer and Mubarak Shah. Temporal synchronization from camera motion. In *Proceedings of Asian Conference on Computer Vision*, pages 515–520, 2004.

[3] Anthony Whitehead, Robert Laganiere, and Prosenjit Bose. Temporal synchronization of video sequences in theory and in practice. *Motion and Video Computing*,

2:132–137, 2005.

[4] Yaron Caspi, Denis Simakov, and Michal Irani. Feature-based sequence-to-sequence matching. *International Journal of Computer Vision*, 68(1):53–64, 2006.

[5] Daniel Wedge, Du Huynh, and Peter Kovesi. Using space-time interest points for video sequence synchronization. In *IAPR Conference on Machine Vision Applications*, pages 190–194, 2007.

[6] Congxia Dai, Yunfei Zheng, and Xin Li. Subframe video synchronization via 3d phase correlation. In *Proceedings of the International Conference on Image Processing*, pages 501–504, 2006.

[7] Sudipta Sinha and Marc Pollefeys. Synchronization and calibration of camera networks from silhouettes. *17th International Conference on Pattern Recognition (ICPR'04)*, 1:116–119, 2004.

[8] P. Tresadern and I. Reid. Uncalibrated and unsynchronized human motion capture : A stereo factorization approach. In *Proc. 22nd IEEE Conf. on Computer Vision and Pattern Recognition, Washington, D. C.*, volume 1, pages 128–134, 2004.

[9] D. W. Pooley, Michael J. Brooks, Anton van den Hengel, and Wojciech Chojnacki. A voting scheme for estimating the synchrony of moving-camera videos. In *ICIP (1)*, pages 413–416, 2003.

[10] Tinne Tuytelaars and Luc Van GOol. Synchronizing video sequences. *Conference on Computer Vision and Pattern Recognition*, 1:762–768, 2004.

[11] 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, 2006.

[12] Bruce Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging understanding workshop*, pages 121–130, 1981.

[13] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593 – 600, 1994.

[14] David Douglas and Thomas Peucker. Algorithms for the reduction of the number of points required to represent a line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

[15] Saul Needleman and Christian Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[16] Martin Fischler and Robert Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[17] CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/.