Through-the-Lens Synchronisation for Heterogeneous Camera Networks

Evren Imre
h.imre@surrey.ac.uk
Adrian Hilton
a.hilton@surrey.ac.uk

Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, UK

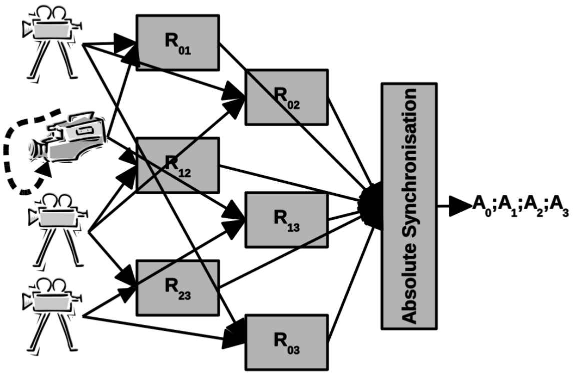

Figure 1: Overview of the algorithm, with one moving and 3 static cameras. R_{ij} blocks compute the relative synchronisation between the ith and the jth cameras. A_i is the absolute synchronisation for the ith camera.

Camera synchronisation involves the temporal alignment of a set of video sequences, independently acquired by two or more cameras. Accurate synchronisation is crucial for a wide variety of applications requiring multi-camera setups, ranging from 3D modelling of dynamic scenes (*e.g.*, featuring a performance, or a sports event) to video surveillance and super-resolution. Conventional synchronisation methods, which typically rely on hardware or audio signals, have practical limitations, imposing constraints on the size and the span of the network [2][1]. Through-the-lens synchronisation offers a robust and flexible way to synchronise a camera network from the content it generates.

In this paper, we propose a bottom-up synchronisation algorithm to estimate a frame rate and an offset for each member of a network composed of 2 or more cameras. Our approach involves the computation of a relative synchronisation estimate between each camera pair, from which the absolute synchronisation parameters of the individual cameras are calculated (Figure 1). The algorithm can handle hybrid networks of static and moving cameras with different resolutions and frame rates, and does not require rigid objects, long trajectories or overlapping fields-of-view beyond 2 cameras. It needs a set of image features on the dynamic scene elements, and the geometric relation between the images (which can be obtained from the static background features).

Relative Synchronisation: The frame indices of the jth camera (t_j) with respect to those of ith (t_i) is defined by the line

$$t_j = \alpha_{ij} t_i + \tau_{ij}, \qquad (1)$$

where α_{ij} is the relative frame rate, and τ_{ij} is the relative offset. The pair $R_{ij} = \{\alpha_{ij};\tau_{ij}\}$ can be estimated by fitting a line to the indices of the corresponding frames via robust methods, such as RANSAC. The index correspondences are established via the Viterbi algorithm, which maximises an image similarity measure across the sequences, while enforcing the ordering constraint (*i.e.*, $\alpha_{ij} \geq 0$). In order to measure the similarity of two images, first the corresponding image features are identified, and then, their median deviation from the geometric relation (*e.g.*, epipolar constraint) is computed. The resulting integer-level correspondences can be refined to subinteger resolution, by minimising the deviation over 3-frame feature trajectories.

Absolute Synchronisation: R_{ij} is computed from the absolute synchronisation estimates for the ith and jth cameras as

$$\begin{array}{ll} \alpha_{ij} = \dfrac{\alpha_j}{\alpha_i} \Rightarrow & \alpha_{ij}\alpha_i - \alpha_j = 0 \\ \tau_{ij} = \tau_j - \dfrac{\alpha_j}{\alpha_i}\tau_i & \end{array}, \qquad (2)$$

where $\{\alpha_i;\tau_i\}$ and $\{\alpha_j;\tau_j\}$ are the synchronisation parameters for the cameras i and j, respectively. For an L camera set, L such linear independent constraints can be stacked into a linear system, whose solution yields the absolute synchronisation parameters. The sets of constraints that satisfy the linear independence requirement appear as the spanning cycles of the graph, where the nodes representing the cameras are linked by the edges representing the corresponding R_{ij}. The absolute synchronisation algorithm solves Equation 2 for each spanning cycle, and generates $\{\hat{R}_{ij}\}$, an estimate of the set of all available relative synchronisations $\{R_{ij}\}$. The absolute synchronisation is computed from the cycle, whose associated $\{\hat{R}_{ij}\}$ minimises the Euclidean distance of the index correspondences to their synchronisation lines. The result is further improved by the inclusion of all consistent R_{ij} to the minimal set, and refinement over their index correspondences.

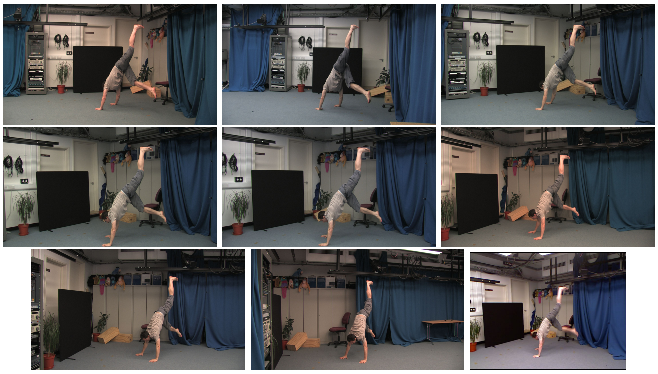

Figure 2: *Acrobatics. Top:* Cameras 0-2. *Middle:* Cameras 3-5. *Bottom:* Cameras 6-7, and the Kinect.

	0	1	2	3	4
α	1	1	1	1	1
τ	0	-11	-15	-31	-9
$\Delta\alpha$	0	5.8×10^{-7}	1.1×10^{-6}	7.6×10^{-7}	1.1×10^{-6}
$\Delta\tau$	0	5.8×10^{-4}	1.3×10^{-4}	3.7×10^{-4}	6.8×10^{-5}
	5	6	7	Kinect	
α	1	1	1	1.2	
τ	-27	-26	-32	≈ 26.8	
$\Delta\alpha$	1.2×10^{-6}	7.1×10^{-7}	1.7×10^{-6}	5.9×10^{-4}	
$\Delta\tau$	2.7×10^{-4}	1.9×10^{-5}	7.6×10^{-4}	6.2×10^{-2}	

Table 1: Absolute synchronisation estimates, with Camera 0 as the reference. α and τ are the ground-truth values of the synchronisation parameters, whereas $\Delta\alpha$ and $\Delta\tau$ indicate the *absolute value* of the estimation error. The ground-truth offset for the Kinect is not known, but manually determined from the sequence.

Experimental Results: The algorithm is tested on 3 datasets, one of which is illustrated in Figure 2. The datasets feature 8 static and moving HD cameras, and a Kinect, capturing a non-rigid object undergoing rapid deformations. The results, presented in Table 1 indicate that the algorithm can achieve very high accuracy despite the challenging content.

Acknowledgements: This work is supported by the TSB project “SYMMM: Synchronising Multimodal Movie Metadata”.

[1] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *Proc. CVPR*, pages 224–231, 2009.

[2] A. Miller. R&d and blue peter- ski rossendale free-viewpoint visualisation. `http://www.bbc.co.uk/blogs/researchanddevelopment/2011/03/rd-and-blue-peter--ski-rossend.shtml`.