

Content-Based Multi-Camera Video Alignment using Accelerometer Data

Antonio Carlos Nazare¹, Filipe Costa², William Robson Schwartz¹

¹Smart Surveillance Interest Group, Smart Sense Laboratory, Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brazil

²CPqD - Image and Speech Processing Management, Campinas, Brazil

Email: antonio.nazare@dcc.ufmg.br, fcosta@cpqd.com.br, william@dcc.ufmg.br

Abstract

Video alignment is an important task for environments with distributed multiple cameras, making possible, for instance, to verify the exact instant that an event happened on different views. The alignment comprises establishing a temporal correspondence among frames captured by different video cameras. Many works have been proposed to solve this problem when the cameras present overlapping of Field of View (FOV) or are located close to each other. However, in this work, we present a novel approach to perform video alignment for cameras without overlapping FOV (i.e., the cameras might be located on different floors of a building). The method employs the sensor data generated by a smartphone (synchronized to a time server), to align multiple videos by finding a temporal matching between the videos which captured a person in the scene and the signal of the smartphone accelerometer carried by this individual, providing the exact time that a movement was performed. To the best of our knowledge, this is the first attempt at performing such type of alignment. According to experimental results, the proposed approach was able to align multiple videos with a length of 30 minutes with errors as low as 160 ms.

1. Introduction

Using of distributed surveillance cameras placed in several locations has exploded over the past years. These systems are used to provide safe environments for people, performing forensic and surveillance related tasks, such as detection of anomalous events or identification and tracking of suspect people [2, 13]. Usually, a huge amount of visual data is collected from multiple cameras and processed for these tasks, which is daunting and prone-to-error. Thus, the use of automatic approaches is desirable for data processing, letting humans act only on the decision-making process [10].

Distributed cameras provide different views captured from multiple environments simultaneously and one of the

major concerns for a multi-camera network is the temporal alignment accuracy between videos [4], which consists on establishing a temporal correspondence among frames captured by different cameras. This process is also known as *Video Alignment* [4].

Retrieving accurate temporal information of each frame is a non-trivial task since this information, when provided, is generated by the internal clock of the cameras, which are not globally synchronized. An automatic approach to camera internal clock synchronization is the use of a time server (e.g. Network Time Protocol¹ server). Although automatic, there are several factors that might influence the accuracy of the NTP synchronization, such as connectivity and network topology [16]. Besides, the precision of NTP synchronization varies according to the model of the devices [19]. Finally, an alternative to video alignment is the manual synchronization or the dedicated hardware employment. Although this can be a viable solution, it is labor-intensive, expensive and, in specific cases, technically unfeasible.

Robust approaches for performing video alignment considering no information regarding their internal clocks is useful for allowing an integrated processing of data captured by multiple distributed cameras. For instance, it might help in tracking subjects across different areas of a large environment (e.g., malls, parks or airports), and it might also be useful in forensics to determine aspects such as, the chronology of a crime captured by cameras, the exact moment that some important actions occurred or track the suspects on the scene to raise evidence.

Once the development of automatic and non-expensive approaches are desirable, several solutions have been proposed in the literature. Some methods are based on the video alignment using the intersection of views between the cameras [1, 8, 18, 20]. Despite the satisfactory results, these approaches are limited by requiring that the cameras have an intersection of views, which is not workable in distributed surveillance systems, where multiple cameras are distributed in a large area. Another alternative used

¹Network Time Protocol (NTP) is an Internet protocol used to synchronize the clocks of connected devices to some time reference.

in multimedia applications is the use of audio for video alignment [7, 9, 15], which although does not require the intersection of camera views, presents the limitation that the cameras must capture the same sound simultaneously. However, assuming the cameras are distant from each other (e.g., at different floors of a building), such capture might not be workable.

In this work, we present a novel approach to perform video alignment with data captured from multiple cameras with a non-overlapping field of view. The proposed approach employs an auxiliary motion sensor device, such as a smartphone, synchronized with an NTP server and is carried by an individual filmed by the video cameras. Specifically, we identify temporal matchings between a video that captured a person in the scene and the signal of the accelerometer carried by this individual, being able to find the correspondence between the visual motion data (i.e., a segment of the video with a person walking), and the accelerometer data. This correspondence allows us to estimate the timestamps of the beginning and the end of the visual motion segment, based on the accelerometer timestamps. Once the timestamps of the start and end of each visual motion segment have been estimated, it is possible to compute the timestamps for the remaining frames of a video. Thus, if we correctly synchronize all frames of all video cameras with timestamps of a common reference (in our case, the person accelerometer data is the reference), we will have all videos synchronized.

To the best of our knowledge, this work is the first attempt of performing multi-camera video alignment based on sensor data considering videos captured at locations without intersection of field of view and that prevent the use of a common audio (e.g., cameras located in different floors of a building). The combination of visual and sensor data has been investigated in the literature, being useful for increasing the effectiveness for certain tasks, such as activity recognition [3], person identification [17], and person detection and tracking [12]. However, different from our purpose, the focus of the aforementioned works is not the video or sensor alignment, since they consider that the visual and sensor data are already synchronized.

Experimental results show the robustness of the proposed approach, which is able to align multiple videos of 30 minutes with errors as low as 160 ms. The evaluated videos were recorded by no-overlapping field of views cameras placed in different floors of a building, differently from works that require the field of views overlapping [1, 8, 18, 20], and without being able of capturing a common audio source, differently from works that require this [7, 9, 15], which makes our proposed method more flexible than currently available approaches.

2. Proposed Approach

The general idea of the proposed approach works as follows. A person carrying a device with a tri-axis accelerometer sensor (e.g., smartphone), synchronized with

a time server, is captured by a set of cameras $\mathcal{C} = \{C_1, C_2, \dots, C_n\}$. While the individual moves in the scene, the accelerometer sensor generates a signal S that is processed to extract the relevant information. Section 2.1 describes the accelerometer data processing.

The same person P_j , who generates the sensor signal S_j , can be captured several times by the same camera C_i while a video V_i is recorded. Each time the person appears in V_i , he/she generates a tracking data containing the visual motion signal of his/her movement. The approach used to extract tracklets is explained in Section 2.2.

Considering the visual motion tracklets of a subject P_j , we can determine the frame timestamps for video V_i by matching the movement information captured from the visual motion tracklets to the region of the signal S_j generated by the accelerometer which is already synchronized with an NTP time server. Therefore, by executing this synchronization of S_j with the videos from n cameras, we can determine the timestamp for all the frames of each video, allowing us to transitively align the videos with each other. The Section 2.3 describes how the matching between the accelerometer signal and the information extracted from the tracklets is performed.

2.1. Sensor Data Processing

After collecting the tri-axis accelerometer signal $S = \{S_x, S_y, S_z\}$ for a subject carrying a sensor device, we extract the accelerometer signal magnitude S^m given by $S^m = \sqrt{S_x^2 + S_y^2 + S_z^2}$. Then, we perform the following steps: (i) resample the signal S^m to a constant frequency of $50Hz$ using a one-dimensional linear interpolation; (ii) apply a *Gaussian Filter* (with $\sigma = 5$), aiming at smoothing the signal and removing noises generated by the capture; and (iii) normalize data between 0 and 1. Figure 1(a) illustrates the resampled and filtered signal (*Filtered Signal*).

As depicted in Figure 1(a), while the individual is moving (e.g., walking), the signal has a high local variance and when it is stationary (e.g., seated), it presents a low variance. We want to capture these both states, *moving* and *stationary*. For this, we apply a signal binarization using a local threshold τ_s , as presented in [6]. Considering blocks of 50 samples (one second of $50Hz$ data) without overlap, we compute the value α for each block $B_i \in \{B_1, B_2, B_3, \dots, B_n\}$ of samples by $\alpha(B_i) = \sigma(B_i) \times \mu(B_i)^{-1}$ where $\mu(B_i)$ is the mean and $\sigma(B_i)$ is the standard deviation of the values in B_i . The samples in B receive value 1 if $\alpha_{B_i} > \tau_s$ and value 0, otherwise. Following Hwang et al. [6], we adopt $\tau_s = 0.05$. The result of signal binarization is showed in Figure 1(a) as *High Signal* and *Low Signal* when sample receives value 1 and 0, respectively. Finally, the binarized signal will be used to perform the matching between the visual and sensor data during the video alignment process (see Section 2.3).

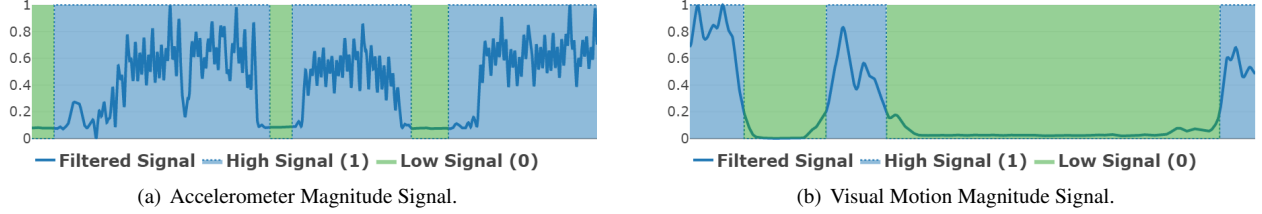


Figure 1: Examples of the processed accelerometer and visual motion signals.

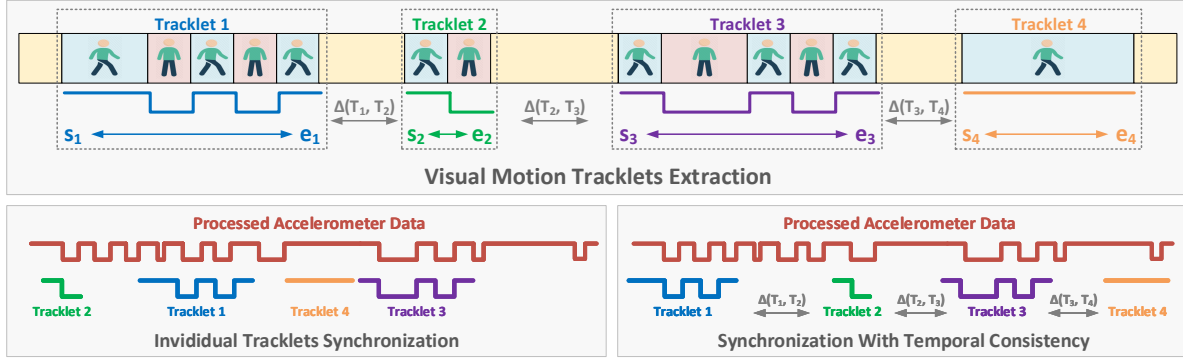


Figure 2: Illustration of visual motion tracklets extraction and their temporal synchronization with the processed accelerometer data.

2.2. Visual Data Processing

The first step to extract tracklets from the visual data is the person detection. We apply a deep learning based object detector called *Faster R-CNN* [14] to detect all individuals present in each video frame. Then, we track the person in the scene considering a single mean-shift of the bounding boxes got using the detector, generating one tracklet for each time that the person appears in the camera view.

As in the accelerometer signal processing (Section 2.1), we intend to capture the intervals at which the tracked person was either moving or stationary. Therefore, we estimate the magnitude of the dense optical flow [5] for all consecutive frames in the tracklet, considering only the person region (the detected bounding box). Then, we normalize the optical flow value by the height of detection bounding box to avoid the interference of the distance between the person and the camera at the moment of the capture. This estimation generates a tracklet temporal signal T^m , where the samples are the magnitude of person's visual motion and $|T^m|$ is the number of frames in the tracklet, in which T^m starts and ends on frames s and e , respectively. The top part of Figure 2 illustrates a set of visual motion tracklets extracted from a video.

To allow the comparison between T^m and the binarized sensor signal we resample T^m at the same sampling frequency ($50Hz$) of S^m . We also apply Gaussian filtering for removing noises (with $\sigma = 5$) and normalize the T^m to set the values between the interval $[0, 1]$. Figure 1(b) illustrates the resampled and the filtered signal (*Filtered Signal*). Then, we apply a signal binarization using a global

threshold τ_t . However, differently from the sensor signal binarization, we do not separate the optical flow signal in blocks, since we assume higher values indicate movement, while lower values indicates the individual is stationary. This way, the value of τ_t has been defined empirically (in this work, it was set experimentally to $\tau_t = 0.15$). Figure 1(b) depicts the binary visual motion tracklet signal as *High Signal* and *Low Signal* when sample receives 1 and 0 values, respectively.

2.3. Visual Motion and Accelerometer Signal Synchronization

The temporal synchronization between the accelerometer and the visual motion captured from tracklets is performed over the processed accelerometer signal S^m (Section 2.1) and the visual motion tracklets (Section 2.1) $\mathcal{T} = \{T_1^m, T_2^m, \dots, T_n^m\}$ extracted from a specific video (Section 2.2). We execute an overlapping sliding function of $T_i^m \in \mathcal{T}$ over S^m , calculating the Hamming Distance [11] in each sliding overlapping index. The index with the smallest distance is a strong candidate for the point of best alignment between signals. For instance, if we have a perfect alignment between T_i^m and S^m , the segments representing the person walking and in movement will be aligned in the visual and in the sensor data, indicating that these signals were generated by the same subject at the same time (i.e., the more the number of aligned segments between video and sensor data, more likely that they were generated simultaneously).

If we perform the overlapping sliding Hamming function considering a single tracklet at the time, we might find an incorrect match between the tracklet and the sensor signal. This is because the smallest Hamming distance response will not necessarily be the at the correct position. For instance, in Figure 2 (*bottom-left*), the match with the smallest Hamming distance between the sensor signal and *tracklet 4* is obtained in an incorrect index. Thus, a temporal relationship between all elements in \mathcal{T} is necessary for a correct synchronization.

To avoid the aforementioned undesired behavior, we consider the temporal gap (distance between two visual motion tracklets) of multiple tracklets of the same person to help us find the correct alignment index. For instance, let T_i^m and T_j^m be two motion tracklets signals, both $\in \mathcal{T}$. Let T_i^m begin at the time s_i and finish at the time e_i , while T_j^m begins at the time s_j and finishes at the time e_j . Let also that $e_i < s_j$ and the gap between the tracklets T_i^m and T_j^m is given by $\Delta(T_i^m, T_j^m) = s_j - e_i$. If we align T_i^m and T_j^m at the same time over S^m , the starting point of the alignment between T_j^m and S^m have to be $\Delta(T_i^m, T_j^m)$ after the ending point of the alignment between T_i^m and S^m (assuming a constant frame-rate for video capture). With this restriction, we can perform a sliding window-based comparison of multiple tracklets at the same time over S^m computing the smallest hamming distance that respect the temporal gaps. The result of this approach is illustrated in Figure 2 (*bottom-right*).

In our experiments, we assume that the person with the accelerometer used to synchronize the cameras will appear in the scenes recorded by the cameras. Considering real surveillance environments, it is unlikely that an individual will stay stationary the whole time he/she is captured by one or more cameras (such case would prevent us from employing the proposed synchronization algorithm). Therefore, it is reasonable to assume that the subject with the accelerometer will walk through the environments covered by the cameras, allowing the application of the proposed approach.

3. Experimental Results

To evaluate the performance of the proposed approach, we conducted experiments in a building, in which we placed a group $\mathcal{C} = \{C_C, C_E, C_K\}$ of three cameras in a *corridor*, a *elevator hall* and a *kitchen*, respectively, with non-overlapping FOV and placed in different floors, depicted in Figure 3. The cameras are neither connected nor synchronized with any common time server. A set $\mathcal{V} = \{V_C, V_E, V_K\}$ of three videos, one from each camera in \mathcal{C} , was captured considering a constant frame-rate of 30 *Frames per Second* (FPS), resolution of 1920×1080 pixels and with an average length of 30 minutes.

To collect the sensor data, three volunteers carried a smartphone each in their pocket and walked entering and exiting the cameras field of view, generating a set of accelerometer signals $\mathcal{S} = \{S_1, S_2, S_3\}$, for



Figure 3: Field of View (FOV) of the cameras used in the experiments.

each person in $\mathcal{P} = \{P_1, P_2, P_3\}$. The volunteers were instructed to walk normally through the environments recorded by video cameras, performing routine activities (e.g., leaving the laboratory and going to the kitchen to get a coffee). All smartphones were synchronized with a common NTP server, thus ensuring that they shared the same time clock. In addition, the accelerometer sensor data captures were performed using a simple Android application², developed by us for this purpose, with a sampling rate of approximately $100Hz$.

To obtain the ground truth of video frame timestamps, we employ the following procedure. Using a laptop synchronized to the time server (the same NTP server used for the smartphones), we generate several sequential *QR codes* that represent the current timestamp and show them to the employed cameras for few seconds. Then, assuming a constant frame-rate of 30 FPS, frames containing the QR code (we decode the information in QR code into timestamps values) are used to compute the ground truth timestamps for all remaining frames. Experimentally, we estimated a maximum error of 0.027 seconds for the ground truth estimation process, which does not interfere in the final results since it is smaller than the frame interval when the videos are captured at 30 FPS (i.e. 30 FPS results in a frame interval of $1/30 = 0.033$ seconds).

The person’s tracklets extracted from the video were obtained as described in Section 2.2 and, for each tracklet an identity in \mathcal{P} was manually attributed (methods of person identification could be used for this purpose, but it is not the focus of this particular work). Tracklets not belonging to the known subjects were discarded. Columns 2, 4 and 6 in Table 1 show the number of extracted tracklets for each subject in each video.

3.1. Results and Discussion

Since the other methods found in the literature have the constraining of FOV intersection or audio sharing, it is not possible to execute and compare to these methods using the data we capture. Therefore, in this section, we will present the quantitative results achieved only by our approach.

The first experiment evaluates the video frame timestamp estimation error (i.e., how well we can attribute a timestamp value to each frame of the video). For each combination $(S_i, V_j) \in \mathcal{S} \times \mathcal{V}$ of sensor signal and video, we estimate the timestamp of the frames in V_j , performing the matching of the signal S_i with all extracted

²The capture application is available for download in the following link: www.sense.dcc.ufmg.br/sensorcap

Table 1: Mean Average Error (MAE), in seconds, of the video timestamps estimation from accelerometer signal.

Signal	V_C – corridor		V_E – elevator hall		V_K – kitchen	
	#tracks	MAE	#tracks	MAE	#tracks	MAE
S_1	7	0.018	6	0.088	3	0.21
S_2	3	0.13	2	0.69	2	0.036
S_3	6	0.14	6	0.11	4	0.086
Fusion	16	0.086	14	0.22	9	0.029

Table 2: Misaligned frames between videos.

Signal	V_C – corridor	V_C – corridor	V_E – elevator
	V_E – elevator	V_K – kitchen	V_K – kitchen
S_1	4	6	2
S_2	16	6	22
S_3	7	7	0
Fusion	4	2	7

tracklets of person P_i (who generated the signal S_i), as described in Section 2.3. The matching process generates a vector of timestamps T_j , where $|T_j|$ is equal to the number of frames of V_j .

To evaluate the estimation error, we calculate the Mean Absolute Error (MAE), given by

$$\text{MAE} = \frac{\sum_{idx=1}^n |T_j[idx] - GT_j[idx]|}{n},$$

between the estimated timestamps (T_j) and the real (ground truth) timestamps GT_j of video V_j (measured in seconds).

The first three rows in Table 1 show the mean absolute error obtained for each combination (S_i, V_j) of individual sensor signal $S_i \in \{S_1, S_2, S_3\}$ and video $V_j \in \{V_C, V_E, V_K\}$, and the number of motion tracklets used in the matching. On average, the MAE was $0.167s \pm 0.191$, demonstrating the effectiveness of the method when applied in videos with length of approximately 30 minutes (i.e., error smaller than 0.2 seconds for 30 minutes of video). We can also see that the results vary for each combination. For instance, the video V_E – *elevator hall* presented the highest error value for the sensor signal S_2 . In this particular case, the person was walking constantly, creating an ambiguous motion signature (see Section 2), and the number of extracted tracklets was small. This case can be considered a limitation of the proposed method.

If we take advantage of the redundancy of persons that appear in the scene and calculate the average vector of timestamps for each video, the results improve. In this approach, to estimate the timestamp of each frame from a video $V_j \in \mathcal{V}$, we calculate the average of the timestamps obtained by the matching of each signal $S_i \in \mathcal{S}$ with V_j . This process allows us to get a better estimation using the redundancy of subjects who appear in the same scene. According to the last row of Table 1 (*Fusion* row), this approach reduces the error for all videos (reducing the average MAE to $0.112s \pm 0.080$).

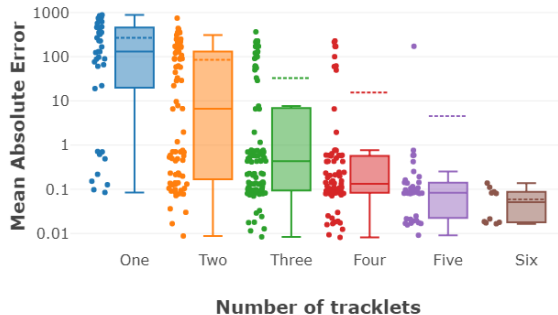


Figure 4: Mean absolute error (in seconds) as a function of the number of motion tracklets. The values are the average for the three persons in the scene.

As described in Section 2, once the timestamps of two videos have been estimated, it is possible to perform frame-to-frame alignment. Table 2 presents the results of the alignment between two videos after the estimation of their vectors of timestamps, showing the absolute number of frames from one video that are misaligned in respect to the other using either individual sensor data or the fusion strategy (as previously explained).

According to the results in Table 2, most of the camera combinations achieved a misalignment smaller than 8 frames. However, there are two cases where the misalignment are larger than 15 frames (half second when capturing at 30 FPS). Although some errors occur, the alignment results show that the method is able to synchronize two different cameras with a significant precision without the constraint of sharing field of view or being nearby cameras. The last row of Table 2 shows the results achieved with the fusion strategy, achieving a maximum error of 7 frames (i.e., $0.233s$).

As described in Section 2.3, we expect we can reduce the error by increasing the number of tracklets of each individual in the scene. Thus, we also investigate the influence of the number of motion tracklets employed on the timestamp estimation process. In this experiment, we employ k tracklets to estimate the video timestamps, with k ranging from 1 up to 6. For each value of k , we tested all possible combinations of k tracklets for all pairs $S_i, V_j \in \mathcal{S} \times \mathcal{V}$ of sensor signals and videos. The individual MAE error results for each test are represented as bullets in Figure 4. A summarization of the results is also showed using boxplots grouped by tracklet number (for a better visualization, the y-axis of the chart is shown in logarithmic scale). The dashed line represents the average of MAE. According to the results, there is a large variability on the error when only one or two tracklets are considered, which is caused by the dependency of the length and motion variability of the chosen tracklets. However, the error decreases significantly when using three or more motion tracklets.

In the last experiment, we assume that we know a *safe temporal range* of sensor signal where the motion tracklets should be aligned. With this assumption, we can

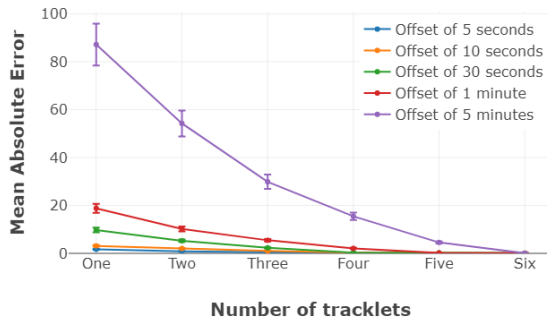


Figure 5: Mean absolute error (in seconds) as a function of the number of motion tracklets. The values are the average for the three persons in the scene.

simulate a common forensics scenario when we know the approximate period of time which an individual carrying the smartphone was captured by the camera, but we do not know exactly the instant that it occurs. For instance, if the interval is 10 seconds, instead of performing the alignment in the whole sensor signal, as in the previous experiments, we perform the correlation of the motion tracklets only considering 10 seconds before and 10 seconds after the candidate alignment. For experimental purposes, we consider the *safe temporal range* an offset of the ground truth alignment. To demonstrate the impact of applying this constraint, we replicate the previous experiment (where we employ t tracklets for timestamps estimation) using different sizes of offset, more precisely, offsets of size 5, 10, 30 seconds and 1 and 5 minutes.

Figure 5 presents the average MAE of the chosen offset for each number of tracks. The results demonstrate that a preliminary alignment helps us to decrease the error even when only few tracklets are considered.

4. Conclusions and Future Directions

This paper presented a novel approach for multiple camera alignment using an auxiliary accelerometer sensor. The experiments performed reported small temporal misalignment between the videos and the accelerometer, which demonstrate the robustness of the proposed synchronization method. Although there is the restriction that a person carries an accelerometer in the scene, we believe that this is an important contribution since, to the best of our knowledge, this is the first attempt to synchronize videos without requiring that the cameras are located close to each other or share the field of view and the results are accurate. Furthermore, our approach do not require any specific synchronization hardware.

Future works include experiments considering more cameras and people for data capture. We also intend to investigate new methods for synchronization using multiple movement sensors, such as accelerometer, gyroscope and magnetometer. Furthermore, another research direction is to extend the approach for dealing with videos with variable frame-rate. Finally, we intend to extend the

proposed method for pointing out the individual who is carrying the accelerometer, considering a scene of a crowd environments, to perform the identification task, for instance.

Acknowledgments

The authors would like to thank the National Council for Scientific and Technological Development – CNPq (Grant 311053/2016-5), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14 and PPM-00540-17), the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project) and Petrobras (Grant 2017/00643-0).

References

- [1] C. Abl, Z. Kukulova, A. Fitzgibbon, J. Heller, M. Smid, and T. Pajdla. On the two-view geometry of unsynchronized cameras. In *IEEE CVPR*, 2017.
- [2] I. Bouchrika, J. N. Carter, and M. S. Nixon. Towards automated visual surveillance using gait for identity recognition and tracking across multiple non-intersecting cameras. *Springer MTA*, 75(2):1201–1221, 2016.
- [3] C. Chen, R. Jafari, and N. Kehtarnavaz. A survey of depth and inertial sensor fusion for human action recognition. *Springer MTA*, 76(3):4405–4425, 2017.
- [4] F. Diego, D. Ponsa, J. Serrat, and A. M. Lopez. Video alignment for change detection. *IEEE TIP*, 20(7):1858–1869, 2011.
- [5] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, 2003.
- [6] I. Hwang, J. Cho, and S. Oh. Vibecomm: Radio-free wireless communication for smart devices using vibration. *Sensors*, 14(11):21151–21173, 2014.
- [7] J. Kammerl, N. Birkbeck, S. Inguva, D. Kelly, A. J. Crawford, H. Denman, A. Kokaram, and C. Pantofaru. Temporal synchronization of multiple audio signals. In *IEEE ICASSP*, 2014.
- [8] T. Kuo, S. Sunderrajan, and B. S. Manjunath. Camera alignment using trajectory intersections in unsynchronized videos. In *IEEE ICCV*, 2013.
- [9] J. Liang, P. Huang, J. Chen, and A. Hauptmann. Synchronization for multi-perspective videos in the wild. In *IEEE ICASSP*, 2017.
- [10] A. C. Nazare and W. R. Schwartz. A scalable and flexible framework for smart video surveillance. *Elsevier CVIU*, 144(C):258–275, 2016.
- [11] M. Norouzi, D. J. Fleet, and R. Salakhutdinov. Hamming distance metric learning. In *NIPS*, 2012.
- [12] S. Papaioannou, A. Markham, and N. Trigoni. Tracking people in highly dynamic industrial environments. *IEEE TMC*, 16(8):2351–2365, 2017.
- [13] R. Poppe. A survey on vision-based human action recognition. *Elsevier IVC*, 28(6):976–990, 2010.
- [14] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(6):1137–1149, 2017.
- [15] P. Shrstha, M. Barbieri, and H. Weda. Synchronization of multi-camera video recordings based on audio, 2007.
- [16] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *IEEE Network*, 18(4):45–50, 2004.
- [17] T. Teixeira, D. Jung, G. Dublon, and A. Savvides. Pem-id: Identifying people by gait-matching using cameras and wearable accelerometers. In *ICDSC*, 2009.
- [18] P. A. Tresadern and I. D. Reid. Video synchronization from human motion using rank constraints. *Elsevier CVIU*, 113(8):891–906, 2009.
- [19] E. Z. Welty, T. C. Bartholomaeus, S. O’Neel, and W. T. Pfeffer. Cameras as clocks. *Glaciology*, 59(214):275–286, 2013.
- [20] F. Zhou and F. D. la Torre. Generalized time warping for multi-modal alignment of human motion. In *CVPR*, 2012.