Automatic Time Skew Detection and Correction

Danil Korchagin
Idiap Research Institute
Martigny, Switzerland

***Abstract*—In this paper, we propose a new approach for the automatic time skew detection and correction for multisource audiovisual data, recorded by different cameras/recorders during the same event. All recorded data are successfully tested for potential time skew problem and corrected based on ASR-related features. The core of the algorithm is based on perceptual time-quefrency analysis with a precision of 10 ms. The results show correct time skew detection and elimination in 100% of cases for a real life dataset of 32 broken sessions and surpass the performance of fast cross correlation while keeping lower system requirements.**

Keywords-time-quefrency analysis; time synchronisation; pattern matching

I. Introduction

The TA2 project (Together Anywhere, Together Anytime) [1] is concerned with investigation of how multimedia devices can be introduced into a family scenario to break down technology and distance barriers. Technically, the TA2 project tries to improve group-to-group communication by making it more natural and by giving the users the means to easily participate in shared activities. In this sense, we are interested in the use of consumer level multimedia devices in novel application scenarios.

Several generic scenarios (e.g., remote playing of a family game, my videos, etc) involve the use of multisource multimodal algorithms. The corresponding algorithms by-turn require data to be in sync. During recording of multisource multimodal database [2], we have experienced several times, that the recorded material was neither synchronised, nor having the same clock (there was positive or negative time skew). While several people proposed “to redo the recordings” as the easiest and “only feasible” solution, we decided to run few experiments to prove that the problem can be resolved on algorithmic level as well, regardless the fact it was decided not to include these recordings into mentioned database.

The time skew can be caused by several factors: due limited carrying capacity during the data capture or, more often, due to unsynchronised clocking of different devices. Unsynchronised clocking in-turn can be caused by many physical level issues, such as temperature variations, variation in intermediate devices, capacitive coupling, material imperfections, etc. As consequence, the clock can travel more slowly for one capturing device and faster for other capturing devices. This, in consequence, can destroy the integrity of the set of the recordings, even if the separate recordings would never be considered as problematic.

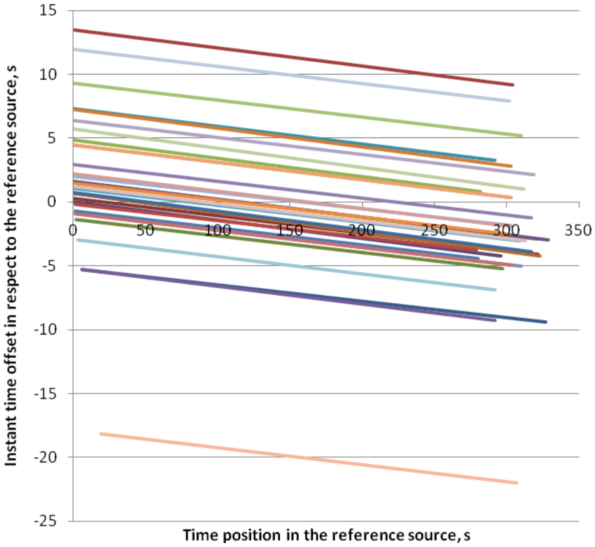

Figure 1. Time skew annotations for experimential dataset of 32 sessions.

In a professional setup, one might expect to be able to use multiple capture devices, and for them all to be synchronised via a common clock or similar [3]. On a network level, a number of protocols (e.g., Network Time Protocol [4]) have been designed to reduce time skew, and produce more stable functions. Some applications (such as game servers) may also use their own synchronisation mechanism to avoid reliability problems due to time skew.

Consumer level devices, however, do not normally provide such capabilities. In our previous work [5] we have shown that an initial shift for the data without time skew issue can be easily resolved by automatic temporal alignment algorithm. Nevertheless in the presence of time skew issue, the data become continuously misaligned during the timeline of the event even it is in perfect sync in the beginning of the data. Further, due different setup of recording devices, we cannot rely in any predictable sense on the video signal. This leaves us with the audio signal from which to infer time skew information.

In this study, we were provided with a real life dataset of 32 broken sessions (see Fig. 1), in which two signals from fixed cameras with different clocks were recording the whole session. If we could show that the time skew could be eliminated reliably, then the project could profit from automatically fixed dataset. If it was too error-prone or computationally onerous, then the dataset would have to be re-recorded with a common clock or similar.

II. Experimental Dataset

All results presented in this paper were achieved on a real life dataset of 64 recordings (32 sessions x 2 desynchronized recordings of 4.5-5.5 min each), containing high quality 1080p25 video and high quality PCM 48 kHz stereo / PCM 44.1 kHz quadro audio. Different sessions were recorded with the same set of hardware. The content consists of a gaming sessions with enabled video chat of socially connected but spatially separated people. All corresponding audio tracks were extracted and converted to 16 kHz mono PCM files with FFMPEG software [6].

Experiments were conducted on a closed set (i.e. we did not consider a rejection mechanism for the recordings that did not correspond to the same session). Nevertheless according to our previous studies on a rejection mechanism [7], the proposed approach can be successfully extended to an open set.

III. Time Skew Detection

Time skew detection corresponds to estimation whether all recordings in the same session have the same absolute time velocity or not (see Fig. 1). Therefore it is enough to answer the question whether the relative time velocity ratio between the recordings from the same session equals to 1.0 or not. Nevertheless to be able to perform time skew correction the relative time velocity ratio has to be estimated precisely.

We define time-quefrency signatures as time-quefrency matrices based on normalised truncated mel-cepstral vectors in steps of 10 ms. A 256 point Discrete Fourier Transform (DFT) is performed on overlapping audio frames of 16 ms in steps of 10 ms and squared to give the power spectrum. The resulting 129 unique bins are then decimated using a filter-bank of 23 overlapping triangular filters equally spaced on the mel-scale. The mel-scale corresponds roughly to the response of the human ear. A logarithm and DFT then yield the mel-cepstrum [8], which is truncated, retaining the lower 12 dimensions except energy. This truncation retains spectral shape and discards excitation frequency. Next, Cepstral Mean Normalisation (CMN) is performed by subtracting from each cepstral vector the mean of the vectors of the preceding (approximately) half second. This has the effect of removing convolutional channel effects. Finally, if the norm of a vector of the 12 mean normalised cepstral coefficients is higher than 1, then the vector is normalised in Euclidean space. This gives us the reduced variance of the search distance space.

If h_i and g_i are desynchronised signals from the session i, and it is known that the signal h_i is located within the signal g_i, then r_h^g, the relative time velocity ratio, is given by:

$$r_h^g = \frac{t_e^g - t_b^g}{t_e^h - t_b^h} \qquad (1)$$

In the equation (1), t_e^h and t_b^h are respectively the gravity points (centres of the corresponding time-quefrency signatures) at the end and at the beginning of the signal h_i. t_e^g and t_b^g are temporally aligned [5] gravity points t_e^h and t_b^h within the signal g_i:

$$\begin{aligned} t_e^g &= \underset{g_i}{\arg\min}(d(t_e^h, g_i)) \\ t_b^g &= \underset{g_i}{\arg\min}(d(t_b^h, g_i)) \end{aligned}, \qquad (2)$$

where d is Euclidean metric.

The initial shift t_b of the signal h_i within the signal g_i is given then by:

$$t_b = t_b^g - \frac{r_h^g l_s^h}{2}, \qquad (3)$$

where l_s^h is the length of the signature for the signal h_i.

The achieved precision (the number of correctly aligned signatures divided by the total number of signatures) of the gravity point temporal alignment resulted in 100% on described dataset. Fig. 2 illustrates estimated instant time offsets in respect to the reference signals not only at defined gravity points, but along all intersection periods. The graph is plotted based on time-quefrency signatures of 10 s. Intersection periods are calculated automatically [7] based on confidence value of temporal alignment (with fixed confidence threshold at 50%).

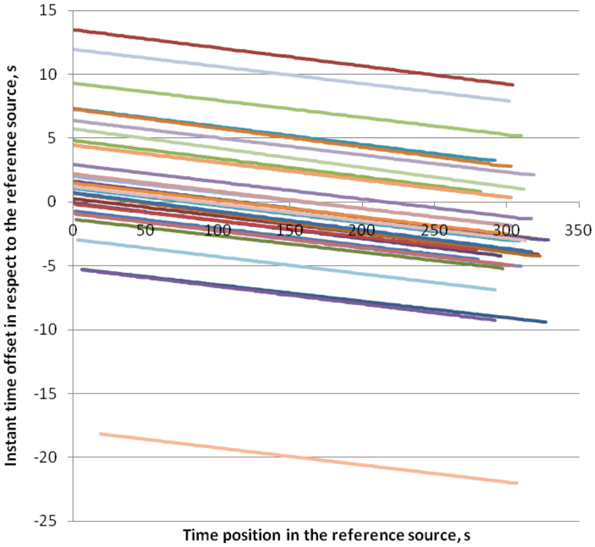

Figure 2. Estimated time skew trajectories.

It is clearly visible that the estimated instant time offsets in general follow the annotations, though a bit less smooth than the annotations. Corresponding standard deviations and other observations from time skew impact are discussed in details in the following chapter.

In addition to defined time-quefrency signatures we compared the results with and without energy component and with well-known fast cross correlation. The corresponding results on gravity point estimation are shown in the table I.

TABLE I. PRECISION OF GRAVITY POINT ESTIMATION

Algorithm	*Precision*
Time-quefrency signatures of 10 s	100%
Time-quefrency with energy signatures of 10 s	100%
Cross correlation (10 s)	48.4%
Cross correlation (30 s)	65.6%

After application of the relative time velocity ratio and initial shift, we were able to confirm 100% precision of time skew correction on described dataset as well. Comparison to other methods is shown in the table II.

TABLE II. PRECISION OF TIME SKEW CORRECTION

Algorithm	*Precision*
Time-quefrency signatures of 10 s	100%
Time-quefrency with energy signatures of 10 s	100%
Cross correlation (10 s)	21.9%
Cross correlation (30 s)	50.0%

Processing time (on an Intel Core 2 CPU 6700 2.66GHz) for the proposed algorithm without multi-core optimisation was 5.4 seconds for automatic temporal skew detection within a session of 5 min with two recordings. It was noticeably faster than time required by fast cross-correlation. We have to notice as well that processing time for the proposed algorithm is directly proportional to the length of the signature and to the length of the session. Memory requirement for the proposed algorithm was 1.5 MB per 5 min session, which is also much lower than required by fast cross-correlation.

IV. TIME SKEW IMPACT ON TEMPORAL ALIGNMENT

The results shown in the table I prove generalisation of time-quefrency based temporal alignment to the case of time skew presence. The results based on cross-correlation have much higher impact from time skew presence (the difference between the precision of these approaches varies up to 52% in time skew presence case versus up to 20% in its absence).

According to the definition of time-quefrency signatures we observe very good precision due to information taken from both domains: temporal and cepstral. In our previous work [5] we have shown that the precision and automatically estimated confidence increase continuously with increasing the length of the signatures in the case of time skew absence. Due the presence of the time skew issue the temporal information does not match precisely anymore. Therefore the longer signatures after a certain point should results in lower precision and, accordingly, in lower confidence.

In Fig. 3 we illustrate how the length of the signature influences the confidence measure on the described dataset. The confidence measures were estimated as a measure of relative variance of the search space via standard deviation for fast cross-correlation and via global maximum for time-quefrency signatures [5].

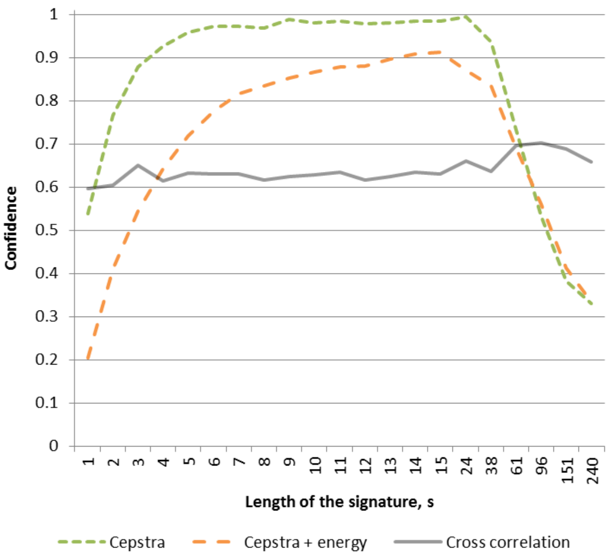

Figure 3. Confidence versus signature / test segment length.

It is clearly visible that the confidence measure of defined time-quefrency signature increases with increasing the length of the signature for the first 9 seconds, keeps its average maximum during the following 15 seconds and start to decrease afterwards (dash dot line). However, the confidence measure of time-quefrency signature is lower when the energy is considered (long dash dot line). Further investigations have proved good robustness of time-quefrency based confidence measure in the case of the time skew presence and resulted in 100% of confidence precision for any data with the confidence higher than 50%. In this sense, confidence estimates for time-quefrency signatures were much more reliable than confidence estimates for cross correlation – we were not able to achieve 100% of confidence precision even for much higher cross-correlation confidence thresholds.

In addition, we have found that longer signatures slightly decrease the smoothness of the estimated instant time skew trajectories. The corresponding standard deviations from expectation on described dataset are shown in the table III.

TABLE III. STANDARD DEVIATION OF TIME SKEW CORRECTION

Algorithm	*σ*
Time-quefrency signatures of 10 s	1.3%
Time-quefrency signatures of 20 s	2.1%
Time-quefrency signatures of 30 s	3.2%

σ – standard deviation

From the table III we can see that the standard deviation improves with shortening of the signatures (smaller values are better). While on described dataset the optimal estimated length of the signature was 10 s, it is worth mentioning that estimated confidence distribution and standard deviation values are dependent on relative time velocity ratio and could be different on other datasets.

V. TIME SKEW CORRECTION

Time skew correction based on estimated relative time velocity ratio can be achieved by different ways: lossy data correction, lossy metadata correction, lossless correction. First two approaches modify original files, therefore the multisource recordings can be analysed directly and replayed with most of the players. Nevertheless it is impossible to say which of several absolute time velocities is correct and which is "broken", we can detect and correct only relative time skew issue. Lossless correction is based on creation of a wrapper for the post-processing software / player and therefore does not affect the initial multisource recordings. The main drawback of the third approach – wrappers are normally compatible only with limited set of multisource players / analysis software. The main advantage of the third approach – it retains initial data and gives a flexibility to change a reference time velocity without lossy impact on the data or metadata.

A. *Lossy data correction*

Lossy video data correction of time skew issue can be achieved with motion compensated frame rate conversion techniques [9], [10]. The video frame rate conversion is also available in many commercial and open source video editing tools (e.g., VirtualDub software [11]). The mentioned open source video editing tool can be used for lossy audio data time skew correction as well via audio sample rate conversion. Lossy data correction allows to keep initial metadata about video fps and audio sample rate in the header, though the quality of the media data slightly degrades each time the correction is applied.

B. *Lossy metadata correction*

Another solution is to keep initial media data and to change only the metadata in the header. This can be achieved via source rate adjustment inside many video editing tools (e.g., VirtualDub software [11] for video metadata correction) or direct hex editing of the header (valid for both audio and video). Most of post-processing algorithms and media players, which rely on metadata information, correctly treat the data regardless “non-standard” values. Nevertheless some compatibility issues can appear in the case a player tries to initialise audio hardware to unsupported sampling rate output.

C. *Lossless correction*

There are many ways to achieve lossless correction via wrapper creation. In this section, we would like to stress most generic solutions from our point of view. One of the solutions would be to use of the native support of audio and video playback in HTML5 web pages [12] to store the information about initial offset and relative time velocity ratio within associated media element attributes “startTime” and “defaultPlaybackRate” for corresponding recordings. Another solution would be to use Time Manipulation module within XHTML+SMIL profile [13] for application of the relative time velocity ratio and Timing module within the same profile for initial offset.

VI. CONCLUSION

We have shown that the time skew within multisource audiovisual data can be detected and eliminated reliably using audio features typical of ASR applications. We have proved experimentally the generalization of temporal alignment and corresponding confidence estimation to the case of time skew presence. We found that good results can be inferred from relatively short time-quefrency signatures. The corresponding results surpass the performance of fast cross correlation, while require less resources. We hope that this work might be useful for the research community as a fast and reliable approach for automatic time skew detection and correction within multisource media datasets.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement no. ICT-2007-214793.

REFERENCES

[1] Integrating project within the European research programme 7, "Together anywhere, together anytime", http://www.ta2-project.eu, 2008.

[2] S. Duffner, P. Motlicek, and D. Korchagin, "The TA2 database: a multi-modal database from home entertainment", Proceedings of the 3rd International Conference on Signal Acquisition and Processing (ICSAP), Singapore, 2011.

[3] J.-M. Verrier, "Audio boards and video synchronisation", Proceedings of the AES UK 14th Conference: Audio - The Second Century, London, UK, 1999.

[4] D.L. Mills, "Internet time synchronization: the network time protocol", IEEE Transactions on Communications, vol. 39 , iss. 10, pp. 1482-1493, 1991.

[5] D. Korchagin, P. N. Garner, and J. Dines, "Automatic temporal alignment of AV data with confidence estimation", Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Dallas, USA, 2010.

[6] Open source multiformat multimedia conversion tool "FFMPEG", http://www.ffmpeg.org.

[7] D. Korchagin, "Out-of-scene AV data detection", Proceedings IADIS International Conference on Applied Computing, vol. 2, pp. 244-248, Rome, Italy, 2009.

[8] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental", In Pattern Recognition and Artificial Intelligence, C. H. Chen, Ed., pp. 374-388, Academic, New York, USA, 1976.

[9] K. Sugiyama, T. Aoki, S. Hangai, "Motion compensated frame rate conversion using normalized motion estimation", Proceedings of IEEE Workshop on Signal Processing Systems Design and Implementation, pp. 663-668, Athens, Greece, 2005.

[10] J.Y. Zhao, D. Eryan Liu, "An efficient motion estimation algorithm for frame rate conversion", Proceedings of Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), pp. 97-100, Shanghai, China, 2010.

[11] Open source video capture and processing program "VirtualDub", http://www.virtualdub.org/.

[12] World Wide Web Consortium (W3C), "HTML5: A vocabulary and associated APIs for HTML and XHTML", 2010, http://www.w3.org/TR/html5/.

[13] World Wide Web Consortium (W3C), "XHTML+SMIL profile", 2002, http://www.w3.org/TR/XHTMLplusSMIL/.