ConCor+: Robust and Confident Video Synchronization using Consensus-based Cross-Correlation

A. Al-Nuaimi [#1], B. Cizmeci [#1], F. Schweiger [#1], R. Katz [*2], S. Taifour [#1], E. Steinbach [#1], M. Fahrmair [*2]

[#] *Institute for Media Technology, Technische Universitaet Muenchen*
Munich, Germany
[1] {anas.alnuaimi,burak.cizmeci,florian.schweiger,staii,eckehard.steinbach}@tum.de

[*] *Docomo Communications Laboratories Europe*
Munich, Germany
[2] {katz,fahrmair}@docomolab-euro.com

***Abstract*—Consensus-based Cross-correlation (ConCor) is a recently presented algorithm for robust synchronization of noisy and corrupted signals. ConCor has a number of interdependent parameters that need to be set correctly to guarantee good performance. In this paper we analyse the effects of the individual parameters on ConCor's behaviour and performance. As a second contribution, we show that a parameter sweep with subsequent majority voting can be used to boost ConCor's performance and produce a trustworthy confidence measure. As a final contribution we show how the proposed extension also allows performing multi-modal (joint audio-video) synchronization of casual multi-perspective video recordings enabling superior matching performance.**

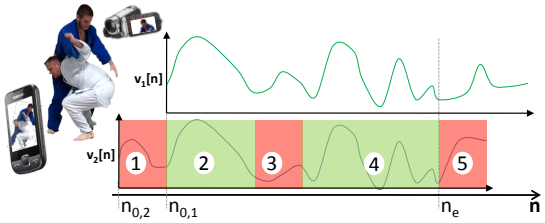

Fig. 1. The objective of video synchronization is to find the offset of v_2 w.r.t v_1 ($\Delta k_o = n_{0,2} - n_{0,1}$). ConCor automatically removes the contributions of segments 1, 3 and 5 (mismatch due to non-overlap, occlusions, coding artifacts, etc.).

I. Introduction

Many media applications require the matching and synchronization of media content. The goal of video synchronization is to find the temporal offset between two related videos, Δk_o, stemming from the different starting times of recording as depicted in Figure 1. We focus in this paper on the topic of video synchronization of *multi-perspective events*. These events refer to activities that are captured simultaneously by different viewers from various viewpoints using mobile devices. Such scenarios occur quite frequently in the context of *User Generated Content* (UGC). Popular examples are street performances or family events. Temporally synchronizing the perspectives allows for a number of applications most notably *Unstructured Video-based Rendering* [1], interactive view switching [2] as well as novel community-based video sharing and editing tools.

The authors of [3] recently presented an algorithm for the robust synchronization of signals. The method uses a consensus forming mechanism to determine which signal parts should participate in the matching operation and is hence called consensus-based cross-correlation (ConCor). Two media synchronization applications, temporal synchronization of two video recordings using their bitrate profiles and image template matching, are presented in [3]. They qualitatively show the benefits of ConCor whose performance is governed by a number of parameters that are heuristically tuned [3]. The performance as a function of the algorithm parameters has not been investigated in [3]. Understanding the effect of the individual parameters allows determining well-performing parameter sets which is crucial for successful deployment in media synchronization applications. Furthermore, large-scale evaluation has not been performed. This is necessary to quantify the gains and improve the algorithm.

In this paper, we analyze the behaviour of ConCor as a function of the different parameters. Based on the analysis we propose a generic signal corruption model which we use to link the interdependent parameters in a meaningful way. We use this model coupled with a mechanism for the fusion of multiple synchronizations – using parameter sweep with majority voting – to robustify ConCor. The resulting extension of ConCor, which we call ConCor+, increases the synchronization performance substantially and furthermore results in a trustworthy result *confidence measure*. Finally, we show that ConCor+ allows for straightforward joint audio/video (AV) synchronization with even better performance.

MMSP'10, October 4-6, 2010, Saint-Malo, France.
???-?-????-????-?/10/$??.??

The remainder of the paper is organized as follows: The main idea of video synchronization via bitrate profile matching is briefly explained in Section II. In Section III, the ConCor algorithm is reviewed and the effects of tuning the introduced algorithm parameters are studied. In Section IV, a signal corruption model that takes the parameter dependencies into account is proposed and the ConCor+ extension is introduced. Joint AV synchronization using ConCor+ is explained in Section V alongside with our evaluation results.

II. Video Synchronization via Bitrate-Profile Cross-Correlation

Today's video compression standards reduce the video bitrate substantially by exploiting the temporal correlation using motion-compensated prediction (MCP): Video parts exhibiting low amounts of scene change are predicted effectively from previous frames requiring small amounts of residual error coding [4]. This results in a smaller video data rate as compared to the parts with unpredictable high dynamic content.

A. Bitrate-based Synchronization

The authors in [5] suggest to use the bitrate profiles (BPs) of compressed video as a one-dimensional representation of a video for the purpose of video synchronization. They argue that the conditional frame entropy over time provides a high-level fingerprint of a video and that the BPs of coarsely quantized P-frames sufficiently approximate this measure. On a high level, they measure the coincidence of scene motion between views over time with the assumption of expecting similar bit rate costs across views. Figure 1 shows how this works in concept. Each perspective is represented using the BP describing the instantaneous bitrate at each frame of the video. High bitrate values imply high amount of motion or scene change. Correlating the bitrate curves can identify the offset Δk_o which maximizes the cross-correlation.

This method has been found to be remarkably view-point invariant and of low complexity but requires an algorithm that can perform cross-correlation that is robust to occasional mismatches and non-overlaps [5]. Such a method (named ConCor) has been developed in [3] and is reviewed in Section III.

B. Video Synchronization Dataset

Our dataset comprises 43 *video sets* of amateur multi-perspective scenes. No restrictions were placed on view angles, neither on camera type nor scene type and environment. Cameras were not required to be static. Each video set contains between 2 and 6 *views*. From here on, the term *video* shall refer to a view and all the data it carries (visual and auditory tracks)[1]. Videos can only be synchronized with other videos in the same video set. A total of 164 synchronizable *video pairs* are available for synchronization. Only synchronization attempts that resulted in an offset within 4 frames from the manually determined ground truth are considered successful. Our dataset is at least 1 order of magnitude larger than those of [3], [6], [5], [7].

[1]Sample video sets can be downloaded at: http://www.lmt.ei.tum.de/team/florian/sync/index.php

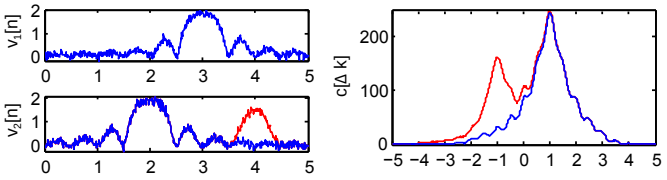

Fig. 2. The red signal in v_2 is an additive corruption. It would lead to the false local maximum highlighted in red in the cross-correlation function. Identifying and removing such corruption can remove false maxima and avoid mismatches.

III. Overview of ConCor

ConCor performs robust signal matching. The idea is to mitigate the effects of distortions to either of the signals to be matched. Example distortions are shown in Figure 2 in red color. Non overlapping parts in the signals can also be viewed as such since they might also lead to erroneous results.

To see how ConCor can handle such distortions we need to observe that the cross-correlation of signals $v_1[n]$ and $v_2[n]$ can be re-written as [3]:

$$\begin{aligned} c[\Delta k] = (v_1 \star v_2)[\Delta k] &= \sum_{n\in\mathcal{K}} v_1(n+\Delta k)\; v_2[n] \\ &= \sum_{n\in\mathcal{K}} v_1[n+\Delta k] \sum_{i}^{m} v_{2,i}[n] \\ &= \sum_{i}^{m}\sum_{n\in\mathcal{K}} v_1[n+\Delta k]\; v_{2,i}[n] = \sum_{i}^{m} c_i[\Delta k] \end{aligned}$$

where $\mathcal{K}$ is the set of sample indices where both $v_1[n+\Delta k]$ and $v_2[n]$ are non-zero, and $v_2[n] = \sum_i^m v_{2,i}$. This implies that partial cross-correlation functions (denoted as $c_i[\Delta k]$ and abbreviated as PCCFs) between v_1 and segments of v_2 (also called *snippets* and denoted as $v_{2,i}[n]$) can be pre-computed and then added up together to form the true cross-correlation. If some of the segments are actually corrupted signal parts that would otherwise cause mismatches and false maxima, it is desirable not to include these in the final summation (c.f. Figure 2). The goal is to calculate an *approximate cross-correlation function* (ACCF) *from the matching parts only*.

Identifying the corrupt (outlier) segments of the signal and discarding their contribution is achieved in ConCor by using a RANSAC-based model fitting [8]. RANSAC can robustly fit data to a given model and discard outliers by iteratively choosing a subset of the data and calculating the model parameters. The remaining data points are then validated against the fitted model. The model with which the highest number of datapoints agree is deemed as the consensus model [8]. In ConCor, the model to be fitted is the index corresponding to the maximum value of the cross-correlation function which identifies the offset between the two signals. Accordingly, a subset of $s(< m)$ PCCFs is summed up to an ACCF whose maximum forms an *offset hypothesis*. The remaining PCCFs can vote for the hypothesis if they have a maximum that lies within $\pm\Delta k_T$ from the offset hypothesis. This process can be repeated for a suitable subset of all possible *combinations*

of s PCCFs and the offset hypothesis which gets the highest number of votes is deemed as the consensus hypothesis, i.e. the found offset. If v_1 and v_2 are the longer and shorter of the two bitrate profiles (BPs) to be temporally matched, we can summarize ConCor as follows:

— SIGNAL SEGMENTATION AND PRE-PROCESSING.

PRE1 Chop signal v_2 in segments $v_{2,i}$ of length M

PRE2 Compute the PCCFs $c_i[\Delta k] = (v_1 \star v_{2,i})[\Delta k]$

PRE3 Set max. number of iterations $N_{max} < \begin{pmatrix} m \\ s \end{pmatrix}$.

— RANSAC-BASED OFFSET DETERMINATION.

RAN0 $i \leftarrow 1$

RAN1 Make a random selection of s PCCFs and compute their sum, the ACCF $\tilde{c}_j[\Delta k]$

RAN2 Extract candidate offsets (locations of local maxima) from that sum.

RAN3 For every offset candidate, evaluate the number of inliers among all PCCFs (An inlier is a PCCF which has a local maxima within $\pm\Delta k_T$ frames from the candidate offset); update $i \leftarrow i + 1$;

RAN4 if $i < N_{max}$ goto RAN1 else goto RES.

— PICK RESULT

RES estimated offset $\tilde{\Delta k}_0 =$ offset with most inliers

IV. CONCOR ANALYSIS AND EXTENSIONS

Based on a parameter analysis of ConCor (Section IV-A) we propose in Section IV-B a signal corruption model which links together the mentioned parameters and takes into account their interdependency. In Section IV-C we propose a scheme for the fusion of multiple synchronization results which are obtained with different parameter settings.

A. ConCor Parameter Analysis

ConCor's performance is a function of the parameters s (number of PCCFs to sum in any iteration), M (segment length) and N_{max} (maximum number of iterations). These parameters are heuristically set in [3] and no quantitative analysis has been performed. Choosing a good set of parameters is not a trivial task. For that we shall analyse their effects.

Number of PCCFs to combine ($\mathbf{s}$): Assuming that the segments are independently chosen, the probability that at a certain trial all s picked segments are not corrupted is:

$$P_{ts} = (1 - P_o)^s \tag{1}$$

where P_o is the probability that one datapoint (a segment in this case) is an outlier. The complimentary probability P_{tf} which expresses the probability of including at least 1 corrupt segment is given as:

$$P_{tf} = 1 - P_{ts} = 1 - (1 - P_o)^s \tag{2}$$

The higher s is, the higher is P_{tf} and hence the bigger the chance of estimating the offset wrongly. Reducing the number of PCCFs to combine at each iteration, however, implies that the ACCF is made up of a small part of the shorter of the two signals and hence may not reliably approximate the true outlier-free cross-correlation signal. Table I presents the percentage of correctly synchronized video pairs as a function of s for the test dataset described in Section II-B. It can be observed that the results do not substantially vary when changing s, however, a more detailed inspection of the synchronization results unveils the following: ConCor manages to synchronize different pairs when changing s. This is better seen under the column $s*$. This represents the subset of all video pairs that were correctly synchronized with *at least one* of the ten configurations $s \in [2, 11]$. The fraction of correct offsets is 10% higher over all configurations taken seperately. This motivates devising a scheme that can fuse multiple synchronization results obtained with different parameter settings (see Section IV-C).

TABLE I
PERCENTAGE OF CORRECTLY SYNCED VIDEO PAIRS AS A FUNCTION OF s ($m = 12$)

s	2	3	4	5	6	7	8	9	10	$s*$
P(%)	58	60	59	58	58	57	58	58	59	68

Maximum number of RANSAC *iterations ($\mathbf{N_{max}}$):* To validate all offset hypotheses, all possible combinations of s PCCF sums out of m PCCFs should be considered:

$$N_{max} < \begin{pmatrix} m \\ s \end{pmatrix} \tag{3}$$

Since this number is typically prohibitive, RANSAC is based on randomly sampling a subset of the entire set of possible combinations [8]. The number of trials can be set to guarantee a certain success probability P_s of finding at least one outlier-free set – assuming there is one. For that, the probability that a picked PCCF is an outlier, P_o, has to be known (or assumed). Ensuring a success probability of P_s implies that N_{max} should be picked such that the failure probability is below $(1 - P_s)$. The failure occurs if the random process fails to pick an outlier-free set during any of the N_{max} iterations. Hence we require:

$$(P_{tf})^{N_{max}} < 1 - P_s \tag{4}$$

By solving expression (4) for the smallest N_{max}, the required number of iterations is obtained:

$$N_{max} = \frac{\ln(1 - P_s)}{\ln(1 - (1 - P_o)^s)} \tag{5}$$

Segment Length ($\mathbf{M}$): A large segment length increases the probability that a chosen segment contains corrupted parts. A small segment length, on the other hand, mitigates this problem but produces non-reliable PCCFs: Consider the extreme case of $M = 1$; A PCCF between a segment of length 1 and v_1 is a scaled version of v_1 and does not carry any alignment information. Table II underscores these expectations: it can be seen that as M is increased, the precision increases. With high segment length, however, this precision drops again. This is also true when the segment length is a constant fraction of the signal length ($M = L_2/m$), where L_2 is the length of the shorter of the two signals.

TABLE II
PERCENTAGE OF CORRECTLY SYNCED VIDEO PAIRS AS A FUNCTION OF CONSTANT SEGMENT LENGTH ($s = 4$).

M	Precision
10	45%
30	51%
50	58%
70	52%

m	Precision
30	53%
20	60%
10	59%

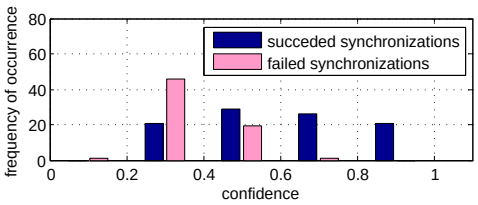

Fig. 3. Frequency of occurrence of the RANSAC-based ConCor confidence values for the $s = 4$ run (Table I).

B. Signal Corruption Model

Running the tests of Table I for a different m produces different results but with the same performance tendency. To cater for the interdependency of s and M, we propose a signal corruption model that links the two together. The model is parametrized by the expected fraction of frames that belong to outlier segments, f. In Figure 1 this would be the ratio of the length of the red highlighted segments w.r.t. the entire length of the signal. Accordingly, the length of the usable part of the shorter signal can be stated as $L_2 \cdot (1 - f)$ where L_2 is the total length of $v_2[n]$. The objective is to use the entire uncorrupted part of the signal for the synchronization. Hence, the upper bound on the snippet length, M, is derived for a given s and f as follows:

$$M \leq \frac{L_2 \cdot (1 - f)}{s} \tag{6}$$

From this inequality, the upper bound is picked as the snippet length so as to keep the computational cost as small as possible. Knowing that $P_o(M{=}1) = f$ and $P_o(M{=}L_2) \approx 1$, we can – for matters of simplicity – linearly interpolate the probability that a segment of length M is an outlier as in:

$$P_o(M) \approx f + \frac{M \cdot (1 - f)}{L_2} = f \cdot \left(1 - \frac{M}{L_2}\right) + \frac{M}{L_2} \tag{7}$$

C. Improved Robustness and Trustworthy Confidence Value using ConCor+ (Parameter Sweep With Majority)

With the model presented in Section IV-B the interdependency of M and s has been accounted for and the N_{max} is readily calculated using expression (5). Yet, for this model to work successfully, a mechanism has to be devised that accurately estimates the fraction of corrupted signal parts. Furthermore, one of either of the two parameters s and M has to be set.

A numerical optimization scheme that determines the optimal s (or M) and f over a large dataset might deliver a parameter set that performs best on average but the goal is to ensure best performance for each individual attempt. Furthermore, we have shown that varying s does not impact aggregate performance significantly, but is beneficial if the multiple synchronization results per pair can be combined (see Table I). This combination can furthermore be motivated by the expectation of a more trustworthy result. This expectation in increase in confidence through multiple runs is inspired by point cloud matching using the *Iterative Closest Point* algorithm (ICP), where the algorithm is rerun multiple times after applying an initial distortion to the point cloud [9]. If the distribution of the registration results shows little variance, then the point cloud is thought to have salient features and the results can be deemed trustworthy. The variance in the registration results is then used as confidence measure. The confidence measure of the original ConCor, given by dividing the number of inlier segments by the total number of segments, has been determined not to be practically useful: Figure 3 shows that failed synchronizations can often return a high confidence value and that successful synchronizations not rarely return low confidences.

Inspired by [9] and motivated by the conclusions from Section IV-A we extend ConCor by a parameter sweep with majority voting and name the extended sync method *ConCor+*. Specifically, we propose to run a series of synchronization attempts on a pair of signals by varying s. By this we ensure that the signals are tested for salient features at various scale levels since changing s implies changing M due to the introduced corruption model. We produce one offset result by determining the majority synchronization result. The confidence value of the resulting majority offset is given by dividing the size of the majority by the total number of individual syncs.
The majority is, however, often not readily obtainable. Simple median filtering or counting-based majority determination have their limitations. Consider for example following set of possible results from a parameter sweep on a video pair: $\mathbf{\Delta k}_{sweep} = [156, 155, 154, 10, 10, 0]$. The majority offset Δk_{maj} determined via a *matching values counter* would be ($\Delta k_{maj} = 10$) which is wrong. Similarly a median operator would be off the correct result. To address this issue, we developed a scheme that determines a majority in which the individual values span a range of maximally t frames.

Once the majority offset has been calculated with its associated majority size, we propose accepting the offset based on the *majority decision criterion*: The majority has to constitute at least 50% of all runs on the pair. If no $\geq 50\%$ majority exists, a `no sync` is output. Consequently, the confidence value for any resulting majority offset is always ≥ 0.5 (as will be observed later in Figure 4).
From here on we shall denote the set of video pairs that successfully sync using ConCor+ as $\mathcal{S}(X_M)$ where X denotes the signal source ($X = V$ for BPs and $X = A$ for the audio signal case, etc.). $\bar{\mathcal{S}}(X_M)$ is the complimentary set of video pairs for which a wrong majority is returned (**pairs returning `no sync` are not included**).

TABLE III
RESULTS FOR PARAMETER SWEEP AND CONCOR+ ($f = 0.4$).

test Description	x	$\lvert\mathcal{S}(x)\rvert$	$\lvert\bar{\mathcal{S}}(x)\rvert$	$P(\%)$	$R(\%)$
ConCor($s = 2$) on BPs	V_2	90	74	54,9	54,9
ConCor($s = 3$) on BPs	V_3	96	68	58,5	58,5
ConCor($s = 4$) on BPs	V_4	97	67	59,1	59,1
ConCor($s = 5$) on BPs	V_5	92	72	56,1	56,1
ConCor($s = 6$) on BPs	V_6	93	71	56,7	56,7
Any of the 5 runs succeded	$V_{2:6}$	133	-	-	81,1
ConCor+ on BPs	V_M	93	10	90,3	56,7

D. Results

We evaluate the gains of using the signal corruption model with ConCor+ ($t = 6$) on the BPs using the previously introduced dataset. Before the results are analysed we shall also introduce the following sets: $\mathcal{S}(O)$ is the set made up of all video pairs. In our dataset $|\mathcal{S}(O)| = 164$ (see Section II-B). $\mathcal{S}(V_i)$ is the subset of correctly synchronized video pairs using their BPs when $s = i$; $\bar{\mathcal{S}}(V_i)$ is the complimentary set of wrongly synchronized video sets ($\bar{\mathcal{S}}(V_i) = \mathcal{S}(O)/\mathcal{S}(V_i)$). $\mathcal{S}(V_{2:6})$ is the subset of all video pairs that were correctly synchronized using the BPs with *at least one* of the five configurations ($s = 2 \rightarrow 6$), hence $\bigcup_{i=2}^{6} \mathcal{S}(V_i) = \mathcal{S}(V_{2:6})$. These definitions allow us to measure *precision* (P) and *recall* (R) for any test (x) as defined by the theory of information retrieval [10]:

$$P(\mathcal{S}(x)) = \frac{|\mathcal{S}(x)|}{|\mathcal{S}(x)| + |\bar{\mathcal{S}}(x)|} \quad R(\mathcal{S}(x)) = \frac{|\mathcal{S}(x)|}{|\mathcal{S}(O)|}$$

As in Table I, Table III shows similar effects of varying s on the performance. However, now M and N_{max} are linked directly using the model introduced in Section IV-B. As a consequence an even higher increase in the number of video pairs that manage to synchronize successfully with at least one of the configurations is observed although less configurations are tested per video pair (5 compared to 10 in Table I). It is seen that the number of synchronizable video pairs with at least 1 of the configurations rises dramatically to 133/164 compared to 97/164 in the best individual configuration ($s = 4$). This means there is much to be gained from using ConCor+. Indeed, the configuration denoted by $\mathcal{S}(V_M)$ shows a dramatic improvement in precision with no degradation in recall thanks to the majority forming process which requires agreement by at least three runs ($\geq 50\%$) to accept a majority offset.

We can now validate the expected improvement in the confidence measure highlighted in Section IV-C. The new confidence measure – the cardinality of the majority – is particularly trustworthy due to the high precision of ConCor+. Figure 4 shows first of all the huge gain in precision (reduction in red area). At the same time it is seen that a wrong offset can rarely occur in association with a high confidence value (c.f. Figure 3).

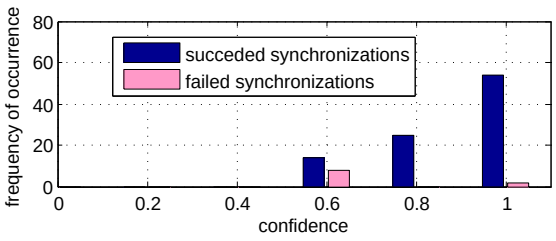

Fig. 4. Frequency of occurrence of confidence values of ConCor+.

V. JOINT AUDIO-VIDEO SYNCHRONIZATION USING CONCOR+

UGC videos most often carry audio tracks as well. Using the original ConCor in [3] does not allow us to make joint use of the audio and video tracks for synchronization purposes since the normal ConCor does not output a trustworthy confidence measure. Hence, any methods for joint AV synchronization that rely on taking the result with the higher confidence, or perform a weighted averaging based on the confidence are likely to fail. ConCor+ outputs a reliable confidence value enabling a simple fusion scheme with hard decisions (taking the result with the higher confidence). A more intelligent approach can be conceived which will be explained later. Before that we will first provide an overview of audio synchronization methods that justifies performing synchronization of the audio tracks using cross-correlation (CC). The implemented AV synchronization method is then explained in further detail. Finally the results for performing AV sync are shown and compared to those stemming from using the audio modality or the video modality only.

A. Principles of Audio Synchronization

A departure from the standard CC approach is used in [11] to synchronize music. Audio features are extracted and a dynamic time warping algorithm is used to align two musical tracks. This approach assumes the tracks being synchronized have similar musical characteristics with possible stretches and compressions in the playback timing. Other approaches are also based on extracting audio features, such as presented in [12] and the one presented in [13] which use dynamic time warping on low-level data. It is also possible to use similar approaches to synchronize a musical track to a musical score as in [14].

The previously mentioned approaches work around stretches and compressions in time, which do not appear in the case of our application. Furthermore, they assume the input is a music track. To overcome this limitation, Shrestha et al. in [6] use an audio-classifier before performing a CC, thus only performing CC on comparable and relevant segments of the audio, with compelling results. ConCor+ provides a similar advantage by performing robust and confident synchronization. It also allows the consistent merging of audio synchronization with bitrate profile-based synchronization, for superior results, as shown in the following.

B. Implemented Approach

Performing CC (which is what ConCor+ also does) can be computationally intensive for audio signals. If a video is 60s long and has a frame rate of 25fps, it would have 1500 video frames (length of BP is 1500). The contained audio, if sampled at 48KHz (which is typical) would be 2,880,000 samples long! CC can quickly overwhelm the memory and require excessive computation time. We choose to subsample the absolute value of the signal to 25Hz since we are only interested in the loudness profile. The latter is also important since audio signals typically have almost no energy in the low frequencies. Hence, we obtain a rough low-frequency representation of the loudness of the audio.

In the first test we perform audio synchronization using normal CC whenever loudness profiles are available (117 of the 164 pairs to be matched have audio data which corresponds to about 71% of all pairs). In the second test, we run ConCor+ on the loudness profiles. Finally, we run a test in which ConCor+ is run on the loudness profiles and the BPs separately. We then concatenate the offsets for audio (whenever available) with those from the video synchronization and determine the majority offset. If no audio offsets are available, only the five video offsets are used.

C. Results

We define the following sets:
1) $\mathcal{S}(A_{CC})$ comprises all video pairs whose loudness profiles synced successfully using cross-correlation and $\bar{\mathcal{S}}(A_{CC})$ to be the complimentary set containing those pairs that failed (which does not include pairs that do not have audio tracks).
2) $\mathcal{S}(A_M)$ represents all the video pairs which successfully synced using their loudness profiles only with ConCor+ and $\bar{\mathcal{S}}(A_M)$ is the set of pairs that returned a wrong majority offset.
3) $\mathcal{S}(AV_M)$ contains all pairs that synced correctly by running ConCor+ on the loudness profiles and the video bitrate profiles separately and performing majority voting on the concatenated offset values. $\bar{\mathcal{S}}(AV_M)$ being again the complimentary set.

Comparing the first two rows in Table IV demonstrates the large increase in precision when using ConCor+ on the loudness profiles instead of normal CC. Moreover the recall increases by about 7%. The results in the second and third row show that audio syncs over all better than video (higher recall although only 71% of all video pairs have audio tracks). The audio signal is after all a natural signal that is truly view-point invariant. Finally it can be seen that the joint AV synchronization using ConCor+ delivers better results than any individual method. Synergetic effects are observed when investigating the results (BPs compensate missing loudness profiles and audio compensates precision issues with video when available).

TABLE IV
JOINT AUDIO-VIDEO SYNCHRONIZATION RESULTS.

Test Description	x	$\lvert\mathcal{S}(x)\rvert$	$\lvert\bar{\mathcal{S}}(x)\rvert$	$P(\%)$	$R(\%)$
cross-correlation Audio	A_{CC}	97	20	82,9	59,2
ConCor+ Audio	A_M	108	8	93,1	65,9
ConCor+ Video	V_M	93	10	90,3	56,7
ConCor+ AV	AV_M	129	6	95,4	75,0

VI. Conclusion

In this paper we investigate the effects of parameter tuning on the behaviour and performance of a recently developed algorithm for robust synchronization, ConCor. Based on the lessons learned we propose a generic signal corruption model that helps us to deal with interrelations of the algorithm's parameters. We use this model in an extension of the algorithm, which we call ConCor+, that involves the fusion of multiple synchronization runs using a majority voting scheme. The proposed method also raises the trustworthiness of the delivered confidence value in the result markedly. We finally show that ConCor+ can be used in a straightforward manner for AV synchronization of casually recorded multi-perspective videos achieving very good performance. We particularly show that the complimentary nature of video and audio raises recall by about 20% compared to video synchronization alone.

References

[1] L. Ballan, G. J. Brostow, J. Puwein, and M. Pollefeys, "Unstructured video-based rendering: interactive exploration of casually captured videos," in *ACM SIGGRAPH 2010 papers*, ser. SIGGRAPH '10. New York, NY, USA: ACM, 2010, pp. 87:1–87:11. [Online]. Available: http://doi.acm.org/10.1145/1833349.1778824

[2] F. Schweiger, E. Steinbach, M. Fahrmair, and W. Kellerer, "CAMP: A framework for cooperation among mobile prosumers," in *IEEE ICME 2009 Workshop on Community driven Mobile Multimedia*, New York, USA, Jun 2009.

[3] F. Schweiger, G. Schroth, M. Eichhorn, E. Steinbach, and M. Fahrmair, "Consensus-based cross-correlation," in *ACM Multimedia*, Scottsdale, AZ, Nov 2011.

[4] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560 –576, july 2003.

[5] G. Schroth, F. Schweiger, M. Eichhorn, E. Steinbach, M. Fahrmair, and W. Kellerer, "Video synchronization using bit rate profiles," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, sept. 2010, pp. 1549 –1552.

[6] P. Shrestha, M. Barbieri, H. Weda, and D. Sekulovski, "Synchronization of multiple camera videos using audio-visual features," *Multimedia, IEEE Transactions on*, vol. 12, no. 1, pp. 79 –92, jan. 2010.

[7] Y. Caspi and M. Irani, "A step towards sequence-to-sequence alignment," in *CVPR*, Hilton Head, SC, USA, June 2000.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[9] J. Nieto, T. Bailey, and E. Nebot, "Recursive scan-matching slam," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 39 – 49, 2007.

[10] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, 1st ed. Addison Wesley, May 1999. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/020139829X

[11] M. Müller, H. Mattes, and F. Kurth, "An efficient multiscale approach to audio synchronization," in *In Proceedings of the 6th International Conference on Music Information Retrieval*, 2006, pp. 192–197.

[12] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *IEEE ICASSP 6th International Conference on Music Information Retrieval*, Taipei, April 2009, pp. 1869 – 1872.

[13] S. Dixon and G. Widmer, "MATCH: A music alignment tool chest."

[14] F. Soulez, X. Rodet, and D. Schwarz, "Improving polyphonic and poly-instrumental music to score alignment," in *ISMIR*, 2003.