© 2018. Published by The Company of Biologists Ltd | Journal of Experimental Biology (2018) 221, jeb173724. doi:10.1242/jeb.173724

METHODS AND TECHNIQUES

Low-cost synchronization of high-speed audio and video recordings in bio-acoustic experiments

Dennis Laurijssen[1], Erik Verreycken[1], Inga Geipel[2,3], Walter Daems[1], Herbert Peremans[4] and Jan Steckel[1,5,*]

ABSTRACT

In this paper, we present a method for synchronizing high-speed audio and video recordings of bio-acoustic experiments. By embedding a random signal into the recorded video and audio data, robust synchronization of a diverse set of sensor streams can be performed without the need to keep detailed records. The synchronization can be performed using recording devices without dedicated synchronization inputs. We demonstrate the efficacy of the approach in two sets of experiments: behavioral experiments on different species of echolocating bats and the recordings of field crickets. We present the general operating principle of the synchronization method, discuss its synchronization strength and provide insights into how to construct such a device using off-the-shelf components.

KEY WORDS: Behavioral experiments, High-speed audio, High-speed video, Synchronization, Data acquisition

INTRODUCTION

Data acquisition in behavioral experiments on a wide range of animal species often relies on multi-modal (i.e. video and single- or multi-channel audio) sensor information (Valente et al., 2007). Indeed, many behavioral experiments on bats (e.g. Geberl et al., 2015; Geipel et al., 2013; Greif et al., 2017; Luo and Moss, 2017; Stilz, 2017; Übernickel et al., 2013), zebra finches (e.g. Ullrich et al., 2016) and even insects such as fruit flies (Coen and Murthy, 2016) rely on capturing synchronized high-speed video and single- or multi-channel audio data. Having an accurate measure of the relative time shift between the individual sensor data streams is essential for subsequent analysis of the sensor data. Determining and compensating for the time shift between multiple-sensor data streams is referred to as time synchronization. The time synchronization of both commercial and custom-built multimodal sensor systems often relies on using a so-called trigger event, e.g. a post-trigger which is provided by the experimenter once the animal has performed its task. This finishes the data capture sequence on multiple devices simultaneously, providing a synchronization point for all the captured sequences. This is often performed by broadcasting a digital transistor-transistor level (TTL) pulse to the dedicated synchronization inputs of each individual data-acquisition device. We argue that this synchronization approach suffers from two important disadvantages. First, this approach depends on the availability of compatible synchronization inputs on the various data-acquisition devices. Furthermore, if the synchronization-handling mechanism is not implemented carefully, precise synchronization cannot be guaranteed, e.g. whenever part of the synchronization system relies on a software component running on a non-real-time operating system. Second, the downside of the digital synchronization pulse-based approach is that the synchronization information is not embedded in the data. By synchronizing either the start or the end of the captured data sequences, the relative time shift between the individual data streams, e.g. audio and video streams, can be deduced. However, this synchronization information is easily lost in the case of truncation of the data sequences. Furthermore, data sequences are often recorded in such a way that a portion of the data is recorded before the so-called trigger event, and a portion after the trigger event (pre- and post-trigger data). The information about the type of the captured data sequence needs to be recorded very carefully in metadata, which increases the risk of data loss or inconsistencies. Again, truncation of the data, i.e. throwing away uninteresting sections of the data sequences, aggravates the risk of inconsistencies.

To overcome these shortcomings of traditional synchronization techniques, we propose a method based on embedding a random 1-bit signal into the data streams directly. This type of signal is exceptionally good for alignment purposes as it exhibits a very narrow autocorrelation function. Embedding this type of synchronization signal into the recorded data sets solves both issues at once: no specialized synchronization input is needed to store synchronization information and the accuracy and precision of the synchronization do not depend on the manufacturer of the recording equipment. In addition, as the synchronization information is embedded in the data streams directly, the synchronization information can be made very robust to truncation and even re-sampling of the sensor data. It should be noted that our proposed approach is similar in concept to the SMPTE timecode system (Poynton, 1996) used in synchronization of television audio and video, the main difference being that we propose to embed a random sequence in the data instead of using structured timecode.

MATERIALS AND METHODS

Topology of the synchronization system

The proposed synchronization system consists of two main parts: a 1-bit signal generator with random periods, and one or more means of transferring the generated random sequences to the sensor modalities that require synchronization. In case of video and multi-channel audio, these means are a blinking LED on the one hand and an electrical copy of the 1-bit synchronization signal on the other hand (e.g. a TTL-compatible version with '−1' represented by 0 V and '+1' represented by 5 V). The blinking LED is recorded using the video camera, and the electrical signal is

[1]CoSys Lab, Faculty of Applied Engineering, University of Antwerp, B-2020 Antwerp, Belgium. [2]Smithsonian Tropical Research Institute, Balboa, Ancón, Republic of Panama. [3]Wissenschaftskolleg zu Berlin, 14193 Berlin, Germany. [4]APL, FTEW-ENM department, University of Antwerp, 2000 Antwerp, Belgium. [5]Flanders Make Strategic Research Centre, 3920 Lommel, Belgium.

*Author for correspondence (jan.steckel@uantwerpen.be)

J.S., 0000-0003-4489-466X

Received 6 November 2017; Accepted 8 December 2017

Journal of Experimental Biology (2018) 221, jeb173724. doi:10.1242/jeb.173724

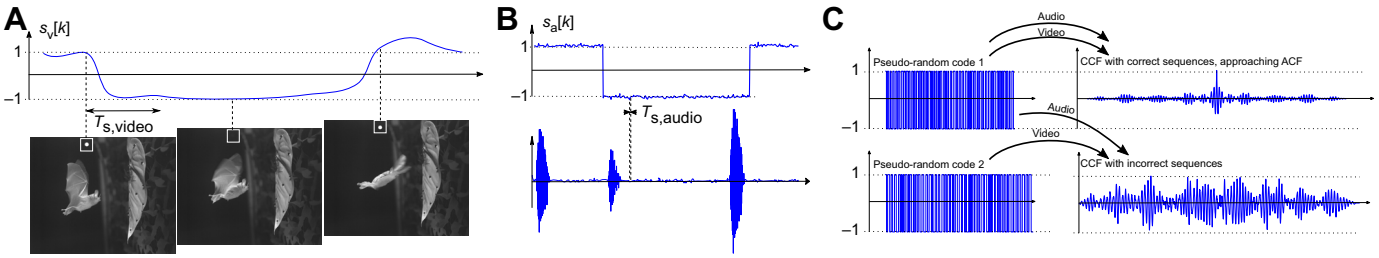

Fig. 1. The synchronization system. (A) The extraction process of the synchronization signal from a video sequence. The intensity of the region of interest is averaged for each time point and put into a time vector $s_v[k]$. This signal is then thresholded to obtain the 1-bit synchronization signal. It should be noted that this signal is sampled much more slowly (between 100 and 5000 Hz) than the audio data (around 500 kHz). (B) The synchronization signal extracted from the audio data in relation to some bat echolocation calls, sampled at 500 kHz. $T_{s,video}$ and $T_{s,audio}$ are the largest sampling period for the video and audio recording system, respectively. (C) Top row: the effect of cross-correlating two sequences that belong together, resulting in a very peaked cross-correlation function (CCF), approaching the autocorrelation function (ACF) of the pseudo-random sequence. Bottom row: what happens when two pseudo-random sequences are correlated which do not belong together. The cross-correlation function is smeared out, allowing the detection of wrongly combined measurement pairs.

recorded with the multi-channel microphone recording equipment, sacrificing one microphone channel. Indeed, many multi-channel microphone array systems can spare a single channel in return for highly accurate synchronization with video data.

Discrete signals consisting of 1-bit random sequences have the advantage of exhibiting an autocorrelation function (ACF) that in the limit of infinite duration approaches the (discrete time) unit impulse (Oppenheim et al., 1997):

$$\delta[n] = \begin{cases} 0 & \text{if } n \neq 0 \\ 1 & \text{if } n = 0 \end{cases}, \qquad (1)$$

where n is the time lag expressed in number of sample periods. Random sequences, even if only pseudo-random, also have very low cross-correlation function values (CCF) with different instantiations of the same (pseudo-random) process. In addition to being straightforward to generate, this property makes them an excellent choice for the synchronization application proposed in this paper. Indeed, as shown in Fig. 1C (upper panel), very accurate time alignment can be achieved for data sequences that belong together: they exhibit a distinctly peaked cross-correlation function. This is further illustrated in Fig. 2A and Fig. 3. Conversely, audio and video data that are incorrectly assumed to match can be easily recognized as non-corresponding through the lack of a distinct impulse-like cross-correlation peak (see Fig. 1C, lower panel).

Pseudo-random number generators

To implement the proposed synchronization system, a pseudo-random number generator (PRNG) may be used. Often, computing environments offer PRNGs that generate number sequences based on fixed algorithms or tables. The numbers generated in the

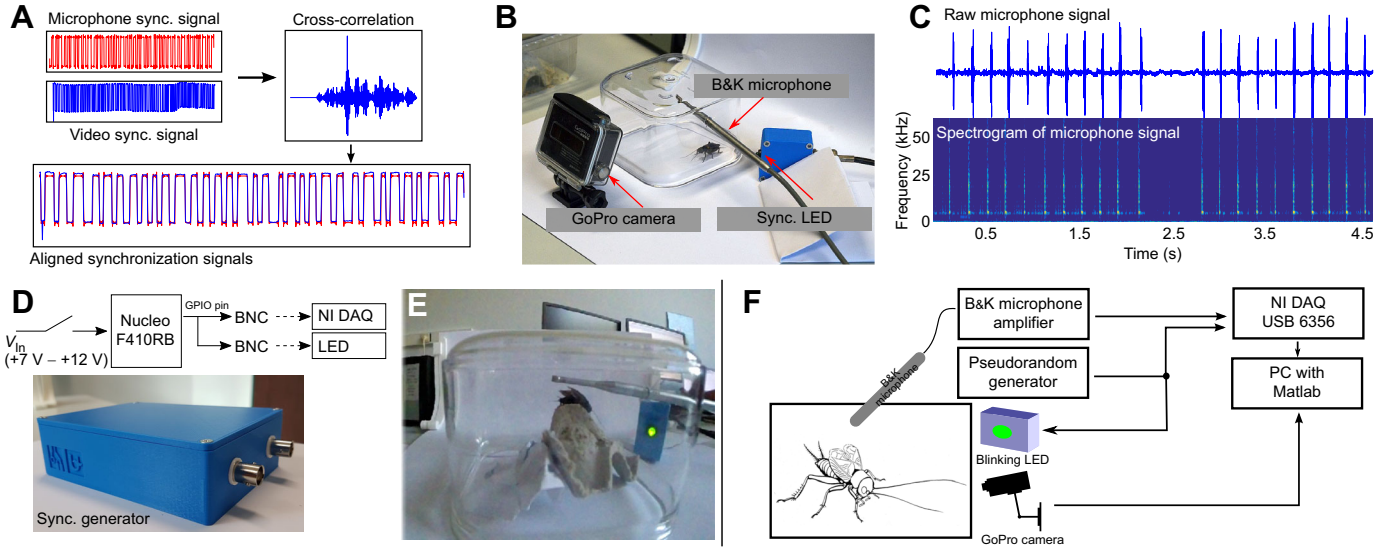

Fig. 2. Overview of the experiment in the lab environment for recording high-speed video and audio from a common field cricket (*Gryllus campestris*). (A) The extracted synchronization signals from the microphone (red) and video (blue) data streams. The cross-correlation between the two sequences exhibits a single narrow peak, which can be used to calculate the time shift between the two signals. The synchronization signals can be aligned using the calculated time shift between the two signals (bottom). (B) The experimental setup for the high-speed recordings of the field cricket: a GoPro camera running at 240 Hz, a B&K microphone and the synchronization LED (blue box) are shown. (C) The recorded microphone signal and spectrogram. (D) The synchronization generator built around a Nucleo F410RB microcontroller. (E) A single frame of the recorded video. (F) An overview of the hardware measurement setup: the B&K microphone is connected via an amplifier to a National Instruments DAQ device. The pseudo-random generator is connected to the blinking LED and to the DAQ device. The data are transferred to a PC running Matlab for further analysis.

Journal of Experimental Biology (2018) 221, jeb173724. doi:10.1242/jeb.173724

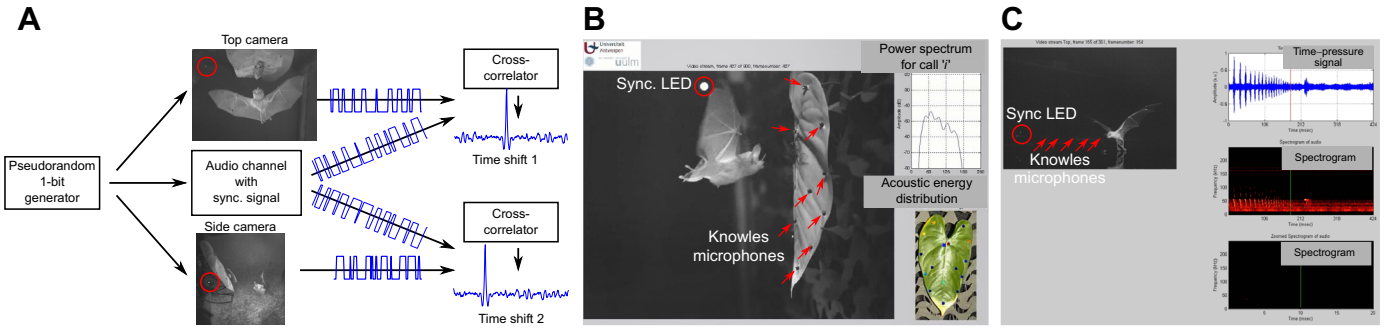

Fig. 3. Overview of the bat behavioral experiment. (A) The overall topology of the synchronization system. A 1-bit pseudo-random generator is coupled either optically through a LED into a video camera or electrically to a multi-channel microphone recording system. Cross-correlating the synchronization signals recorded by the two sensor modalities allows calculation of the exact time shift required for aligning the two sensor data sequences. Note also that by placing a separate LED in the field of view of each camera, the synchronization of multiple camera recordings can be easily accommodated. (B) An experimental setup where a *Micronycteris microtis* is capturing a dragonfly on a leaf. Sixteen Knowles FG-type microphones are mounted in the leaf surface (indicated by red arrows), which allows calculation of the distribution of the acoustic energy on the leaf surface. Video is recorded using a high-speed (500 Hz) infrared (IR) camera, and the IR-LED used for synchronization is indicated by a red circle. (C) The experimental setup used to document a *Noctilio leporinus* capturing a fish from a body of water. Again, Knowles FG-type microphones are mounted near the edge of the water surface (indicated by red arrows), and the IR-LED is indicated with a red circle. Also shown is the time–pressure signal of one of the microphone channels, and the corresponding spectrogram.

sequence exhibit a large entropy, i.e. they appear to be random, although the sequence itself is deterministic (and periodic). However, when implemented properly, the period of these generators will be very large [e.g. the Mersenne Twister PRNG (default in Matlab), exhibits a periodicity of $2^{19937}-1$ (Matsumoto and Nishimura, 1998)]. Such a PRNG can be used to generate 1-bit (pseudo-random) sequences. Assuming a 32-bit PRNG, we can derive a 1-bit pseudo-random signal by using the 32-bit random values r as values for a timer-scaler to calculate a time period (P_{wait}) lying in between a predefined minimum and maximum period (P_{min} and P_{max}). The synchronization system will pause for a period of P_{wait} before toggling the 1-bit output value, thus creating the 1-bit pseudo-random signal. The equation used for calculating the timer waiting period is:

$$P_{\text{wait}} = P_{\text{min}} + \frac{r}{2^{32}}(P_{\text{max}} - P_{\text{min}}). \quad (2)$$

Given this model, the key question is how to pick a good value for P_{min} and P_{max}. The basis for a good selection is summarized in Fig. 4. The conclusions that can be drawn from the figure lead to the following selection procedure. (1) Select the minimal period P_{min} that the output of the pseudo-random generator remains in the same state (either 1 or 0) to be $P_{\text{min}}=2T_{\text{s}}$, where T_{s} is the largest sampling period (corresponding to the slowest sampling rate) in the entire recording system. This obeys Shannon's sampling requirement to avoid missing transitions in the synchronization signal. (2) Determine the smallest fragment length L (in number of samples) one ever wants to select from a signal while still being able to synchronize it with other related signals. (3) Given L, choose as large a $P_{\text{max}}/P_{\text{min}}$ as possible, without reducing the estimated number of transitions below 10. The number of transitions K can be estimated using:

$$K = \frac{2 \cdot L \cdot T_{\text{s}}}{P_{\text{min}}\left(\frac{P_{\text{max}}}{P_{\text{min}}} + 1\right)}. \quad (3)$$

The lower limit of 10 transitions can be considered to be constant with respect to the fragment length. Fig. 5 illustrates the relationship between the number of transitions and the strength of the autocorrelation peak.

Note that in Fig. 4, the total fragment length has been chosen to be rather short (only 100 samples), leading to a very limited range of proper P_{min} and P_{max} values, and a relatively high risk of meeting an odd failure of the synchronization principle. The short length has been specifically chosen to illustrate these effects. In practical cases, the length of the fragment will be significantly greater, resulting in a larger valid range and a vanishing chance of

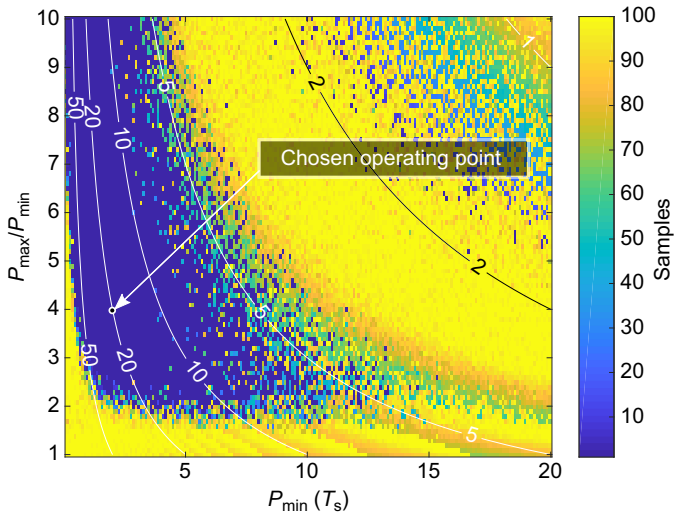

Fig. 4. Maximum alignment error of a 100-sample signal fragment observed over N=1000 simulations as a function of P_{max} and $P_{\text{max}}/P_{\text{min}}$. Maximum alignment error (N=1000) is shown as the number of sample periods (T_{s}). A uniform sample desynchronization in $[-T_{\text{s}}/2, T_{\text{s}}/2]$ was assumed. Contours are superimposed on the graph, indicating the expected number of transitions of the synchronization signal on the total fragment length of 100. As can be clearly seen, choosing $P_{\text{min}} \leq 2T_{\text{s}}$ (where P_{min} is minimum period) is disadvantageous as it causes transitions to be skipped in the recorded signal fragment. This corresponds to the requirement of obeying Shannon's sampling theorem. In addition, taking $P_{\text{max}}/P_{\text{min}}$ (where P_{max} is maximum period) too low (close to 1) increases the risk of yielding periodic synchronization signals for smaller fragments. Likewise, the combination of P_{min} and P_{max} should be such that on average at least 10 transitions remain in the smallest fragment one wants to consider. The operating point chosen for the example in the text has been indicated on the graph.

Journal of Experimental Biology (2018) 221, jeb173724. doi:10.1242/jeb.173724

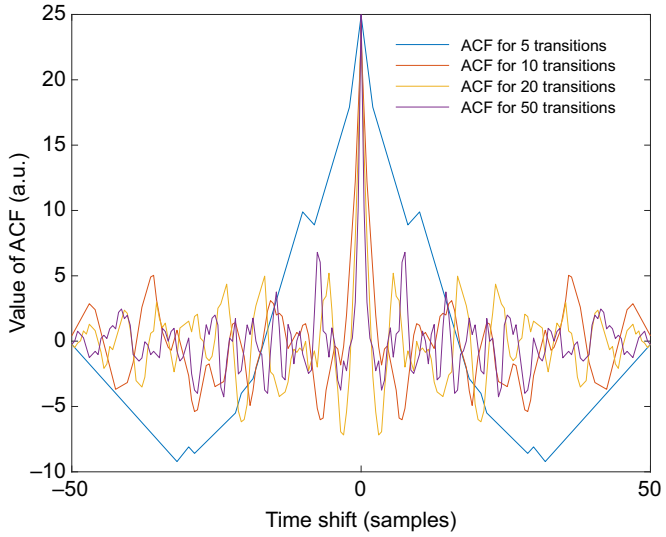

Fig. 5. Illustration of the relationship between the number of transitions in the synchronization signal and the strength of the autocorrelation peak. ACF, autocorrelation function; a.u., arbitrary units.

a failing synchronization. In practice, we commonly use fragment lengths above 500 samples, easily accommodating 50 transitions or more.

We can illustrate the procedure outlined above with a practical example. Most often, the slowest recording modality is a video camera; even high-speed video cameras typically have measurement rates between only 100 and 5000 Hz. We choose the minimal period P_{min} to be at least twice the sampling period of the slowest device. For example, for a camera with a sampling rate of 100 Hz, we set the minimal period P_{min}=20 ms. The maximal period is chosen to be four times the minimal period: P_{max}=80 ms. This allows for a good synchronization of fragments as small as 500 ms.

As hardware for an implementation, an ST-Microelectronics Nucleo F410RB development board was chosen. This development board features an ARM Cortex M4 micro-controller together with a USB programmer and debugger. The board can be powered using the USB connection or using an external DC voltage (between 7 and 12 V). The STM32F410 micro-controller includes a great number of peripherals on a chip. The most interesting peripheral for this application is the integrated hardware true-random number generator (TRNG), that generates random 32-bit values using analog noise. This is an improvement over the PRNG that has been assumed before, though it is in no way required for a well-functioning system. A pseudo-code description of the software running on the micro-controller is presented in Listing 1 (below). For easy replication of the proposed synchronization system, we also provide Arduino-compatible code (Listing 2), which can be easily implemented to build synchronization devices. In order to feed the 1-bit random signal to multiple external measurement systems, the micro-controller output pin is connected to multiple Bayonet Neill–Concelman (BNC) connectors. In the experimental setup shown in Fig. 2, one synchronization output is connected to a National Instruments DAQ device, acquiring both the microphone signal and the synchronization signal, while the other is connected to a LED driver placed in the field of view of a GoPro camera.

Listing 1 : Pseudo-code of an embedded 1-bit pseudo-random generator

```
initialize peripherals;

set output pin state;
generate random wait period;
start timer;

while (1){
  if (timer period elapsed){
    stop timer;
    toggle output pin state;
    generate random wait period;
    start timer;
  }
}
```

Listing 2 : Arduino implementation of a 1-bit pseudo-random generator on output pin 13

```
void setup ()
{
  pinMode (13, OUTPUT);
}
void loop ()
{
  digital Write (13, HIGH);
  //Wait for a random period between 20 ms and 80 ms
  delay (random (20,80));
  digitalWrite (13, LOW);
  //Wait for a random period between 20 ms and 80 ms
  delay (random (20,80));
}
```

Note that in many cases DAQ devices come with high-speed digital to analog (DA) channels or digital output channels that may be used to generate the pseudo-random synchronization signal. In addition, most studio audio devices provide a large number of DA channels that can be used for driving other audio device inputs and even directly driving a LED for video synchronization (in case the output is DC coupled and is able to source a sufficient amount of current). As many such devices support DA conversion rates up to 192 kHz or higher, the proposed method can be implemented yielding accurate synchronization, without the need to build separate hardware. One only needs to generate an audio sequence according to the principles outlined above.

Aligning multiple data sequences

To synchronize the separate data sequences, the extraction of the synchronization signals $s_a[k]$ from the audio signal and $s_v[k]$ from the video signal is required (where k is the time step). Note that, usually, different sensor modalities will have different sample rates: high-speed video of the order of 100–5000 Hz, and audio of the order of 8–500 kHz, depending on the frequency range of interest. Therefore, the extracted synchronization signals need to be resampled to arrive at a common sampling frequency. For practical reasons, i.e. video data cannot be easily upsampled to the audio sampling frequency for memory efficiency reasons, we chose the lowest sampling frequency as the reference rate in our implementation. The resampling step results in a subsampled audio synchronization signal $s_a^r[k]$. As P_{min} was chosen to be at least three times the sampling period of the slowest device, this subsampling will not give rise to aliasing. Next, we calculate the

Journal of Experimental Biology (2018) 221, jeb173724. doi:10.1242/jeb.173724

CCF between the two synchronization signals:

$$s_c[n] = \sum_k s'_a[k] \cdot s_v[k+n] \quad (4)$$

for all possible values of the time shift n. Next, we determine the maximum of the cross-correlation function $s_c[n]$ and the corresponding time shift $n*$ between the audio and the video synchronization signals:

$$n* = \arg\max_n s_c[n]\,. \quad (5)$$

This time shift $n*$ can then be used to align the audio and video sequences.

Extracting the synchronization signal $s_a[k]$ from the audio data is trivial, as it is directly captured by one of the channels of the audio recording equipment. Extracting the video synchronization signal $s_v[k]$ requires a bit of extra image processing. In our implementation, given the video sequence $f_{x,y}[k]$, we select a square region of interest $R_{x,y}$ defined around the position of the LED in the image. Next, for every frame at time step k, we calculate the average intensity in the region $R_{x,y}$:

$$s'_v[k] = \frac{\sum_{(x,y)\in R_{x,y}} f_{x,y}[k]}{N}, \quad (6)$$

where N is the number of pixels contained in the region $R_{x,y}$. Finally, we subtract the mean from the raw video signal $s'_v[k]$ to obtain the video synchronization signal $s_v[k]$:

$$s_v[k] = s'_v[k] - \frac{\sum_k s'_v[k]}{M}, \quad (7)$$

where M is the length of the raw video synchronization signal $s'_v[k]$. The extraction of the synchronization signal is illustrated in Fig. 1A,B.

RESULTS AND DISCUSSION

Illustration: cricket songs

To illustrate the efficacy of the proposed approach, we performed two sets of experiments. The first experiment consisted of recording a calling field cricket (*Gryllus campestris*) using a GoPro Hero 3+ camera capturing video at 240 frames s^{-1}. The audio was recorded using a Brüel & Kjær (B&K) 1/8 in microphone, and the signals were recorded using a National Instruments USB 6356 DAQ device. Fig. 2 shows the setup in more detail. We performed audio and video recordings of 5 s, and extracted the synchronization signals from audio and video separately. Using custom-written Matlab code, we performed alignment of the audio and video data. Separate wave (audio) and MPEG-4 (video) files were written to the hard disk using a 10× lower sampling rate. Using a video-editing tool (Wondershare Filmora), the two sequences were combined, and the resulting video is shown in Movie 1. When observing the video, the motion of the cricket's wings during stridulation is synchronized with the recorded sound.

Illustration: bat behavioral experiments

The second illustration is a series of bat behavioral experiments performed during the EU-FP7 project 'ChiRoPing', on Barro Colorado Island, Panama. We performed experiments on *Micronycteris microtis*, *Macrophyllum macrophyllum* and *Noctilio leporinus*, one gleaning and two trawling bats, respectively. All required research permissions were obtained from the Panamanian Environmental Agency (ANAM) and the Smithsonian Institutional Animal Care and Use Committee (IACUC; 2008-11-06-24-08). The acoustic data were recorded with a custom-made 16-channel microphone array based around Knowles FG-23329-p07 condenser microphones. The video data were recorded with a high-speed camera (CamRecord CR600×2, Optronis GmbH, Kehl, Germany) at 500 frames s^{-1} (*M. microtis*) and 250 frames s^{-1} (*N. leporinus*), respectively. Synchronization was performed using our proposed synchronization mechanism, and audio and video data were combined in single video files. The results of these measurements are shown in Movie 1 and annotated screen-shots of the *M. microtis* and *N. leporinus* recordings can be seen in Fig. 3.

Conclusions

In this paper, we have described and demonstrated a flexible and low-cost synchronization method for multi-modal bio-acoustic experimental data. Our proposed synchronization method relies on embedding synchronization information into the sensor data directly. The main advantages are (1) that no manufacturer-provided synchronization method is needed, (2) that synchronization information is not easily lost, i.e. no manual recording of metadata is required, and (3) different sensor data streams can be easily checked for correspondence. As the method does not rely upon manufacturer-standardized synchronization mechanisms, it can be easily extended to other sensing modalities using vibration sensors, force sensors, etc., by electrically coupling them into a sacrificial channel of a multi-channel recording device. The synchronization method can even be extended to synchronize with optical 3D tracking equipment, e.g. the Qualisys Miqus cameras, by using an 830 nm infrared LED. We have provided an example of how to construct such a synchronization device using off-the-shelf hardware components, and provided a pseudo-code implementation for the pseudo-random generator.

Currently, we have only demonstrated synchronization with multi-channel recording systems through a sacrificial data channel. In the case where multi-channel recording equipment is not available, other means of inserting the synchronization data can be devised. For example, the synchronization information could be embedded in the least-significant bit (LSB) of the recorded data, which is often occupied by noise in real-world recordings. This, however, would require alteration of the recording hardware, making this approach less straightforward. The synchronization information might also be inserted acoustically by using amplitude-shift keying (ASK) modulation (or more advanced modulation schemes) in a section of the acoustic spectrum which is not relevant to the biological experiment, resembling the approach used in the cameras using a blinking LED. This ASK-modulated signal can also be inserted electrically into the analog-to-digital converter of the recording device, requiring a small electronic circuit to sum the microphone signal with the ASK-modulated signal.

The proposed approach opens the opportunity to synchronize an arbitrary number of data-acquisition systems and sensor streams, as no intrinsic limitation is present in the proposed architecture. During our bat behavioral experiments, we routinely synchronized up to four high-speed cameras with two 16-channel microphone arrays. We argue that the flexibility and robustness of our proposed approach, in combination with the fact that it can be constructed using off-the-shelf components, makes it a useful tool that can be applied in a broad range of biological behavioral experiments in which the combined recording of multi-sensor data streams is required.

Acknowledgements
The authors would like to acknowledge the entire ChiRoPing consortium for the fruitful discussions, collaboration and experimentation leading to this synchronization mechanism and the illustrative data. The videos of the bat experiments were produced during the ChiRoPing EU-FP7 project. More details can be found on the ChiRoPing website (http://www.chiroping.org).

Competing interests
The authors declare no competing or financial interests.

Author contributions
Conceptualization: H.P., J.S.; Methodology: D.L., H.P.; Software: D.L., E.V., J.S.; Validation: D.L., E.V., J.S.; Investigation: D.L., E.V., I.G., J.S.; Resources: I.G.; Data curation: J.S.; Writing - original draft: D.L., E.V., I.G., W.D., H.P., J.S.; Visualization: J.S.; Supervision: W.D., H.P., J.S.; Project administration: W.D., H.P., J.S.; Funding acquisition: H.P.

Funding
The authors gratefully acknowledge the support of the Industrial Research Fund (Industrieel Onderzoeksfonds) of the University of Antwerp. Part of this study was funded through the ChiRoPing project (Seventh Framework Programme of the European Union, IST contract number 215370).

Supplementary information
Supplementary information available online at
http://jeb.biologists.org/lookup/doi/10.1242/jeb.173724.supplemental

References
Coen, P. and Murthy, M. (2016). Singing on the fly: sensorimotor integration and acoustic communication in Drosophila. *Curr. Opin. Neurobiol.* **38**, 38-45.
Geberl, C., Brinkløv, S., Wiegrebe, L. and Surlykke, A. (2015). Fast sensory-motor reactions in echolocating bats to sudden changes during the final buzz and prey intercept. *Proc. Natl. Acad. Sci. USA* **112**, 4122-4127.
Geipel, I., Jung, K. and Kalko, E. K. V. (2013). Perception of silent and motionless prey on vegetation by echolocation in the gleaning bat Micronycteris microtis. *Proc. Biol. Sci.* **280**, 20122830.
Greif, S., Zsebők, S., Schmieder, D. and Siemens, B. M. (2017). Acoustic mirrors as sensory traps for bats. *Science* **357**, 1045-1047.
Luo, J. and Moss, C. F. (2017). Echolocating bats rely on audiovocal feedback to adapt sonar signal design. *Proc. Natl Acad. Sci. USA* **114**, 10978-10983.
Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. Model. Comput. Simul.* **8**, 3-30.
Oppenheim, A. V., Willsky, A. S. Nawab, S. H. (1997). *Signals & Systems*. Upper Saddle River: Prentice Hall.
Poynton, C. A. (1996). *A Technical Introduction to Digital Video*. Hoboken, NJ: John Wiley & Sons.
Stilz, P. (2017). How glass fronts deceive bats. *Science* **357**, 977-978.
Übernickel, K., Tschapka, M. and Kalko, E. K. V. (2013). Flexible echolocation behavior of trawling bats during approach of continuous or transient prey cues. *Front. Physiol.* **4**, 96.
Ullrich, R., Norton, P. and Scharff, C. (2016). Waltzing Taeniopygia: integration of courtship song and dance in the domesticated Australian zebra finch. *Anim. Behav.* **112**, 285-300.
Valente, D., Wang, H., Andrews, P., Mitra, P. P., Saar, S., Tchernichovski, O., Golani, I. and Benjamini, Y. (2007). Characterizing animal behavior through audio and video signal processing. *IEEE Multimedia* **14**, 32-41.