The Quantification of Gesture-speech Synchrony: A Tutorial and Validation of Multi-modal Data Acquisition Using Device-based and Video-based Motion Tracking

Wim Pouw[1, 2**], James P. Trujillo[3, 4**], James A. Dixon[1]

** shared first authorship

1. Center for the Ecological Study of Perception and Action, University of Connecticut
2. Department of Psychology, Education, & Child Studies, Erasmus University Rotterdam
3. Donders Institute for Brain, Cognition, and Behaviour, Radboud University
4. Centre for Language Studies, Radboud University

Author note: WP and JPT contributed equally to the current manuscript and share first authorship. All data and analyses code used for this paper, are available on the Open Science Framework: https://osf.io/rgfv3/.

Abstract (words: 232)

There is increasing evidence that hand gestures and speech synchronize their activity on multiple dimensions and time scales. For example, gesture's kinematic peaks (e.g., maximum speed) are coupled to prosodic markers in speech. Such coupling operates on very short timescales at the level of syllables (200 ms), and therefore requires high resolution estimation of gesture kinematics and speech acoustics. High-resolution speech analysis is common for gesture studies given its classic ties with (psycho)linguistics. However, the field has lagged behind in the objective study of gesture kinematics (e.g., compared to research on action). Often, kinematic peaks in gesture are measured by eye, where a "moment of maximum effort" is determined by several raters. In the current paper, we provide a tutorial on more objective and time-effective methods to quantify temporal properties of gesture kinematics, where we focus on common challenges and possible solutions that come with the complexities of studying multimodal language. We further introduce and compare, using an actual gesture dataset (392 gesture events), the performance of two video-based motion-tracking methods (deep learning vs. pixel change) against a high-performance wired motion-tracking system (Polhemus Liberty). We show that videography methods perform well in the temporal estimation of kinematic peaks, and thus provide a cheap alternative to expensive motion-tracking systems. We hope that the current paper incites gesture researchers to embark on the widespread objective study of gesture kinematics and its relation to speech.

Key words: motion tracking, video recording, deep learning, gesture & speech analysis, multimodal language

Introduction

There is an increasing interest in the way co-occurring hand gestures and speech coordinate their activity (Esteve-Gibert &Guellaï, 2018; Wagner, Malisz, & Kopp, 2014). The nature of this coordination is exquisitely complex, as it operates on multiple levels and time scales. For example, on the semantic level, referential gestures can augment (or highlight) speech by iconically denoting state of affairs that are *not* expressed in speech. Twirling the finger when saying "they went down the stairs" can thereby indicate that the stair was a spiral staircase (McNeill & Duncan, 1998). Twirl the fingers a couple of seconds too early, and the multimodal message becomes unstable. Therefore, timing matters. Even more so, on the prosodic level, where it has been found that gesture-speech coupling occurs on even shorter timescales (200 ms). It has been found, in the pronunciation of the nonsense word "baba", when stress is put on the first (***ba***ba) or last syllable (ba***ba***), the pointing gesture's maximum extension is coordinated to align more closely with the stressed syllable (Rochet-Capellan, Laboissiere, Galvan, & Schwartz, 2008; see also Esteve-Gibert & Prieto, 2014; Krivokapić, Tiede, Tyrone, 2017; Rusiewicz, Shaiman, Iverson, & Szuminsky, 2014). On the biomechanical level, it has further been found that gesture's moments of peak physical impetus (lasting ~50 ms) entrain fundamental frequency and the amplitude envelope during phonation (Pouw, Harrison, & Dixon, 2018). Thus, there are many levels and timescales that defines the gesture-speech relationship (for overviews see, Kendon, 2004; Wagner et al., 2004).

It can be argued that the study of the temporal dynamics of gesture-speech coordination has relatively lagged behind in the objective study of gestural movements or *kinematics*, especially compared to the degree to which state-of-the-art (psycho)linguistic

methods are employed for the study of speech (e.g., Loehr, 2012, Shattuck-Hufnagel & Ren, 2018). This manifests itself in the relative scarcity (as compared to other research on action) of published studies that have applied motion tracking in *gesture-speech* research (Danner, Barbosa, & Goldstein, 2018; Leonard & Cummins, 2010; Krivokapić, Tiede, Tyrone, & Goldenberg, 2016; Krivokapic, Tiede, Tyrone, 2017; Parrel, Goldstein, Lee, & Byrd, 2014; Rochet-Capellan, Laboissier, Galvan, & Schwartz, 2008; Rusiewicz, Shaiman, Iverson, & Szumisky, 2014; Treffner & Peter, 2002; Queck et al., 2002, Zelic, Kim, & Davis, 2015). It can be argued that the absence of motion-tracking in the standard methodological toolkit of the multimodal language researcher has further led to imprecisions and conceptual confusions. For example, in the quantification of how tightly gestures couple with prosodic markers (e.g., pitch accent) in speech, researchers have pinpointed relevant kinematic events in gesture by manually identifying the point of "maximum effort" from video recordings (Wagner et al., 2014). Others have used further clarifying definitions of Kendon, which suggests that the researchers should search for the "kinetic goal" or the "peak of the peak" of the gesture (see Loehr, 2004, p. 77). Although, such kinematic judgments are generally made by several raters allowing for some measure of objectivity, the resolution of the kinematics of gesture is necessarily constrained when working with non-quantitative definitions. Wagner highlights that non-quantitative definitions have led to conceptual confusions that have made the literature markedly difficult to digest:

> "[the maximum effort is studied] with varying degrees of measurement objectivity and with varying definitions of what counts as an observation of maximum effort.

> Most definitions evoke a kinesthetic quality of effort or peak effort (Kendon, 2004) correlated with abrupt changes in visible movement either as periods of movement acceleration or strokes (Kita, van Gijn, & van der Hulst, 1998), as sudden halts or hits (Shattuck-Hufnagel, Veilleux, & Renwick, 2007), or as maximal movement extensions in space called apexes (Leonard & Cummins, 2008)."
>
> Wagner et al. (2014), p. 221 (original emphasis)

It is further clear that mainstream hand-coded methods for identifying properties of gesture kinematics are notoriously time-consuming, but are notably still being proposed (Hilliard & Cook, 2017) given the absence of (tutorials about) viable, validated, and easy-to-implement alternatives. As Wagner and colleagues (2014) and Danner (2017) also acknowledge, the time-consuming aspect of multimodal research has had the implication that studies on gesture-speech synchrony are typically performed with single or a limited number of subjects that are intensively studied with a micro-level approach (e.g., Leonard & Cummins, 2009; Loehr, 2012; Shattuck & Huffnagel, 2018). Thus, time-intensiveness of current methodology limits the amount of data that can be generated, which then comes to weaken generalizability of effects to the larger population and the role of individual differences therein.

It is clear therefore that if the laborious task of rater-judgments of, for example, gesture kinematics can be replaced with reliable and objective automated methods, time-scales relevant for multimodal language can be studied, conceptual confusions can be prevented, time-resources can be spared, and researchers will be more likely to be drawn to the large-scale study of gesture kinematics in multimodal language. Indeed, the number

and volume of calls for technical innovations in research on multimodal language is increasing (Danner et al., 2018; Krivokapic et al., 2017; Wagner et al., 2014). In the current paper, we would like to contribute to this joint effort to innovate multimodal research by not only introducing and validating novel methods for studying gesture-speech dynamics (e.g., Beecks et al., 2015; Danner et al., 2018; Schueller et al., 2017; Krivokapic et al., 2017), but also by providing a tutorial for overcoming some common challenges when doing multimodal research.

Overview and Goals

Our overall goal for this paper is to make the objective study of multimodal research more approachable. In the first part of the paper, we address some key steps that need to be taken in order to study multimodal communication in a quantified and replicable manner (Part I). We specifically focus on the kinematic study of gesture, and how data can be processed so that gesture can be studied in relation to speech. At some critical junctures, in tutorial fashion, we provide solutions or code for data-recording and data-processing challenges. Our second goal, in Part II of this paper, is to provide a validation of cheap motion-tracking videography methods (a pixel differentiation and a deep neural network approach) by comparing performance of these approaches with a high-performance standard: a wired motion-tracking system called the Polhemus Liberty. By validating that videography methods are suitable for use in multimodal research, we hope that we can further fuel the widespread kinematic study of gesture. Indeed, we believe with the current innovations in computer vision (e..g, Mamidanna, Cury, et al., 2018; Cao, Simon and Wei, 2016) a treasure-trove of research opportunity arises for the multimodal researcher. Not

only can high-resolution kinematic analyses now be done with a video camera alone, but also with video-data that was not initially intended for the study of kinematics. Thus we hope that current innovations in videography methods and the increasing prevalence of sharing of data amongst researchers, together with efforts to make multimodal research more approachable will lead to a new age in the search for structure in multimodal language.

Part I: Three Key Steps to Quantifying Speech-gesture Synchrony

In Part 1 of this manuscript, we will describe the key steps towards quantifying speech-gesture synchrony. These steps cover the initial planning decision (i.e., which motion-tracking approach to use) including a discussion of several motion-tracking approaches currently available, followed by a brief tutorial on synchronizing audio-visual data streams, and concluding with a suggestion for how to annotate and assess the combined data.

Step 1. Deciding on a Motion-Tracking Method

The first challenge for the multimodal language researcher is how to decide which type of motion-tracking system is needed to answer a particular research question. In this section, we provide an overview of the principal motion-tracking methods currently available. Table 1 provides a summary overview of these methods as well as a suggestion for their applications.

Motion-tracking methods can be broadly separated into two categories: video-based tracking, which utilizes standard video recordings to measure movements, and device-

based tracking, which requires specialized hardware to measure movements. Note that we do not wish to give an exhaustive overview of all motion-tracking methods, but rather provide an introduction to some of the more widely known or commonly employed methods. The aim is to provide the reader with an overview of the common approaches to motion tracking, as well as to introduce some specific implementations of these methods.

Video-based Tracking

Video-based tracking has a long history in the field of computer vision and therefore there are many available approaches. This approach is an attractive option for many because it can be applied to video data that has already been required. As such, it can be a powerful tool for multimodal language researchers when applied to the large corpora of video data that is already available. This approach is also highly accessible, as much of the software is freely available. While video-based tracking can provide an accessible and useful measure of movement, it is limited by the field-of-view of the original recording, as well as by the fact that 3D movement must be estimated from 2D data.

Perhaps the simplest approach to estimating movement, pixel-based motion tracking typically takes advantage of a process known as 'optical flow'. In this approach, movement is quantified based on the change of pixels from one frame to the next. Starting with a 2-dimensional vector field representing all pixels in the scene, the rate and location of changes in the brightness of these pixels lead to the calculation of speed and direction of movement within the scene.

Overall, pixel-differentiation has been shown to provide a reliable measure of movement (Paxton & Dale, 2013; Romero, Amaral, Fitzpatrick, Schmidt, Dunca, & Richardson, 2017) and can be used to capture movement in specific areas of the visual

scene (Danner, Barbosa & Goldstein, 2018). However, this method is particularly vulnerable to changes in background, such as movement or changes in lighting, and may not be able to capture smaller movements or movements towards the camera (Danner et al., 2018). Furthermore, motion tracking of multiple individuals is challenging if individuals move in a close proximity, as the bodily regions of interest that are tracked are likely to overlap, leading to inaccuracies in movement estimates of the individuals. In sum, if dealing with suitable quality video data and movements that are likely to be well tracked by this approach, pixel differentiation can provide easily accessible and robust measure of continuous movement.

Novel methods that will likely revolutionize the study of movement with video data utilize deep learning methods, like OpenPose. OpenPose was developed by Cao, Simon and Wei (2016) as a method of computer-vision based estimation of bodies from 2D still frames or videos. The method uses a form of deep learning (see LeCun, Bengio & Hinton, 2015 for a review of this topic), specifically convolutional neural networks (LeCun, Boser, Denker, et al., 1990), to predict the location of body parts as well as to ensure that body parts are consistently assigned to the correct individual. OpenPose offers an advantage over more simplistic pixel-based methods of tracking by allowing the simultaneous tracking of any number of individuals present in a scene. As the neural network is trained to detect specific body parts, it is also more robust to background noise and images involving multiple people moving and interacting at once.

The currently available version of OpenPose provides multi-person tracking of the body, face, and hands. The library uses predefined key-points (eg. shoulder, elbow, wrist), with the number and exact location of the keypoints varying slightly depending on the

library used. The method therefore provides an excellent solution for estimating movement of the body or hands in 2D, with an off-the-shelf (ie. pre-trained) network ready to use for estimation of standard points of interest on the body.

Similar to OpenPose, Deeplabcut (Mathis et al., 2018) uses deep learning to estimate movement from video data. DeepLabCut is a derivative of an earlier approach, called DeeperCut (Insafutdinov et al., 2016), which estimated whole body poses (ie. body part positions) from video data. While no direct comparisons have been made between DeeperCut and OpenPose, both have independently shown excellent performance in detecting human poses in video and image data. DeepLabCut utilizes the feature detectors from DeeperCut and re-trains these detectors on new inputs. The feature detectors are the readout layers from DeeperCut that provide the predicted location of a body part. By training these feature detectors on new inputs, DeepLabCut effectively 'rewires' the network, creating feature detectors for a new input. This approach is appealing because this rewiring of the network allows the researcher to define which objects or body parts should be detected, granting a large amount flexibility to the researcher.

One of the main advantages to DeepLabCut is that its use of a pre-trained network and an 'extremely deep neural network' (Mathis et al., 2018) leads to a relatively small amount of data required to train the model. While most pose estimation models, such as DeeperCut and OpenPose, require thousands of labelled images for training, DeepLabCut achieves high performance with only ~200 labelled images. Although the model training requires more work from the researcher than off-the-shelf pose estimators like OpenPose, it provides a powerful tool for researchers who wish to define their own points of interest to track.

Device-based Tracking

While video based tracking methods are useful for extracting movement information from data that is already acquired, such as video corpora, the gold-standard for capturing human movement remains device-based tracking. This is primarily due to the generally higher *temporal resolution* of device-based recordings and the ability to track movement in three dimensions, providing a recording of gestures that are not confined to the vertical and horizontal planes, i.e., these recordings have a higher spatial resolution. While many devices are available, utilizing a variety of different approaches, we focus on three specific subtypes of device-based tracking: wired motion tracking, optical motion tracking, and markerless motion tracking. We provide a specific example of each subtype, selected based on what we believe are the most widely used.

Device-based wired motion-tracking. Currently, the Polhemus Liberty (Vermont, USA; http://www.polhemus.com; Liberty Latus Brochure, 2012) is one of the most widely used forms of motion capture in the study of human action and gesture. The Polhemus system utilizes a set of electromagnetic sensors that can be attached to the body, with movement captured by a set of receptors that together define the recording space. With a temporal resolution of 240Hz (samples per second) per sensor, a resolution 0.0012 mm for ideal conditions (30 cm range; total possible range is 180cm), the system provides a fine-grained capture of movement. As the recordings are based on electromagnetic fields, the system also does not suffer from occlusions, so even complex movements are captured robustly.

Although the Polhemus Liberty is a powerful tracking tool, researchers may be limited in how much overall body movement can be captured due to the cost and setup of

many wired sensors. Additionally, due to its reliance on an electromagnetic field, the system cannot be used in conjunction with other electronic methods such as EEG, and care should be taken to keep metals away from the system's tracking range. Wired motion-tracking is therefore especially useful for measuring gestures in specific body parts, which may not be confined to two dimensions of movement, and for analyses requiring a fine-grained resolution of movement.

Device-based optic marker motion tracking. Optic tracking typically uses infrared light that is captured by several cameras, with the 3D position of a physical marker being calculated based on the multiple viewpoints of the camera array. One type of system uses wired, infrared light-emitting diodes that are placed on key points on the body. A popular example of this type of system is the Optotrak system(Northern Digital, Waterloo, Canada). The Optotrak is known for high reliability (States & Pappas, 2006), high temporal resolution (max 400Hz), and high spatial resolution (approximately 0.1mm at 2.25m distance from the cameras). While Optotrak motion tracking is of high quality, the practical implementation may require more some care and expertise. For example, States and Pappas (2006) discuss the problem that tracking quality deteriorates when the markers are tilted away from the plane of the sensors. Additionally given that participants need to wear markers, this system maybe not ideal when working with children or sensitive populations.

In contrast to wired (optic) motion tracking, fully optic, markered motion tracking does not use wired sensors, but rather requires participants to wear reflective markers that are subsequently tracked by an array of cameras that emit infrared light. Similar to wired motion tracking, optical tracking is known for its high precision, with the Vicon

system providing a temporal resolution of 100Hz as well as sub-millimeter spatial resolution. The Vicon system in particular has consistently been shown to be one of the most reliable and precise motion tracking systems (Richards, 1999; Vigliensoni & Wanderley, 2012), even exceeding the Polhemus Liberty (Vigliensoni & Wanderley, 2012).

While optical tracking provides high precision, it also requires calibration and somewhat intrusive markers to be placed on participants body. Additionally, while precision may be higher for Vicon than Polhemus, the Vicon system is more prone to tracking loss due to occlusion of the reflective markers. Researchers considering these methods should therefore consider the types of movements that participants may produce, and how they may respond to the physical attachment of markers.

Device-based Markerless Motion Tracking. A somewhat new addition to motion tracking technology is the use of markerless devices. One such device is the Leap Motion, which uses three infrared emitters and two infrared cameras. Measuring the deformation of the (reflected) infrared light allows the Leap Motion to capture 3D shapes in its field of view. The device has a high spatial (0.4 - 1.2mm) and a reasonable temporal resolution (mean = 40Hz) and has been shown to be quite reliable in its spatial tracking (Weichert, Bachmann, Rudak et al., 2013). The primary limitations of this device are its relatively small field of view (Guna, Janus, Pogačnik et al., 2014), which requires gestures to be produced directly above the device, and its inconsistent sampling rate (Guna et al., 2014). The high spatial resolution therefore makes the Leap Motion ideal for experiments measuring fine-grained hand and finger movement in a confined area in space. However, it is less ideal for capturing gestures in a less constrained environment or for larger movements involving the limbs or other body parts.

Similar in concept to the Leap Motion, the Microsoft Kinect uses a combination of infrared depth cameras with computer vision algorithms, allowing a markerless estimation of key points on the whole body using 3D tracking. The primary advantage of the Kinect is that it provides unobtrusive 3D motion tracking of the major articulators, such as head and gross arm/hand movements. The system is also relatively low cost and very mobile, allowing one to capture data outside of a confined lab setting. There are also open-source Kinect recording softwares available, such as OpenKinect (https://doi.org/10.5281/zenodo.50641), making the system a highly accessible motion-tracking device. Kinect based tracking has been shown to be reliable for several tasks when compared to gold-standard systems of assessment. For example, Otte and colleagues (2016) show excellent agreement between the Kinect and Vicon systems for clinical assessments of motor function. Additionally, Kinect-based measures of gesture kinematics have also been validated against human coder assessments of the same features (Trujillo, Vaitonyte, Simanova & Ozyurek, 2018).

With a temporal resolution of 30Hz, the Kinect does not offer the fine-grained resolution of markered or wired tracking systems. While the Kinect can provide reliable tracking of larger articulators, such as arm and hand gestures, there is also evidence that Kinect is much less precise in capturing fine-grained movements such as finger-tapping (Romero et al., 2017) when compared to systems such as Polhemus or Vicon (Vigliensoni & Wanderley, 2012). The mobility and non-intrusive nature of Kinect must therefore be carefully weighed against its reduced precision.

Table 1. Overview of Motion Tracking Methods

Video-Based			
Method	**Key Features**	**Cost Level**	**Application**
Pixel-differentiation	*Simple to compute* *Requires very stable background*	*Low*	*Calculation of overall movement and velocity in relatively constrained data*
Computer-vision	*Can track very specific parts of the scene (eg. hands, face)* *Computationally costly*	*Low*	*Tracking specific body parts and/or movements of multiple people*
Device-Based			
Wired	*High precision and robust against occlusion* *Limited by number of wired sensors that can easily be attached*	*High*	*Focus on small number of articulators where precision is needed and occlusion may be a problem for other methods*
Optical (markered)	*Gold-standard precision* *Requires calibration and for participants to wear visible markers*	*High*	*High precision tracking of multiple body parts on one or multiple participants*
Markerless (single-camera)	*Non-invasive, 3D tracking* *Lower precision and tracking stability*	*Moderate*	*Mobile setup for whole body tracking, where fine-grained precision is less necessary*

Step 2 Recording Phase: Synchronization of Audio-visual-motion Recording Streams

Having decided on what type of motion-tracking method is applicable to your research question, the next hurdle is to have the multimodal data streams synchronized in their recording. There are three streams of data that are most relevant for multimodal language research: audio, video, and motion tracking. Motion-trackers are often stand-alone devices, without in-built audio recording capabilities. If audio recording is provided (such as Kinect) the audio quality is often subpar for acoustic analyses. Thus, in most cases motion-tracking needs to be synchronized in some way with the audio and the video stream. To complicate matters further, although generic video cameras do record audio (and thus have audio and visual streams synchronized by default), the quality of the audio recording is often surpassed by more specialized audio recording equipment. Furthermore, specialized stand-alone audio equipment is often preferable as this equipment can be tailored for high performance recording in specific situations, e.g., in noisy environment one can filter surrounding noises using condenser cardioid microphones. Thus, if one prefers high-grade tracking of motion, audio, and video, one is often confronted with having to synchronize data streams recorded with specialized non-singular equipment[1].

Yet, if built-in synchronization of video is not possible (for example, because motion tracking is not done via a video camera that has high-quality audio options), how do we

[1] Note that some video cameras allow for an audio plugin, which automatically synchronizes the specialized audio with the video stream. This method can be particularly useful when tracking of motion is performed on the videodata alone (as no post-hoc audio-visual-motion synchronization is needed anymore).

synchronize the separate recording of audio and movement? A possible solution is activating the recording of the audio stream and the motion-tracking (e.g.., Polhemus) stream via a single PC system. For the dataset that we use below, we handled near-simultaneous activation (within a few milliseconds; also dependent on system specifications) of the recording of Polhemus motion-tracking data and the microphone using a C++ script made openly available by Michael Richardson (Richardson, n.d.) which we further modified for audio recording using SFML audio packages (toolbox SFML for C++; for a link to this modified script for audio-motion tracking with a Polhemus Liberty system see folder "c++ code Polhemus+Audio" at https://osf.io/rgfv3/).

Although we use this particular solution, there are many more possibilities and high-performing methods to synchronize audio and motion tracking (and video). Most notable in this case is 'Lab Streaming Layer' (Kothe, 2014; for a demo see Kothe, 2013)which provides a way to synchronize recordings from multiple PC or Laptop systems at once, which can be further customized and implemented using the Python as well as C++ programming languages. Lab Streaming Layer is particularly suitable when, for example, a simultaneous recording is needed with more than one microphone or two motion tracking systems. Indeed, recording from two microphones often requires a specialized device (e.g., PC with two sound cards installed), or otherwise requires two separate devices that need to be synchronized in their recording as well (next to video and motion tracking data). Lab Streaming Layer in this case is ideal as it can coordinate recording of multiple devices from multiple systems, which makes it an ideal solution for dyadic or multi-person language research. Yet, with all these solutions some programming skills are required that might not be immediately at hand. Luckily, for the researcher interested in multi-person language

interaction, audio and video recording from multiple systems can also be synchronized post-hoc in an easy way as we will introduce below.

If the motion-tracking and the audio are synchronized in their recording onset and offset, such as in our case, the video recording still needs to be synchronized in the post-processing phase with the audio and motion data. This synchronization is necessary in most cases, because in the annotation phase (using ELAN) we want to have both the motion-tracking data and high-grade audio aligned with our video data so as to be able to make accurate decisions about gesture-speech typology. For example, ELAN allows one to import movement time-series data into your time line (Crasborn, Sloetjes, Auer, & Wittenberg, 2006) such that gesture initiation can be aligned with motion tracking output (see discussion below on gesture annotation).

For present purposes, accurate synchronization of audio and motion tracking with video is of utmost importance for the our validation study, because we have to compare videography motion tracking methods (deep learning, and pixel change) with the Polhemus. If video is not aligned with the Polhemus, we cannot estimate with certainty how video-based methods perform in comparison to the Polhemus (as there is added noise of misalignments between video and the motion-tracking + audio).

We obtained an easy solution for post-hoc audio synchronization using Adobe Premiere Pro CC 2015[2]. Adobe Premiere Pro CC allows you to load in multiple audio-visual or multiple audio-only streams, as to then apply the function "synchronize audio" for these streams. The synchronization of the audio is performed by Adobe Premiere Pro by aligning

[2] Note that previous Adobe Premier versions may not support this function. For example Adobe Premiere Pro Cs6 does not have have an audio synchronization function.

the temporal structure of waveform *A* (e.g., in-build camera audio) with the temporal structure of waveform *B* (e.g., microphone audio). Given that in our case the camera and the microphone have recorded a single (or at least partially shared) audio event (speech of the participant), the alignment of the waveforms is possible. By aligning the waveforms of the **a)** camera-audio and the **b)** microphone, coincidentally, the **c)** video and **d)** motion tracking+audio data are also aligned (given that **a** was already synchronized with **c**, and **b** was already synchronized with **d**). Using this *chaining technique* of synchronization allows one to synchronize a host of devices and data streams, as long as each system has a "mother" audio stream that can link to the other systems' "mother" audio streams. What if a researcher wants to align two separately recorded cardioid microphone streams with different contents (such that not enough information in the audio waveforms of the microphones overlap)? A chaining solution would be to use the in-built audio of a camera (which does capture both audio streams), and synch both microphone-streams to the camera rather than directly to each other (leading to synchronized microphone data). There are other solutions to this as well, but the chaining technique is quite useful for post-hoc synchronization.

Step 3. Post-processing Phase: Creating Gesture Annotations and Merging Gesture and Speech Data with Motion Tracking Time Series

Annotating Events of Interest

Once the experiment is completed and raw data are recorded, the researcher often still needs to isolate some relevant events from the data based on expert judgment. For

example, prosodic markers may need to be applied using the ToBi method (Beckman & Elam, 1997), which require several expert raters to judge, for example, when pitch accents were unfolding (e.g., Loehr, 2012; Shattuck-Hufnagel & Ren, 2018). Similarly, gesture events may need to be identified, which will always involve some researcher intervention to decide whether a particular gesture type is applicable (e.g., beat versus iconic gestures).

ELAN is a well-known, powerful research tool that can be used during the annotation phase (Lausberg & Sloetjes, 2009). We will not go into how to use ELAN, but we do want to highlight two important ELAN functionalities that are particularly helpful when doing motion-tracking research. First, the time-series data from the motion tracker can be uploaded to ELAN which allows you to continuously monitor movement trajectories as a visual aid during the annotation of bodily gestures (see Crasborn, Sloetjes, Auer, & Wittenburg, 2006; for an example from Pouw & Dixon, 2018 see https://osf.io/5h3bx/). This visual aid is particularly superior to the raw video for deciding *when* a gesture initiates and ends, because of the minimalist representation of movement and because of the sampling rate of the motion tracker, which is likely yield higher visual and temporal resolution for the gesture coder.

More objective estimates for gesture-initiation and termination can be used when one has motion-tracking data, which can be employed within ELAN as well. One simple approach using raw motion capture data is the Elan Plugin for Automatic Annotation (EPAA). The EPAA allows the researcher to automate gesture detection by providing some arbitrary cut-off for when a particular movement reaches a speed or velocity threshold. For example, Hassemer (2016) used EPAA by applying a cut-off speed of the hands of greater than 10cm/s to allow for a first automated pass for likely gesture events, which was then

further modified by expert judgment. The motion-tracking approaches described in Step 1 allow several alternative approaches, typically based on peaks in movement speed. For example, methods for video-based, semi-automatic gesture identification approaches have recently been described: Danner et al. (2018) employ pixel differentiation to identify gesture strokes based on peaks in movement found in a specified area of the video (see *Step 1. Video based tracking - pixel-differentiation* for more detail on the exact implementation), whereas de Beugher, Brône & Goedemé (2017) introduce a custom hand-detection algorithm paired with a gesture recognition approach based on displacement from automatically calculated rest positions. Alternatively, Trujillo et al. (2018) describe an approach using several kinematic features extracted from 3D motion tracking, such as Kinect, to support gesture annotation.

Merging Speech and Motion tracking

After having extracted a particular speech time series like the Fundamental Frequency (F0) track with, for example PRAAT (Boersma, 2001), this data needs to be aligned with the motion-tracking time series. A challenge that may arise in merging speech and motion-tracking data is that the sampling rate of some speech property and the sampling rate of the motion tracker may be different. For example, in the current validation study we have videography motion tracking which samples at 29.97Hz, and we have F0 track of speech that we sampled on 240Hz.

Thus we have the following files:

Data Example 1. Raw speech (SP_DATA) and motion-capture (MOC_DATA) time series

Data frame = SP_DATA

time_ms	F0
0	106.0
4	105.3
8	104.1
13	103.2
17	$F0_5$
.	.
.	.
End time	$F0_n$

Data frame = MOC_DATA

time_ms	x	y
33	710.7	871.2
67	709.9	868.1
133	709.3	865.5
167	x_4	y_4
.	.	.
.	.	.
End time	x_n	x_n

Note. The left data frame called SP_DATA shows an example of the fundamental frequency of speech in Hertz (i.e., pitch track) which samples at 240 per 1000 milliseconds (about every 4.167 milliseconds). The right data frame called MOC_DATA is an example of 2D position motion capture data, which samples at 29.97 samples per 1000 milliseconds, a sampling rate that is common for videography methods (about every 33.367 milliseconds).

In this case we not only need to align the datasets, but we also need to up-sample motion-capture data (MOC_DATA to 240Hz) or down-sample speech data (SP_DATA to 29.97Hz) if we want to end up with a fully merged speech+motion datafile. For our analyses below we want to keep a high sampling rate of speech, similar to the sampling rate of the original Polhemus experiment (240Hz),so we will up sample our motion-capture data.

First, to merge the motion capture data with the speech data, the following function from R base called 'merge' will align the datasets and merge them in a single data frame called 'merged_file':

R code Example 1. Merging two data frames

```
--------R Code--------
merged_file <- merge(SP_DATA, MOC_DATA, by.x= "time_ms", by.y =
"time_ms", all = TRUE)
--------------------------------
```

Note. This R Code constructs a new data frame called 'merged_file', wherein the MOC_DATA time series data will be merged with SP_DATA on the basis of the reference time variables 'time_ms' present in both datasets. "All = TRUE" indicates that new rows will be created whenever 'time_ms' from the speech data and 'time_ms' from the MOC_DATA are not identical; for those rows only data is present for one of the data streams (the other will have NA's).

Applying this function will give you the following file (Data Example 2), where each observation from the speech and motion tracking data sets are now collected at some time *t* (or 'time_ms') in the merged dataset, and observations are merged together on one single row if possible (i.e., when at time *t* both F0 and motion-capture observations are made). For example, if at some time *t* (e.g., time_ms = 234 in Data Example 2) a motion-capture observation is present but no F0 observation, then a row will be constructed with only data for the motion-capture observation for that row.

Data Example 2. Raw speech (SP_DATA) and motion-capture (MOC_DATA) time series

---
dataframe = merged_file

---

time_ms	F0	x	y
0	106.0	710.7	871.2
4	105.3	NA	NA
8	104.1	NA	NA
13	103.2	NA	NA
17	101.9	NA	NA
21	100.5	NA	NA
25	100.0	NA	NA
29	99.8	NA	NA
33	99.7	709.9	868.1
.	.	.	.
.	.	.	.
233	168.1	NA	NA
234	NA	708.8	822.2
238	168.3	NA	NA
.	.	.	.
End time	$F0_n$	x_n	y_n

Note. An example of SP_DATA and MOC_DATA merged. Note that from 4 to 29 milliseconds speech is repeatedly sampled but no x and y values are recorded for those times (NA's are given, i.e, "Not Available"). Coincidentally, at 33 milliseconds there is an observation for both F0 and motion-capture as the sampling intervals overlapped at that point in time. But at some point the sampling intervals do not align anymore, such that at 234 milliseconds there is an observation for motion-capture but this does not exactly align with the observation 1 millisecond earlier (233) for F0; thus two separate rows are constructed in this case.

If both speech and motion tracking are sampled on exactly the same time schedule, i.e., identical sampling rate and start time, then the merging function as applied above would be the end product for us. However, in the current case there are different sampling rates. Indeed, in most cases there will be some measurement of F0 at time *t* while there is no complementary motion tracking data at time *t* (see Data Example 2). Since we have NA's

that are embedded by actual observations for motion tracking, and we know the time steps in milliseconds from one known observation to another, we can linearly interpolate x and y values for the unknown observations. We can do this by using na.approx() function from R package 'zoo' (Zeileis, & Grothendieck, 2005).

R code Example 2. Code for linear approximation of motion tracking data

```
--------R Code--------
#step 1 download and install zoo package
require(zoo)
#step 2 upsample x, and y by linear interpolation
merged_file$x_up <- na.approx(merged_file$x, x=merged_file$time_ms,
rule=2)
merged_file$y_up <- na.approx(merged_file$y, x=merged_file$time_ms,
rule=2)
#step 3 keep only observations for F0 and interpolated x_up and y_up
merged_file <- subset(merged_file, !is.na(F0) )
--------R Code End--------
```

Note. "#" indicates a commenting out of the code (the code on this line will not be run by the compiler). This R code firstly loads in the R package 'zoo' (Step 1). Subsequently, it applies the linear interpolation function na.approx two times, which saves two new up sampled variables 'x_u'p and 'y_up' in the original 'merged_file' dataframe (Step 2). The na.approx takes as its first argument the variable to be interpolated, the second argument for x provides the time index (which is 'time_ms') for the interpolation procedure. The argument type=2 refers to the procedure that if begin and endpoints that cannot be interpolated (because begin and endpoints are not embedded by observations), these values will be extrapolated and given nearest value. In step 3, we remove

rows that were not of the original sampling rate of F0, effectively keeping all original F0 sampling intervals at a sampling rate of 240Hz, now also with merged or interpolated x and y values.

Applying this code from example 2 will give you the following updated 'merged file' whereby x and y values are up sampled as shown in 'x_up' and 'y_up' (through linear interpolation[3]) as to accommodate the sampling rate of speech data (240 Hz).

Data Example 3. Fully merged data with linearly interpolated motion tracking data

---

dataframe = merged_file

---

time_ms	F0	x	y	x_up	y_up
0	106.0	710.7	871.2	710.7	871.2
4	105.3	NA	NA	710.6	870.8
8	104.1	NA	NA	710.5	870.4
13	103.2	NA	NA	710.4	870.0
17	101.9	NA	NA	710.3	869.7
21	100.5	NA	NA	710.2	869.3
25	100.0	NA	NA	710.1	868.9
29	99.8	NA	NA	710.0	868.5
33	99.7	709.9	868.1	709.9	868.1
.	.	.	.	.	.
233	168.1	NA	NA	708.78	822.5
238	168.3	NA	NA	708.67	821.2
.	.	.	.	.	.
End time	$F0_n$	x_n	y_n	x_up_n	y_up_n

Note. This data example shows the results of up sampling 'x' and 'y' into 'x_up' and 'y_up' using linear interpolation, as to accommodate the sampling rate of F0. Values shown in red are interpolated values.

[3] Another interpolation methods such as cubic spline interpolation is possible as well.

Now that we have merged the speech data and with the up-sampled motion-tracking data, we still need to isolate speech+mocap time series for particular events of interest. For example, if we want to know the moment where gesture *A* reaches its highest vertical point (positive peak y), we want to evaluate a subset of time series values which map onto gesture *A*. Thus we have to merge (ELAN) annotations that marked some temporal regions of interest (e.g., gesture type, or gesture identifier event) into the time series. To do this we can let ELAN generate or make a file that has an begin and end time for a particular event tier (gesture event identifier in example below) and we can upload this into R. We thus have our latest 'merged_file' within which we want to incorporate the 'annotation_data'.

Data Example 4. Annotation file (annotation_data) to be merged with the speech+motion tracking data (merged_file)

Data frame = annotation_data

begintime	endtime	tier
4	15	A
.	.	B
.	.	C
.	.	.
.	.	.
.	.	.
$begintime_n$	$endtime_n$	$tier_n$

Data frame = merged_file

time_ms	x_up	y_up
0	710.7	1077.02
4	710.6	1078.25
8	710.5	1079.25
13	.	.
17	.	.
.	.	.
time_ms	3.41	1040.12

Note. The 'annotation_data' file shows a hypothetical gesture event 'A' occurring between 4 and 15 milliseconds. We want to place these annotations into our merged speech and motion tracking data, such that for every gesture-speech observation that occurred during an event *x* we will have a variable that marks that observation as belonging to event *x*.

R code Example 3. Loading annotation data into timeseries

```
--------R Code--------
#CUSTOM FUNCTION
load.event <- function(original_time, annotations)
{
  output <- vector()
  for(i in annotations[,1])
  {
        output <- ifelse((original_time >=
        annotations$begintime[annotations[,1] == i] & original_time <=
        annotations$endtime[annotations[,1] == i]),
        as.character(annotations[,3][annotations[,1]==i]), output)
  }
  return(output)
}

#apply function
merged_file$g_event <- load.event(merged_file$time_ms, annotation_data)
--------R Code End--------
```

Note. This R code constructs a custom made function which assesses, for an 'original_time' vector, which of those values are occurring during some event as given in the 'annotations' vector. Specifically, this function loops over the rows of the annotation data, and loads an event marker, indicating whether an event was happening or not, into a new vector (called 'output'). This is done for each row of the original time series ('original_time'). The final line of code applies this function by entering in the relevant arguments, namely the time_ms vector of the 'merged_file data' and the 'annotation_data'.

Applying our custom made R function (R code Example 3) will give us a new variable in the 'merged_file' called 'g_event' (for **g**esture **event**) which marks, for each observation in the time series, whether at that time an event was occurring based on the annotation begin time and end time, which is indicated by an 'A' at that time point.

Data example 5. The final merged speech and mocap datafile, now with annotations

---

dataframe = merged_file

---

time_ms	x_up	y_up	g_event
0	710.7	1077.02	NA
4	710.6	1078.25	A
8	710.5	1079.25	A
13	.	.	A
17	.	.	NA
.	.	.	.
time_ms	x_up_n	y_up_n	g_event_n

Note. Applying the R code example 3 produces a new variable called ('g_event') which represents that during observations at 'time_ms' 4, 8, and 13 a hypothetical event A (highlighted in red) was occurring. Where NA is given, no 'g_event' was occurring.

This final Data Example 5 is a very workable end-version of the data. A host of functions can now be applied that take some time series for a given event and compute some measure from it. For example, we can fill a vector 'peaks_y' where for each of the indices we want a maximum vertical position observed during a gesture event (R code Example 4).

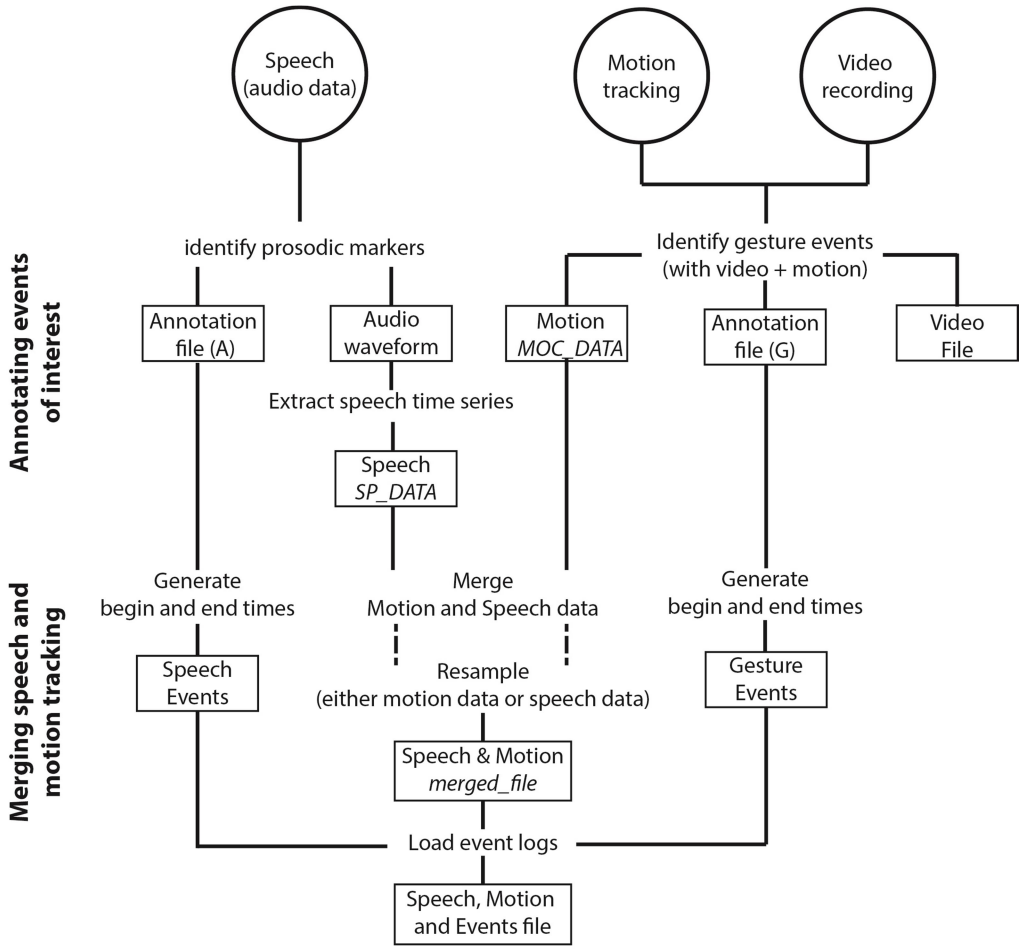
R code Example 4. Generating a vector with maximum vertical peaks for each gesture event

```
--------R Code--------
peaks_y <- vector()
for (i in unique(merged_file$g_event[!is.na(merged_file$g_event)]))
        {peaks_y <- c(peaks_y, max(merged_file$y_up[merged_file$g_event==i],
        na.rm = TRUE))}
--------R Code End--------
```

Note. This function extracts all maximum vertical (y_up) positions observed for each g_event (excluding NA's) and orders these values in a vector called peaks_y (from first observed g_event to last observed g_event). If we would take the mean of 'peaks_y' vector we would have the average maximum height for all gestures observed. In the next section we will summarize the research pipeline as we have presented here.

Summary Research Pipeline

Figure 1. Schematic overview of post-processing steps



3. Part II: Comparing performances in temporal estimation of gesture kinematics

In Part I, we provided an overview of the general pipeline of decisions and methodological steps towards quantified gesture-speech synchrony research. In Part II, we will provide an example of using such methods to measure gesture-speech synchrony and in doing so will also provide an assessment of the quality of two video-based motion tracking methods compared to a gold-standard wired tracking method. We choose to limit our comparison to these three methods in order to assess whether such video-based tracking is of sufficient precision to take advantage of the wealth of video-based data already available. The Polhemus system provides an established standard of quality against which to compare the video-based methods.

Validation Analyses

Dataset

The current dataset is from an exploratory study by Pouw & Dixon (see for preliminary preprint report: Pouw & Dixon, 2018c), wherein participants retold a 5-minute cartoon that they had just watched (a classic gesture induction method: McNeil, 1992). During the retelling of the cartoon, participant's index finger of the dominant hand was motion-tracked with a Polhemus Liberty (240 Hz; resolution ~0.13 mm spatial resolution under ideal conditions). We also recorded recorded speech with a cardioid microphone (RT20 Audio Technica Cardioid microphone) and made video recordings (Sony Digital HD Camera HDR-XR5504) to allow for gesture categorization. In the current study, we use these video recordings to additionally track motion with videography methods. This

dataset consists of 392 gesture events. We compare the tracking results from our markered gold standard, Polhemus, to two video-based methods: Deep learning and pixel differencing.

Speech Acoustics

We extracted the Fundamental Frequency from the speech data (i.e., pitch track time series) at 240Hz, with a pitch range = 75-300 Hz. This lower and upper bound was adjusted for male voice range around 85-155Hz (there were only males in this sample).

Videography methods

Pixel change

Instantaneous pixel change is a quantification of the amount of visual change in the video data, and has been found to be a reliable estimation of gross-body movement that sometimes matches even low-cost motion tracking equipment (e.g., Kinect) and further correlates well with outputs from other more expensive motion-tracking technology (Romero et al., 2017). We computed the instantaneous pixel change on the video data (sampling rate = NTSC standard sampling rate camera = 29.97 frames per second) using a Python script recently developed and made available by Brookshire and colleagues (2017), for code see github link: https://github.com/gbrookshire/ivc. We applied a low-pass second-order butterworth filter using R package "signal" (Ligges et al., 2015) of 10Hz to the pixel change time series. Given that we want to maintain the resolution of speech acoustics (240Hz) to make a fair comparison to Polhemus, we up sampled the pixel change time series to 240 Hz.

Deep learning motion tracking ('Deeplabcut'); Minimal vs. Highly trained network

We trained a pre-trained deep neural network (DNN) called "ResNet" with 50 layers (He, Zhang, Ren, & Sun, 2016) for pose estimation for 250.000 iterations. More than 200.000 iterations is a typical amount needed until learning gains plateau as stated by Mathis and colleagues (2018). This DNN yielded 1.73 average pixel difference for the training set, and 2.64 pixel average difference for the test set between human-made estimation of the right hand index finger, versus the estimation made by the DNN (note test pictures were 800 x 1000 = 8000 pixels). A full example clip of the DNN motion tracking can be seen here (https://osf.io/9hku8/). For a tutorial on DeepLabCut and code see github link from Mathis and colleagues: https://github.com/AlexEMG/DeepLabCut.

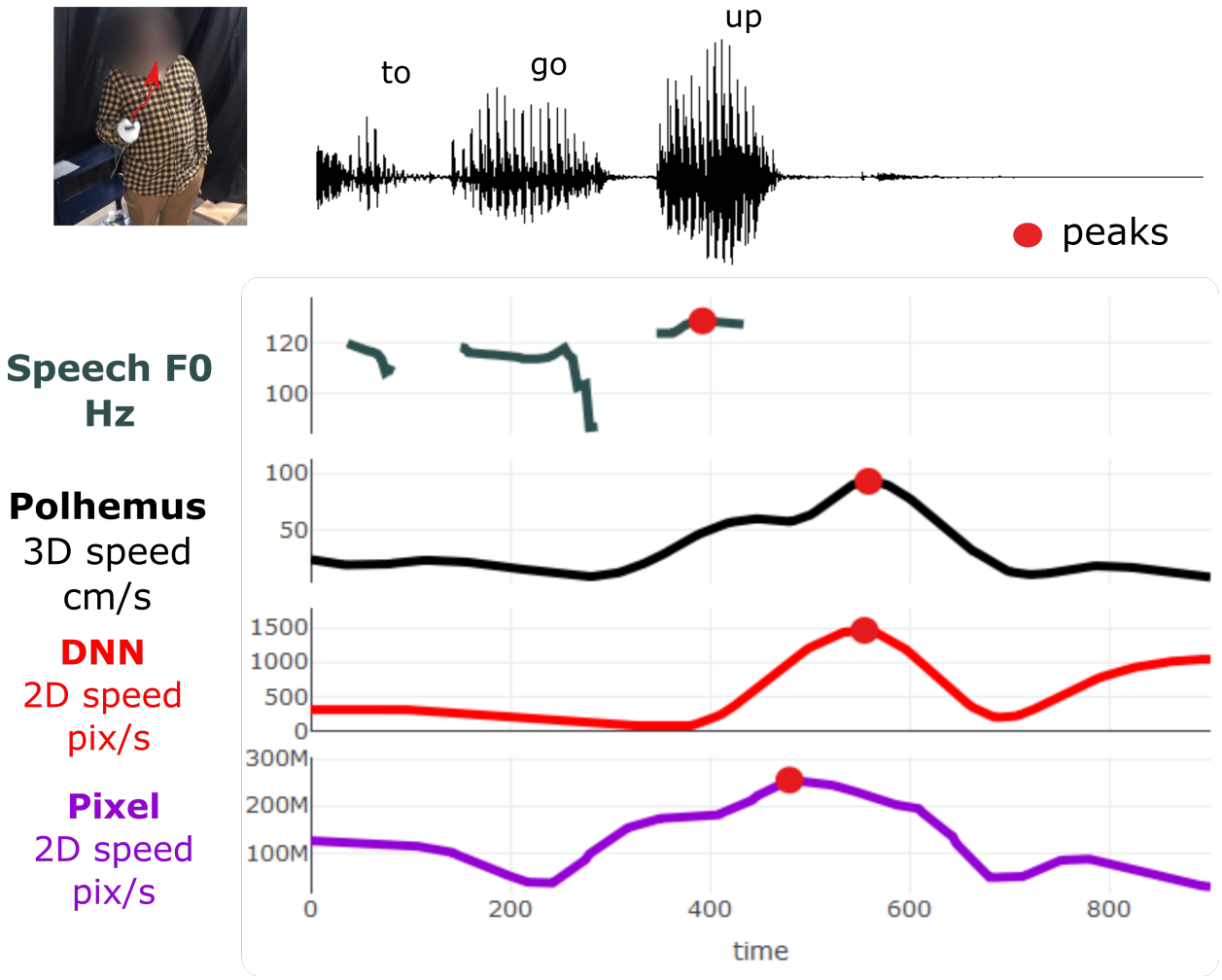
Temporal estimation of kinematic peak and acoustic peak

For the current analyses, we want to know whether using videography methods to temporally estimate some kinematic event in gesture, relative to peak F0, is comparable in performance to 3D high-resolution motion tracking. To make this comparison, we determined for each gesture event when the highest peak in pitch occurred (as an anchor point for speech)[4], and when the highest peak in speed was observed (peak speed)

[4] It is open for discussion whether peak in F0 is a good anchor point (rather than say peak amplitude or ToBi prosody label based point estimates). However, this is not something that should be of current concern. As long if we have a formally defined anchor point in speech, we can do the planned comparisons of the motion tracking methods in terms of gesture-speech synchrony estimations.

occurred as determined by the Polhemus (3d speed pixel change method (2d speed) and Deep neural network method (2d speed).

Figure 1. Example gesture event peak speed per method



Note. Example of a gesture event lasting 800 ms (see video here: https://osf.io/aj2uk/) from the dataset (event 10 from participant 2). Red dots indicate the positive maxima peaks in the respective datastreams, Fundamental Frequency in Hertz (F0), Polhemus 3D velocity in centimeters per second, DNN 2D velocity in pixels position change per second, Pixel method 2D velocity in summed pixel change per second. Note velocity is directional speed, whereas speed is non-direction specific velocity.

Results

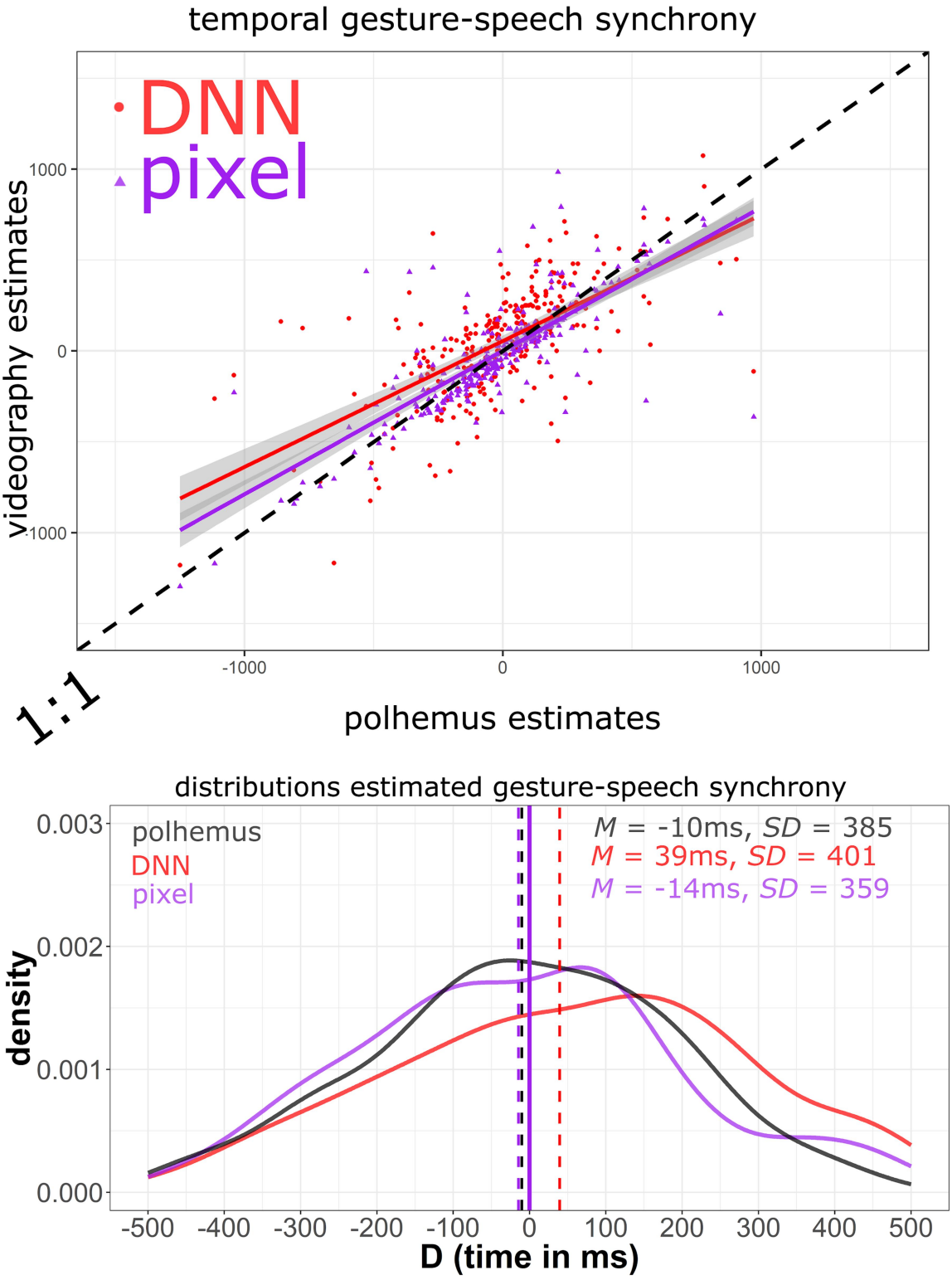
Gesture-speech synchrony estimates: peak speed peak F0

Table 2 and Figure 3 and 4, provide an overview of the performance of the videography methods relative to the Polhemus. Both the pixel and the DNN method show statistically reliable (p's < 2.2e-16) and high correlations (r's > .73) in the estimates of the temporal offsets of peak speed and peak F0. Correlation between the two videography methods DNN and Pixel was $r = 0.747$, $p < .0001$.

Table 2. Results and comparisons estimation peak speed versus peak F0 in gesture

	Polhemus	**DNN**	**pixel**
Estimate mean *(SD)* asynchrony	-10ms *(385)*	39ms *(401)*	-14ms *(359)*
Correlation Polhemus			
r		**.756**	**.797**
95%CI		[.700 - .803]	[.750 - .837]
p		< .00001	< .00001

Figure 3. Results and comparisons estimation peak speed versus peak F0 in gesture

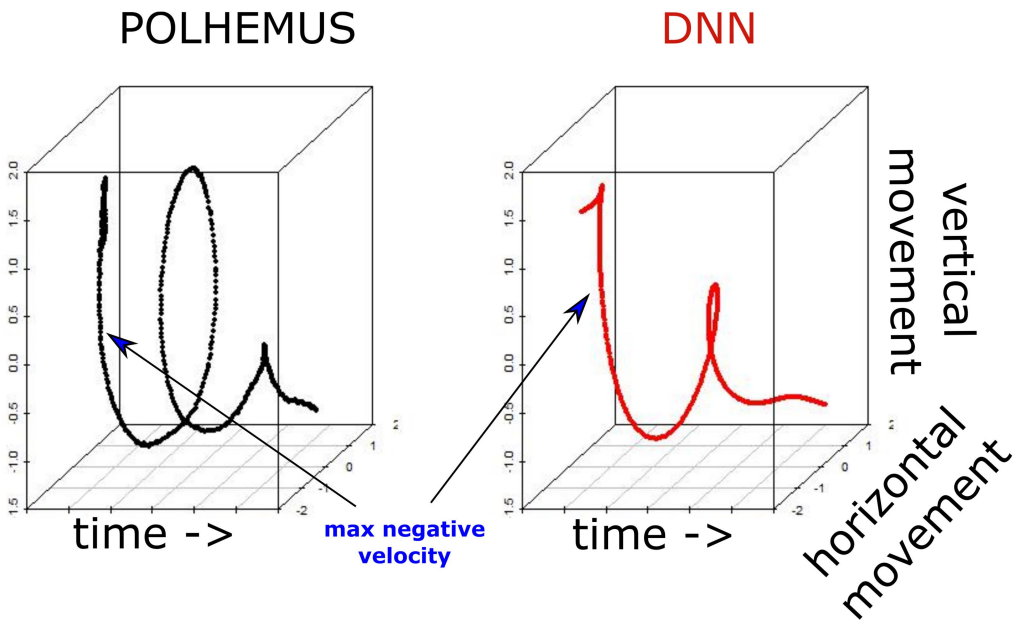


Note Figure 3. Upper panel: Videography estimates of gesture-speech synchrony (vertical axis) are compared to Polhemus estimates of synchrony. Purple dots indicate pixel change method performance relative to Polhemus, and red dots indicate deep neural network (DNN) performance relative to Polhemus. A 1:1 slope (as indicated by black line of identity) would indicate that a identical performance of videography and performance. Dots along the region of the identity line indicate comparable approximations of gesture-speech synchrony for the different methods. Note, that some points are excluded that fell far from the point cloud (for the full graph see: https://osf.io/u9yc2/). Lower panel: smoothed density distributions for the estimated gesture-speech synchrony estimates per method, with means (dashed vertical lines) indicating average gesture-speech synchrony.

Positional data comparison

The key advantage of the DNN motion tracking over pixel method is that, over and above the quantification of movement, positional information is provided as well. We could for example explore movement trajectories (e.g., Huffnagel & Ren, 2018) for gestures that make some kind of rotational movement (Figure 4). Or we could be interested in the maximum vertical point of gestures (e.g., Trujillo et al., 2018). To assess of the DNN versus the Polhemus we instead estimated for each gesture the moment where the maximum downward speed (or maximum negative speed) was reached as way to probe a moment where a physical impact (or a beat) of a gesture might be produced. For this analyses we yielded a correlation of performance of the Polhemus versus the DNN, $r = .754$, 95%CI[.697, .801], $t\ (270) = 18.85$, $p < .0001$.

Figure 4. Example trajectory as measured by Polhemus versus DNN



Note. An example of an Iconic gesture with a circling motion (axis z-scaled) as registered by Polhemus and the DNN. This type of positional information is not available when using the Pixel change method. For our comparison we look at the moment where a negative velocity is highest; where a gesture reached its highest speed when moving downward.

Discussion

In the first part of the paper, we have provided a methodological overview of common challenges in multimodal language research. Our further goal was to make explicit the issues one needs to consider before running an actual experiment, and provide a basic tutorial of some procedures in the post-processing phase. In the second part, we have assessed performance of videography methods, including novel deep-learning motion tracking, with a common standard motion tracking system (Polhemus Liberty). Specifically, for purposes of estimating gesture-speech synchrony, we showed that both pixel change methods and deep neural network motion tracking are performing very well relative to a Polhemus Liberty wired motion tracking system. Deep learning motion tracking has the further advantage of being able to track the 2D position of the gesture, rather than only a quantification of the amount of movement, as is the case for pixel differentiation methods.

Although performance of the deep learning motion tracking was high in our study, some parameters may need to be adjusted in order to make this technique more reliable for future studies. For example, performance might be enhanced by using a larger training dataset, providing more accurate position judgments of hand positions from a second independent coder, or by interpolating the hand position when DeepLabCut indicates low certainty during tracking. However, for current purposes we show that DeepLabCut performs very well in our task of estimating gesture-speech synchrony.

We think this current validation of video-based tracking is important because it shows that reliable motion tracking for gesture-speech synchrony analyses can be done without the need to collect additional data. Of course, physical motion tracking systems, whether optical or electromagnetic, will remain superior to video-based tracking that relies

on a single point of view. However, given the current high performance of videography methods, we think such methods promises to be a major step forward in terms of efficiency and reliability of tracking meaningful movements.

Implications and Applications

We hope that the current paper contributes to the study of multimodal language in more diverse populations and with increasingly larger samples to accommodate the study of individual differences. To serve these goals, the steps we have described here can be applied to any type of motion tracking data that can be reliably synchronized with audio/video data. Multimodal language researchers can apply these quantitative methods to data that has already been acquired, or they can choose to take motion tracking requirements into account when collecting new data. The use of already acquired data is particularly useful given the large number of video corpora that have been generated and maintained over the years. Additionally, markerless motion tracking (whether video-based based or device-based) can be quite valuable for capturing movements of more sensitive populations (e.g., Eigsti & Pouw, 2018; Romero, Fitzpatrick, Roulier, Duncan, Richardson, Schmidt, 2018). Finally, although we have assessed deep learning motion-tracking performance in terms of the temporal estimation of kinematic peaks, this method can be especially useful for gesture trajectory analyses (e.g., Shattuck-Hufnagel & Ren, 2018), and are likely to replace methods that require annotations by hand (e.g., Hilliard & Cook, 2017).

Summary

The temporal relationship between speech and gesture is an integral part of the study of multimodal language. Although methods are now available for objectively

quantifying both movement and speech, bringing these two streams of data together for meaningful and reliable analyses is non-trivial. We have provided an overview of the key steps that must be taken in order to conduct such research and have described different approaches that are available at each step. Finally, we have provided a direct, step-by-step example (and code) for such an analysis, which we use to validate the use of video-based motion tracking techniques for quantifying speech-gesture synchrony. We hope that this overview will provide a useful resource for multimodal language researchers interested in applying quantitative methods to the study of speech and gesture.

References

Beckman, M. E., & Ayers, G. (1997). Guidelines for ToBI labelling. *The OSU Research Foundation*, *3*, 30.

Beecks, C., Hassani, M., Hinnell, J., Schüller, D., Brenger, B., Mittelberg, I., & Seidl, T. (2015, August). Spatiotemporal similarity search in 3d motion capture gesture streams. In *International Symposium on Spatial and Temporal Databases*(pp. 355-372). Springer, Cham.

Boersma, P. (2001). PRAAT, a system for doing phonetics by computer. *Glot International* 5 (9/10), 341-345.

Crasborn, O., Sloetjes, H., Auer, E., & Wittenburg, P. (2006). Combining video and numeric data in the analysis of sign languages with the ELAN annotation software. In C. Vetoori (Ed.), *Proceedings of the 2nd Workshop on the Representation and Processing of Sign languages: Lexicographic matters and didactic scenarios* (pp. 82-87). Paris: ELRA.

Danner, S. G., Barbosa, A. V., & Goldstein, L. (2018). Quantitative analysis of multimodal speech data. *Journal of Phonetics*, *71*, 268-283.

Eigsti*, I. & Pouw, W. (2018). Explicit synchrony of speech and gestures in autism spectrum disorder. Poster presented at the*10th annual meeting for the Society for the Neurobiology of Language*, Quebec City, August 16-18, 2018.

Esteve-Gibert, N., & Guellaï, B. (2018). Prosody in the auditory and visual domains: A developmental perspective. *Frontiers in Psychology*, *9*, 338. doi: 10.3389/fpsyg.2018.00338

Esteve-Gibert, N., & Prieto, P. (2013). Prosodic structure shapes the temporal realization of

intonation and manual gesture movements. *Journal of Speech, Language, and Hearing Researc*h, 56(3), 850-864. doi: 10.1044/1092-4388.

Guna, J., Jakus, G., Pogačnik, M., Tomažič, S., & Sodnik, J. (2014). An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *Sensors (Basel)*, 14(2): 3702:3720. doi: 10.3390/s140203702

Hassemer, J. (2016). *Towards a theory of gesture form analysis. Imaginary forms as part of gesture conceptualisation, with empirical support from motion-capture data.* Doctoral dissertation, Rheinische-Westfälische Technische Hochschule Aachen.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hilliard, C., & Cook, S. W. (2017). A technique for continuous measurement of body movement from video. *Behavior Research Methods*, *49*(1), 1-12. doi: 10.3758/s13428-015-0685-x

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M. & Schiele, B. (2016). DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. *European Conference on Computer Vision,* 34–50

Kothe, C. (2014a). Lab Streaming Layer (LSL). Available online at: https://github.com/sccn/labstreaminglayer

Kothe, C. (2013, 07-31). Demo 1 The Lab Streaming Layer. Retrieved from URL https://www.youtube.com/watch?v=Y1at7yrcFW0&t=539s

Krivokapić, J., Tiede, M. K., & Tyrone, M. E. (2017). A kinematic study of prosodic structure in articulatory and manual gestures: Results from a novel method of data collection. *Laboratory Phonology*, *8*(1), 1-36. doi: 10.5334/labphon.75.

Lausberg, H., & Sloetjes, H. (2009). Coding gestural behavior with the NEUROGES-ELAN system. *Behavior Research Methods, Instruments, & Computers*, *41*(3), 841-849. doi: 10.3758/BRM.41.3.841.

Leonard, T., Cummins, F. (2010). The temporal relation between beat gestures and speech. *Language and Cognitive Processes*, 26(10), 1457–1471. doi: 10.1080/01690965.2010.500218.

Ligges, U., Short, T., Kienzle, P., Schnackenberg, S., Billinghurst, D., Borchers, H.-W., Carezia, A., Dupuis, P., Eaton, J.W., Farhi, E., Habel, K., Hornik, K., Krey, S., Lash, B., Leisch, F., Mersmann, O., Neis, P., Ruohio, J., Smith III, J.O., Stewart, D., Weingessel, A., 2015. Package 'Signal'. R Foundation for Statistical Computing.

Loehr, D. P. (2012). Temporal, structural, and pragmatic synchrony between intonation and gesture. Laboratory Phonology, 3(1), 71-89. doi: 10.1515/lp-2012-0006.

Mathis, A., Mamidanna, P., Abe, T., Cury, K. M., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). Markerless tracking of user-defined features with deep learning. arXiv preprint arXiv:1804.03142.

Mathis, A., Mamidanna, P., Cury, K.M., Abe, T., Murthy, V.N., Mathis, M.W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with

deep learning. *Nature Neuroscience*, 21(9): 1281-1289. https://doi.org/10.1038/s41593-018-0209-y

McNeill, D., & Duncan, S. (1998). Growth Points in Thinking-for-Speaking. *Language and Gesture,* 141-161.

Otte, K., Kayser, B., Mansow-Model, S., Verrel, J., Paul, F., Brandt, A.U., & Schmitz-Hübsch, T. (2016). Accuracy and reliability of the Kinect version 2 for clinical measurement of motor function. *PLOS ONE*, 11(11): e0166532. Doi: 10.1371/journal.pone.0166532.

Paxton, A., & Dale, R. (2013). Frame-differencing methods for measuring bodily synchrony in conversation. *Behavior research methods*, *45*(2), 329-343. doi: 10.3758/s13428-012-0249-2

Pouw, W., Harrison, S. J. & Dixon, J. A. (under review). Gesture-Speech Physics: The Biomechanical Basis of Gesture-Speech Synchrony. Preprint PsyArxiv doi: 10.31234/osf.io/tgua4

Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X. F., Kirbas, C., ... & Ansari, R. (2002). Multimodal human discourse: gesture and speech. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *9*(3), 171-193.

Richards, J.G. (1999). The measurement of human motion: A comparison of commercially available systems. *Human Movement Science,* 18(5): 589-602.

Richardson, M. J. (n.d.). Retrieved from http://xkiwilabs.com/software-toolboxes/)

Rochet-Capellan, A., Laboissiere, R., Galvan, A., Schwartz, J. (2008). The speech focus position effect on jaw-finger coordination in a pointing task. *Journal of Speech, Language, and Hearing Research*, *51*(6), 1507–1521. doi: 10.1044/1092-4388.

Romero, V., Fitzpatrick, P., Roulier, S., Duncan, A., Richardson, M. J., & Schmidt, R. C. (2018). Evidence of embodied social competence during conversation in high functioning children with autism spectrum disorder. *PloS One*, 13(3), e0193906. doi: 10.1371/journal.pone.0193906

Rusiewicz, H. L., Shaiman, S., Iverson, J. M., & Szuminsky, N. (2014). Effects of perturbation and prosody on the coordination of speech and gesture. *Speech Communication*, *57*, 283-300. doi: 10.1016/j.specom.2013.06.004.

Schueller, D., Beecks, C., Hassani, M., Hinnell, J., Brenger, B., Seidl, T., & Mittelberg, I. (2017). Automated pattern analysis in gesture research: similarity measuring in 3D motion capture models of communicative action. *Digital Humanities Quarterly*, *11*(2), 1-14.

Shattuck-Hufnagel, S., & Ren, A. (2018). The prosodic characteristics of non-referential co-speech gestures in a sample of academic-lecture-style speech. *Frontiers in Psychology*, *9*. doi: 10.3389/fpsyg.2018.01514

States, R.A., & Pappas, E. (2006). Precision and repeatability of the Optotrak 3020 motion measurement system. Journal of Medical Engineering & Technology, 30(1): 11-16.

Treffner, P., Peter, M., & Kleidon, M. (2008). Gestures and phases: The dynamics of speech-hand communication. *Ecological Psychology*, *20*(1), 32-64. doi: 10.1080/10407410701766643.

Trujillo, J.P., Vaitonyte, J., Simanova, I. et al.(2018). Toward the markerless and automatic analysis of kinematic features: A toolkit for gesture and movement research. *Behavior Research Method*s, 1-9. doi: 10.3758/s13428-018-1086-8

Vigliensoni, G., & Wanderley, M. (2012). A Quantitative Comparison of Position Trackers for the Development of a Touch-less Musical Interface. *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression*, 103-108.

Wagner, P., Malisz, Z., & Kopp, S (2014). Gesture and speech in interaction: An overview. *Speech Communication*, *57*, 209-232. doi: 10.1016/j.specom.2013.09.008.

Weichert, F., Bachmann, D., Rudak, B., & Fisseler, D. (2013). Analysis of the accuracy and robustness of the Leap Motion Controller. *Sensors,* 13(5), 6380-6393; doi:10.3390/s130506380.

Zeileis, A., & Grothendieck, G. (2005). zoo: S3 infrastructure for regular and irregular time series. *arXiv preprint math/0505527*.

Zelic, G., Kim, J., & Davis, C. (2015). Articulatory constraints on spontaneous entrainment between speech and manual gesture. *Human Movement Science*, *42*, 232-245. doi: 10.1016/j.humov.2015.05.009