Gesture Annotation with a Visual Search Engine for Multimodal Communication Research

Sergiy Turchyn,[1] Inés Olza Moreno,[2] Cristóbal Pagán Cánovas,[2]
Francis F. Steen,[3] Mark Turner,[4] Javier Valenzuela,[5] Soumya Ray[1]
[1]Department of EECS, Case Western Reserve University, Cleveland, Ohio, USA
[2]Institute for Culture and Society, University of Navarra, Spain
[3]Department of Communication, University of California-Los Angeles, California, USA
[4]Department of Cognitive Science, Case Western Reserve University, Cleveland, Ohio, USA
[5]Departamento de Filología Inglesa, University of Murcia, Spain

Abstract

Human communication is multimodal and includes elements such as gesture and facial expression along with spoken language. Modern technology makes it feasible to capture all such aspects of communication in natural settings. As a result, similar to fields such as genetics, astronomy and neuroscience, scholars in areas such as linguistics and communication studies are on the verge of a data-driven revolution in their fields. These new approaches require analytical support from machine learning and artificial intelligence to develop tools to help process the vast data repositories. The Distributed Little Red Hen Lab project is an international team of interdisciplinary researchers building a large-scale infrastructure for data-driven multimodal communications research. In this paper, we describe a machine learning system developed to automatically annotate a large database of television program videos as part of this project. The annotations mark regions where people or speakers are on screen along with body part motions including head, hand and shoulder motion. We also annotate a specific class of gestures known as timeline gestures. An existing gesture annotation tool, ELAN, can be used with these annotations to quickly locate gestures of interest. Finally, we provide an update mechanism for the system based on human feedback. We empirically evaluate the accuracy of the system as well as present data from pilot human studies to show its effectiveness at aiding gesture scholars in their work.

1 Introduction

Human communication has many different facets. People communicate not just through spoken language, but through gesture, facial expression, posture, tone of voice, pacing, gaze direction and touch. We learn to communicate using precisely timed movements, delicately modulated sounds, interpreting the mental states of others from moment to moment, dynamically coordinating with others, and maintaining a high level of contextual awareness (Clark 1996; Duranti and Goodwin 1992). Still, the majority of research into communication until now has focused on written language and speech. This is at least partly because data is easy to obtain and share in this case. Conversely, the full range of communicative behavior must be recorded with resource-intensive audiovisual technologies. Naturalistic data can be difficult to obtain; artificial collections from lab recordings take their place. Large-scale datasets are required for systematic study, yet no single researcher has the required time or resources to create them. Further, even if such data are collected, researchers in the humanities who study linguistics or communication may lack supporting computational tools to help analyze the data.

An international academic collaboration, the Distributed Little Red Hen Lab (*http://www.redhenlab.org*) project, is working to enable the transition of the study of multimodal human communication to large scale data-driven approaches. We are inspired by fields such as astronomy, genetics and neuroscience that have undergone a similar transformation. As part of this project, we collect data on naturalistic multimodal communication on a large scale, provide computational and storage tools to manage data and aid in knowledge discovery, and provide means of iterative improvement through integrating the results and feedback of researchers into the project.

Red Hen's primary data sources for multimodal and multilingual communication are television recordings. Fortunately, section 108 of the U.S. Copyright Act authorizes libraries and archives to record and store any broadcast of any audiovisual news program and to loan those data, within some limits of due diligence, to researchers for the purpose of research. The NewsScape Archive of International Television News (*http://newsscape.library.ucla.edu*) is Red Hen's largest; as of August 2017, it included broadcasts from 51 networks, totaling 340,000 hours and occupying 120 terabytes. The collection dates back to 2005 and is growing at around 5,000 shows a month. It is an official archive of the UCLA Library. Under Red Hen, it has been expanded to record television news in multiple countries around the world, curated by local linguists participating in the Red Hen project. The NewsScape archive now includes, in rough order of representation, broadcasts in English, Spanish, German, French, Norwegian, Swedish, Danish, Continental Portuguese, Brazilian Portuguese, Russian, Polish, Czech, Ital-

ian, Arabic, and Chinese. The system is fully automated and scales easily, using Raspberry Pi capture stations running custom open-source software.

Most of the programs above are associated with closed caption transcripts, and Red Hen has an array of tools to identify linguistic elements in these transcripts. These tools include Stanford CoreNLP (*http://stanfordnlp.github.io/CoreNLP*), a set of natural language processing utilities providing parts of speech, lemmas, and named entities in half a dozen languages; the FrameNet project (*http://framenet.icsi.berkeley.edu*) to annotate frames; the SEMAFOR project (*http://www.ark.cs.cmu.edu/SEMAFOR*) to perform an automatic analysis of the frame-semantic structure of English text and provide frame names, frame elements, and semantic role labeling results. These results can be accessed using search tools developed for the project.

To aid researchers in gesture or communication studies analyze this data, however, as well as linguistic elements, visual elements of the scenes need to be annotated. For example, for researchers interested in "co-speech" gestures, it is useful to indicate those parts of a video where the speaker is visible on the screen. For researchers interested in gestures of specific types, such as "timeline gestures," annotations indicating the presence of such gestures or fragments of such gestures would be valuable. In this paper, we describe an initial approach to such a visual search engine for Red Hen. The goal is to enable a researcher to quickly build a dataset that contains gestures of interest by searching through the annotations of scenes provided by our system.

2 Background and Related Work

A gesture is a group of body movements used as part of communication (Cooperrider and Goldin-Meadow 2017). Movements unrelated to communication, such as eating, are not considered to be gestures. Gestures are very closely connected to language. Gesture timing often matches speech timing. Gestures often precede the corresponding speech, or happen at the same time, but very rarely happen after it. Gestures often convey similar meaning as the speech. For example, when talking about shooting, a person might use his hand as an imaginary gun to convey the same meaning. Gesture and language are so connected, that a stuttering person also pauses their gestures to maintain the timing.

Even though gesture meaning is related to speech meaning, they are not the same. Gestures can convey additional information. For example, when talking about the layout of a building, gestures can indicate the relative location of each room that is not mentioned in speech.

Within artificial intelligence, gestures have been studied in computer vision, human-computer or human-robot interaction and social robotics. A variety of approaches are used (Mitra and Acharya 2007). Hand gestures are a common target. Generally, hand positions are detected and used as an input to a hidden Markov model or a neural network. For example, one study (Molchanov et al. 2016) uses depth, color, and stereo IR cameras to detect hands for human-computer interaction. This study uses a convolutional neural network to detect spatio-temporal features in the short clips coming from each camera and classifies them as a gesture or not. Another similar field is hand stroke gesture detection (Ye and Nurmi 2015), which tries to recognize gesture-based input to a computer through (typically) a touchscreen, Kinect or similar device. Work in human-robot interaction (Fong, Nourbakhsh, and Dautenhahn 2003; Jaimes and Sebe 2007) has considered building gesture-based interfaces (Gleeson et al. 2013) or understanding gestures using reinforcement learning (Yanik et al. 2014). Other related work focuses on recognizing mental states from facial expressions or speech (Chen and Huang 2000; Busso et al. 2004).

There have been various gesture recognition challenges such as the 2013 Multi-modal Gesture Recognition Challenge (Escalera et al. 2013). The provided dataset includes gestures recorded in an artificial settings. People in front of a white wall or a whiteboard perform various gestures. There is always one person in the frame. The camera position is fixed. The gesturing person is facing the camera. The goal of the challenge was to recognize gestures using audio, skeletal model, user mask, depth, and RGB data. The top algorithms in the challenge used hidden Markov models, neural networks, and random forests.

Overall, gesture recognition has been studied in different subfields of AI. However, several aspects of the task we study make it different and in some ways more challenging. Rather than HCI/HRI, we are focusing on communication between humans, which tends to be far more fluid and subtle. A significant part of HCI/HRI work uses artificially recorded videos (people were asked to perform a gesture on purpose) or tasks where gestures are artificial, specific and predetermined. Instead, we focus on television programs such as news and talk shows, where people interact in a far more natural manner. Much work also assumes the availability of complex features or sensors such as skeleton models, depth cameras or Kinect videos. Often there is often only one person on screen, and often facing the camera. These assumptions do not hold in our setting. Instead, the task involves analyzing videos with split screens, post-processing effects and arbitrary camera motion. Finally, and most importantly, we currently have little labeled data in this setting. Our approach is designed to bootstrap itself by enabling annotation by gesture researchers which can then be fed back to improve the system over time.

ELAN

To make our system's annotations available to gesture researchers, we use a tool called ELAN. The European Distributed Corpora Project (EUDICO) Linguistic Annotator (ELAN) (Lausberg and Sloetjes 2009) is a state-of-the-art tool for gesture annotation. It allows a user to examine a video and create annotations as desired (Figure 1). Annotations are grouped in tiers that can have hierarchical relationships. ELAN provides controls to jump from annotation to annotation in a particular tier, as well as search for a phrase across all the annotations.

ELAN annotation files have extension ".eaf" that stands for ELAN Annotation Format. EAF files are written in Extensible Markup Language (XML). They contain the information a user creates using the ELAN interface: tiers, tier

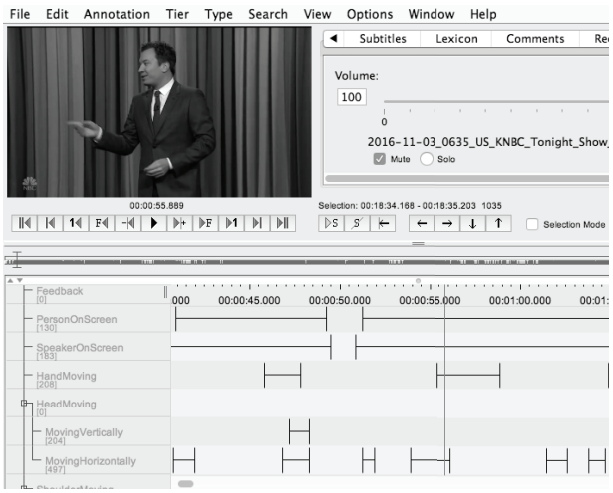

Figure 1: Output of our system shown in ELAN.

types, and annotations, indexed by frame. In our approach, we provide pre-annotated EAFs to users which can be used to quickly search through a video using ELAN's interface.

3 Our Annotation System

To design the annotation system, we first consider how gesture scholars annotate videos. Suppose the annotator wants to find gestures of interest in a long video. First, she would need to find segments where a person was on screen. Large video segments can be eliminated in this step. In some cases, she may be interested in whether someone who is speaking is on screen. Second, she would check whether the person or speaker was doing the gesture of interest. If there is such a gesture, the researcher annotates specific attributes such as axis and direction of the gesture movement.

A perfect system would detect entire gestures. However, gestures are often very subtle and complex and the area of gesture research is new enough that the space of possible gestures is not yet fully explored. Further, we do not have large labeled datasets to train classifiers on individual gestures. As a result, in our work, we focus on annotating *gesture fragments*. These are head, hand, and shoulder motions that typically accompany gestures of interest. We also annotate the presence of people and speakers on screen (the latter requiring the detection of lip motion). We use a combination of supervised algorithms and unsupervised heuristics to detect these fragments. Furthermore, we use a small dataset of videos to train a classifier for a specific gesture type, a timeline gesture, and provide annotations for this.

An example of our system's output is shown in Figure 1, which shows the ELAN interface. The tiers of annotations, such as "PersonOnScreen", etc. were output by our system. The bars next to these tiers indicate frames which contain the event annotated. Using ELAN's interface one can quickly jump between or search for specific events.

We now describe the individual detectors in our system.

Person and Speaker Detection

We detect people based on faces. We define a person on screen as a person whose face is present in the frame. The OpenCV library (Bradski 2000) implements face detection based on a pretrained Haar cascade classifier (Viola and Jones 2001), a series of classifiers of varying complexity that are applied from the least to the most complex.

We detect whether the speaker is on screen by looking at motion in the lip region of a person (Figure 2 left). Lip detection also uses a trained Haar cascade classifier from prior work (Castrillón Santana et al. 2007). If the cascade classifier does not find them in the bottom part of the face, our approach guesses the approximate lip region. This allows us to get lip motion even if the lips were not detected. We also have an additional alignment procedure as suggested by prior work (Bendris, Charlet, and Chollet 2010). Given lip locations in two consecutive frames, we can adjust the lip location in the second frame to better match the first frame based on average Manhattan distance between the pixels.

Lip motion detection is based on the optical flow as suggested by prior work (Bendris, Charlet, and Chollet 2010). Optical flow indicates how much individual pixels move between two frames. After we detected the lips, we calculate Farneback optical flow (Farnebäck 2003) between the grayscale lip regions and threshold based on the mean and the standard deviation. A higher optical flow means more activity inside the lip region and likely lip motion.

Head Motion Detection

We detect head motion based on head positions. To separate vertical and horizontal motion we look at the motion angle. For each consecutive pair of frames, we get the largest vertical and horizontal motions for all people on screen. We use a chunking approach to determine which motion we consider significant. For each chunk (of length around 3-5 seconds) we calculate the average motion based on the magnitudes from the step described above. If the magnitude is bigger than the mean by a chosen threshold, we annotate the frame with head motion.

Hand Motion Detection

To detect hands, we start by getting the skin color for each person found (a person found is equivalent to a frontal face found). In the HSV color space we zoom into the face and take the median and standard deviation of each color channel after removing outliers in the H dimension. We use fixed skin color ranges to threshold the image and get skin patches. We use a background subtraction technique (Zivkovic 2004; Zivkovic and der Heijden 2006) to ignore the static background. After getting the skin pixels, we look for contours using OpenCV's built-in contour detector algorithm (Suzuki and Abe 1985). Using positions and mean colors of skin patches, we assign them to different people in the frame with k-means clustering. For each person, we pick the hands in the frame by looking at their hands in the previous frame, or looking at distance to an expected hand location in case the previous frame has no hand information.

The hand positions are used to detect hand motion. For each person in a frame we find them in the previous frame

Figure 2: Examples of individual detectors. a) Head and lip detection. b) Hand detection. c) Shoulder detection. The thin light grey lines are candidate shoulders. The thicker white lines are the chosen shoulder lines based on symmetry and color difference.

by looking for similar face positions. Then, if found, we get motion in pixels for both hands. We take the maximum motion for all people on screen and compare it to two thresholds. The lower threshold removes insignificant motion. The upper threshold removes noise (when detections are noisy, they jump around the frame from one place to another). Figure 2 (center) shows an example of hand detection.

Shoulder Motion Detection

Shoulder detection makes use of the shoulder shape often being similar to a straight line. We use the Canny edge detector (Canny 1986) and the Hough transform (Matas, Galambos, and Kittler 2000) to find candidate lines for shoulders and filter them to remove unlikely ones based on positions and angles. Since shoulder lines separate clothes from background, we pick the lines most likely corresponding to the shoulders by looking at color difference on both sides, as well as symmetry. Figure 2 (right) shows an example of shoulder detection.

To get motion, we first get the exact motion by comparing neighboring frames, and later post-process it to find significant shoulder motion. We look at the motion for each shoulder separately. Since each shoulder line has two points, we use the minimum y-axis motion of both line ends as the motion of a single shoulder. To decide what motion is significant, we compare each frame to its neighbors. We use the same chunking approach as in head motion. This discovers shoulder motion larger than normal, because the chunk average approximates the normal shoulder motion of the person.

Timeline Gestures

We have a small dataset of timeline gestures with 63 positive and 77 negative examples. A timeline gesture in our case is a lateral hand gesture that indicates a time interval. These gestures often accompany statements such as "from beginning to end," "from start to finish" etc. The positive examples came from gesture researchers who had previous used the Red Hen data. The negative examples were acquired by us by isolating frames where no gestures occurred.

The features come from face and hand motion. For a set number of frames before and after frame i, we compute mean and standard deviation of distances, as well as mean angle of face and hand motion. Mean angles are calculated using the equation $\mathrm{atan2}(\sum_{i=1}^{n}\sin(a_i), \sum_{i=1}^{n}\cos(a_i))$.

Timeline gestures are recognized using an SVM classifier with the RBF kernel (Bishop 2006) implemented by the Scikit-learn Python library (Pedregosa et al. 2011). The classifier is applied to a single frame at a time, although frames around it are used for the features. We only try to detect timeline gestures on the intervals where there is a person on screen and the hands are moving. The classifier makes an assumption that there is one person on screen, so only the first person is used in the multiple people case.

Postprocessing and Feedback

The detector outputs are further refined by smoothing. First, a sliding window is used to only mark frames as positive when a certain percentage of each sliding window is positive. Second, we merge annotations close to each other. Last, we remove very short annotations. This improves annotation quality and removes noise.

One of the main goals of the system is improvement based on annotation feedback. Each detector has a set of parameters and their values have a significant influence on the results. When someone uses our annotations, they can label frames where the annotation was incorrect in some way. These annotations are used by the detectors as data and a hill climbing search is carried out in the space of detector parameters to maximize balanced accuracy (the average of the true positive and true negative rates).

4 Empirical Evaluation

Here we present experiments to evaluate the performance of individual detectors, as well as the entire system in that researchers using it should be able to annotate gestures with less effort than if they did not use it.

Accuracy of Individual Detectors

We evaluate individual detectors on two one-hour long videos. The two videos are "Late Night Show with James Corden" (LNS) and "CNN New Day" (CNN ND). The system took 3.7 and 4.9 hours to annotate these two videos on a MacBook Pro. We picked a talk show and a news video because these two videos have significant differences. For example, talk shows more frequently have people talking to each other while not facing the camera, but the number of artifacts due to post-processing effects is much smaller. The

Task	Person on Screen		Speaker on Screen		Vertical Head Motion		Hand Motion		Shoulder Motion	
Video	LNS	CNN ND	LNS	CNN ND	LNS	CNN ND	LNS	CNN ND	LNS	CNN ND
Balanced Accuracy	0.760	0.846	0.714	0.790	0.593	0.663	0.610	0.536	0.537	0.512
Precision	0.937	0.970	0.799	0.882	0.487	0.465	0.807	0.292	0.659	0.127
Recall	0.722	0.792	0.599	0.714	0.218	0.391	0.303	0.173	0.078	0.027
Random Class. Precision	0.806	0.804	0.531	0.588	0.120	0.124	0.529	0.194	0.069	0.019

Table 1: Performance of the system on two videos.

opposite happens in news videos. Evaluating both kinds of videos shows how the detectors perform in both settings. For these videos, we manually annotated their frames for each detector. Table 1 shows the results for our detectors. We use balanced accuracy because the class labels are not balanced (most frames do not have the associated fragments), and also report precision and recall. We also report the precision of the random classifier as a baseline (its recall is 50%).

From the table, the "Person on Screen" detector is quite accurate. The detector performs better on news videos. This is because in the news setting the reporters almost always face the camera. In the talk show setting people are often further away from the camera and are facing different directions. However, the performance, especially precision, is still high. The "Speaker on Screen" detector also shows good results on news videos and is worse on talk shows, where speakers often do not face the camera, making lip detection hard. In general, these detectors miss people at different angles than frontal or profile, and sometimes fire for other artifacts (including Halloween Jack-o'-lanterns).

Head motion detection has a precision of about 50% and similar recall. The false positives are often small head motions that are not significant enough for a human eye. The false negatives mostly come from the head motion from the side or another unusual angle that still have a vertical component and so are identified as vertical motion.

The talk show setting has significantly more hand motion and is also easier to detect. The reason for this is that in the news setting, post-processing effects often hide part of the hands and make it impossible to see their motion. People in the talk show move their hands almost constantly (especially with multiple people on screen), so the precision is much higher. Even if the hand detector misdetects the hands, hand motion will likely coincide with some real hand motion.

There are two situations where our hand detector does not perform well. First, when people wear short-sleeved clothes, the entire arm can be seen and hands cannot be separated by only looking at the skin color. Another problem is people not facing the camera, so that if the face is misdetected, the position of the hands is not correctly assessed.

The shoulder motion detection results are also very different between the two videos. The precision is significantly higher in the talk show setting. A key reason is that news videos often have "active" backgrounds such as other videos being shown in the background, which makes it challenging to detect which lines belong to the shoulder.

Overall, some of these results seem acceptable, and others can be improved as more annotated data is collected. However, as we show in the next section, our approach can already reduce the effort involved in annotation.

Effort Reduction during Gesture Annotation

We carried out a pilot study to test whether the system helps reduce effort when annotating gestures. Three subjects A, B, and C performed two experiments (Olza, Valenzuela, and Pagán 2017). We provided EAF files of video clips produced by the system where the annotators looked for head movements. Each clip was 4 minutes long.

First, all subjects annotated several clips for 75 minutes to calibrate their individual performance without using our system. Subjects B and C were selected for the second part of the experiment because their performance was similar.

The second task for B and C was to annotate two 4 minute clips. B started with empty EAF files. C started with our pre-annotated EAF files. B completed the task in 1:07:55 while C did it in 00:33:47. In other words, the annotator using our system completed the task twice as fast as the one not using it. The resulting annotations were checked for quality. We found that the error rate was the same in each case. While B did have slightly more detailed annotations, the time saved by the system was still considered significant.

In the second experiment, subjects B and C switched roles. C started with an empty EAF file. B started with a pre-annotated EAF file. The task for B was to revise the system's annotations, as well as annotate the basic features of each head movement. The task for C was to just annotate the basic features of each (vertical and horizontal) head movement. Both subjects worked with one 4 minute video.

B completed the task in 20:05 minutes. C completed the task in 21:14 minutes. Their performance was also similar. However, we found that B misunderstood the task and annotated hand detection and hand movement in addition to head movement. Thus, B did more work in the same time as C.

These two experiments indicate that, though in a preliminary state, when using the system the annotation task can be completed with less effort. We expect that these results will improve as the system becomes more accurate over time.

5 Conclusion and Future Work

Human communication is multimodal, and large datasets to study it carefully are becoming available. As part of the Distributed Little Red Hen Lab, we have described an effort to create a visual search engine to allow gesture scholars to more easily analyze massive quantities of video. The work is preliminary, and we feel that dealing with the challenging data characteristics will require novel ideas from machine learning and computer vision. However, even at this point, we feel it shows promise in helping the disciplines of

linguistics, communication and related areas transform into data-driven sciences.

The authors wish to thank Peter Uhrig for discussions related to the article.

References

Bendris, M.; Charlet, D.; and Chollet, G. 2010. Lip activity detection for talking faces classification in tv-content. In *The 3rd International Conference on Machine Vision (ICMV)*, 187–190.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Bradski, G. 2000. OpenCV Python library. *Dr. Dobb's Journal of Software Tools*.

Busso, C.; Deng, Z.; Yildirim, S.; Bulut, M.; Lee, C. M.; Kazemzadeh, A.; Lee, S.; Neumann, U.; and Narayanan, S. 2004. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, ICMI '04, 205–211.

Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8(6):679–698.

Castrillón Santana, M.; Déniz Suárez, O.; Hernández Tejera, M.; and Guerra Artal, C. 2007. Encara2: Real-time detection of multiple faces at different resolutions in video streams. *Journal of Visual Communication and Image Representation* 130–140.

Chen, L. S., and Huang, T. S. 2000. Emotional expressions in audiovisual human computer interaction. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, volume 1, 423–426 vol.1.

Clark, H. 1996. *Using Language*. Cambridge University Press.

Cooperrider, K., and Goldin-Meadow, S. 2017. Gesture, language, and cognition. In *The Cambridge Handbook of Cognitive Linguistics*, Cambridge Handbooks in Language and Linguistics. Cambridge University Press. chapter 8.

Duranti, A., and Goodwin, C. 1992. *Rethinking Context: Language as an Interactive Phenomenon*. Cambridge University Press.

Escalera, S.; Gonzàlez, J.; Baró, X.; Reyes, M.; Lopes, O.; Guyon, I.; Athitsos, V.; and Escalante, H. 2013. Multimodal gesture recognition challenge 2013: Dataset and results. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, 445–452. New York, NY, USA: ACM.

Farnebäck, G. 2003. *Two-Frame Motion Estimation Based on Polynomial Expansion*. Berlin, Heidelberg: Springer Berlin Heidelberg. 363–370.

Fong, T.; Nourbakhsh, I.; and Dautenhahn, K. 2003. A survey of socially interactive robots. *Robotics and autonomous systems* 42(3):143–166.

Gleeson, B.; MacLean, K.; Haddadi, A.; Croft, E.; and Alcazar, J. 2013. Gestures for industry intuitive human-robot communication from human observation. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 349–356.

Jaimes, A., and Sebe, N. 2007. Multimodal human-computer interaction: A survey. *Computer vision and image understanding* 108(1-2):116–134.

Lausberg, H., and Sloetjes, H. 2009. Coding gestural behavior with the neuroges-elan system. *Behavior Research Methods* 41(3):841–849. http://tla.mpi.nl/tools/tla-tools/elan.

Matas, J.; Galambos, C.; and Kittler, J. 2000. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding* 78(1):119 – 137.

Mitra, S., and Acharya, T. 2007. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 37(3):311–324.

Molchanov, P.; Yang, X.; Gupta, S.; Kim, K.; Tyree, S.; and Kautz, J. 2016. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4207–4215.

Olza, I.; Valenzuela, J.; and Pagán, C. 2017. Automatic visual analysis and gesture recognition: Two preliminary pilots. Technical report, University of Navarra, Institute for Culture and Society.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Suzuki, S., and Abe, K. 1985. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing* 30(1):32 – 46.

Viola, P., and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 511–518.

Yanik, P. M.; Manganelli, J.; Merino, J.; Threatt, A. L.; Brooks, J. O.; Green, K. E.; and Walker, I. D. 2014. A gesture learning interface for simulated robot path shaping with a human teacher. *IEEE Transactions on Human-Machine Systems* 44(1):41–54.

Ye, Y., and Nurmi, P. 2015. Gestimator: Shape and stroke similarity based gesture recognition. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, 219–226.

Zivkovic, Z., and der Heijden, F. 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters* 27(7):773–780.

Zivkovic, Z. 2004. Improved adaptive gausian mixture model for background subtraction. In *Proceedings of the International Conference on Pattern Recognition*.