BOOK REVIEW

Sylvie Gibet, Nicolas Courty, & Jean-François Kamp (Eds.) (2006). *Gesture in human–computer interaction and simulation*. Berlin, Heidelberg, & New York: Springer.

Reviewed by Jacob Eisenstein

The international Gesture Workshops bring together a broad spectrum of interdisciplinary research related to gesture, emphasizing computational models and applications. The proceedings of the 2005 workshop are summarized in *Gesture in human–computer interaction and simulation*, edited by Sylvie Gibet, Nicolas Courty, and Jean-François Kamp. This collection of papers touches on a broad range of topics at the intersection of gesture and computer science; I explore three main areas. First, many computational approaches to gesture have treated it as a sequence of semaphores, particularly in recognition; I discuss articles from this collection that attempt to incorporate more flexible and fluid models. Next, I describe the state of the art in automatic hand tracking and gesture detection, which I believe is well-surveyed by this collection of papers. Finally, I summarize the interaction techniques employed in the current generation of gestural user interfaces. While leaving out some areas discussed in the proceedings, this focus allows for brevity and relevance to this journal's audience.

Beyond gesture as semaphore

Much of the early treatment of gesture synthesis and recognition by computer scientists focused on gesture as a sequence of semaphores to be produced or decoded (cf. Quek et al., 2002). This has been the case for both gesture synthesis and recognition, and for both gesticulation and sign language (e.g., Starner et al., 1998). This approach, while a necessary simplification for engineering purposes, does not well reflect the way in which people use gestures in normal conversation. Gesturing in everyday interaction is a fluid combination of improvised and conventionalized forms, rarely performed as isolated semaphore-like actions (Kendon, 2004). Formal sign languages are also known to be more continuous and multi-faceted than a simple sequence of semaphores (Neidl et al., 2000).

Several of the papers in this collection represent steps towards a more nuanced computational treatment of gesture and sign language. Addressing gesticulation, Hartmann, Mancini, and Pelachaud investigate how gesture synthesis can be

ISSN 1568–1475 / E-ISSN 1569–9773 © John Benjamins Publishing Company

modified to carry a desired emotional content while maintaining the original semantics. Building on top of an existing hand gesture synthesis system, they parameterize various aspects of the gesture: spatial and temporal extent, fluidity, power, repetition, and frequency of gesture. In evaluating their system, they found that viewers were aware of changes to spatial and temporal extent, but were less cognizant of the other parameters. The authors did not evaluate whether the modified gestures still appeared natural; one wonders whether changes to the "temporal extent" of a gesture might disturb gesture-speech synchrony. In another paper in this volume, Mancini, Bresin, and Pelachaud develop a mapping between emotion and these same expressive parameters. This mapping is used to control the facial gestures of an embodied conversation agent (ECA); the result is that the agent's face can respond to the emotional content of music. The question of whether humans actually recognize the intended emotions is left to future work.

For sign language, several papers develop formal models that move beyond the sign-as-semaphore approach. Lenseigne and Dalle note that although sign language synthesis and recognition originally focused on simply producing the signs in the lexicon, spatial and temporal relationships between signs are also used to convey semantics. They develop a formal model of this phenomenon, in which semantic entities are grounded by various features of the gesture, depending on their semantic classification; for example, dates and events can be grounded by temporal information, whereas places and objects are grounded by spatial location. They argue that this model may eventually help to make sign language generation more fluent.

Similarly, Braffort and Lejeune note that in French Sign Language, spatial relationships between entities are not conveyed through specific lexicalized signs, but through the spatial arrangement of the signs for the entities themselves. They present a formal grammar for expressing such relationships, built on combinations of basic operators and relators, which express things such as whether an entity is spatially located, whether that location can serve as a landmark, and how the location relates to another localised entity. The authors have developed a protoype that translates written language into their formalism, and then synthesizes the signs necessary to express the sentence; however, at present this tool works only for simple isolated utterances such as "the cat is in the car." It would be interesting to see whether the same ideas could be applied to the synthesis of gesticulation.

While this research shows a movement towards richer models of gesture synthesis than the "gesture-as-semaphore" model, the collection includes little matching research on the recognition side. This may be because gesture semaphores are still difficult to recognize, relegating research on recognition of more nuanced models of gesture to future work. Another explanation is that most sign language testbeds are composed of single, isolated signs, rather than complete sentences.

The whole-sentence corpus currently under development by the National Center for Sign Language and Gesture Resources may ultimately push recognition research towards richer models (Neidl et al., 2000). However, in this collection, the articles on sign language recognition focused on improving recognition of individual words. Wang, Chen, and Gao synthesized new examples in Chinese Sign Language by combining existing examples using the genetic operators of crossover and mutation; these additional training examples are shown to improve performance on the test set. Zahedi, Keysers, and Ney focused on robustness to signer variability, by automatically clustering training examples according to the signer's "accent." In both cases, performance improved by a statistically significant amount, but it is unclear how well individual sign recognition will generalize to understanding fluent sign language sentences.

Paralleling the work of Hartmann, Mancini, and Pelachaud on emotion in gesture synthesis, Ilmonen and Takala describe a system for detecting emotional content from the motion of an orchestral conductor. Conductors wore a "data suit" containing magnetic motion tracking sensors. A broad set of features was extracted by transforming position and orientation into velocity, acceleration, and curvature spectrograms, histograms, and filterbank outputs. These features were then fed to an artificial neural network, which is a computer program that is capable of learning to classify observations after being "trained" on correctly-labeled examples. The resulting system was able to distinguish six types of emotional content with reasonable accuracy when the tempo was held constant. It is unclear whether the system learned features that would be general across conductors, or whether some customization to each conductor's personal style is necessary. Although the gestures employed in conducting are conventional and are different in kind from gesticulation, this research is an example of simultaneously recognizing multiple streams of information from gesture: in this case, both tempo and emotion.

Finally, Kranstedt et al. attempt to provide a richer account of deictic gestures than is typically understood in computer science approaches. They describe a compositional semantics for using deictic gestures to ground references, and argue that this model can serve two purposes: in recognition, using gesture to resolve ambiguous references; and in synthesis of multimodal referential utterances. Of particular interest to the authors is the inherent imprecision involved in pointing, which they describe as a cone of reference, rather than a line of singular width. It is this imprecision that allows speakers to reference regions rather than single objects; using hand trackers, Kranstedt et al. are currently studying the mechanics of this phenomenon.

Automatic hand and body tracking

The papers in this collection provide a good survey of the state of the art in hand and body tracking. This may be relevant to researchers outside of computer science who want to precisely quantify the mechanics of gesture movement without tedious annotation. In this section, I attempt to summarize the various techniques for hand and body tracking described in this collection.

The primary way in which hand tracking techniques can be classified is by distinguishing whether tracked beacons or computer vision is used. Tracked beacons, provided by companies such as Polhemus and Vicon, are special hardware devices that track 3D position and motion. By mounting several of these beacons on the body and hands, precise movement trajectories can be captured. In addition, instrumented gloves with bend sensors can capture the hand posture. Alternatively, computer vision can be used to attempt to infer hand motion and posture directly from video.

Tracked beacons and instrumented gloves generally yield more accurate data than computer vision, and have faster update rates. Wang, Chen, and Gao describe a system using CyberGloves and mounted trackers to recognize isolated signs from Chinese Sign Language, with a vocabulary of 5100 signs; this is a substantially greater vocabulary than any of the vision-based systems described in this collection. Similarly, as already discussed, Ilmonen and Takala used a "data suit" of mounted beacons to track the motion of musical conductors. However, despite the accuracy of such specialized hardware, computer vision is preferable from a usability standpoint, as the gesturer can move naturally, without gloves, cables, or beacons to attach.

The pipeline approach to vision-based body tracking

Perhaps the most prevalent architecture for hand tracking from computer vision is described by Dias et al., in their presentation of the Open Gestures Recognition Engine. This approach, known as a "pipeline architecture," involves transforming an image by feeding it through a series of successively higher-level vision processes, with each process treated as a black box. First, the static background is subtracted from the image, removing the parts of the image that do not move. Since lighting changes over time, modern background subtraction algorithms update their background model adaptively to take such changes into account — one method for adaptive background subtraction is described by Cassel, Collet, and Gherbi in this volume. The danger of adaptive background subtraction is that if the user remains still for too long, the user's body may become part of the background — this was not an issue for Cassel, Collet, and Gherbi, as they track fast-moving gymnasts and acrobats. It should also be noted that conventional approaches to background subtraction will fail utterly in the presence of any camera motion, panning, or zooming. Thus, such techniques can rarely be applied directly to, say, video from television or movies.

After the background is subtracted, the next step is often to attempt to identify a specific part of the foreground, typically the hand. In Dias et al., a color histogram model is used — the color values at individual pixels are compared with labeled examples of the hand (or whatever is to be tracked). In chrominance-based color spaces, such as YCrCB, skin tones occupy a tightly bounded region, making such color spaces superior to RGB. Using only color features, the tracking system will be brittle to the presence of other skin-colored objects in the scene; such systems may impose the restriction that users wear long-sleeve shirts to eliminate the forearm, and that only one person appears in the image. To solve this problem, features other than color may be used, such as shape histograms (Ankerst et al., 1999).

In whole-body tracking, the object to be tracked is often the only foreground object, making color histograms unnecessary — this was the case in Cassel, Coolet, and Gherbi, and also in Park and Lee's study on whole-body action sequences (e.g., sitting, running, jumping). Whether or not color-based object detection is applied, the result is a map indicating the likelihood of each pixel belonging to the object being tracked. Because of inherent errors in computer vision, this map will be noisy; isolated pixels throughout the image will be recognized as the target object, and "holes" in the target object may also occur. Thus, the next stage is to apply some type of smoothing operation to the pixel map. Cassel, Collet, and Gherbi describe one such operator, which they call "block filtering."

All of the steps described thus far can be applied to a single image; the final step is to perform temporal smoothing across several images. Objects in the real world move smoothly over time, so the key idea of temporal smoothing is that the tracked object's next position should be near to its previous observed location. Dias et al. use the Camshift algorithm (Allen et al., 2004), which restricts the search for the tracked object to a window bounded by the previous observed position. In Cassel, Collet, and Gherbi's paper, the Kalman filter is applied. The Kalman filter assumes that the change in the object's position from frame to frame will be governed by a Gaussian distribution, and the error in a system's estimate of the object's position will also be governed by a (different) Gaussian distribution. If these assumptions are correct, the Kalman filter is guaranteed to produce the maximum likelihood estimate of the true system state. While the Camshift algorithm offers no such guarantee, it is faster. Also, the assumption that tracking error is governed by a Gaussian distribution is unlikely to hold for hand tracking;

"distracters" that look like the hand may occur anywhere in the image, and there is no reason that they should be distributed normally about the hand's true position. An alternative to the Kalman filter that makes no Gaussian assumption is the particle filter, which is described by Moeslund and Nørgaard in this collection.

Alternatives to the pipeline approach

While the pipeline approach is ubiquitous in the literature, there are drawbacks. Most significant is that errors are propagated through the pipeline: a mistake in background subtraction will impair tracking, which may confuse the temporal smoother. Since temporal smoothing depends on continuity from frame to frame, a misidentification in a single frame can cause the tracker to be irrevocably lost. This collection contains several alternatives to the pipeline approach; while they lack the generality of the pipeline approach, they are often quite robust for the specific problem that they address.

Moeslund and Nørgaard describe a system for wearable computing that identifies two hand postures in the presence of a cluttered, non-static background. They mount a camera on the user's head that points downward to capture the hands beneath. The camera moves with the user's head, so traditional background subtraction approaches will not work. To identify the hand, a b-spline is fitted to the image, by matching known features of the interior region (the hand) — no assumptions are made about the properties of the background itself. They are able to recognize two forms of pointing gestures — one with the index finger and thumb extended from the hand, one with the index finger extended and the thumb tucked in.

Simpler approaches may also be available depending on the characteristics of the dataset. For example, if the background is controlled to contain only a dark solid color, much simpler techniques can be used. Zahedi, Keysers, and Ney are able to identify skin tones simply by thresholding pixels by their brightness, because their corpus ensures a black background. Since there are only three speakers in their dataset, it is unclear whether this approach would generalize to situations in which the speaker wears bright clothes or has dark skin. Similarly, Burns and Mazzarino employ a dataset containing only the hand and a black background. While this may seem a unrealistic scenario, it may in fact be well-suited to gestural interaction on tabletop displays (Wu & Balakrishnan, 2003). They compare several different image-processing techniques to identify the locations of individual fingertips. The circular Hough transform identifies circular regions in images, and was found to be the most accurate way to identify fingertips, although it was also the most computation-intensive. The authors also investigate lower-cost methods based on geometrical properties of the hand contour. All methods were found to be capable of identifying fingertips at a rate of thirty frames per second, using standard commodity computer hardware available in 2005.

Gestural interaction techniques

Traditionally, most gestural user interfaces have fallen into one of two categories: direct manipulation, with the tracked hand functioning as a pointer (Wexelblat1995); or, whole-gesture semaphores (Väänänen & Böhm, 1993) that function as gestural buttons, triggering specific events (cf. Quek et al., 2002). Neither approach appears to utilize the true strengths of gesture, which include the ability to express multiple pieces of information simultaneously through several features, and to qualitatively describe spatial and temporal relationships between multiple entities. However, it is not yet clear how such expressivity can be harnessed to make useful human–computer interaction techniques.

In their article in this book, Fabre et al. describe a bimanual gestural user interface for constructing 3D geometrical scenes, comprised of geometric shapes. The hands are tracked using two CyberGloves with tracking beacons. Interaction is performed via a set of six static, semaphoric gestures, but some of these gestures can be parameterized, blending in properties of the direct manipulation approach. For example, to select a region of space, the two hands make pinching gestures, and then the positions of each hand are used to set the corners of the bounding box. Another bimanual interaction technique is to use the dominant hand to select an object in the scene, while using the non-dominant hand to select a command from a menu. Navigation through the 3D scene is performed using a "raycast" metaphor, which utilizes both the posture and position of the two hands. The user's motion through space is governed by the direction and magnitude of a ray, which is manipulated through bimanual gesture. The posture of the non-dominant hand sets the magnitude of the ray; when the non-dominant hand is closed in a fist, the ray shrinks to nothing. When the non-dominant hand is open, the user can extend the ray by moving the dominant hand in relation to the non-dominant hand.

Arfib, Courturier, and Filatriau describe a series of user interfaces for controlling the parameters of music synthesis. None of these interfaces use free-hand gestures, but they could in theory be extended to do so. The first interface allows the user to create a sonic waveform by tracing a path along a non-linear surface — this is an example of the sort of direct-manipulation interaction already discussed. The second interface supports bimanual interaction, where one hand uses a stylus to set the properties of a vibrating string, and the other hand uses a touchpad to "play" the string. This appears to be a combination of direct manipulation (the touchpad) with more indirect techniques in the stylus. The final interface is perhaps the most naturalistic and unconventional: the user controls "breathing" textures by moving two tracked batons up and down in alternating patterns. Thus, the rate and fluidity of the breathing texture are controlled by the gesture.

Summary

The research described in *Gesture in human–computer interaction and simulation* reveals a variety of interesting links between the the empirical study of gesture and the development of computer science applications. Many of the innovations described in this collection of articles are possible only because of improvements in the underlying technologies of computer vision and animation. As these technologies mature, computer scientists will be increasingly free to incorporate more sophisticated models, making communication between engineers and empirical researchers of gesture even more relevant. At the same time, such collaboration may enable psychologists and linguists to benefit from these technological advances, offering new computational tools for their research.

Acknowledgements

The author thanks Randall Davis and Adam Kendon for helpful feedback and edits.

References

Allen, John G., Richard Y. D. Xu, & Jesse S. Jin (2004). Object tracking using camshift algorithm and multiple quantized feature spaces. In *CRPIT '36: Proceedings of the pan-sydney area workshop on visual information processing* (pp. 3–7). Darlinghurst, Australia: Australian Computer Society, Inc.

Ankerst, Michael, Gabi Kastenmüller, Hans-Peter Kriegel, & Thomas Seidl (1999). 3D shape histograms for similarity search and classification in spatial databases. In *SSD '99: Proceedings of the 6th international symposium on advances in spatial databases* (pp. 207–226). London, UK: Springer-Verlag.

Kendon, Adam (2004). *Gesture: Visible action as utterance*. Cambridge: Cambridge University Press.

Moeslund, Thomas B. & Erik Granum. (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81, 231–268.

Neidl, Carol, Judy Kegl, Dawn MacLaughlin, Benjamin Bahan, & Robert G. Lee (2000). *The syntax of American Sign Language: Functional categories and hierarchical structure*. Cambridge, MA: The MIT Press.

Quek, Francis, David McNeill, Robert Bryll, Susan Duncan, Xin-Feng Ma, Cemil Kirbas, et al. (2002). Multimodal human discourse: gesture and speech. *ACM Transactions on Computer–Human Interaction (TOCHI)*, 9 (3), 171–193.

Starner, Thad, Joshua Weaver, & Alex Pentland (1998). A wearable computer based American Sign Language recognizer. *Lecture Notes in Computer Science*, 1458, 84–96.

Turk, Matthew (2004). Computer vision in the interface. *Communications of the ACM*, 47 (1), 60–67.

Väänänen, Kaisa & Klaus Böhm (1993). Gesture-driven interaction as a human factor in virtual environments — an approach with neural networks. In R. Earnshaw, M. Gigante, & H. Jones (Eds.), *Virtual reality systems* (pp. 93–106). New York: Academic Press, Ltd.

Wexelblat, Alan (1995). An approach to natural gesture in virtual environments. *ACM Transactions on Computer–Human Interaction (TOCHI)*, 2 (3), 179–200.

Wu, Mike & Ravin Balakrishnan (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *UIST '03: Proceedings of the 16th annual ACM symposium on user interface software and technology* (pp. 193–202). New York: ACM Press.