Continuous Recognition of One-Handed and Two-Handed Gestures using 3D Full-Body Motion Tracking Sensors

Per Ola Kristensson, Thomas F.W. Nicholson, and Aaron Quigley
School of Computer Science
University of St Andrews, St Andrews, United Kingdom
{pok,tfwn,aquigley}@st-andrews.ac.uk

ABSTRACT
In this paper we present a new bimanual markerless gesture interface for 3D full-body motion tracking sensors, such as the Kinect. Our interface uses a probabilistic algorithm to incrementally predict users' intended one-handed and two-handed gestures while they are still being articulated. It supports scale and translation invariant recognition of arbitrarily defined gesture templates in real-time. The interface supports two ways of gesturing commands in thin air to displays at a distance. First, users can use one-handed and two-handed gestures to directly issue commands. Second, users can use their non-dominant hand to modulate single-hand gestures. Our evaluation shows that the system recognizes one-handed and two-handed gestures with an accuracy of 92.7%–96.2%.

Author Keywords
Gesture recognition, motion tracking, wall-sized displays

ACM Classification Keywords
I.5.5 Pattern Recognition: Implementation—Interactive systems

General Terms
Experimentation, Human Factors

INTRODUCTION

3D full-body motion tracking sensors enable users to interact with TV sets and wall-sized displays using their own body. However, an open research question is how to efficiently interact with displays at a distance using such sensors. In this work we use the Microsoft Kinect 3D full-body motion tracking sensor to design a bimanual continuous gesture interface that recognizes one-handed and two-handed gestures while they are being articulated by the user. The Kinect is a markerless sensor system that tracks the user's body as a skeleton structure with 20 joints. We modify and extend a recently published template-based continuous 2D gesture recognition algorithm [3] so that it can process 3D output from a Kinect sensor. In doing so we are the first to present a general bimanual gesture recognizer for the Kinect sensor that can continuously recognize arbitrarily defined gesture templates. Prior work on the Kinect have focused on pose estimation [9] and the latest Kinect Beta SDK (1.0.12) does currently not support gestures.

Gestures are natural components in human-human interaction and human neuromuscular control is well-evolved to simultaneously facilitate fast learning and accurate recall of gestures [7]. In the human-computer interaction (HCI) field, a variety of 2D gesture recognition algorithms have been developed for touchscreens and tablets. These either use a statistical model (e.g. [8]), or various forms of template matching (e.g. [1, 3, 10]). Outside the HCI field, gesture recognition typically refers to 3D gesture recognition [4]. Such gestures may be detected using a variety of sensors, such as cameras and three-axis accelerometers. Common recognition techniques include conditional density propagation, hidden Markov models, and dynamic time warping [4].

In this paper we first introduce the concept of an input zone as a gesture delimitation device for 3D full-body motion tracking sensors. We provide two reliable methods for determining whether the user's hands are inside or outside this zone. Thereafter we describe how we adapted a continuous recognition algorithm [3] for the Kinect and extended it so that it can recognize two-handed gestures. We show that our system recognizes one-handed and two-handed gestures with an accuracy of 92.7%–96.2% on held-out test data. Then we present a new interface for simultaneous bimanual gesturing and control that enables users to gesture with one hand and make selections among the predicted gesture templates with the second hand. Finally, we motivate our system by proposing a range of applications. For example, we designed a text entry method that enables users to write in thin air.

BIMANUAL CONTINUOUS GESTURE INTERACTION

We continuously track both hands of the user via the skeleton structure we obtain from the Kinect. For each hand and sample point we receive a tuple (x, y, z, t), in which x, y and z are the 3D coordinate estimates of the hand (in meters), and t is a timestamp. We smooth this signal using a moving average with an optimal window size learned from training data.

For our gesture recognition interface to work we need to be able to reliably determine the beginning and the end of the

IUI'12, February 14–17, 2012, Lisbon, Portugal.

gesture articulation by the user. To perform this delimitation we use a zoning technique. When the user is interacting with the Kinect we continuously measure the distance between the user's hand and the Kinect. When this distance is below a set threshold the hand is defined to be within what we call the *input zone*. A similar approach has previously been used to delineate 3D selections [6]. We experimented with various thresholds. One solution is to use the absolute distance between the user's hand and the Kinect. We implemented this solution and used a threshold of 1.5 meters in front of the Kinect. While this solution proved to be robust, it may in some situations be advantageous to have a flexible input zone that is invariant of the distance between the user's body and the Kinect. We therefore also developed a method for determining whether the user's hand is within the input zone based on a relative measure. It uses a binary classifier that is defined as the following decision rule:

$$z(v_b, v_h, d_h) = \begin{cases} 1, & \text{if } v_b < \gamma_{vb} \wedge v_h < \gamma_{vh} \wedge d_h > \gamma_{dh} \\ 0, & \text{otherwise} \end{cases},$$

in which v_b is the speed of the user's body, v_h is the speed of the user's hand, d_h is the distance between the user's hand and the user's body, and γ_{vb}, γ_{vh} and γ_{dh} are empirically determined parameters. When $z = 1$ the user's hand is within the input zone. We set the thresholds for γ_{vb}, γ_{vh} and γ_{dh} by estimating them from data gathered from four volunteers. The participants held a wireless mouse in their right hand. They were explained the concept of an input zone and then asked to use a forward-motion to move their hand into the input zone, then move the hand laterally, and thereafter pull their hand out of the input zone. We instructed participants to press and hold down the left mouse button when they felt that their hand was within the input zone and to release the left mouse button when they felt they were outside of the zone. Based on this data we estimated the parameter values using cross-validation. We evaluated the binary classifier by investigating how accurately it could classify a sample point as being inside or outside the input zone using three tolerance thresholds: 5, 10 or 15 sample points to the left or to the right of the boundary. Cross-validation showed that the accuracy was 88.0%, 91.2% and 93.4% respectively.

There is some noise in the signal when users are entering and exiting the input zone. We therefore discard a portion of the beginning and end of the trace, a process known as de-hooking in the handwriting recognition community. The de-hooking parameters are also learned from training data.

While the hand is within the input zone the system concatenates the projected (x, y) positions of the hand into a progressively increasing input vector:

$$\mathbf{I}_i = [(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)].$$

When a new sampling point is received the system computes a posterior probability for gesture template ω_k as:

$$P(\omega_k|\mathbf{I}_i) = \frac{P(\omega_k)P(\mathbf{I}_i|\omega_k)}{\sum_n P(\omega_n)P(\mathbf{I}_i|\omega_n)},$$

where $P(\omega_k)$ is the prior probability and $P(\mathbf{I}_i|\omega_k)$ is the likelihood. If there are reasons to believe certain gestures are more likely than others (for example, due to context), the prior can reflect this. However, in the experiments reported in this paper the prior is uniform. The likelihood is found by searching for the sub-segment of the template ω_j that maximizes the likelihood of a distance function in combination with an end-point bias [3]. The continuous recognition algorithm uses two features for the distance function: the mean Euclidean distance and the mean turning angle between two point vectors. The relative weighting between these two features is controlled via a mixture weight parameter. For details we refer the reader to the complete description of the continuous gesture recognition algorithm [3].

In this paper we extend the continuous gesture recognition algorithm so that it can also recognize two-handed gestures. In this case we have two input vectors $\mathbf{I}_i^{(l)}$ and $\mathbf{I}_j^{(r)}$ for the left- and right-hand gesture articulations respectively and we are interested in the posterior probability for a bimanual gesture $\omega_k^{(lr)}$. Assuming conditional independence under a joint model this posterior probability is:

$$P^{(lr)}\left(\omega_k^{(lr)}|\mathbf{I}_i^{(l)}, \mathbf{I}_j^{(r)}\right) = \frac{P\left(\omega_k^{(lr)}\right)P\left(\mathbf{I}_i^{(l)}|\omega_k^{(l)}\right)P\left(\mathbf{I}_j^{(r)}|\omega_k^{(r)}\right)}{\sum_n P\left(\omega_n^{(lr)}\right)P\left(\mathbf{I}_i^{(l)}|\omega_n^{(l)}\right)P\left(\mathbf{I}_j^{(r)}|\omega_n^{(r)}\right)},$$

in which $P\left(\omega_k^{(lr)}\right)$ is the prior for the bimanual gesture, and $P\left(\mathbf{I}_i^{(l)}|\omega_k^{(l)}\right)$ and $P\left(\mathbf{I}_j^{(r)}|\omega_k^{(r)}\right)$ are the likelihoods for the left- and right-hand parts of the bimanual gesture template respectively.

We also created a bimanual interface that enables users to simultaneously using the dominant hand to gesture and the non-dominant hand to modulate the recognition results. Figure 1 shows a user gesturing using this interface. In the figure the user is writing text in thin air by gesturing Graffiti letters (cf. Figure 2). While the user is gesturing the system continuously updates the four most likely predicted gesture templates to the left in the display. By moving the non-dominant hand up or down the user can select among these alternatives. We found that this style of interaction is particularly effective if a user requires very high recognition accuracy.

Figure 1. A user is writing in thin air by gesturing Graffiti letters and selecting among alternative predicted letters to the left of the display.

EVALUATION

We tested recognition accuracy on three gesture sets: the $1 gesture set [10], the Graffiti gesture set, and a bimanual gesture set which we created ourselves. The $1 gesture set and

the Graffiti gesture set have been previously used in recognition experiments (e.g. [3, 10]). The gesture sets are illustrated in Figure 2.

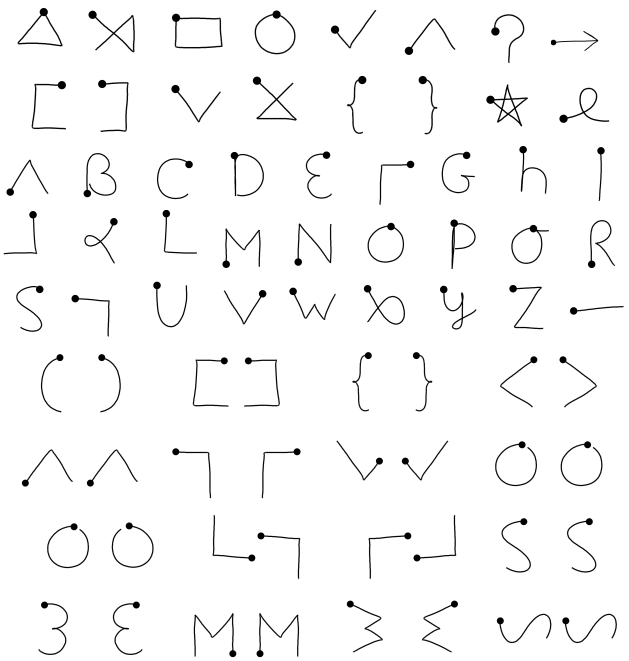

Figure 2. The gesture sets used in the evaluation. The beginning of a gesture is indicated by a solid dot. Top: the set of one-handed gestures from the $1 gesture set [10]. Middle: the set of one-handed gestures from the Graffiti alphabet. Bottom: the set of two-handed gestures we created ourselves.

We used a Microsoft XBox 360 Kinect sensor connected to a Windows 7 laptop. Each session was divided into a practice and a testing part. In the practice part we explained to the participants the concept of the input zone and demonstrated how to articulate gestures with the Kinect. In the testing part participants were shown a series of gestures on the screen and asked to reproduce them as quickly and as accurately as possible. To ensure reproducibility we used an absolute distance of 1.5 meters between the participant's hand and the Kinect to decide whether the participant's hand was inside or outside the input zone.

We recruited 18 volunteers from our university campus. Their ages ranged between 18–35. None of the participants had any prior experience of using a Kinect sensor. The 18 participants were randomly split into two groups with nine participants each to avoid effects due to fatigue and motor learning confounding the results. Participants in the first group were asked to to articulate the gestures in the Graffiti and the bimanual gesture sets. Each Graffiti gesture was performed once and each gesture in the bimanual gesture set was performed five times. In total, each participant in this group performed 27 + $16 \times 5 = 107$ gestures. Participants in the second group were asked to to articulate the gestures in the $1 set. Each gesture in the gesture set was performed five times. In total, each participant in this group performed $16 \times 5 = 80$ gestures. This data collection task lasted approximately 20–30 minutes for each participant.

Recognizer (E,T,E+T) and input style	Accuracy	Δ
One-handed (E)	90.3%	·
One-handed (T)	92.3%	2.0%
One-handed (E+T)	92.7%	0.4%
Two-handed (E)	86.2%	·
Two-handed (T)	96.2%	10.0%
Two-handed (E+T)	96.2%	0%

Table 1. Accuracy and absolute gains in accuracy for complete one-handed and two-handed 2D gestures. One-handed gesture recognition was tested with the $1 [10] and the Graffiti gesture sets. Two-handed gesture recognition was tested with a gesture set we designed ourselves. E: Euclidean distance only, T: turning angle only, E+T: combination.

In total we collected 1683 one-handed and two-handed gestures. We split the data into a training and test set by randomly designating the gestures collected from four participants in each group (eight participants in total) as the training set and the rest of the gestures as the test set. The held-out test set was only used for final evaluations. Using the training set we searched for the optimal mixture weight for the continuous gesture recognition algorithm and the two additional parameters we have introduced in this paper: the window size of the moving average used for smoothing, and the proportion of the input gesture which is de-hooked. We used the parameter configuration that maximized accuracy on the training set and evaluated the algorithms on the held-out test set.

Table 1 summarizes the recognition results for complete gestures. The system recognized complete one-handed gestures with an accuracy of 92.7%. This is similar to prior work [3] on recognition of pen stroke gestures drawn on a Tablet PC, which found that continuous recognition of complete gestures resulted in 94.5% accuracy. Also consistent with prior work [3], the turning angle feature resulted in higher accuracy than Euclidean distance. Combining both features using an optimal mixture weight resulted in a negligible gain in accuracy.

The system recognized the two-handed gestures with an accuracy of 96.2%, which is slightly higher than the accuracy obtained for one-handed gestures. Again, the turning angle feature resulted in higher accuracy than Euclidean distance. Combining both features made no difference. Bimanual gesture recognition was easier, which is unsurprising since under an appropriate probabilistic model two simultaneous input gestures provide additional information to the recognizer.

Figure 3 plots average accuracy as a function of the proportion of a complete one-handed or two-handed gesture. The solid lines in the figure show the performance of the continuous gesture recognition algorithm. The dashed lines show the performance of a baseline algorithm that only recognizes complete gesture templates. The baseline algorithm uses optimal parameter values learned from the training set. Cross marks denote the accuracy for top-1 matches while squares denote the accuracy for top-3 matches.

As is evident in the figure, the continuous recognizer is much better at predicting the participants' intended gestures at all stages of the articulation process. As a reference point, the continuous gesture recognition algorithm was able to achieve an accuracy of 46.0% for one-handed and 55.0% for two-

handed gestures when participants had gestured only 20% of the complete gesture (top-1 matches). This was an absolute gain in accuracy of 39.3% and 38.4% compared to the baseline for one-handed and two-handed gestures respectively.

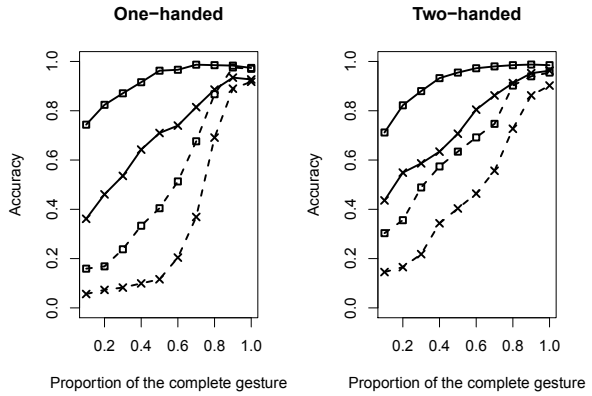

Figure 3. Accuracy for one-handed and two-handed gestures as a function of the proportion of the complete gesture for the continuous gesture recognizer (solid lines) and a baseline recognizer (dashed lines). Cross marks denote the accuracy for top-1 matches while squares denote the accuracy for top-3 matches.

DISCUSSION

As in previous work (e.g. [1, 3, 10]), we only tested a few gesture sets. Therefore, our results should be interpreted with some care. Unfortunately there is no open collection of gesture sets available for standardized comparisons.

We chose to delimit gestures via an input zone. Participants felt this was natural and they quickly adapted to it. With 92.7–96.2% accuracy for single-hand and two-handed gestures our system is directly usable as a command interface for interactions with displays at a distance, such as when users are playing video games or interacting with wall-sized displays.

A potential application of our system would be as an input technique in hospital operating rooms (ORs). ORs require hospital workers to maintain the boundary between sterile and non-sterile environments [2]. Our system enables hospital workers to access and edit information while ensuring this boundary is kept at all times. Also, unlike most gesture recognition systems, our system performs recognition continuously while the user is gesturing. This enables hospital workers to verify that the gesture will be correctly recognized as the intended command before completing it. This minimizes the risk of recognition errors as hospital workers can choose to abort the gesture if it is not properly recognized and thus prevent an unintended command from being committed.

Another application of our system is as a text entry method for writing in thin air. The 27 gestures in the Graffiti alphabet we tested suffice to perform rudimentary text entry tasks, such as writing a short message or filling out a medical form. Figure 1 shows a user writing in thin air using Graffiti. Our gesture interface makes it possible to implement freehand Graffiti input techniques for thin air that have previously only been explored via a specialized glove-based system [5].

CONCLUSIONS

We have presented a bimanual gesture recognition system for 3D full-body motion tracking sensors, such as the Kinect. Our system continuously recognizes arbitrarily defined gesture templates in real-time on a standard laptop. Our evaluation showed that the system recognizes one-handed and two-handed gestures with 92.7–96.2% accuracy.

To enable comparisons against future gesture recognition algorithms we have shared our collected data. Further, to help designers, developers and researchers we have also released the C# source code for our recognition algorithms under an open source license. Data and code can be downloaded here: `http://pokristensson.com/kinect.html`.

ACKNOWLEDGEMENTS

This work was supported by the Engineering and Physical Sciences Research Council (grant number EP/H027408/1) and the Scottish Informatics and Computer Science Alliance.

REFERENCES

1. Appert, C., and Bau, O. Scale detection for a priori gesture recognition. In *Proc. CHI 2010*, ACM Press (2010), 879–882.
2. Johnson, R., O'Hara, K., Sellen, A., Cousins, C., and Criminisi, A. Exploring the potential for touchless interaction in image-guided interventional radiology. In *Proc. CHI 2011*, ACM Press (2011), 3323–3332.
3. Kristensson, P. O., and Denby, L. C. Continuous recognition and visualization of pen strokes and touch-screen gestures. In *Proc. SBIM 2011*, ACM Press (2011), 95–102.
4. Mitra, S., and Acharya, T. Gesture recognition: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37* (2007), 311–324.
5. Ni, T., Bowman, D., and North, C. AirStroke: bringing unistroke text entry to freehand gesture interfaces. In *Proc. CHI 2011*, ACM Press (2011), 2473–2476.
6. Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. The Go-Go interaction technique: non-linear mapping for direct manipulation in VR. In *Proc. UIST 1996*, ACM Press (1996), 79–80.
7. Rosenbaum, D. A. *Human Motor Control*. Academic Press, San Diego, CA, USA, 1991.
8. Rubine, D. Specifying gestures by example. In *Proc. SIGGRAPH 1991*, ACM Press (1991), 329–337.
9. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. Real-time human pose recognition in parts from single depth images. In *Proc. CVPR 2011*, IEEE Press (2011), 1297–1304.
10. Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proc. UIST 2007*, ACM Press (2007), 159–168.