Gesture Recognition Using Laser-Based Tracking System

Stéphane Perrin, Alvaro Cassinelli and Masatoshi Ishikawa
University of Tokyo, Ishikawa Hashimoto Laboratory
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
alvaro,sperrin,ishikawa@k2.t.u-tokyo.ac.jp

Abstract

This paper describes a finger gesture recognition system based on an active tracking mechanism. The simplicity of this tracking system is such that it would be possible to integrate the whole system on a chip, making it an interesting input interface for portable computing devices. In this context, recognition of gestural characters allows information to be input in a natural way. The recognition of three dimensional gestures is also studied, opening the way to a more complex interaction mode and to other kinds of applications.

1. Introduction

Input of information is becoming a challenging task as portable electronic devices become smaller. Alternatives to the keyboard have been proposed for portable devices such as personal digital assistants (PDAs). For example, input of data is often done through a touch sensitive screen using a prescribed input method, such as Graffiti®. The next step is to remove the need for an input device such as a stylus, thus allowing input using only the fingers [17]. This paper proposes a system whose main purpose is to provide a user with a natural way of interacting with a portable device or a computer through the recognition of finger gestures. This system can form part of a so-called Perceptual User Interface (PUI [16]), whereby the gestures would be one mode of interaction, along with speech, for example. The system could be used for Virtual Reality or Augmented Reality [10, 14] systems. The kind of gestures that should be considered will vary depending on the applications.

First, in this paper, a system that solves the problem of actively tracking gestures is described. This active tracking system, described in detail in [7], allows us to directly obtain the position of a tracked finger. The system, which is depicted in Fig.1, is based on a wide-angle photodetector and a collimated laser beam generated by a laser diode and steered by means of a two micro-mirrors. It is interesting to note that this tracking system does not require the user to hold any special device (such as gloves which are commonly used for such systems [15]). Moreover, this system offers the possibility of tracking 3D gestures. Next, this simple active tracking system is used as an input method of data, that is characters, similarly to the use of a stylus on some PDAs. Gesture Recognition (GR) itself is performed using Hidden Markov Models (HMMs) as described in [8]. Several problems arise and solutions are proposed, including other ways to perform the GR.

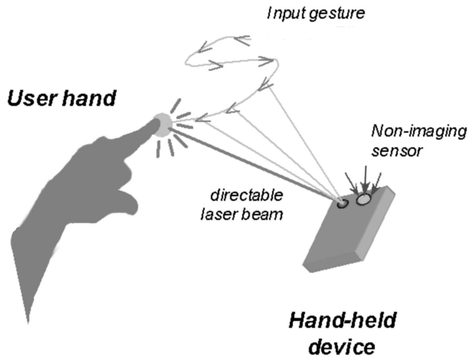

Figure 1. The proposed active tracking system as a human-machine interface for hand-held devices.

Finally, other applications of the described system are proposed, some of them using the ability to track 3D gestures.

This paper is organized as follows. Section 2 gives a brief description of the gesture tracking system with a study of its ability to track 3D gestures. Section 3 focuses on gestural character recognition itself using HMMs. This section discusses solutions to the problems encountered in GR and proposes other methods. In the conclusion section, the most relevant results are summarized and future research directions outlined.

2. Active Tracking

2.1. Two Dimensional Tracking

Tracking is based on the analysis of a temporal signal corresponding to the amount of backscattered light measured during a laser *saccade*, i.e. a rapid laser scan generated in the neighborhood of the tracked object (see Fig.2). While tracked, the object continuously backscatters some laser light (Fig.2.a). When the object moves, the backscattered signal is lost and tracking fails (Fig.2.b). The system then generates a local scanning saccade (Fig.2.c), and re-centers the laser at the new backscattering position (Fig.2.d). If this process is repeated rapidly enough, the object will always be within the reach of a small saccade.

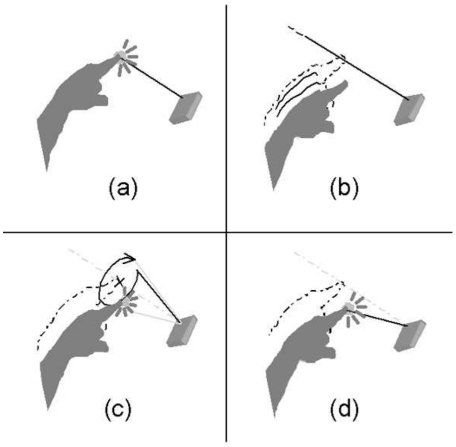

Figure 2. Principle of the non-imaging tracking system based on a laser saccade.

A circular saccade was selected because this trajectory is easy to generate and has good symmetry properties that translate into reduced algorithm complexity. A detailed description of the tracking algorithm used is given in [7].

The performance of the system was evaluated by measuring the maximum speed an object could move without being lost by the tracking system. The tracked object was a circular piece of white paper, $R_{obj} = 9$ mm in radius, tracing a circular trajectory at different uniform speeds. The distance between the mirrors and the object remained constant.

It is important to note that the maximum speed experimentally obtained was about 2.75 m/s for a tracked object whose size is approximately the same as a finger tip (usually between 1.5 cm and 2 cm in diameter). This speed value is higher than the typical speed of a finger performing gestures. Therefore, the present system is able to easily track a finger tip.

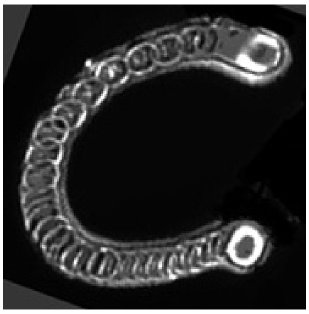

Figure 3. Example of the tracking of a finger gesture (character C).

An illustration of the tracking algorithm appears in Fig.3 which shows the tracking of a gesture performed by a user with one of his fingers. Only the backscattered laser light is shown.

All the gestures are expected to be performed at an approximately constant distance from the mirrors. By measuring the azimuth and elevation angles of the steering mirrors, the two dimensional position of the tracked object can be computed.

2.2. Three Dimensional Tracking

To determine the azimuth and elevation of the tracked finger, it is necessary to have enough contrast between the backscattered signal and the background illumination. Also, the absolute value of the backscattered intensity can be used to estimate the distance from the tracking system to the finger, without resorting to complex telemetric techniques such as triangulation or time-of-flight measurements (this is possible because the working distance in the target application remains relatively small - on the order of tens of centimeters). The maximum working distance and the achievable precision of the estimated depth, are both dependent on the noise characteristics of the backscattered signal as well as the illumination background. Since synchronous detection is not implemented in our prototype, dark room conditions were necessary to minimize background noise. The background noise intensity, I_B, was assumed to be a Gaussian distribution with a measured mean $\langle I_B \rangle = 7$ nW and variance $\sigma^2_{I_B} = 0.1$ nW. The characteristics of the backscattered signal are modeled here as a simple ”coplanar” configuration, represented in Fig.4. The intensity of the laser source was fixed to $I_L = 370$ μW (after reflection at the galvano-mirrors). We verified that the laser light could be, for our purposes, considered to be isotropically backscattered by the skin on a finger-tip. However, the spatial characteristics of the backscattered light (speckle) vary considerably depending on the micrometer-scale position of the

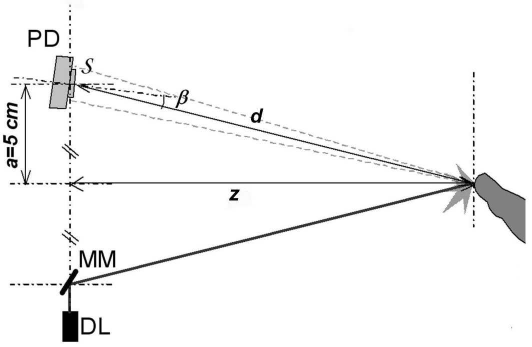

Figure 4. Relevant geometrical parameters of the setup. S: active surface of the photodetector. PD: Photo-Detector. MM: Micro-Mirrors. DL: Diode Laser.

incident point on such a microscopically wrinkled surface. This accounts for a significant fluctuation of the amount of light integrated by a distant, small photodetector with an active area $\mathcal{S} = 25$ mm^2. The random variable $\mathcal{R}$ will model the object's reflectivity. From the geometrical configuration, it follows that the intensity of the signal backscattered by the objecy I_O as a function of the depth z is approximated by equation 1.

$$I_O(z) = \frac{\mathcal{R}.I_L.\mathcal{S}.\cos\beta}{(a^2+z^2)} + I_B \tag{1}$$

(This simplified equation assumes that the finger-tip stays somewhere close to the photodetector-source median plane.) Because in our configuration the photodetector is such that β is small, the approximation $\cos\beta \approx 1$ also holds. Thus, the mean intensity of the backscattered signal as a function of the depth is $\langle I_O(z)\rangle = \frac{I_L.\mathcal{S}}{(a^2+z^2)}\langle\mathcal{R}\rangle + \langle I_B\rangle$, and its standard deviation is just $\sigma_{I_O}(z) = \frac{I_L.\mathcal{S}}{(a^2+z^2)}\sigma_{\mathcal{R}} + \sigma_{I_B}$, since the background noise and the reflectivity noise are independent phenomena. It was verified that the proposed model agrees well with the experimental results.

2.2.1. Maximum Working Distance The binary signal obtained after thresholding the photodetected signal should be as meaningful as possible: a measured intensity of magnitude $\langle I_O(z)\rangle$ most probably indicates that the laser beam is actually hitting the objet's surface, while a signal of magnitude $\langle I_B\rangle$ most probably indicates a miss. In our present algorithm, the binarisation threshold $I_{th}(z)$ is fixed in such a way that the probability of an erroneous result is independent of the actual situation. In other words, the binary signal is modelled as a binary symmetric noisy channel. This means that the tails of the noise distributions of $I_O(z)$ and I_B are equal at $I_{th}(z)$. Assuming Gaussian noise, as described above, the depth-dependent ”symmetric” threshold is therefore given by equation 2.

$$I_{th}(z) = \left|\frac{\langle I_O(z)\rangle .\sigma_{I_B} + \langle I_B\rangle .\sigma_{I_O}(z)}{\sigma_{I_B} + \sigma_{I_O}(z)}\right| \tag{2}$$

(assuming that $\langle I_B\rangle < I_{th}(z) < \langle I_O(z)\rangle$). In particular, if both variances are negligible or equal, then the symmetric threshold is, as expected, equidistant from the mean background and to the mean backscattered intensity.

The confidence $Conf$ of the estimation resulting from the use of this threshold is equal to the cumulative distributions up to $I_{th}(z)$, of any of these noisy variables and is given by equation 3.

$$Conf(z) = \frac{1}{2}\left\{1 + erf\left(\frac{I_{th}(z) - \langle I_B\rangle}{\sigma_{I_B}\sqrt{2}}\right)\right\} \tag{3}$$

The robustness of the tracking is directly related to this quantity. As was verified, the tracking robustness decreases with distance. If a minimum confidence of 95% is sought for a proper discrimination between the object and the background, then in our present configuration (and using a Class-I equivalent laser source) the maximum working distance is about 166 cm. This is more than enough for the application considered.

2.2.2. Depth Resolution It is easy to understand that, if the variance of the measures were constant, as the finger moves away from the system, the precision of the computed distance would decrease. However, the variance of the backscattered signal not only varies, but can also grow large as the finger approaches the system. In fact, for a fixed confidence $Conf$ of the depth discrimination, the achievable resolution $\Delta(z, Conf)$ can be precisely computed as a function of the distance. By using a binarisation threshold computed for $I_O(z)$ and $I_O(z+\Delta)$ and by fixing the value for the confidence, an equation is obtained that can be solved for $\Delta(z, Conf)$. Fig.5 represents the achievable resolution for several values of $Conf$ as a function of the distance (the system can discriminate a point whose depth is z from a point situated $\Delta(z, Conf)$ away with a confidence equal to $Conf$). We see from the figure that the system has optimal depth resolution when the finger is placed at a distance of around 5.5 cm from the mirrors. Two points whose depths differ by 5.5 mm can be discriminated with 95% confidence, and a finer resolution of 1.7 mm is achieved with 70% confidence. The system can reach sub-millimetric precision with a lower confidence of 60%.

At 30 cm, the resolution drops to the rather unusable level of 20 cm (for 95% confidence). Then, if one is seeking a 95% confidence for depth discrimination, only a few ”levels” (in the depth direction) would be available in an operating area limited to 10 or 20 centimeters wide. However,

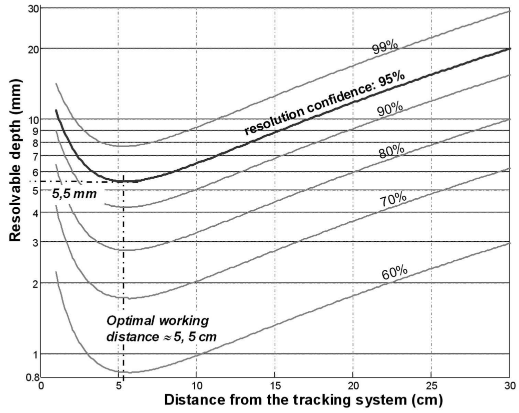

Figure 5. The achievable resolution for a given confidence, as a function of the distance.

a very simple technique can be used to drastically reduce the variance of the backscattered signal: the unavoidable intensity variations due to the speckle-generating reflection can be simply averaged by using a slow integrating photodetector, while maintaining the laser active for a slightly longer duration, even when the mirrors have started moving towards the new sampling position. Using such a technique, it should be possible to obtain millimeter-scale depth resolution up to 10 or 20 cm from the system.

Fig.6 represents a proof of principle experiment demonstrating the ability of the system to perform 3D tracking of a finger.

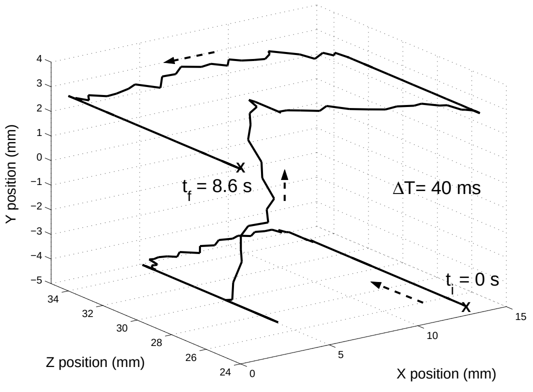

Figure 6. Experimental result of the 3D tracking of a finger.

3. Gesture Recognition

3.1. Gestural Character Recognition with HMMs

The first application of the system is the input of 2D gestural characters in a similar way to what is done using a stylus and the Graffiti® method. Six characters from the modern Latin alphabet were selected (A, B, C, D, E and S, as shown in Fig.7) in order to evaluate the performance of the proposed system. The chosen method for recognition was based on HMMs. Details of HMMs can be found in the literature [9]. The software used for the GR was HTK [19].

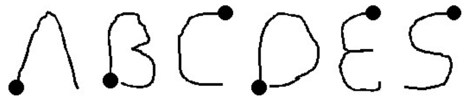

Figure 7. The gestural equivalents of the chosen characters.

In order to recognize gestures with HMMs, the first step is to build models that characterize the gestures that will be modelled as a sequence of values for the extracted features. Such models are then used for recognizing other gestures. In order to build the models, that is, the HMMs, they have to be trained by using data similar to the data to be recognized later. Once a model kind has been chosen (a 5 states left-right model was chosen), the parameters of the HMMs are determined by means of this training, which is a process of estimation and re-estimation.

To train the HMMs, we have to provide data files that characterize the gestures. The first step is thus to extract features from the gestures. The choice of the extracted features is important as these features will model the gestures and are used to train the HMMs. The features must be carefully chosen in order to obtain an efficient GR system. The choice of the features depends on some constraints, such as the kind of features that can be extracted from the raw data and the invariance desired [1, 5]. In our case, the raw data is the angles of each mirror. However, since we want the recognition to be as size-invariant as possible (note that in our case, the recognition with HMMs is translation-invariant anyway), the angles themselves were not used as features to characterize gestures. Instead, speed, acceleration and direction were used. A discussion on the choice of these particular features is provided in [8].

The computation of the three features (speed, acceleration and direction) is done using a sliding window of a given size W that is moved by W sample steps along all sample data, which are the angles. That is, the speed, acceleration and direction are computed using a given number of samples, which is the window size. Raw data is obtained every 2 ms which was found to be too short to extract relevant features. This is the main reason why sub-sampling was performed, features were obtained every 10 ms (that is, $W = 5$).

The direction is computed from the slope of the regression line of all points included in the sliding window for each of its positions. The slope is then converted to an angle and then to an integer value representing a given angular sector. Sixteen sectors were used in our case. The speed and acceleration are computed directly from the position (obtained from the angles), considering that gestures are performed in a plane always roughly at the same distance from the mirrors. These three features model the gestures and are used by the recognition software for training and then recognition.

3.1.1. Explicit Transitions The main problem when using such a method is that the recognition system has no clues for determining when the user is performing a character and when he is performing a transition gesture between two characters. The first solution is to model all the possible transitions and build an extensive database. However, two problems occur with this approach. First, the number of possible transitions among x different characters is x^2. This number quickly becomes huge when considering the whole modern Latin alphabet, for example. The second problem is that the more characters to be recognized, the greater the number of transitions to be modeled. Moreover, similarities appear between these transitions, which can be extremely similar to characters, thus leading to a considerable number of errors.

The second solution is to indicate transitions explicitly, that is, to tell the system when the user is performing a relevant gesture (a character) and when he is performing an irrelevant gesture (a transition between two characters). This was the chosen solution and the user had to push a button during the acquisition of a character, and release it when performing a transition. For convenience, the third dimension could be used for performing this action. (A small, quick movement of the finger back and forth could be interpreted as pushing a button or a switch.)

The Fig.8 shows an example of the tracking of the finger of a user drawing the character B.

3.1.2. Results The results obtained in such conditions showed several problems. First, only gestural characters whose first and last points were not at the same position (that is, all considered characters except D and B, see Fig.7) could be successfully recognized. Then, there was some confusion in the recognition of the character C. When several C's were performed in a row, they looked like several A's performed in a row, leading to confusion between C's and A's. Secondly, the HMM-based method was unable to distinguish a short pause inside a character gesture from the pause between two gestures. HMM-based methods are dynamic methods which recognize the evolution of the states of an observation. If temporal continuity is lost, these methods fail to perform a valid segmentation. A method to obtain better results could be to divide the actual morphemes, that is whole characters, into smaller morphemes such as segments or curves.

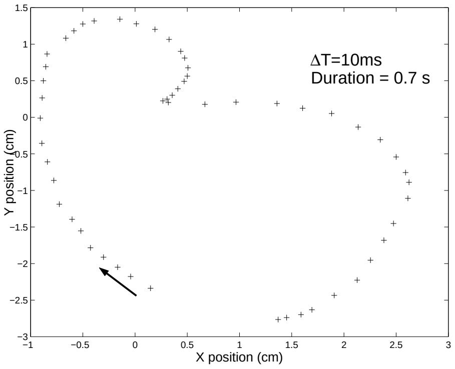

Figure 8. Experimental result of the 2D tracking of the character B.

3.2. Other Recognition Methods

HMMs proved to be well-adapted to hand gestures recognition [13, 18]. However, in the case of finger gestures, and especially handwritten characters recognition, other methods appear to be more suited. Instead of recognizing sequences of characters, one can consider recognizing sequences of words. Doing this prevents the user to perform a special gesture between each character, as suggested before. Each word would be ”written” in the air by the user with one of his fingers (see Fig.9, top and bottom). Once a word is completed, the following word would be written roughly at the same position. The movement made by the user from the end of a word to the beginning of the next one can be recognized by the recognition system as a ”space” between words (see Fig.9, middle). The recognition system itself could be extremely similar to existing handwriting recognition systems (based on Time Delay Neural Networks [3, 11, 12] or other pattern recognition techniques [4]). The problem of lifting the pen between some letters can be solved by using the third dimension.

4. Conclusions and Future Work

A system aimed at tracking a finger without the use of any special device for gestural character input was proposed

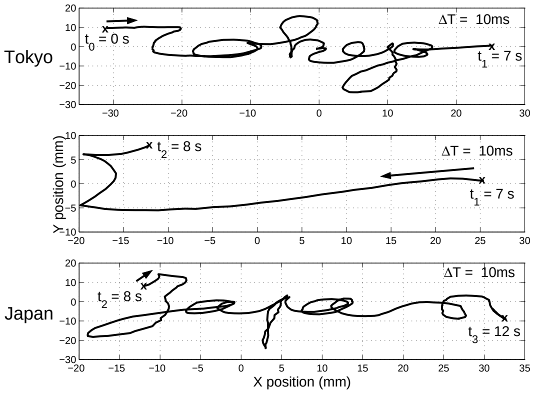

Figure 9. Experimental result of the 2D tracking of two handwritten words (”Tokyo Japan”) with the transition.

and successfully demonstrated. Moreover, it was shown that 3D gesture tracking is possible. Gesture recognition with the proposed HMM-based method is possible but more suitable recognition methods should be considered, performing recognition either on a character basis or on handwriting (word) basis.

Further research will be conducted on an efficient interface based on 3D gesture recognition. For example, the use of the third dimension allows switching between different modes, such as from character input mode to a mouse-like mode (where a pointer is moved according to the finger movement). Alternatively, in drawing application, switching from a pencil tool to an eraser tool, for example, is possible. This switching can be done in a similar way to that used for indicating the explicit transition, as described above. Of course, the third dimension can be used for more complex applications that fully make use of this feature of the system. One application could be to define a complete 3D alphabet in order to allow a more natural and richer interaction language with a machine. Another could be to allow the user to input 3D shapes or drawings, as shown on Fig.6. For more specialized applications, the 3D capacity offers users a natural and powerful way to perform some tasks, such as controlling the zooming of a map.

A further step is to use gestures along with other modalities, such as speech. Such a multimodal system [2, 6] could provide a user interface that would combine the complexity and the naturalness of human interaction.

References

[1] L. Campbell et al. Invariant features for 3-d gesture recognition. *Proc. FG’96*, pages 157–162, 1996.

[2] W. Feng. Using handwriting and gesture recognition to correct speech recognition errors. *Proc. 10th Int’l Conf. Adv. Sci. and Tech.*, pages 105–112, Mar. 1994.

[3] S. Jaeger, S. Manke, and A. Waibel. Npen++: An on-line handwriting recognition system. *Proc. 7th Int’l Workshop on Frontiers in Handwriting Recognition*, pages 249–260, Sep. 2000.

[4] A. K. Jain, R. P. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. PAMI*, 22(1):4–37, Jan. 2000.

[5] H. Morrison and S. J. McKenna. Contact-free recognition of user-defined gestures as a means of computer access for the physically disabled. *Proc. 1st Cambridge Workshop on Universal Access and Assistive Technology*, pages 99–103, Mar. 2002.

[6] S. Oviatt. Multimodal interfaces for dynamic interactive maps. *Proc. Conf. Human Factors in Computing Systems*, pages 95–102, 1996.

[7] S. Perrin, A. Cassinelli, and M. Ishikawa. Laser-based finger tracking system suitable for MOEMS integration. *Proc. IVCNZ’03, New Zealand*, pages 131–136, 26-28 Nov. 2003.

[8] S. Perrin and M. Ishikawa. Quantized features for gesture recognition using high speed vision camera. *Proc. SIBGRAPI’03, Brazil*, pages 383–390, Oct. 2003.

[9] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, Jan. 1986.

[10] Y. Sato, Y. Kobayashi, and H. Koike. Fast tracking of hands and fingertips in infrared images for augmented desk interface. *The 4th Int’l Conf. Automatic Face and Gesture Recognition*, pages 462–467, Mar. 2000.

[11] M. Schenkel, I. Guyon, and D. Henderson. On-line cursive script recognition using time delay neural networks and hidden markov models. *Proc. ICASSP ’94*, 2:637–640, 1994.

[12] G. Seni, R. K. Srihari, and N. M. Nasrabadi. Large vocabulary recognition of on-line handwritten cursive words. *IEEE Trans. PAMI*, 18(7):757–762, 1996.

[13] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. Technical Report TR306. 5, MIT Media Lab, 1995.

[14] T. Starner et al. The perceptive workbench: Computer-vision-based gesture tracking, object tracking, and 3d reconstruction for augmented desks. *Machine Vision and Applications*, 14:59–71, 2003.

[15] D. J. Sturman and D. Zeltzer. A survey of glove-based input. *IEEE Computer Graphics and Applications*, 14:30–39, 1994.

[16] M. Turk and G. Robertson. Perceptual user interfaces. *Communications of the ACM*, 43(3):33–34, 2000.

[17] C. Von Hardenberg and F. Bérard. Bare-hand human-computer interaction. *Proc. ACM Workshop on Perceptive User Interfaces*, Nov. 2001.

[18] Y. Wu and T. S. Huang. Vision-based gesture recognition: A review. *Lecture Notes in Computer Science*, 1739, 1999.

[19] S. Young et al. *The HTK Book (for HTK Version 3.1)*. Cambridge University, Dec. 2001.