Ibero-American Symposium on Computer Graphics - SIACG (2006)
P. Brunet, N. Correia, and G. Baranoski (Editors)

Recognizing Hand Gestures with CALI

Ricardo Jota[†] Alfredo Ferreira[‡] Mariana Cerejo José Santos Manuel J. Fonseca Joaquim A. Jorge

Inteligent Multimodal Interfaces Group
Department of Information Systems and Computer Science
INESC-ID/IST/Technical University of Lisbon
http://immi.inesc-id.pt

Abstract

Human computer interaction techniques that do not rely on devices are often perceived as more natural by users. Many of these, include hand pose recognition as an interaction technique appealing to users. In this paper we describe and evaluate two techniques for hand pose recognition, based on CALI, a general library for gesture recognition. This library was initially designed for calligraphic recognition, however recent usage shows that CALI is able to support other applications. One unexplored research area includes its application to hand pose recognition, even though there are already different approaches to the subject using techniques such as Hidden Markov Models or Model-based tracking. We developed and tested a new approach to recognize hand poses taking advantage of the features obtained from CALI. To explore this approach we implemented two techniques. The first recognizes bare-hands by their outer contours, the second uses color marks on each fingertip to track the hand and recognize its pose. Experimental results show that both approaches present recognitions rates around 93%.

Categories and Subject Descriptors (according to ACM CCS):
H.5.2 [Information Interfaces and Presentation]: User Interfaces - *Input Devices and Strategies*
I.3.6 [Pattern Recognition]: Implementation - *Interactive Systems*

1. Introduction

Since put-that-there [Bol80] was presented, in 1980, researchers were drawn to the possibility of controling computers without resorting to the keyboard/mice duo. This lead to a rising number of multi-modal interfaces being presented every year. Some of those works argued that one way to improve interaction is to use our own natural interaction tools: our hands. However, different paths were taken. Some researchers choose to use gloves to better capture hand postures while others followed the bare-hand path. Both paths bring different approaches on gesture interaction, the first focus on having robust tracking methods available while the second focus on having the user free from any interaction device, gloves included. This, theoretically, allows the user to shift attention between different tasks without disposing of any interaction devices, being less intrusive and making the user interaction more seamlessly.

We believe that gloves are almost as awkward to the user as other interaction devices. However, using bare-hands for interaction usually depends on complex detection algorithms and tracking methodologies, unlike the glove-based technique that relies on the hardware to provide most of the information. To improve the efficiency of bare-hand interaction, simpler and faster hand pose detection algorithms and tracking methodologies are required.

We present two techniques to identify hand poses. Towards this, we use a generic recognition library called CALI published by Fonseca *et al.* [FFJ05]. CALI was initially devised for recognition in calligraphic interfaces [FPJ02]. Following CALI success on hand-drawn recognition, it was generalized to classify more general shapes. This generic version of CALI has been used mainly in shape classification for retrieval uses. In the present work, instead

[†] R. Jota was supported by the Portuguese Foundation for Science and Technology, grant reference SFRH/BD/17574/2004.
[‡] A. Ferreira was supported by the Portuguese Foundation for Science and Technology, grant reference SFRH/BD/17705/2004.

of using CALI to identify specific shapes or gestures from sketches or classify shapes for retrieval, we suggest widen its application to hand pose recognition. We propose a recognition strategy to be used in the two techniques. In the first technique we use the hand silhouette while in the second we use fingertips information to produce a polygon representation of the hand pose. With the proposed strategy we expect to achieve recognition rates, at least, similar to the ones produced by existing approaches to hand pose recognition, but requiring less complex algorithms or fewer computation time, along with simpler hardware.

The paper is organized as follows: after a short discussion on related work, we describe the proposed recognition strategy and the two techniques. Next, we present experimental results and compare our results to other known techniques. Finally, we explain our conclusions and define future work.

2. Related Work

Hand gestures, along with methods for hand usage in human-computer interaction [SZ93], were presented as valid for human-computer interaction. A huge variety of recognition methods has been documented. Quek and Zhao [QZ96] and others [WH00] used inductive learning in order to reduce computation time but this required a large training set. Nolker *et al.* [NR96] used Hidden Markov Model to identify simple gestures. Her work also needs a large training set. By 1998 one of the first papers describing model-based tracking for gesture recognition was presented [LH98], the main problem with model-based tracking is that its computation algorithm weight does not allow for a real-time recognition. By the end of the nineties, the main research paths were defined, one used model-based recognition while the other pursued appearance-based approaches.

More recently some works [SKK00, vHB01, OSK02] focus on tracking fingertips as a gestures recognition strategy. This technique require two steps, fingertips detection and gesture recognition, hence slowing down the system. Although this strategy has proven successful, systems where fingertips are not needed the first step may be redundant. Sato [SSK01] also presented a neural network approach, which also required a good training set. In 2003 Wu and Balakrishnan [WB03] published a work using hand gestural interaction, in this work they use a touch surface to aid the gesture recognition. Using a touch surface detracts from the work recognition, even though the work has a good recognition rate. Because most scenarios don't have access to a touch surface we do not view this as a desired setup. Rivière and Guitton [dlRG03, dlRG05] use model-based tracking to recover postures and image moments to extract translation and rotation for 3D objects. It is not clear whenever the work is rotation independent or if its recognition speed allows real-time. Kim and Fellner [KF04] use marked fingertips and infrared light to track hand motion and recognize gestures, they applied their work to 3D object manipulation and deformation. Malik *et al.* [ML04, MRB05] uses hand gestures over a tabletop as a two-hand input device for large displays from a distance. They consider fingertips and gesture recognition as two completely distinct processing steps.

Recently, Lawson and Duric [LD06] proposed the use of deficits of convexity to recognize hand gestures. In order to accomplish this, they analyze the gesture silhouette and the gesture convex hull, their recognition is both scale and rotation independent, like ours, but they can only recognize gestures that have non–convex silhouettes, thus limiting the set of identifiable gestures. From all the related work Lawson's is the most similar to ours. One advantage from our work, over Lawson's, is that, by using CALI's features we are able to recognize convex silhouettes.

Our work present most of the features required by any interactive recognition system, such as, real-time recognition, scale and rotation independency, low training requirements and a good recognition rate. Some of the works presented here have some of these qualities, but, to our knowledge, none have all of them.

3. Recognition Strategy

We present two techniques for hand pose identification that rely on CALI to extract geometric features from hand silhouettes and from polygons produced by connecting fingertips. CALI is a general, simple, fast, and robust recognition library, initially devised for recognition in calligraphic interfaces [FPJ02], recently generalized to classify geometric shapes for retrieval [FFJ05]. To classify shapes, CALI computes a set of geometric attributes from which derive features such as area and perimeter ratios from special polygons. CALI starts the calculation of geometric features by computing the *convex hull* (*ch*) of the shape. Then, it computes three special polygons from the *convex hull*: the *Largest Area Triangle* (*lt*), the *Largest Area Quadrilateral* (*lq*) inscribed in the *convex hull* and the *Smallest Area Enclosing Rectan-*

Feature	***Description***
A_{ch}	Area of the convex hull
A_{er}	Area of the (non-aligned) enclosing rectangle
A_{lq}	Area of the largest quadrilateral
A_{lt}	Area of the largest triangle
A_{st}	Area of the stroke
H_{er}	Height of the (non-aligned) enclosing rectangle
P_{ch}	Perimeter of the convex hull
P_{er}	Perimeter of the enclosing rectangle
P_{lq}	Perimeter of the largest quadrilateral
P_{lt}	Perimeter of the largest triangle
T_l	Total length, *i.e.* perimeter of original polygon
W_{er}	Width of the (non-aligned) enclosing rectangle

Table 1: *List of relevant geometrical features.*

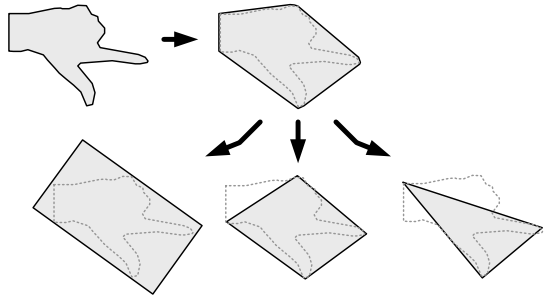

Figure 1: *Special polygons computed from an hand silhouette.*

gle (*er*). Figure 1 depicts an example of polygons extracted from a hand silhouette.

CALI combines the geometric features (listed in Table 1) computed from the shape's special polygons to produce a feature vector that describes the shape. Such feature vector is called descriptor. This descriptor allows the classification of shapes independently of their size, rotation or translation.

By using this descriptors, our scheme for recognizing hand gestures with CALI (see Figure 2) supplies a mechanism to recognize hand poses requiring only minimal training. This technique is composed by five different components. The image processing component performs some computer vision operations on captured frames to produce an image suitable for vectorization. The vectorization component is responsible for converting the resulting image to vector format, producing a scribble representing the hand pose. The CALI component extracts a set of more than thirty features from this scribble, depending on the recognition technique, and we select different subsets of these features to create a geometry feature vector (descriptor). This descriptor is computed using relationships between the relevant features listed in Table 1. The vectorization and image components differs between the two proposed techniques. It implements distinct operations when using the hand silhouette or the finger tips techniques. We describe these operations in the next sections.

Finally, the classification component is used during the training phase for storing descriptors in the database. For each hand pose identified in the training an average descriptor is obtained using the mean feature values for that pose. The matching component compares these descriptors with the one produced during the recognition phase, yielding a ordered set of suggested poses. The suggested set is created by performing a range search using Euclidean distances between the average descriptor and the descriptor of the pose to recognize. The resulting distance is used to sort results.

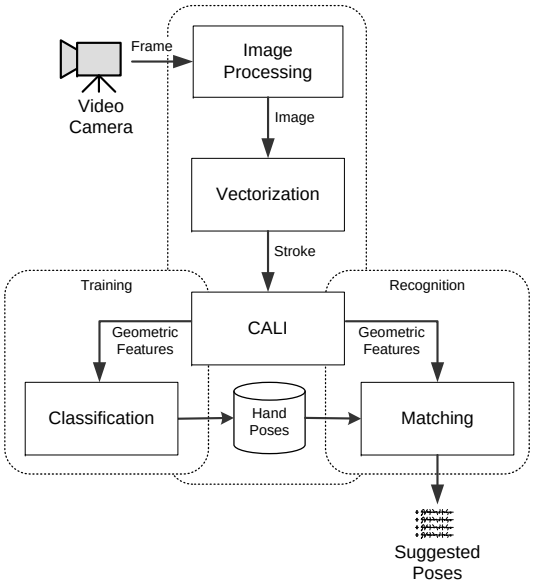

Figure 2: *System architecture*

3.1. Using the Hand Silhouette

As previously stated, we argue that the best interaction tools are our hands. In this technique, like in Lawson's technique, we choose to use the hand contour to represent the pose. However our technique drifts away from Lawsons in the way the hand silhouette is used. Lawson uses deficits of convexity while we use CALI features. In short, our strategy focus on obtaining the hand's shape, convert it to a stroke and use CALI as a recognition method for hand poses. Figure 3 illustrates the steps performed to generate strokes from hand images. Using a controlled environment allows us to threshold the image according to a certain value, creating a binary image. In the vectorization step the connected components are obtained and the image's biggest connected component is selected. Afterwards the biggest connected component is approximated into a polygonal curve. The curve is simplified using the Douglas-Peucker algorithm, whose resulting points are used to create a stroke. The resulting stroke is then fed to CALI that classifies the stroke producing a feature vector describing the hand pose.

Figure 3: *Hand silhouette recognition.*

$$\left[\frac{T_l}{P_{ch}} \quad \frac{P_{ch}^2}{A_{ch}} \quad \frac{A_{st}}{A_{ch}} \quad \frac{H_{er}}{W_{er}} \quad \frac{A_{ch}}{A_{er}} \quad \frac{A_{lt}}{A_{ch}} \quad \frac{A_{lt}}{A_{er}} \quad \frac{A_{lq}}{A_{ch}} \quad \frac{A_{lq}}{A_{er}} \quad \frac{A_{lt}}{A_{lq}} \quad \frac{P_{lt}}{P_{ch}} \quad \frac{P_{ch}}{P_{er}} \quad \frac{P_{lt}}{P_{er}} \quad \frac{P_{lq}}{P_{ch}} \quad \frac{P_{lq}}{P_{er}} \quad \frac{P_{lt}}{P_{lq}} \right]$$

Figure 4: *Hand Silhouette's geometric feature vector.*

>From early experiments, we verify that the features enumerated in Figure 4 and Table 1 yielded the best results. The chosen features include, mostly, area or perimeters relations. Nonetheless the $\frac{A_{st}}{A_{ch}}$ and $\frac{H_{er}}{W_{er}}$ are the two most defining features. The former differs open fingers shaped poses apart from closed fingers shaped poses, the latter allows poses with similar convex hull to be correctly recognized. The other fourteen features are, mainly, used to achieve a better recognition rate in borderline cases.

3.2. Using the FingerTips

Our other technique relies on fingertip detection to identify hand poses. By connecting fingertips we create a polygon that is, later on, classified using CALI. In the first step we use color segmentation to obtain each mark position. The image is converted from RGB to HSV, thus creating two gray scale images, one with Hue values and other with Saturation values. These images are compared to pre-defined filters to create a segmented image. In order to clean noise and smooth the results a set of morphological operation are applied. Each of the connected components included in the resulting image corresponds to a different mark. This information is retained as it is relevant to stroke creation. In the second step each component's centroid is obtained using the image moments and the five resulting points are orderly connected to create a stroke. The chosen order can be seen on Figure 6, all the fingers are connected starting from the bottom one to the top one in a clockwise direction. This stroke is then passed to CALI and it's features returned and used in hand pose recognition.

We select the most defining features produced by CALI to describe hand poses using only fingertips position. Figure 5 shows the geometric feature vector used in this technique. One of the most relevant aspects from our analysis shows that the $\frac{T_l}{P_{ch}}$ feature distinguish closed hand poses apart from open hand poses. Since experimental results showed its importance in correctly detecting hand poses we included it twice in the feature vector, increasing its relevance. We added five other features to recognize between each type of pose. For example, the $\frac{H_{er}}{W_{er}}$ identifies fist-like poses. It also helps distinguish open finger poses from closed finger poses.

$$\left[\frac{H_{er}}{W_{er}} \quad \frac{T_l}{P_{ch}} \quad \frac{P_{ch}^2}{A_{ch}} \quad \frac{A_{lq}}{A_{er}} \quad \frac{P_{lt}}{P_{ch}} \quad \frac{T_l}{P_{ch}} \quad \frac{P_{lt}}{P_{er}} \right]$$

Figure 5: *Fingertip's geometric feature vector.*

4. Experimental Results

In order to evaluate the hand pose recognition of both techniques, we performed experiments with users in a controlled environment. The setup of these experiments was composed by an inexpensive commodity web camera (Logitech® Webcam® Messenger™) mounted side-by-side with a regular white lamp that illuminates the hand as it makes gestures against a uniform background. Depending on the technique under evaluation, this background was black or white for the hand silhouette or fingertips, respectively. We used this setup to minimize the image pre-processing stage in these experiments, since we do not focus on hand or mark detection, but on pose recognition.

Our main objective on these experiments was to measure the recognition rate of both presented techniques. To that end, we extended the test set proposed by Lawson [LD06] with two additional hand poses. Thus, our test set comprises a total of eight hand poses, presented in Figure 7. Despite neither our techniques nor these experiments aims web browser control, for better understanding, we kept the pose names used by Lawson. From left to right, poses are named "Point", "Click", "Home", "Back", "Stop", "Scroll", "Four" and "Wait".

4.1. User Testing

These experiments involved ten users, who were briefly introduced to the experiment. Additionally, we explicitly ask users to be as comfortable as possible while posing their hand. After this, users participate in the experiment sessions for the hand silhouette technique evaluation first and then for the fingertips technique evaluation sessions.

For each technique, the experiment was divided into two distinct sessions. In the initial training session, all users performed twice the eight hand poses sequentially, in a predefined order. Our prototype extracted the features from each pose and classified it accordingly. From these training we collected a set of twenty entries for each pose, which made up a total of 160 entries for each technique.

Figure 6: *Fingertips recognition.*

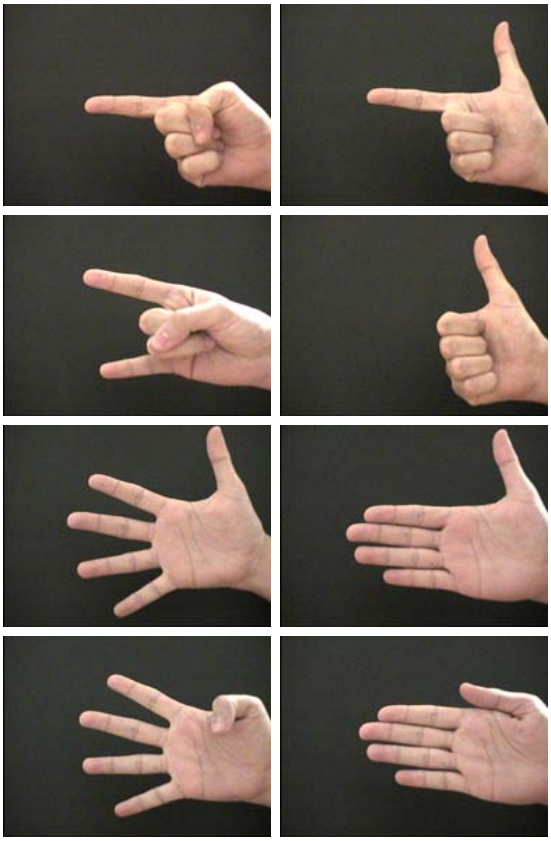

Figure 7: *Selected Hand Poses for User Tests*

After all users finished the training session, we considered the prototype trained. Then, users were called again for the second session. In the recognition session we ask users to carry out two distinct sets of eight poses to evaluate the recognition rate of our techniques. Each of these recognition sets include all the test poses ordered differently. From recognition sessions, we collected information on pose recognition for each technique.

4.2. Analysis

Although we had a limited number of user, the hand silhouette technique proved successful, presenting a 92.5% recognition rate. As one can verify in Table 2 all poses have a recognition rate higher than 80%. Our results show that 30% of the users generate 83% of the recognition failures, as depicted in Figure 8. On the other hand, 70% of the users achieved recognition rate around 100%. We believe that this difference is due to a lack of precision during user's hand pose execution. However, further testing will be needed to prove our beliefs.

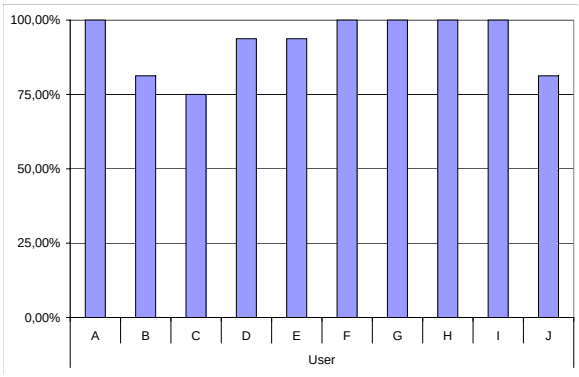

Figure 8: *Silhouette's recognition rate by user.*

Even so, some of the mismatches were expected: "Click" can be viewed as a sum of both "Back" and "Point" thus the dispersion between recognitions seems natural. The "Point" convex hull is very similar to "Back" convex hull, in a rotation-independent algorithm. This fact might justify the 15% confusion between "Point" and "Back".

Although some poses were wrongly recognized, looking at the overall results, we conclude that, in general, CALI is able to recognize hand poses with a good recognition rate, making this technique suitable to use in interactive systems.

The fingertips results were also positive, achieving a recognition rate of 93,15%. Like the previous technique, all poses have a recognition rate above 80%. However, unlike the hand silhouette technique, all users have, on average, a recognition rate of 90%. The low recognition rate of "Four" puzzled us, but, after verifying the training we found a couple of "Four" pose examples that did not quite follow the pose (the thumb finger was in the wrong position), therefore lowering the recognition rate. "Point" also had this problem, various users position the thumb finger differently from what we expected. As one of our testing rules was make users as

	Point	*Click*	*Home*	*Back*	*Stop*	*Scroll*	*Four*	*Wait*
Point	95%	–	–	5%	–	–	–	–
Click	15%	80%	–	5%	–	–	–	–
Home	–	–	100%	–	–	–	–	–
Back	10%	–	–	90%	–	–	–	–
Stop	–	–	–	–	95%	–	5%	–
Scroll	–	–	–	5%	–	95%	–	–
Four	–	–	–	–	15%	–	85%	–
Wait	–	–	–	–	–	–	–	100%

Table 2: *Silhouette's confusion table*

comfortable as possible we did not ask them to shift their thumb to the required position. As one can expect different thumb position results in different polygons.

When comparing the two techniques we come to the conclusion that CALI is suitable for hand pose recognition either using simple, non-intersecting, polygons (bare-hand) and more complex, intersecting, polygons (fingertips). Although, our results show a globally slightly lower recognition rate than Lawson's, our techniques identify a larger set of poses. As Lawson concludes, their recognition is limited to hand poses with deficits of convexity. The "Wait" pose, we included in our test set, should give Lawson a very low recognition rate given its almost null deficits of convexity. We believe that our results can be improved by performing tests with a bigger number of users, minimizing the effect of "bad" poses.

5. Conclusions and Future Work

We demonstrated the effectiveness of CALI as a hand pose recognition algorithm using different techniques. Experimental results showed us good prospects on using CALI on real world interaction environments. Even though the results were promising, some future work will focus on refining our matching algorithm. To that end, we intend to integrate in our approach k-nearest neighbor and neural networks methods. We used a simple search range algorithm in matching, when comparing features obtained with the training set. Probably the K-nearest neighbor algorithm or using simple neural networks with CALI features as input will provide better results.

The two techniques presented in this paper were tested in a controlled environment because in the current stage of our research we focus mainly on evaluating the viability of using CALI to recognize hand poses. However, using proper image pre-processing computer vision techniques we could easily extend the present system to work in a generic indoor (or even outdoor) environment.

Thus, in a near future, the proposed techniques will be tested in more realistic environments, integrated in interaction applications. The fingertip technique will be used in a virtual painting scenario where kids paint on a projected wall using their fingers. In this application gestures will be used to identify interaction modes (paint, erase, smudge, select,etc.). The hand silhouette technique will be used in a back-projection wall scenario. Here, the gestures will be captured using an infra-red camera, as suggested by Matsushita and Reikimoto [MR97], and will be used to interact with 3D modeling applications. This way we expect to validate the results presented here in an interactive environment with real users working on real applications.

	Point	*Click*	*Home*	*Back*	*Stop*	*Scroll*	*Four*	*Wait*
Point	80%	–	–	–	10%	5%	–	5%
Click	–	95%	–	–	–	–	–	5%
Home	–	–	100%	–	–	–	–	–
Back	–	–	–	100%	–	–	–	–
Stop	–	–	–	–	95%	5%	–	–
Scroll	–	–	–	–	5%	95%	–	–
Four	–	–	–	–	–	–	85%	15%
Wait	–	–	–	–	5%	–	–	95%

Table 3: *Fingertips confusion table*

Acknowledgments

This work was partially supported by the Portuguese Science Foundation Grant decorAR - POSC/EIA/59938/2004 and the European Commission FP6 grant IMPROVE (Improving Display and Rendering Technology for Virtual Environments) - IST-2003-004785.

References

[Bol80] BOLT R. A.: Put-that-there: Voice and gesture at the graphics interface. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1980), ACM Press, pp. 262–270.

[dlRG03] DE LA RIVIÈRE J.-B., GUITTON P.: Hand posture recognition in large display vr environments. In *Proceedings of the 5th International Gesture Workshop* (April 2003), Springer.

[dlRG05] DE LA RIVIÈRE J.-B., GUITTON P.: Image-based analysis for model-based tracking. In *Proceedings of Mirage* (March 2005).

[FFJ05] FONSECA M. J., FERREIRA A., JORGE J. A.: Generic Shape Classification for Retrieval. In *Proceedings of the 6th. IAPR International Workshop on Graphics Recognition(GREC 2005)* (August 2005).

[FPJ02] FONSECA M. J., PIMENTEL C., JORGE J. A.: CALI: An Online Scribble Recognizer for Calligraphic Interfaces. In *Proceedings of the 2002 AAAI Spring Symposium - Sketch Understanding* (Palo Alto, USA, Mar. 2002), pp. 51–58.

[KF04] KIM H., FELLNER D. W.: Interaction with hand gesture for a back-projection wall. In *CGI '04: Proceedings of the Computer Graphics International (CGI'04)* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 395–402.

[LD06] LAWSON E., DURIC Z.: Using deficits of convexity to recognize hand gestures from silhouettes. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)* (Setubal, Portugal, February 2006), Springer.

[LH98] LIEN C.-C., HUANG C.-L.: Model-based articulated hand motion tracking for gesture recognition. *Image and Vision Computing Volume 16*, Number 2 (February 1998), 121–134.

[ML04] MALIK S., LASZLO J.: Visual touchpad: a two-handed gestural input device. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces* (New York, NY, USA, 2004), ACM Press, pp. 289–296.

[MR97] MATSUSHITA N., REKIMOTO J.: Holowall: designing a finger, hand, body, and object sensitive wall. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology* (New York, NY, USA, 1997), ACM Press, pp. 209–210.

[MRB05] MALIK S., RANJAN A., BALAKRISHNAN R.: Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology* (New York, NY, USA, 2005), ACM Press, pp. 43–52.

[NR96] NOLKER C., RITTER H.: Illumination independent recognition of deitic arm postures. *Proc. 24th Annual conf. of the IEEE Indutrial Electronics Society* (1996).

[OSK02] OKA K., SATO Y., KOIKE H.: Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition* (Washington, DC, USA, 2002), IEEE Computer Society, p. 429.

[QZ96] QUEK F. K. H., ZHAO M.: Inductive learning in hand pose recognition. In *FG '96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)* (Washington, DC, USA, 1996), IEEE Computer Society, p. 78.

[SKK00] SATO Y., KOBAYASHI Y., KOIKE H.: Fast tracking of hands and fingertips in infrared images for augmented desk interface. In *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000* (Washington, DC, USA, 2000), IEEE Computer Society, p. 462.

[SSK01] SATO Y., SAITO M., KOIK H.: Real-time input of 3d pose and gestures of a user's hand and its applications for hci. In *VR '01: Proceedings of the Virtual Reality 2001 Conference (VR'01)* (Washington, DC, USA, 2001), IEEE Computer Society, p. 79.

[SZ93] STURMAN D. J., ZELTZER D.: A design method for whole-hand human-computer interaction. *ACM Trans. Inf. Syst. 11*, 3 (1993), 219–238.

[vHB01] VON HARDENBERG C., BÉRARD F.: Bare-hand human-computer interaction. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces* (New York, NY, USA, 2001), ACM Press, pp. 1–8.

[WB03] WU M., BALAKRISHNAN R.: 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* (New York, NY, USA, 2003), ACM Press, pp. 193–202.

[WH00] WU Y., HUANG T. S.: View-independent recognition of hand postures. *CVPR - IEEE Computer Society Conference on Computer Vision and Pattern Recognition Volume 2* (June 2000), 2088.