Gesture Recognition Using 3D Appearance and Motion Features

Guangqi Ye, Jason J. Corso, Gregory D. Hager
Computational Interaction and Robotics Laboratory
The Johns Hopkins University
grant@cs.jhu.edu

Abstract

We present a novel 3D gesture recognition scheme that combines the 3D appearance of the hand and the motion dynamics of the gesture to classify manipulative and controlling gestures. Our method does not directly track the hand. Instead, we take an object-centered approach that efficiently computes the 3D appearance using a region-based coarse stereo matching algorithm in a volume around the hand. The motion cue is captured via differentiating the appearance feature. An unsupervised learning scheme is carried out to capture the cluster structure of these feature-volumes. Then, the image sequence of a gesture is converted to a series of symbols that indicate the cluster identities of each image pair. Two schemes (forward HMMs and neural networks) are used to model the dynamics of the gestures. We implemented a real-time system and performed numerous gesture recognition experiments to analyze the performance with different combinations of the appearance and motion features. The system achieves recognition accuracy of over 96% using both the proposed appearance and the motion cues.

1 Introduction

Gestures have been one of the important interaction media in current human-computer interaction (HCI) environments [3, 4, 11, 12, 14, 16, 18, 21, 24, 25, 26]. Furthermore, for 3D virtual environments (VE) in which the user manipulates 3D objects, gestures are more appropriate and powerful than traditional interaction media, such as a mouse or a joystick. Vision-based gesture processing also provides more convenience and immersiveness than those based on mechanical devices.

Most reported gesture recognition work in the literature (see Section 1.1) relies heavily on visual tracking and template recognition algorithms. However general human motion tracking is well-known to be a complex and difficult problem [8, 17]. Additionally, while template matching may be suitable for static gestures, its ability to capture the spatio-temporal nature of dynamic gestures is in doubt. Alternatively, methods that attempt to capture the 3D information of the hand [11] have been proposed. However, it is well-known that, in general circumstances, the stereo problem is difficult to solve reliably and efficiently.

Human hands and arms are highly articulate and deformable objects and hand gestures normally consist of 3D global and local motion of the hands and the arms. Manipulative and interaction gestures [14] have a temporal nature that involve complex changes of hand configurations. The complex spatial properties and dynamics of such gestures render the problem too difficult for pure 2D (e.g. template matching) methods. Ideally we would capture the full 3D information of the hands to model the gestures [11]. However, the difficulty and computational complexity of visual 3D localization and robust tracking prompts us to question the necessity of doing so for gesture recognition.

To that end, we present a novel scheme to model and recognize 3D temporal gestures using 3D appearance and motion cues without tracking and explicit localization of the hands. Instead we follow the site-centered computation fashion of Visual Inteface Cues (VICs) paragigm [3, 24].

We propose that interaction gestures can be captured in a local neighborhood around the manipulated object based on the fact that the user only initiates manipulative gestures when his or her hands are close enough to the objects. The advantage of this scheme is that it is efficient and highly flexible. The dimension of the volume of the local neighborhood around the manipulated object can be adjusted conveniently according to the nature of the particular interaction environment and the applicable gestures. For example, in a desktop interaction environment where the interaction elements are represented as small icons on a flat panel and manipulative gestures are only initiated when the user's hand is near the surface of the panel, we only need to observe a small volume above the panel with the icon sitting at the center of the bottom. The height and diameter of the volume is also limited to be able to capture enough visual cues to carry out successful gesture recognition.

The remainder of this paper is structured as follows. In Section 2 we present a novel method to efficiently capture the 3D spatial information of the gesture without carrying out a full-scale disparity computation. We discuss how

to learn the cluster structure of the appearance and motion features via an unsupervised learning process in Section 3. Two ways to model the dynamics of the gestures, i.e., forward HMMs [10, 19] and multilayer neural networks [6], are also presented. In Section 4 we demonstrate our real-time system that implements the proposed method and present the results of gesture recognition.

1.1 Related Work

[22] gives a general overview of the state of the art in gesture analysis for vision-based human computer interaction. Robust hand localization and tracking, modeling the constraints of hand motion and recognizing temporal gesture patterns are among the most difficult and active research areas. Compared to other technique, such as neural network, rule-based method [14], HMM [23, 24] and its extension [2] is a popular scheme to model temporal gestures.

Many HCI systems [12, 14, 16, 21, 22] have been reported that enable the user to use gestures as a controlling or communicative media to manipulate interaction objects. The hand or fingertips are detected based on such cues as visual appearance, shape and even human body temperature via infrared cameras, etc. A variety of algorithms have been applied to track the hand [22], such as the Kalman filter and particle filter [5].

With a model-based approach [1, 13], it is possible to capture the gesture in higher dimensionality than 2D. In [1] the 3D hand model is represented as a set of synthetic images of the hand with different configurations of the fingers under different viewpoints. Image-to-model matching is carried out using Chamfer distance computation. One of the difficulties of this approach is to construct a good 3D model of the hand that can deal with variance between different users. Furthermore, efficient algorithms are necessary to handle the matching between models and input images. Another approach to capture 3D data is to use special cameras [11], such as 3D cameras or other range sensors. However, the hardware requirement limits its application to general HCI systems.

2 Capturing 3D Features of Manipulative Gestures

Manipulative and controlling gestures have a temporal 3D nature involving the interaction between human hands and other objects. Example subjects include the tools and toys in a VE, interaction elements in a HCI interface, etc. One of the most difficult problems in visual modeling of temporl gestures is data collection. Here we propose an efficient scheme to capture 3D gesture appearance and motion in an object-centered fashion. We assume that the manipulative gestures can be captured in a local space around the manipulated object. The assumption is valid [24] in many HCI scenarios, such as a WIMP-style interface [20].

2.1 3D Gesture Volume

Given a pair of rectified stereo images of the scene, a disparity map can be computed using a standard correspondence search algorithm. Since we only care about the local neighborhood around the object, we can constrain the stereo search to a limited 3D space around the object. This brings about two advantages: (1) we only care about the small patch of the image centered at the object, and (2) we only need to search through a small number of disparities (depths), which is a volume around the depth of the object. To speed up the disparity computation, we further simplify the process using block matching technique.

Formally, let I_l and I_r be a pair of rectified images of the scene. We split the images into tiles of equal size of $w \times h$. Here w and h refer to the width and height of the tile, respectively. Suppose we only consider a local area of size of $m \times n$ patches, starting at patch (x_0, y_0). Given a discrete parallax search range of $[0, (p-1) \times w]$, we can characterize the scene using a $m \times n \times p$ volume V as:

$$V_{x,y,z} = Match(I_{l(x_0+x,y_0+y)}, I_{r(x_0+x+z,y_0+y)}) \quad (1)$$
$$x \in [0, m-1], y \in [0, n-1], z \in [0, p-1]$$

Note that in the previous equation, the image index indicates a patch of the image, not a particular pixel.

We convert the color images into hue images to reduce the impact of changes in lighting intensity because hue is a good color-invariant model [9]. Furthermore, we perform a comprehensive color normalization process [7] on each image to overcome the variance of illumination and lighting geometry among different interaction sessions. These techniques ensure the relative stability of the appearance feature under different imaging conditions. Different block matching algorithms can be applied, such as sum of squared difference and sum of absolute differences.

Following this scheme, we can extract the features of the image as a very simple vector with the size of $m \times n \times p$. The typical size of the extracted appearance vector is from 125 to 1000. In contrast, the size of the original image is 640×480 and the size of the local image around a typical object in our experiments is approximately 150×150. Thus, this feature extraction scheme reduces the size of the the input data greatly.

Figure 1 shows examples of the stereo image pair and the extracted 3D feature of the scene. It can be seen that the extracted feature volume characterizes the different configuration of the user's hand with respect to the target interaction subject.

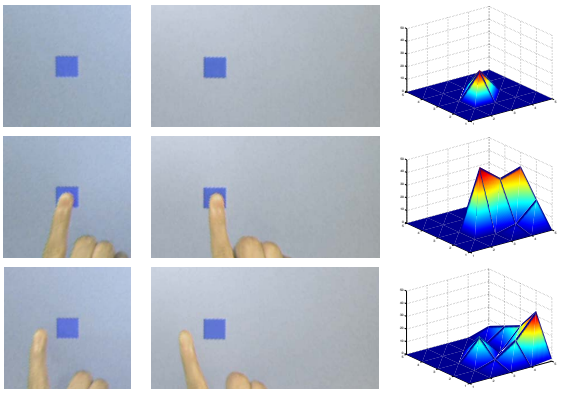

Figure 1: Examples of the image pair and extracted appearance feature. Left and middle column display left images and right images of the scene, respectively. Right column shows the bottom layer of the feature volume (i.e., $V_{x,y,z}$ with $z = 0$)

2.2 Motion by Differencing

Since we represent the 3D appearance of the gesture images using feature vectors, one simple way to capture the motion information of the gestures is to compute the the displacement in this feature space. In our real-time experiment, the change between consecutive frames is normally very small because of the high frame rate. Thus we compute the difference between the appearance feature of the current frame and that of several frames before.

$$Motion_i = V_i - V_{i-k}, \;\; i = k+1, ..., M \qquad (2)$$

One way to combine the appearance feature and the motion feature is to concatenate the two vectors to form a larger vector. This new vector contains both the static and temporal information of the gesture.

Given an image sequence that contains a particular manipulative gesture, we convert the sequence into a series of vectors, or points in the gesture volume or gesture motion space. Thus, the gesture can be conceptualized as a directed path connecting these points in the appropriate order. Intuitively we can model the gesture by learning the parameters of such a path. However, this appearance or motion space is still a relatively high-dimension space, making the learning and recognition difficult to handle.

3 Learning the Gesture Structure

3.1 Unsupervised Learning of the Cluster Structures of 3D Features

Given a training set of image sequences containing a group of valid gestures, we build a new set of appearance or motion vectors following the technique described in the previous section. It can be expected that there will be much redundancy of information because the training set contains repeatable gestures and there are only a limited number of gestures in the set. Actually in our current experiment containing six manipulative gestures, using a PCA technique on the 125-dimension appearance feature that we extracted from over 600 gesture sequences, we can achieve an average reconstruction error of less than 5% using only 8 eigenvectors. Therefore, we are able to characterize the appearance or motion feature of the gestures using data of much lower dimensionality without losing the capability to discriminate between them.

One of the popular ways to model temporal signals is to learn a statistical model [6]. However, the size of training data needed for statistical learning normally increases exponentially with the dimensionality of input features. This curse of dimensionality is one of the reasons that visual modeling of gestures is difficult. Thus we propose to reduce the dimensionality of the 3D feature by learning its cluster configuration.

Since the raw 3D feature data is of high dimensionality and difficult to cluster via intuitive ways, we propose a unsupervised method to learn the cluster structure. Basically we implement a K-means algorithm to learn the centroid of each of the clusters of the feature sets. The choice of the number of clusters is empirical. We then represent each vector using a symbol that indicates its cluster identity. In our data sets comprised of 6 gestures, we use less than 20 clusters to describe each of the feature set, including the appearance set, the motion set and the appearance-motion combination set.

3.2 Gesture Modeling Using HMM

We use typical forward HMMs to model the dynamics of the temporal gestures. The input to the HMMs is the gesture sequence represented as a series of symbols with each symbol indicating the cluster identity of current frame. The basic idea is to construct a HMM for each gesture and learn the parameters of the HMM from the training sequences that belong to this gesture using the Baum-Welch algorithm [10, 15]. The probability that each HMM generates the given sequence is the criterion of recognition. The gesture sequence is recognized as the class with the highest probability. Rejection of invalid gestures is based on the thresholding of the best probability. If the highest probability that a sequence achieves on all HMMs is lower than a threshold, the sequence will be rejected. This threshold is chosen to be smaller than the lowest probability that each HMMs generates the sequences that belong to that class in the training set.

In our experiment, we use a 6-state forward HMM to

model each of the six manipulative gestures. Figure 2 shows the topology of the HMMs.

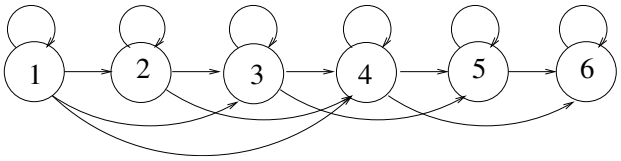

Figure 2: HMM structure for the interaction gestures

The choice of the number of the states in the forward HMM is based on the intuitive analysis of the temporal properties of the gestures to be modeled. In our current experiment, each of the gestures can be decomposed of less than 6 distinct stages. For example, if we use 3 spatial layers to represent the vicinity of an manipulated object, the gesture of swiping an icon to the left can be viewed as such a configuration sequence of the hand: (1) entering the outer layer of the vicinity of the icon, (2) entering the inner layer (3) touching the icon to select it and (4) swiping the icon by moving the finger to the left side of the icon. Ideally, each of the distinct stages can be modeled by a certain state of the forward HMM. The parameter sets of the trained HMMs verify our expectation, in which the observation probability of each symbols of a gesture is dominantly high in one of the states and very small in other states. Generally speaking, a dynamic process with n stages can be modeled using an n-state forward HMM with similar topology. For example, in [19], four-state HMMs are used to recognize American Sign Language.

3.3 Gesture Modeling Using A Multilayer Neural Network

Another way to learn the gestures is to use multilayer neural networks. The input to the neural network is the whole gesture sequence, which is now a sequence of symbols. The output is the identity of the gesture. To meet the requirement of the neural network, we need to fix the length of each input sequence. We align each sequence to a fixed length by carrying out sub-sampling on those sequences that are longer than the predefined length and interpolation on those that are shorter. The parameters of the network are also learned from training data using the standard backpropagation algorithm [6].

In our current system, the neural network consists of 3 layers, i.e., the input and output layer and the hidden layer. The number of nodes in the hidden layer is chosen to be 50.

4 Experimental Results

4.1 Experimental Setup

We use 4D touchpad as our experimental platform. We use two color cameras to observe the interaction desktop which is presented as a flat panel on which the interaction elements are rendered. The system is calibrated based on a homography technique so that, for the rendered icons on the panel, a correspondence of the dimension and position of the icon is established between the rendered image and the images captured from the cameras. The user interacts with the objects on the panel using manipulative and controlling gestures. Figure 3 shows the configuration of our experiment platform.

Figure 3: The 4D Touchpad HCI platform.

In our current experiments, we collect gesture sequences consisting of 6 interactive gestures, i.e., pushing a button, twisting a dial clockwise, twisting a dial anti-clockwise, toggle a switch, swiping an icon to the left and swiping an icon to the right.

We implement the system on a PC with dual Pentium III processors. The system achieves real-time speed; the processing is limited by the cameras (30Hz). The system processes the continuous video in the following fashion. For each captured image pair, the appropriate appearance and/or motion features are extracted and the corresponding cluster identity of current features is computed based on trained cluster centroids. We begin the recording of a sequence when the cluster identity represents a valid hand configuration instead of a scene where the hand has not entered the vicinity of the target icon, which we call an "empty" configuration. We carry out the recognition of current sequence and notify the user when a valid gesture is recognized. The recording of current sequence is then terminated and the system enters a new cycle. Another case for ending current sequence is that the system continuously observes empty configuration for several frames.

4.2 Gesture Recognition Results

To perform training of the HMMs and the neural network, we record over 100 gesture sequences for each of the 6 ges-

tures. A separate test set contains over 70 sequence of each gesture.

We carry out the training and testing on several feature sets. These different sets are characterized by the dimensionality of our 3D gesture volume described in Section 2 and different combination of the appearance and motion cues.

1. **Appearance Only** ($5 * 5 * 5 = 125$-D)
   In this set, we only use the appearance feature with the dimensionality as $5 * 5 * 5 = 125$. We carry out the K-means on the training set of these features using 8 cluster centers.

2. **Appearance Only** ($10 * 10 * 10 = 1000$-D)
   Similar to the first set. But the dimensionality of the appearance feature is $10 * 10 * 10 = 1000$.

3. **Motion Only** ($10 * 10 * 10 = 1000$-D)
   We compute the motion feature by taking the difference between two 1000-D appearance vectors. We use 15 cluster centers to represent the cluster structure.

4. **Concatenation of Appearance and Motion**
   In this set, we concatenate the 125-D appearance feature with the 1000-D motion vector to form a 1125-D vector. We carry out the K-means on this appearance-motion feature set using 18 clusters.

5. **Combination of Appearance** (125-D) **and Motion**
   We carry out K-means on the 125-D appearance feature and 1000-D motion features separately. Then each frame is represented as a 2-D discrete vector containing both the appearance cluster identity and motion cluster character.

6. **Combination of Appearance** (1000-D) **and Motion**
   Similar to the previous setting except that we use the 1000-D appearance feature.

We perform the training and testing on these sets for the HMM models and the neural network. For the neural network, we align each gesture sequence to the fixed length of 20. For the HMM models, we also carry out comparison experiments between using the same aligned sequences as the neural network and applying the raw unaligned sequence. Table 1 shows the gesture recognition results for all the feature sets and both gesture models. For each model we report both the recognition accuracy on the training set and that on the test set.

The results show that aligning the sequences to the same length improves the recognition accuracy. It can also be seen that the motion feature alone seems to perform slightly worse than those with appearance cues. However, combining appearance features with the motion features achieves the best recognition accuracy for our current gesture set.

Table 1: Gesture recognition results for different feature spaces

Set	HMM	NN	Unaligned
1	99.5 99.5	100.0 98.8	99.4 99.4
2	99.5 100.0	98.4 94.4	98.4 98.0
3	98.4 98.1	97.7 86.3	97.9 98.8
4	98.9 99.0	98.9 87.7	96.7 96.1
5	100.0 100.0	100.0 96.6	98.2 97.3
6	99.8 99.8	99.8 97.1	99.2 99.5

Another interesting comparison between the HMM model and neural network shows that our multilayer neural network tends to over-train on the feature sets. The neural network model achieves equivalent or higher accuracy on the training set as the HMM model, but perform worse on the test set. During the training of the HMMs, the Baum-Welch algorithm runs for less than 5 iterations before the overall system entropy reaches a local minimum. While during the neural network training process, the backpropagation algorithm typically runs for over 1000 iterations. We stop the the procedure when the decrease of the output error between consecutive runs is lower than a threshold, which is typically a very small number such as 0.00001.

Alternatively, one could stop the backpropagation algorithm interactively by measuring the performance on a validation set after each iteration and halting the training process if the classification on this validation set degenerates. However, we choose a fixed threshold to preserve the generality of the method and keep the training process automatic.

We also compare the gesture modeling using HMM based on the raw sequences and those using collapsed sequences. Each raw sequence containing a gesture is packed in such a way that we only record a symbol if it is different to its previous one. In essence, we only record the order of the appearance of each feature, excluding the duration in the original temporal sequence. This is similar to the rule-based and state-based gesture modeling [2, 22]. Table 2 shows the gesture recognition results based on the datasets of collapsed sequences.

Table 2: Gesture recognition results for collapsed sequences

Feature Sets	Training	Test
Appearance(125-D)	89.3%	88.8%
Appearance(1000-D)	88.3%	86.1%
Motion(1000-D)	98.4%	96.6%
Concatenation	90.8%	89.0%
Combination 1	94.2%	96.8%
Combination 2	99.8%	98.8%

Compared to the results using raw sequences, the gesture recognition using collapsed sequences perform slightly worse. Still, for the combination of the appearance and the motion features, this scheme of gesture modeling based only on key frames achieves very good recognition performance.

5 Conclusions

In this paper we present a novel real-time 3D gesture recognition system that combines the 3D appearance of the hand and the motion dynamics of the gesture to classify manipulative and controlling gestures. Instead of tracking the user's hand, we capture the 3D appearance of the local volume around the manipulation subject. Motion is computed as the difference of the appearance feature between frames. We reduce the dimensionally of the 3D feature by employing unsupervised learning. We implement a real-time system based on the 4D touchpad platform and test the system using two different approaches to model the temporal gestures, i.e., forward HMMs and multilayer neural networks. By combining the appearance and motion cues, both HMM models and the neural network achieves an recognition accuracy of over 96%. The proposed scheme is a flexible and efficient way to capture the 3D visual cues in a local neighborhood around the object. The experiment results show that these local appearance and motion features capture the necessary visual cues to recognize different manipulative gestures.

Our future research will address more complex gestures, such as those gestures involving two hands. In our current experiment setup, the manipulated objects lie on a 2D plane. We intend to perform experiments on objects with 3D volume, such as a cube on a desktop. We also plan to investigate other ways to model the gesture dynamics, such as HMMs that achieve mimimal classification errors.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0112882. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Vassilis Athitsos and Stan Sclaroff. Estimating 3D Hand Pose from a Cluttered Image. In *Computer Vision and Pattern Recognition*, volume 2, pages 432–439, 2003.

[2] Aaron Bobick and Andrew Wilson. A State-based Approach to the Representation and Recognition of Gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1325–1337, 1997.

[3] Jason J. Corso, Darius Burschka, and Gregory D. Hager. The 4DT: Unencumbered HCI With VICs. In *Proceedings of CVPRHCI*, 2003.

[4] James Davis and Aaron Bobick. The Representation and Recognition of Action Using Temporal Templates. In *Computer Vision and Pattern Recognition*, pages 928–934, 1997.

[5] Jonathan Deutscher, Andrew Blake, and Ian Reid. Articulated Body Motion Capture by Annealed Particle Filtering. *Computer Vision and Pattern Recognition*, 2, 2000.

[6] Richard Duda, Peter Hart, and David Stork. *Pattern Classification*. John Wiley and Sons, Inc, 2001.

[7] Graham D. Finlayson, James Crowley, and Bernt Schiele. Comprehensive Colour Image Normalization. In *Proceedings of the European Conference on Computer Vision*, number 1, pages 475–490, 1998.

[8] D. Gavrila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.

[9] Theo Gevers. Color based object recognition. *Pattern Recognition*, 32(3):453–464, 1999.

[10] Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1999.

[11] S. Malassiotis, N. Aifanti, and M. Strintzis. A Gesture Recognition System Using 3D Data. In *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmisssion*, pages 190–193, 2002.

[12] Kenji Oka, Yoichi Sato, and Hideki Koike. Real-Time Fingertip Tracking and Gesture Recognition. *IEEE Computer Graphics and Applications*, 22(6):64–71, 2002.

[13] Vasu Parameswaran and Rama Chellappa. View Invariants for Human Action Recognition. In *Computer Vision and Pattern Recognition*, volume 2, pages 613–619, 2003.

[14] F. Quek. Unencumbered Gesture Interaction. *IEEE Multimedia*, 3(3):36–47, 1996.

[15] Lawrence Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[16] Aditya Ramamoorthy, Namrata Vaswani, Santanu Chaudhury, and Subhashi Banerjee. Recognition of Dynamic Hand Gestures. *Pattern Recognition*, 36:2069–2081, 2003.

[17] J.M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: An application to human hand tracking. In *Computer Vision – ECCV '94*, volume B, pages 35–46, 1994.

[18] Christopher Rwen, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–784, 1997.

[19] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. Technical Report TR-375, M.I.T. Media Laboratory, 1996.

[20] Andries van Dam. Post-wimp user interfaces. *Communications Of The ACM*, 40(2):63–67, 1997.

[21] Christian von Hardenberg and Francois Berard. Bare-Hand Human-Computer Interaction. In *Workshop on Perceptive User Interfaces*, 2001.

[22] Ying Wu and Thomas S. Huang. Hand Modeling, Analysis, and Recognition. *IEEE Signal Processing Magazine*, 18(3):51–60, 2001.

[23] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing Human Actions in Time-sequential Images Using Hidden Markov Model. In *Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[24] Guangqi Ye, Jason J. Corso, Darius Burschka, and Gregory D. Hager. Vics: A modular vision-based hci framework. In *Proceedings of 3rd International Conference on Computer Vision Systems(ICVS 2003)*, pages 257–267, 2003.

[25] Guangqi Ye and Gregory D. Hager. Appearance-based visual interaction. Technical report, 2002. CIRL Lab Technical Report, Department of Computer Science, The Johns Hopkins Univeristy.

[26] Zhengyou Zhang, Ying Wu, Ying Shan, and Steven Shafer. Visual Panel: Virtual Mouse Keyboard and 3D Controller with an Ordinary Piece of Paper. In *Workshop on Perceptive User Interfaces*, 2001.