Appeared in Virtual Reality, 8(4):242-252, Sept. 2005.
(The original publication is available at www.springerlink.com)

Analysis of Composite Gestures with a Coherent Probabilistic Graphical Model

Jason J. Corso, Guangqi Ye, Gregory D. Hager

Computational Interaction and Robotics Lab
The Johns Hopkins University
Baltimore, MD 21218
{jcorso|grant|hager}@cs.jhu.edu

Received: / Revised version:

Abstract Traditionally, gesture-based interaction in virtual environments is composed of either static, posture-based gesture primitives or temporally analyzed dynamic primitives. However, it would be ideal to incorporate both static and dynamic gestures to fully utilize the potential of gesture-based interaction. To that end, we propose a probabilistic framework that incorporates both static and dynamic gesture primitives. We call these primitives Gesture Words (GWords). Using a probabilistic graphical model (PGM), we integrate these heterogeneous GWords and a high-level language model in a coherent fashion. Composite gestures are represented as stochastic paths through the PGM. A gesture is analyzed by finding the path that maximizes the likelihood on the PGM with respect to the video sequence. To facilitate online computation, we propose a greedy algorithm for performing inference on the PGM. The parameters of the PGM can be learned via three different methods: supervised, unsupervised, and hybrid. We implemented the PGM model for a gesture set of 10 GWords with 6 composite gestures. The experimental results show that the PGM can accurately recognize composite gestures.

Key words human computer interaction – gesture recognition – hand postures – vision-based interaction – probabilistic graphical model

1 Introduction

Recently, the development in Virtual Reality (VR) technologies [5] has taken us to 3-D virtual worlds and prompted us to develop new human-computer interaction (HCI) techniques. Many of the current VR applications employ such traditional HCI media as joysticks, wands, or other tracking technologies (magnetic trackers [3], optical trackers [1], etc.). However, many of these techniques encumber the user with hardware that can potentially reduce the realism (and effect) of the simulation. These limitations have presented us with the challenge to design and implement new HCI techniques that are natural and intuitive. Spoken language [18], haptics [16, 30, 44, 45, 46] and vision [11, 26, 40] have been popular choices to replace traditional interaction media. Computer vision holds great promise: vision-based interfaces would allow unencumbered, large-scale spatial motion. Furthermore, rich visual information provides strong cues to infer the motion and configuration of the human hands and arms.

Many gesture-based visual interfaces have been developed [13, 24, 38, 36, 43, 47]. According to the nature of the gestures in the vocabulary, the gestures in existing interfaces can be classified into two categories: static hand postures and dynamic gestures. Static postures [4, 20, 23, 28, 35, 39] model the gesture as a single key frame, thus discarding any dynamic characteristics. For example, in recent research on American Sign Language (ASL) [34, 48], static hand configuration is the only cue used to recognize a subset of the ASL consisting of alphabetical letters and numerical digits. The advantage of this approach is the efficiency of recognizing those gestures that display explicit static spatial configuration. However, it has an inherent shortcoming in handling dynamic gestures whose temporal patterns play a more important role than their static spatial arrangement.

Dynamic gestures contain both spatial and temporal characteristics, thus providing more challenges for modeling. Many models have been proposed to characterize the temporal structure of dynamic gestures: including temporal template matching [7, 21, 25, 33], rule-based and state-based approaches [8, 28], hidden Markov models (HMM) [24, 29, 34, 41, 42] and its variations [9, 24, 37], and Bayesian networks [32]. These models combine spatial and temporal cues to infer gestures that span a stochastic trajectory in a high-dimensional spatio-temporal space.

Most current systems model dynamic gestures qualitatively. That is, they represent the identity of the gesture, but they do not incorporate any quantitative, parametric information about the geometry or dynamics of the motion involved. To overcome this limitation, a parametric HMM (PHMM) [37] has been proposed. The PHMM includes a global parameter that carries an extra quantitative representation of each gesture. This parameter is included as an additional variable in the output probabilities of each state of the traditional HMM.

It seems clear that to fully harness the representative power of human gestures, static postures and non-parametric and parametric, dynamic gestures must be integrated into a single coherent gesture model. For example, visual modeling of ASL is still limited by the lack of capabilities to handle the composite nature of gestures. To that end, we present a novel framework that integrates static postures, unparameterized dynamic gestures and dynamic parameterized gestures into a coherent model.

In this framework, a graphical model is used to model the semantics and temporal patterns of different parts of a complex gesture; essentially, the graphical model is a high-level language (or behavioral) model. In the model, each stage of the gesture is represented as a basic language unit, which we call a Gesture Word (GWord). A GWord can be modeled as either a static posture, unparameterized dynamic gesture or a parameterized gesture. A composite gesture is composed of one or more GWords with semantic constraints. These constraints are represented in the graphical model, with nodes denoting GWords and edges describing the temporal and linguistic relationship between GWords. The parameters of the model can be learned based on heuristics or via a probabilistic framework based on recorded training data. Online gesture recognition is carried out via greedy inference on the graphical model. Here, online means that the algorithm does not have access to future video frames.

Our proposed framework is related to work in the field of activity modeling. Bregler [10] abstracted human activity in a three-layered model. In the data-driven approach, regions of coherent motion are used as low-level features. Dynamic models capture simple movements at the mid-level, and HMMs model the high-level complex actions. Pentland and Liu [27] proposed Markov Dynamic Models which couple multiple linear dynamic models (e.g. Kalman filters) with a high-level Markov model. Ivanov and Bobick [17] proposed a probabilistic syntactive approach to activity modeling. In their two-layered model, a discrete symbol stream is generated from continuous low-level detectors and then parsed with a context-free grammar. Galata et al. [15] proposed an approach to learn the size of structure of the stochastic model for high-level activity recognition.

The main contribution of this work is to investigate a high-level language model to integrate the three different low-level gesture forms in a coherent manner. We extend the state-of-the-art in gesture modeling by relaxing the assumption that the low-level gesture primitives have a homogeneous form: e.g. all can be modeled with a HMM.

2 Modeling Composite Gestures

Probabilistic graphical models (PGM) are a tool for modeling the spatial and temporal characteristics of dynamic processes. For example, HMMs and Bayesian networks are commonly used to model such dynamic phenomena as speech and activity. PGMs provide a mathematically sound framework for learning and probabilistic inference.

However, most previous work in gesture and activity recognition assume a consistent model for all low-level processes (GWords). We propose to use PGMs to integrate multiple, heterogeneous low-level gesture processes into a high-level composite gesture. Intuitively, we combine multiple GWords to form a *Gesture Sentence* that corresponds to a complete interaction task. For example, grasping a virtual object → moving it → dropping the object (Figure 1).

In the remainder of this section, we define notation in Section 2.1 and present our construction of the composite gestures using PGMs in Section 2.2. In Section 2.3 we discuss different types of GWords. We formulate the learning of the PGM in Section 2.4. The gesture inference is discussed in Section 2.5.

2.1 Definitions

Let the image $\mathbf{I} \doteq \{\mathcal{I}, I, t\}$ be a finite set of pixel locations $\mathcal{I}$ (points in $\mathbb{R}^2$) together with a map $I : \mathcal{I} \rightarrow \mathcal{X}$, where $\mathcal{X}$ is some arbitrary value space, and t is

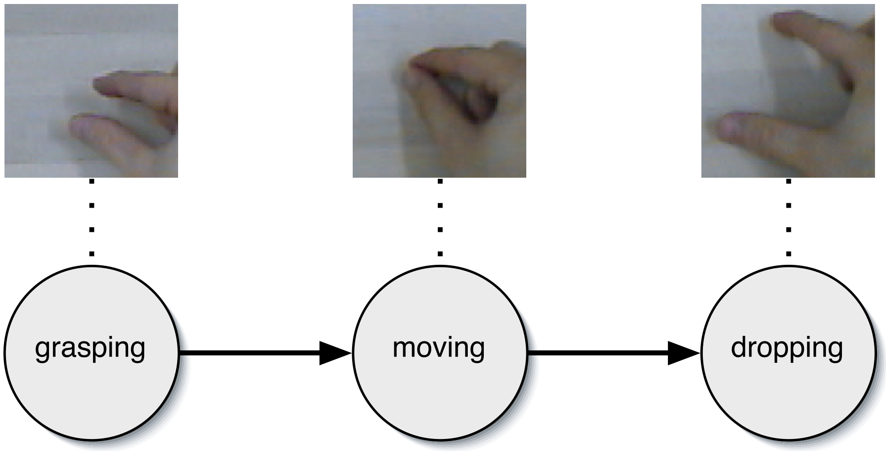


Figure 1 Composite gesture example with a corresponding graphical model.

a time parameter. Define $\mathcal{S} = \{\mathbf{I}_1 \ldots \mathbf{I}_m\}$ to be a sequence of images with length $m \geq 1$. Let $G = \{\mathcal{V}, \mathcal{E}\}$ be a directed graph representing the gesture language model. Each node $v \in \mathcal{V}$ in the graph corresponds to a GWord which belongs to a vocabulary $\mathcal{V}$ of size $|\mathcal{V}|$. Associated with each node v is a probability function $P(\mathcal{S}|v)$, which measures the observation likelihood of $\mathcal{S}$ for a given GWord v. Each edge $e \in \mathcal{E}$ is a probability function $P(v_j|v_i)$, where $v_j, v_i \in V$. Intuitively, the edge models the temporal relationship between successive gesture units in the composite gesture.

2.2 The Gesture Language

We use a *bigram model* to capture the dynamic nature of the gesture language. The bigram model represents the linguistic relationship between pairs of GWords. Formally, given a vocabulary $\mathcal{V}$, define a GWord sequence $\mathcal{W} = \{s, v_1, \ldots, v_k, t\}$ where $v_i \in \mathcal{V}$ and s, t are two special nodes (*dummy gestures*) that act as the graph source and sink. Thus, a gesture is a path through the PGM starting at the source node and ending at the sink node. In Figure 2, we give an example PGM that can model 6 gestures. For example, the path $s \rightarrow 1 \rightarrow 3 \rightarrow 6 \rightarrow t$ is a candidate Gesture Sentence.

We embed the bigram language model into the PGM by associating nodes with individual GWords and assigning transition probabilities from the bigram model. For convenience, let $P(v_1) \doteq P(v_1|s)$, which can be considered as the priors of a GWord. Then the probability of observing the sequence in the bigram model is

$$P(\mathcal{W}) \doteq P(s, v_1, \ldots, v_k, t) = P(v_1) \prod_{i=2}^{k} P(v_i|v_{i-1}) \tag{1}$$

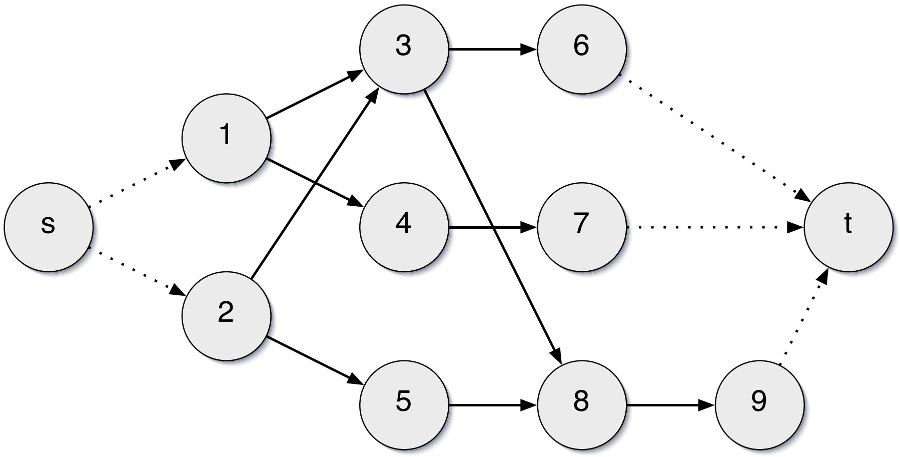


Figure 2 Example PGM used to represent the gesture language model. Each path beginning at node s and ending at node t is a valid Gesture Sentence.

As defined in Section 2.1, each node of the PGM models a specific GWord with its corresponding observation likelihood. Given an image sequence $\mathcal{S}$ we can construct a candidate segmentation (Section 2.5) that splits the sequence into p subsequences $\{\mathcal{S}_1 \ldots \mathcal{S}_p\}$. We correspond each of the subsequences to a GWord thus creating a Gesture Sentence $\mathcal{W}$. Assuming conditional independence of the subsequences given the segmentation and the observation likelihood of a subsequence only depends the corresponding GWord, the observation likelihood of the sequence is

$$P(\mathcal{S}|\mathcal{W}) = \prod_{i=1}^{p} P(\mathcal{S}_i|v_i) \tag{2}$$

Then, the overall probability of observing the Gesture Sentence is

$$\begin{aligned} P(\mathcal{W}|\mathcal{S}) &\propto P(\mathcal{W}) \cdot P(\mathcal{S}|\mathcal{W}) \\ &= P(v_1) \prod_{i=2}^{p} P(v_i|v_{i-1}) \cdot \prod_{i=1}^{p} P(\mathcal{S}_i|v_i) \end{aligned} \tag{3}$$

with the special source and sink node probabilities defined as $P(s) = 1$, $P(t|v \in \mathcal{V}) = 1$, $P(v \in \mathcal{V}|s) = P(v)$.

2.3 The Three Low-level Gesture Processes

As discussed in Section 1, there are three main approaches to modeling gestures in current virtual reality systems: static postures, non-parametric dynamic, or parametric dynamic. In the PGM presented earlier, each node corresponds to one of these three types of gesture processes.

2.3.1 Static Posture Static postures are based on the recognition of single discriminative frames of video. Hence, static postures simplify gesture processing by discarding all temporal information. For example, in the current literature, most alphanumeric symbols in ASL are represented as static postures [48]. Commonly used approaches to model the postures include appearance-based templates, shape-based models, and 3D model-based methods.

2.3.2 Non-parametric Dynamic Non-parametric dynamic gestures capture temporal processes that carry only qualitative information; no quantitative information (e.g. length of hand-wave) is present. Hence, these gestures are potentially more discriminative than static postures because of the additional temporal dimension. For example, the 'j' and 'z' letters in the ASL have a temporal signature; i.e. the spatial trajectory of the finger over time is used to discriminate between the 'i' and the 'j'. Hidden Markov models [29, 34, 41, 42] and motion history images [7] are common methods used to model non-parametric dynamic gestures.

2.3.3 Parametric Dynamic Parametric dynamic gestures are the most complex among the three types because they not only incorporate a temporal dimension but also encode a set of quantitative parameters. For example, in explaining the height of a person using an outstretched hand, the distance between the ground and the hand gives a height estimate. Parametric hidden Markov models [37] have been proposed to model a single spatial variable. However, most of the techniques are based on visual tracking.

The parametric dynamic gestures bring an added degree of difficulty to the recognition process because they can have too high a degree of temporal variability to be captured by a standard model like an HMM. For example, Figure 1 shows a composite gesture for grabbing, moving, and dropping a virtual object. In general, the moving gesture will appear quite arbitrary because the user has the freedom to navigate the entire workspace and also pause for variable amounts of time before dropping the object.

2.4 Learning the PGM

In this paper, we assume that the learning and the implementation of the individual low-level gesture units are handled separately (in Section 3 we discuss our implementations) and the observation probabilities of these units are normalized on the same scale. Here, we address the problem of learning and inference on the high-level gesture model. Specifically, we learn the parameters of the bigram language model (Equation 1). We describe three basic techniques to learn the bigram model: supervised, unsupervised, and hybrid.

2.4.1 Supervised Learning Given a set of n labeled GWord sequences $\mathcal{L} = \{\mathcal{W}_1 \dots \mathcal{W}_n\}$ with $\mathcal{W}_i = \{s, v_{(i,1)}, \dots, v_{(i,m_i)}, t\}$ where $m_i + 2$ is the length of sequence $\mathcal{W}_i$ and $v_{(i,j)} \in \mathcal{V}$. The GWord prior is given by

$$P(v_k) = \frac{\sum_{i=1}^{n} \delta(v_k, v_{(i,1)})}{n} \tag{4}$$

where $\delta(\cdot)$ is the Kronecker delta function and $v_k \in \mathcal{V}$. The prior computes the probability that a Gesture Sentence begins with a certain GWord. The bigram transition probability is given by the following equation.

$$P(v_l|v_k) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m_i-1} \delta(v_k, v_{(i,j)}) \cdot \delta(v_l, v_{(i,j+1)})}{\sum_{i=1}^{n} \sum_{j=1}^{m_i-1} \delta(v_k, v_{(i,j)})} \tag{5}$$

Intuitively, Equation 5 measures the transition probability from a GWord v_k to another GWord $v_l \in \mathcal{V}$ by accumulating the number of bigram pairs $v_k \rightarrow v_l$ and normalizing by the number of bigrams beginning with v_k.

2.4.2 Unsupervised Learning Given a set of n unlabeled image sequences $\mathcal{U} = \{U_1 \ldots U_n\}$. We generate an initial bigram model M_0 in a uniform fashion based on the PGM. We can use additional heuristics based on the specific application to refine the uniform initialization. We train the bigram model using an EM-like [22] iterative algorithm.

1. $M \leftarrow M_0$
2. Compute the best labeling (Section 2.5) for each sequence in $\mathcal{U}$ based on the current bigram model M.
3. Using the supervised learning algorithm (discussed previously), refine the bigram model M.
4. Repeat until a fixed number of iterations is reached or the change of the bigram model in successive iterations is small.

2.4.3 Hybrid Learning Given a set of labeled GWord sequences $\mathcal{L}$ and a set of unlabeled image sequences $\mathcal{U}$. We generate an initial bigram model M_0 using the labeled sequences with the supervised learning algorithm discussed above. Then, we refine the bigram model in an iterative manner similar to the one used in unsupervised learning.

1. $M \leftarrow M_0$
2. Compute the best labeling (Section 2.5) for each sequence in $\mathcal{U}$ based on the current bigram model M. Call the labeled sequences $\hat{\mathcal{U}}$.
3. $\mathcal{T} = \bigcup(\mathcal{L}, \hat{\mathcal{U}})$.
4. Using the data $\mathcal{T}$ perform the supervised learning algorithm (discussed previously) to refine the bigram model M.
5. Repeat until a fixed number of iterations is reached or the change of the bigram model in successive iterations is small.

2.5 Inference on the PGM

Given an image sequence $\mathcal{S}$ of length m and a PGM with an embedded bigram model, we construct the inference problem as the search for the best labeling $\mathcal{L}$ of $\mathcal{S}$ that maximizes the overall probability given in Equation 3. Formally, the inference problem is stated as

$$\{v_1^* \dots v_p^*\} = \arg\max_{\mathcal{W}=f(\mathcal{S})} P(\mathcal{W}) \cdot P(\mathcal{S}|\mathcal{W}) \tag{6}$$

where $\mathcal{S} \doteq \{\mathcal{S}_1 \dots \mathcal{S}_p\}$, $f(\mathcal{S}) = \{v_1 \dots v_p\}$ is a one-to-one mapping from a sequence segmentation to a Gesture Sentence, and p is unknown. Let $g(\cdot)$ be the mapping from subsequence $\mathcal{S}_i$ to a GWord v_i; it is computed using the maximum-likelihood criterion:

$$g(\mathcal{S}_i) = \arg\max_{v_j \in \mathcal{V}} P(\mathcal{S}_i|v_j) \tag{7}$$

Theoretically, the inference problem in Equation 6 could be solved by an exhaustive search. However, the combinatorial complexity is prohibitive. Furthermore, the fundamental differences in the three types of low-level gesture processors makes the optimization more difficult. In addition, online processing is a prerequisite for human-computer interfaces. Thus, we propose a sub-optimal, greedy algorithm.

Initialize the algorithm by setting $v_0 = s$ and $\mathcal{S}_0 = \emptyset$. At stage t in the algorithm processing, we search for the best transition from v_t to v_{t+1} which maximizes path probability, defined as the product of the transition probability $P(v_{t+1}|v_t)$ and the observation probability $P(\mathcal{S}_{t+1}|v_{t+1})$. The beginning of subsequence $\mathcal{S}_{t+1}$ is set as the end of $\mathcal{S}_t$. To determine the end of the subsequence S_{t+1} and thus make the greedy path choice, we incrementally increase the length of the subsequence until the path to one of the children c meet both of the following two conditions.

1. The observation probability of the child passes a threshold τ_c. We discuss a supervised technique for learning the node thresholds below.
2. The path probability of c is highest among all of the children of node v_t. Formally, $c = \arg\max_{v_{t+1}} P(v_{t+1}|v_t) \cdot P(\mathcal{S}_{t+1}|v_{t+1})$.

In Figure 3 we show a graphical depiction of a stage in the middle of the greedy algorithm. In the figure, at stage $t+1$, child c_2 of node v_t is chosen. We see that at the end of stage $t+1$ the end of sequence $\mathcal{S}_{t+1}$ has been determined.

We learn the individual node thresholds using a supervised technique. Given a set of labeled GWord sequences and segmented image sequence pairs $(\mathcal{W}_i, \mathcal{S}_i) \in \mathcal{D}$. we pose the problem of determining the threshold τ_v for GWord $v \in \mathcal{V}$ as finding the minimum observation probability for all occurrences of v:

$$\tau_v = \min_{(\mathcal{W}_i,\mathcal{S}_i)\in\mathcal{D}} \quad \min_{v_i \in \mathcal{W}_i \text{ and } \delta(v_i,v)} P(\mathcal{S}_i|v) \tag{8}$$

First, we initialize all the thresholds to 0, $\tau_v = 0, \forall v \in \mathcal{V}$, to handle the case where v does not occur in $\mathcal{L}$. Then, for all GWords $v \in \mathcal{V}$ we compute τ_v according to Equation 8.

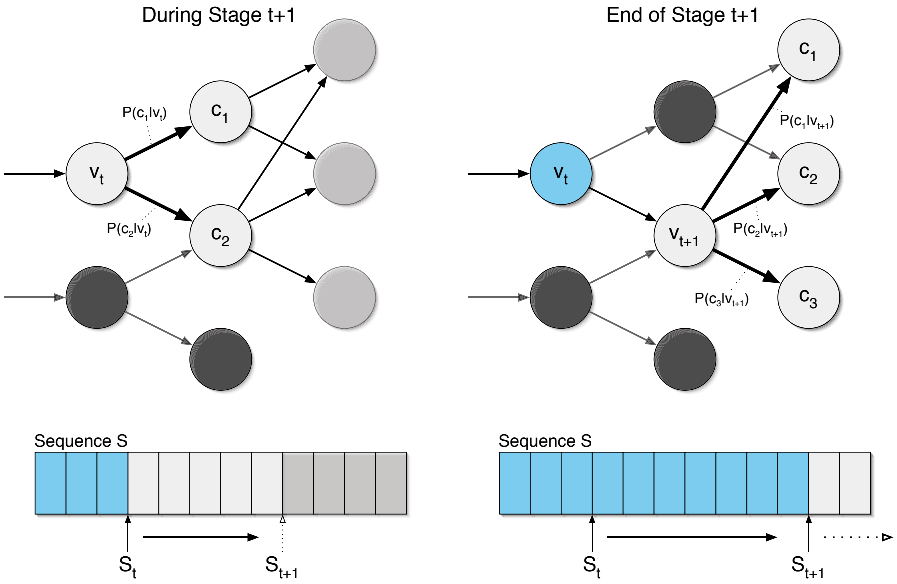


Figure 3 Graphical depiction of two stages of the proposed greedy algorithm for computing the inference on the PGM. Dark gray nodes are not on the best path and are disregarded, and blue represents past objects on the best path.

3 Experimental Setup

We analyze the proposed model for recognizing composite gestures by constructing a gesture set and the corresponding PGM. We employ the Visual Interaction Cues (VICs) paradigm [42] (Section 3.1) to structure the vision processing and use the 4D Touchpad [13] (Section 3.2) as the experimental platform.

3.1 The Visual Interaction Cues Paradigm

The VICs paradigm [42] is a methodology for vision-based interaction operating on the fundamental premise that, in general vision-based human computer interaction (VBI) settings, global user modeling and tracking are not necessary. As discussed earlier, typical vision-based interaction methods attempt to perform continuous, global user tracking to model the interaction. Such techniques are computationally expensive, prone to error and the re-initialization problem, prohibit the inclusion of arbitrary numbers of users, and often require a complex gesture-language the user must learn. However, under the VICs paradigm, we focus on the components of the interface itself instead of on the user.

We motivate the paradigm with a simple, real-world example. When a person presses the keys of a telephone while making a telephone-call, the telephone maintains no notion of the user. Instead, it only recognizes the result of a key on the keypad being pressed. In contrast, typical methods for VBI would attempt to construct a model of the user's finger, track it through space, and perform some action recognition as the user pressed the keys on the telephone. It is likely that in such processing, the computer system would also have to be aware of the real-world geometric structure of the telephone itself. We claim that this processing is not necessary.

Let $\mathcal{W}$ be the space in which the components of the interface reside. In general, $\mathcal{W}$ is the 3D Euclidean space $\mathbb{R}^3$ but it can be the Projective plane $\mathbb{P}^2$ or the Euclidean plane $\mathbb{R}^2$. Define an interface component mapping $M : \mathcal{C} \to \mathcal{X}$, where $\mathcal{C} \subset \mathcal{W}$ and $\mathcal{X} \doteq \{\mathcal{I} \vee A(\mathcal{I})\}$ with $\mathcal{I}$ the image as defined in Section 2.1 and $A(\cdot)$ being an arbitrary function, $A : \mathbb{P}^2 \to \mathbb{P}^2$. Intuitively, the mapping defines a region in the image to which an interface component projects (see Figure 4).

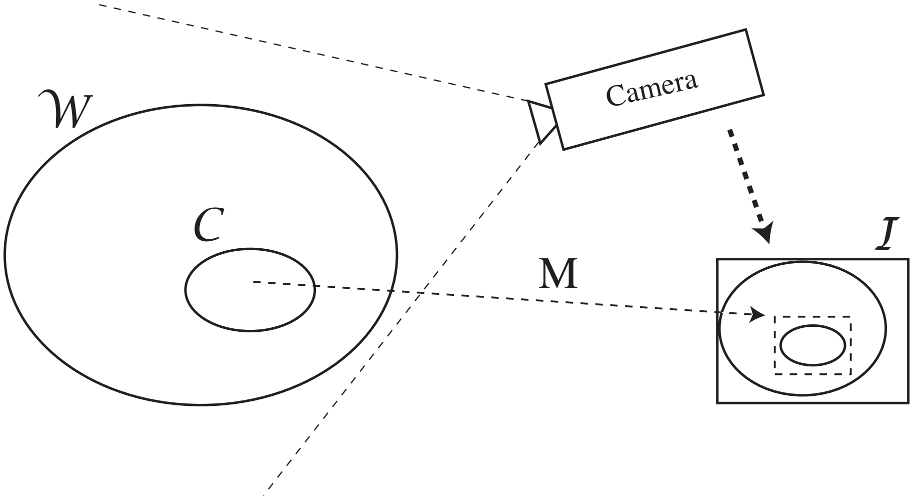


Figure 4 Schematic explaining the principle of local image analysis for the VICs paradigm: M is the component mapping that yields a region of interest in the image $\mathcal{I}$ for analyzing actions on component $\mathcal{C}$

If, for each interface component and the current image, a map is known, detecting a user action reduces to analyzing a local region in the image. This fundamental idea of local image analysis is the first principle of the VICs paradigm.

The second principle of the VICs paradigm concerns the computational methods involved in analyzing the image(s). Each interface component defines a function-specific set of image processing components that are ordered in a simple-to-complex fashion such that each level of increasing interaction-detection precision (and increasing computational cost) is executed only if the previous levels have validated the likely existence of an expected object in this ROI. Such a notion of simple-to-complex processing is not novel; for example, in early image processing, pyramidal schemes were invented that perform coarse-to-fine analysis of images [2]. However, it is integral to the VICs paradigm.

3.2 The 4D Touchpad

In this section, we explain a VICs platform [13] that has been constructed based on the 3D-2D Projection interaction mode [42]. Here, a pair of wide-baseline cameras is directed at a flat-panel display. This setup is shown in Figure 5 (left). The platform incorporates four dimensions of data: two for the physical screen, a third from the binocular vision, and a fourth from the temporal VICs processing.

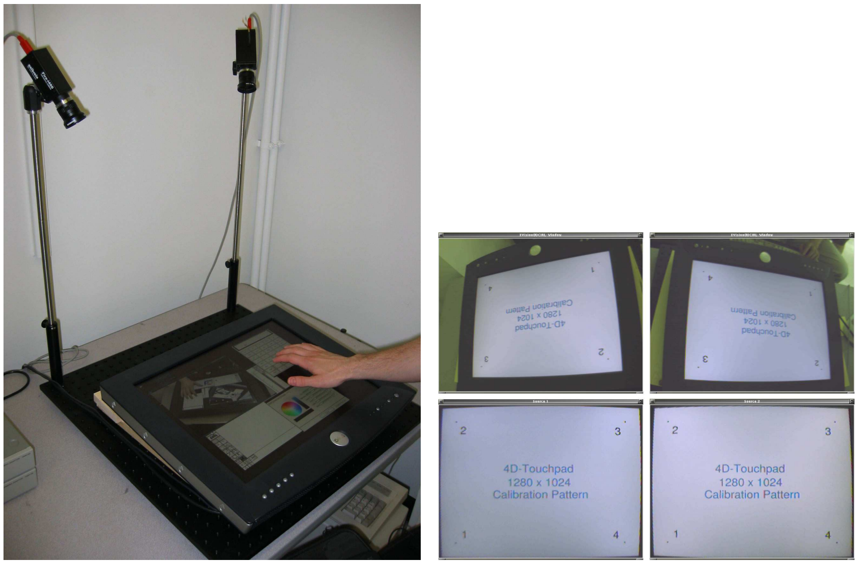


Figure 5 (left) 4D Touchpad Platform. (right) Example rectification process for the 4D Touchpad. Upper row contains the original images with the rectified images below.

Since the cameras $c = 1, 2$ are fixed, the interface component mapping for the system can be computed during an initial calibration stage. We assume the optics can be modeled by perspective projection. Let $\mathcal{F} \subset \mathbb{P}^2$ be the coordinate frame of the flat-panel screen. Define $H_c : \mathcal{I}_c \to \mathcal{F}$ the mapping from each input image $\mathcal{I}_c$ to the flat-panel frame $\mathcal{F}$. We employ a homography [14] for the mapping. Since the VICons exist in frame $\mathcal{F}$, each interface component mapping is simply the identity. This transformation process is known as rectification, and we show an example of it in Figure 5 (right). Radial distortion is evident in the rectified images; in the current system, we do not include any radial distortion correction. While doing so would complete the rectification procedure, in practice we find it unnecessary.

The rectification process warps both camera images in a way that all points in the plane of the flat-panel screen appear at the same position in both camera images. This can be used for stereo calculation; the resulting space is $\mathbb{P}^2 \times \mathbb{Z}$ with 0 disparity being in the plane of the flat-panel. Disparity is defined as the absolute distance between corresponding points in the two rectified images.

3.3 Gesture Set

The goal of the proposed framework is to facilitate the integration of different types of gestures (Section 2.3) and thus, natural interaction in virtual environments. To that end, we present an experimental gesture set with ten elements (GWords) with each of the three gesture types represented.

3.3.1 Low-Level Gwords The gesture set is designed to be used in general manipulative interfaces where actions such as selecting, grasping, and translating are required. Table 1 contains graphical depictions of each GWord. For dynamic gestures, we show three example images during the progress of the gesture.

- **Press**. Press is the static posture of a single finger activating the interface component.
- **Left**. Left is a dynamic, non-parametric motion of a finger to the left with respect to the interface component.
- **Right**. Right is a dynamic, non-parametric motion of a finger to the right with respect to the interface component.
- **Back**. Back is a dynamic, non-parametric retraction of the finger off the interface component.
- **Twist**. Twist is a clockwise twisting motion of a finger atop the interface component (dynamic, non-parametric).
- **Grab 1**. The first grabbing gesture is the dynamic, non-parametric motion of two fingers approaching the interface component open and closing once they have reached it.
- **Grab 2**. The second grabbing gesture is the dynamic, non-parametric motion of two fingers approaching the interface component open and remaining open upon reaching it.
- **Track**. Track is a parametric gesture that tracks two translational degrees-of-freedom.
- **Rotate**. Rotate is a parametric gesture that tracks one rotational degree-of-freedom.
- **Stop**. Stop is a static posture represented by an open hand atop the interface component.

3.3.2 Probabilistic Graphical Model With the algorithms presented in Section 2, we construct and train a probabilistic graphical model to be the *interaction language*. Figure 6 is a graphical depiction of the PGM; for clarity, we have not drawn any edges with zero probability in the bigram language model from supervised

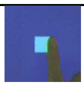
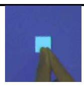
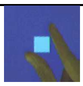
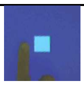
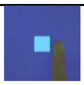
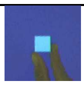
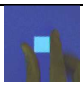
GWord	Press	Left	Right	Back	Twist	Grab 1	Grab 2	Track	Rotate	Stop
Stage 1										
Stage 2										
Stage 3										

Table 1 Example images of basic GWords.

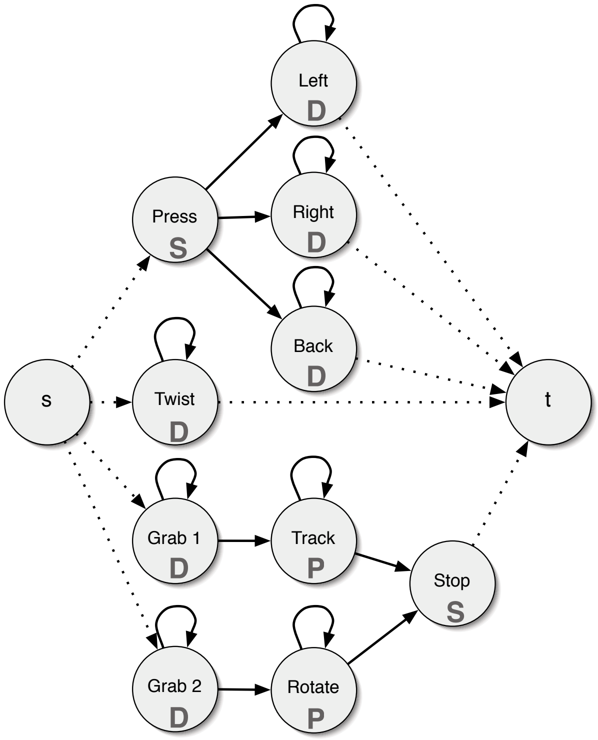


Figure 6 The probabilistic graphical model we constructed for our experimental setup. Edges with zero probability are not drawn. The nodes are labeled as per the discussion in Section 3.3. Additionally, each node is labeled as either **P**arametric, **D**ynamic, non-parametric, or **S**tatic posture.

learning. A simple Gesture Sentence is thus **Press** → **Left**: the user approaches an interface component with an outstretched finger and then swipes his or her finger to the left. For example, such composite gesture could be used to delete an interaction component. A more complex Gesture Sentence involving all three types of low-level GWords is **Grab 1** → **Track** → **Stop**. This Gesture Sentence could be widely used in VR to grab and move virtual objects.

3.3.3 Implementation of Low-Level GWords As discussed in Section 2.3, we include three types of low-level gesture processing: static posture, non-parametric dynamic, or parametric dynamic. In this section we discuss the construction of these low-level processors for our experimental setup. However, from the perspective of the PGM framework, the specific construction of the low-level processors is arbitrary.

Static Posture. Static postures are based on the recognition of single discriminative frames of video. A multitude of potential methods exist in the literature for such recognition: SIFT keys [19] and Shape Contexts [6] for example. Exploiting the principle of local image analysis from the VICs paradigm (Section 3.1), we use a common technique from machine learning called neural network processing [31]. We train a standard three-layer binary (on/off) network. We fix a local image neighborhood of 128 x 128 pixels corresponding to the VICon region in the image defined by its interface component mapping. As input to the network, we choose a coarse sub-sampling (16 x 16) and take non-overlapping pixel-neighborhood averages. We employ the intensity only (Y channel in YUV images).

Non-parametric Dynamic. We model the dynamics of the motion of the finger using discrete forward HMMs. For a complete discussion of our technique, refer to [42,43]. Instead of directly tracking the hand, 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 interaction component. The appearance feature is represented as a discrete volume with each cell describing the similarity between corresponding image patches of the stereo pair. The motion cue is captured via differentiating the appearance feature between frames. A K-means based vector quantization [18] algorithm is used to learn the cluster structure of these raw visual features. Then, the image sequence of a gesture is converted to a series of symbols that indicate the cluster identities of each image pair. A 6-state forward HMM is used to model the dynamics of each gestures. The parameters of the HMM are learned via the standard forward-backward algorithm based on the recorded gesture sequences. The gesture recognition is based on the probability that each HMM generates the given gesture image sequence.

Parametric Dynamic. The implementation of a parametric, dynamic processor is dependent on the task for which it is to be used. For example, in our gesture set, we require both a translational and a rotational processor. Again, many potential techniques exist for tracking the local motion of an image patch or pair of image patches. In our experiments, we used a *filtered-detection* algorithm [12]: for each frame of video, we detect the feature(s) of interest and use a linear Kalman filter to model the dynamics of motion. For example, in the case of the translational processor, we detect the image point where the two grasping fingertips (thumb and index finger) meet. Assuming we can detect the same exact point every frame, tracking this grasping-point provides the two translational degrees-of-

Prior											
GWord	Left	Right	Back	Twist	Grab 1	Grab 2	Track	Rotate	Press	Stop	t
	0	0	0	0.167	0.167	0.167	0	0	0.5	0	0

Bigram Model											
GWord	Left	Right	Back	Twist	Grab 1	Grab 2	Track	Rotate	Press	Stop	t
Left	0.94	0	0	0	0	0	0	0	0	0	0.06
Right	0	0.93	0	0	0	0	0	0	0	0	0.07
Back	0	0	0.84	0	0	0	0	0	0	0	0.16
Twist	0	0	0	0.93	0	0	0	0	0	0	0.07
Grab 1	0	0	0	0	0.94	0	0.06	0	0	0	0
Grab 2	0	0	0	0	0	0.94	0	0.06	0	0	0
Track	0	0	0	0	0	0	0.96	0	0	0.04	0
Rotate	0	0	0	0	0	0	0	0.95	0	0.05	0
Press	0.33	0.33	0.33	0	0	0	0	0	0	0	0
Stop	0	0	0	0	0	0	0	0	0	0.7	0.3
t	0	0	0	0	0	0	0	0	0	0	1

Table 2 Language Model (Priors and Bigram) using supervised learning.

freedom. While it is difficult (or impossible) to detect exactly the same point every frame, in practice, the Kalman filter handles small variations in the point detection.

4 Experimental Results

Figure 6 shows our vocabulary of 6 possible composite gestures. To quantitatively analyze the PGM, we recorded a training set of 100 video sequences each corresponding to one of the 6 gestures. The length of the sequences vary from 30 to 90 frames (at 10 frames-per-second). These sequences were not used in training the low-level gesture units. For the supervised training, we manually labeled each frame of the video sequences with a GWord. For unsupervised learning, we initialized a uniform language model and used the algorithm in Section 2.4.2 to refine the model. After 2 iterations, the bigram model converged.

We compare the language models after supervised and unsupervised learning in Tables 2 and 3, respectively. The bigram models are presented as adjacency matrices such that each row represents the probability of transitioning from a GWord (leftmost column) to other GWords (or itself). It can be seen that the 2 PGM bigram models have similar structure. It shows that even without good heuristics or labeled data, our unsupervised learning algorithm can still capture the underlying language model from raw gesture sequences.

However, there are differences worth mentioning. For example, the prior for **Stop** from unsupervised learning is 0.03, but there are no sequences in the training corpus that begin with it. This is caused by the failure of the inference algorithm given a uniform bigram language model. Second, we see a difference in the self-transition probability for the **Press** GWord. In the labeled data, we fixed the dura-

Prior

GWord	Left	Right	Back	Twist	Grab 1	Grab 2	Track	Rotate	Press	Stop	t
	0	0	0	0.1	0.11	0.16	0	0	0.6	0.03	0

Bigram Model

GWord	Left	Right	Back	Twist	Grab 1	Grab 2	Track	Rotate	Press	Stop	t
Left	0.91	0	0	0	0	0	0	0	0	0	0.09
Right	0	0.88	0	0	0	0	0	0	0.0	0	0.12
Back	0	0	0.83	0	0.01	0	0	0	0	0	0.16
Twist	0	0	0.0	0.95	0	0	0	0	0	0	0.05
Grab 1	0	0	0	0	0.82	0	0.14	0	0	0.02	0.02
Grab 2	0	0	0	0	0	0.77	0.04	0.15	0	0.04	0
Track	0	0	0	0	0.02	0	0.77	0.03	0	0.16	0.02
Rotate	0	0	0	0	0	0.01	0.03	0.90	0	0.06	0
Press	0.02	0.02	0.03	0	0	0	0	0	0.91	0	0.02
Stop	0	0	0	0	0	0	0	0	0	0.77	0.23
t	0	0	0	0	0	0	0	0	0	0	1

Table 3 Language Model (Priors and Bigram) using unsupervised learning.

Gesture Sentence	Supervised %	Unsupervised %
Press → Left	97.3	97.3
Press → Right	85.7	78.6
Press → Back	88.9	90.4
Twist	96.4	96.4
Grab 1 → Track → Stop	93.3	82.1
Grab 2 → Rotate → Stop	97.9	97.9

Table 4 Recognition accuracy of the PGM used in our experimentation.

tion of **Press** to one frame, but with a uniform bigram model, a static posture can last for several consecutive frame via self-transition.

During testing, we used the proposed greedy inference algorithm to analyze the video sequences. In Table 4, we present the recognition accuracy for the gestures for both language models. For each sequence, we compared its known composite gesture identity with the GWord output of the PGM. We consider the output correct if it matches the GWord sentence at every stage.

We can see from the results that the proposed high-level gesture language modeling can recognize compositions of heterogeneous low-level gestures. These composite gestures would be impossible to recognize using traditional unimodal techniques, while the PGM formulation takes advantage of high-level linguistic constraints to integrate fundamentally different low-level gesture units in a coherent probabilistic model.

However, the recognition accuracy for gesture **Press** → **Right** and gesture **Press** → **Back** are relatively poor. From visual inspection of the recognition algorithm's output, we find that this is due to the greedy algorithm. The **Left**, **Right**, and **Back** are modeled with HMMs and trained with relatively long sequences (e.g.

20 frames). However, during inference, the greedy algorithm jumps to a conclusion based on an shorter subsequences (e.g. 7 frames). In our experiments, we see a bias toward the **Left** GWord for these incomplete subsequences.

The recognition results from the supervised and the unsupervised learning are comparable. This suggests that our linguistic approach to gesture recognition can perform well without a heuristic prior or manually labeled data. Hence, our method is less susceptible to the curse of dimensionality which, in our case, is that the amount of data (labeled, for supervised learning) required for learning generally increases exponentially with the number of GWords.

5 Conclusion

We have presented a linguistic approach to recognize composite gestures. The composite gestures consist of three different types of low-level units (GWords): static, posture-based primitives; non-parametric dynamic gestures; and parametric, dynamic gestures. We construct a coherent model by combining the GWords and a high-level language model in a probabilistic framework which is defined as a graphical model. We have proposed unsupervised and supervised learning algorithms; our results show that even with a random initialization, the PGM can learn the underlying gesture language model. By combining the PGM and the greedy inference algorithm, our method can model gestures composed of heterogeneous primitives.

Our approach allows the inference of composite gestures as paths through the PGM and uses the high-level linguistic constraints to guide the recognition of composite gestures. However, the proposed greedy inference algorithm will make locally optimal decisions since it is operating online. Furthermore, even in the offline case, the heterogeneous, low-level gesture processes make an exhaustive search through all composite gesture sequences computationally prohibitive.

The experiments in this paper include a relatively small gesture vocabulary of 10 low-level GWords and 6 composite gestures. While we have found the bigram model sufficient to capture the linguistic constraints of this vocabulary, it is unclear if it will scale well with larger gesture vocabularies. In future work, we intend to investigate its scalability and other more context-rich language models. In addition, we are currently integrating the gesture model into a VR system. We plan to perform human factors experiments to analyze the efficacy of our gestural modeling in the system.

Acknowledgements We thank Darius Burschka for his help with the Visual Interaction Cues project. This work was in part funded by a Link Foundation Fellowship in Simulation and Training and by the National Science Foundation under Grant No. 0112882.

References

1. 3rd Tech. Hiball-3100 sensor, `http://www.3rdtech.com/HiBall.htm`.

2. P. Anandan. A Computational Framework and an Algorithm for the Measurement of Visual Motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.
3. Ascension Technology Corporation. Flock of birds, `http://www.ascension-tech.com/products/flockofbirds.php`.
4. 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.
5. Ronald T. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(11):1–38, 1997.
6. Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape Context: A New Descriptor for Shape Matching and Object Recognition. In *Neural Information Processing*, pages 831–837, 2000.
7. Aaron Bobick and James Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.
8. 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.
9. M. Brand, N. Oliver, and A.P. Pentland. Coupled Hidden Markov Models for Complex Action Recognition. In *Computer Vision and Pattern Recognition*, pages 994–999, 1997.
10. Christoph Bregler. Learning and Recognizing Human Dynamics in Video Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
11. Q. Cai and J. K. Aggarwal. Human Motion Analysis: A Review. *Journal of Computer Vision and Image Understanding*, 73(3):428–440, 1999.
12. Jason J. Corso. Vision-Based Techniques for Dynamic, Collaborative Mixed-Realities. In *Research Papers of the Link Foundation Fellows. Ed. Brian J. Thompson. University of Rochestor Press in association with The Link Foundation*, volume 4. 2004.
13. Jason J. Corso, Darius Burschka, and Gregory D. Hager. The 4DT: Unencumbered HCI with VICs. In *IEEE Workshop on Human Computer Interaction at Conference on Computer Vision and Pattern Recognition*, 2003.
14. Olivier Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1993.
15. Aphrodite Galata, Neil Johnson, and David Hogg. Learning Variable-Length Markov Models of Behavior. *Computer Vision and Image Understanding*, 83(1):398–413, 2001.
16. B. Insko, M. Meehan, M. Whitton, and F. Brooks. Passive haptics significantly enhances virtual environments. Technical Report 01-10, Department of Computer Science, UNC Chapel Hill, 2001.
17. Y. A. Ivanov and Aaron F. Bobick. Recognition of Visual Activities and Interactions by Stochastic Parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.
18. Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1999.
19. David Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
20. 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.
21. Stephen J. Mckenna and Kenny Morrison. A Comparison of Skin History and Trajectory-Based Representation Schemes for the Recognition of User-Specific Gestures. *Pattern Recognition*, 37:999–1009, 2004.
22. Todd K. Moon. The Expectation-Maximization Algorithm. *IEEE Signal Processing Magazine*, pages 47–60, 1996.

23. Kai Nickel and Rainer Stiefelhagen. Pointing Gesture Recognition based on 3D-Tracking of Face, Hands and Head Orientation. In *Workshop on Perceptive User Interfaces*, pages 140–146, 2003.
24. Kenji Oka, Yoichi Sato, and Hideki Koike. Real-Time Fingertip Tracking and Gesture Recognition. *IEEE Computer Graphics and Applications*, 22(6):64–71, 2002.
25. Vasu Parameswaran and Rama Chellappa. View Invariants for Human Action Recognition. In *Computer Vision and Pattern Recognition*, volume 2, pages 613–619, 2003.
26. Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997.
27. Alex Pentland and Andrew Liu. Modeling and Prediciton of Human Behavior. *Neural Computation*, 11(1):229–242, 1999.
28. F. Quek. Unnencumbered Gesture Interaction. *IEEE Multimedia*, 3(3):36–47, 1996.
29. Lawrence Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
30. M. Salada, J. E. Colgate, M. Lee, and P. Vishton. Validating a novel approach to rendering fingertip contact sensations. In *Proceedings of the 10th IEEE Virtual Reality Haptics Symposium*, pages 217–224, 2002.
31. Robert J. Schalkoff. *Artificial Neural Networks*. The McGraw-Hill Companies, Inc., 1997.
32. Yifan Shi, Yan Huang, David Minnen, Aaron Bobick, and Irfan Essa. Propagation Networks for Recognition of Partially Ordered Sequential Action. In *Computer Vision and Pattern Recognition*, volume 2, pages 862–869, 2004.
33. Min C. Shin, Leonid V. Tsap, and DMitry B. Goldgof. Gesture Recognition Using Bezier Curvers for Visualization Navigation from Registered 3-D Data. *Pattern Recognition*, 37(0):1011–1024, 2004.
34. 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.
35. Carlo Tomasi, Slav Petrov, and Arvind Sastry. 3D Tracking = Classification + Interpolation. In *Proc. Int'l Conf. Computer Vision*, pages 1441–1448, 2003.
36. Christian von Hardenberg and Francois Berard. Bare-Hand Human-Computer Interaction. In *Workshop on Perceptive User Interfaces*, 2001.
37. Andrew Wilson and Aaron Bobick. Parametric Hidden Markov Models for Gesture Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):884–900, 1999.
38. Christopher Wren, 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.
39. Ying Wu and Thomas S. Huang. View-independent Recognition of Hand Postures. In *Computer Vision and Pattern Recognition*, volume 2, pages 88–94, 2000.
40. Ying Wu and Thomas S. Huang. Hand Modeling, Analysis, and Recognition. *IEEE Signal Processing Magazine*, 18(3):51–60, 2001.
41. 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.
42. Guangqi Ye, Jason J. Corso, Darius Burschka, and Gregory D. Hager. VICs: A Modular HCI Framework Using Spatio-Temporal Dynamics. *Machine Vision and Applications*, 2005. to appear.

43. Guangqi Ye, Jason J. Corso, and Gregory D. Hager. Gesture Recognition Using 3D Appearance and Motion Features. In *Proceedings of CVPR Workshop on Real-Time Vision for Human-Computer Interaction*, 2004.
44. Guangqi Ye, Jason J. Corso, Gregory D. Hager, and Allison M. Okamura. VisHap: Augmented Reality Combining Haptics and Vision. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 3425–3431, 2003.
45. Y. Yokokohji, J. Kinoshita, and T. Yoshikawa. Path planning for encountered-type haptic devices that render multiple objects in 3d space. *Proceedings of IEEE Virtual Reality*, pages 271–278, 2001.
46. T. Yoshikawa and A. Nagura. A touch/force display system for haptic interface. *Presence*, 10(2):225–235, 2001.
47. 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.
48. H. Zhou, D.J. Lin, and Thomas S. Huang. Static Hand Postures Recognition based on Local Orientation Histogram Feature Distribution Model. In *Proceedings of CVPR Workshop on Real-Time Vision for Human-Computer Interaction*, 2004.