# *Continuous Body and Hand Gesture Recognition for Natural Human-Computer Interaction*

# Continuous Body and Hand Gesture Recognition for Natural Human-Computer Interaction

YALE SONG, DAVID DEMIRDJIAN, and RANDALL DAVIS
Massachusetts Institute of Technology

We present a new approach to vision-based continuous gesture recognition that tracks body and hand movements and predicts gesture labels from unsegmented and unbounded input in a unified framework. The system uses a stereo camera for body and hand tracking and does not rely on special markers, allowing more natural interaction. Body poses are reconstructed in 3D space using a generative model-based approach and a multi-hypothesis Bayesian estimation framework. Hand poses are classified using an example-based approach and a multi-class support vector classifier. Gestures are recognized using a discriminative dynamic hidden-state inference framework and a heuristic approach for sequence segmentation and successive labeling.

The system extends existing computer vision and machine learning techniques with three novel approaches: (1) exploiting static and dynamic attributes of motion for body pose estimation; (2) using a Gaussian temporal-smoothing kernel for gesture recognition; and (3) using a two-layered heuristic approach for segmenting and labeling gestures from continuous input.

We tested our system in a real-world human-computer interaction scenario using 10 body-and-hand gestures. Based on the tests, (1) we show that combining body and hand signals significantly improves the recognition accuracy; (2) we identify which features of body and hands are most informative; (3) we show that using a Gaussian temporal-smoothing kernel significantly improves performance; and (4) we show that our two-layered heuristic approach improves continuous gesture segmentation and labeling. Also, we show that our system is able to achieve the recognition accuracy of 93.7% for isolated gestures and 88.37% for continuous gestures.

Categories and Subject Descriptors: I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Motion*; I.5.5 [**Pattern Recognition**]: Implementation—*Interactive Systems*

General Terms: Algorithms, Design, Experimentation

Additional Key Words and Phrases: Vision-based body and hand tracking, continuous gesture recognition

## 1. INTRODUCTION

For more than 40 years, human-computer interaction has been focused on the keyboard and mouse. Although this has been successful, as computation becomes increasingly mobile, embedded, and ubiquitous, it is far too constraining as a model of interaction. As both research and commercial applications have shown, gesture-based interaction is the wave of the future. In consumer electronics, for example,

Authors' address: Y. Song, D. Demirdjian, and R. Davis, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar St., Cambridge, MA 02139; email: {yalesong,demirdji,davis}@csail.mit.edu.

there is an emerging interest in gesture-based video game controllers, such as Microsoft Kinect, Nintendo Wii, and Sony PlayStation Move.

Gestural interaction has a number of clear advantages. First, it uses equipment we always have on hand: there is nothing extra to carry, misplace, or leave behind. Second, it can be designed to work from actions that are natural and intuitive, so there is little or nothing to learn about the interface. Third, it lowers cognitive overhead, a key principle in human-computer interaction: Gesturing is instinctive and a skill we all have, so it requires essentially no thought, leaving the focus on the task, as it should be, not on the interaction modality.

Human gesture is most naturally expressed with body and hands, ranging from the simple gestures we use in normal conversations, to the more elaborate gestures used by baseball coaches giving signals to players; soldiers gesturing for tactical tasks; and police giving body and hand signals to drivers. Current technology for gesture understanding is, however, still sharply limited, with body and hand signals typically considered separately, restricting the expressiveness of the gesture vocabulary and making interaction less natural.

Interactive gesture understanding should be able to process continuous input seamlessly, i.e., no awkward transitions, interruptions or indications of disparity between gestures. We use the terms *unsegmented* and *unbounded* to clarify what we mean by continuous input. Continuous input is unsegmented, i.e., there is no indication of signal boundaries, such as gesture start and end. Continuous input is also unbounded, i.e., the beginning and the end of the whole sequence is unknown, regardless of whether the sequence contains a single gesture or multiple gestures. Interactive gesture understanding from this continuous input, which is both unsegmented and unbounded, thus needs to be done *successively*, i.e., prediction of a gesture class label should be done as new observations are made.

We present a new approach to vision-based continuous gesture recognition that attends to body and hands, allowing a richer gesture vocabulary and more natural human-computer interaction. The three main components of our system are a 3D upper body pose estimator, a hand pose classifier, and a continuous gesture recognizer. We implemented our system by using and extending a variety of computer vision and machine learning techniques.

3D body pose estimation is performed by constructing a kinematic body model and estimating poses in a multi-hypothesis Bayesian inference framework, using a particle filter [Isard and Blake 1998]. Hand poses are classified by extracting a histogram of oriented gradients (HOG) features [Dalal and Triggs 2005] and learning a multi-class Support Vector Machine (SVM) classifier [Vapnik 1995]. Finally, gesture recognition is performed using a discriminative hidden-state inference framework, a latent-dynamic conditional random field (LDCRF) [Morency et al. 2007].

We extend existing techniques for pose tracking and gesture recognition with three novel approaches. First, when estimating body pose, we use both static attributes of motion (i.e., 3D visible surface and contour point cloud) and a dynamic attribute of motion, a motion history image (MHI) [Bobick and Davis 1996] that allows us to capture the discrepancies in the dynamics of motion. Second, for gesture recognition, we incorporate a Gaussian temporal-smoothing kernel [Harris 1978] into the HCRF formulation to capture long-range dependencies and make

our system less sensitive to the noise from estimated time-series data, while not increasing the dimensionality of input feature vectors. This keeps the computational complexity the same as the original HCRF model. Third, for labeling a continuous gesture sequence, we developed a two-layered heuristic approach which performs sequence segmentation and successive gesture labeling simultaneously.

To evaluate our system on a realistic scenario, we conducted a set of experiments using 10 body-and-hand gestures from the Naval Air Training and Operating Procedures Standardization (NATOPS) aircraft handling signals database [Song et al. 2011b]. Based on our tests, (1) we show that combining body and hand poses significantly improves the gesture recognition accuracy; (2) we indicate which body and hand features are most informative for this recognition task; (3) we show that a Gaussian temporal-smoothing significantly improves the gesture recognition accuracy; and (4) we show that our two-layered heuristic approach to continuous gesture recognition significantly improves gesture segmentation and successive labeling tasks when used in conjunction with an LDCRF model.

Section 2 reviews some of the related work in pose tracking and gesture recognition, highlighting the novelties in our system, Section 3 gives an overview of our gesture recognition system, Section 4 describes body and hand tracking, Section 5 describes gesture recognition using a Gaussian temporal-smoothing kernel as well as our two-layered heuristic-approach for continuous gesture recognition, and Section 6 describes experiments and results. Section 7 concludes with a summary of contributions and suggesting directions for future work.

Some of the material presented in this paper has appeared in earlier conference proceedings [Song et al. 2011b; 2011a]. In this article, we give an in-depth description of the system as well as a detailed analysis from the experiments. In addition, we describe a new approach to continuous gesture recognition that has not been covered in the previous papers.

## 2. RELATED WORK

The topics covered in this paper range broadly from body and hand pose tracking to gesture recognition. This section briefly reviews some of the most relevant work; comprehensive review articles covering the material in detail include [Aggarwal and Cai 1999; Gavrila 1999; Moeslund and Granum 2001; Moeslund et al. 2006; Poppe 2007; Erol et al. 2007; Mitra and Acharya 2007].

Gesture-based interfaces typically require precise body and/or hand tracking methods. This is commonly done by wearing specially designed markers or devices (e.g., data glove [Zimmerman et al. 1986] or colored gloves [Wang and Popović 2009; Yin and Davis 2010]). However, the most natural form of gestural interaction would not require additional markers or sensors attached to the body. In our work, to avoid obtrusive and unnatural interaction, our system was built not to require any marker to be attached to the human body, but to perform motion tracking based solely on data from a single stereo camera.

Several successful vision-based tracking approaches have been reported, falling generally into two categories: model-based methods, which try to reconstruct a pose model in 3D space by fitting the model to the observed image [Deutscher et al. 2000; Demirdjian and Darrell 2002; Sminchisescu and Triggs 2003; Lee and Cohen 2006],

and example-based methods, which assume a pose vocabulary and try to learn a direct mapping from features extracted from images to the vocabulary [Brand 1999; Shakhnarovich et al. 2003; Mori and Malik 2006]. Model-based methods are in general not affected by a camera viewpoint, do not require a training dataset, and are generally more robust in 3D pose estimation. Example-based methods require a large training dataset and in general are more sensitive to camera viewpoints, but once a mapping function is learned, classification can be performed efficiently.

In gesture recognition, reconstructing body pose in 3D space provides important information, such as pointing direction, motivating our use of model-based approach for body pose estimation. Hand poses, by contrast, are more categorical, i.e., it is typically not crucial to distinguish fine-grained details of hand pose in order to understand a body-and-hand gesture. Therefore, we take an example-based approach to hand pose classification.

Previous efforts at body-and-hand tracking and gesture understanding include Buehler et al. [2009], which presented a vision-based arm-and-hand tracking system for sign language recognition. Upper body poses were estimated in 2D space using a generative model, with a combination of pictorial structures and HOG descriptors. Similar to our work, estimated wrist positions were used to determine hand positions and left/right hand assignment, but hand poses were not classified explicitly. Also, body poses were reconstructed in 2D space, losing some of the important features in gesture recognition (e.g., pointing direction).

There have also been active efforts to build a robust inference framework for pattern analysis tasks based on discriminative learning. Lafferty et al. [2001] introduced conditional random fields (CRFs), a discriminative learning approach that does not make conditional independence assumptions. Quattoni et al. [2005] introduced hidden conditional random fields (HCRFs), an extension to CRFs that incorporates hidden variables. Many other variants of HCRFs have been introduced since then [Sutton et al. 2004; Gunawardana et al. 2005; Wang et al. 2006; Morency et al. 2007], but most of these were tested on single-signal pattern recognition tasks (e.g., POS tagging [Lafferty et al. 2001], object recognition [Quattoni et al. 2005], body gesture recognition [Wang et al. 2006; Morency et al. 2007], and phone classification [Gunawardana et al. 2005]) and paid less attention to dealing with noisy input signals.

In this work, we demonstrate that discriminative hidden-state learning approaches are well suited to multi-signal (i.e., body and hand) gesture recognition tasks, and that significant recognition accuracy gains can be achieved by performing Gaussian temporal-smoothing.

Learning temporal patterns from body-and-hand dual-signal input sequences can be quite challenging due to the long-range dependencies among observations and the low signal-to-noise ratio (SNR). Previous work on HCRFs for gesture recognition [Wang et al. 2006] approached the first issue, capturing long-range dependencies, by defining a temporal window and concatenating signals within the window, creating a single large input feature at the cost of increasing dimensionality.

We take a slightly different approach and in doing so resolve both the first and second issues. Instead of concatenating neighboring signals, we use a Gaussian temporal-smoothing kernel to compute a weighted mean of neighboring input fea-

tures, not only capturing long-range dependencies but also making the framework less sensitive to noise. This approach keeps the dimensionality of input feature vectors unchanged, and hence our approach has the same computational complexity as the original HCRF model [Quattoni et al. 2005].

Another challenge in continuous gesture recognition lies in predicting gesture labels in real-time when an input sequence is both unsegmented and unbounded. Recent work has showed promising results on simultaneous sequence segmentation and labeling with bounded input sequence data: Sutton et al. [2004] introduced a dynamic conditional random field (DCRF), a conditionally-trained undirected sequence model with repeated graphical structure and tied parameters. Their model showed promising results on a natural language chunking task, performing parameter estimation using loopy belief propagation using a training dataset that contained no hidden node. However, parameter estimation and inference become difficult when the model is given a dataset that contains hidden nodes in its underlying structure [Morency et al. 2007].

Morency et al. [2007] presented a latent-dynamic conditional random field (LDCRF), an extension to DCRFs with a disjoint set of hidden state variables per label, capturing sub-structure of a class sequence and learning dynamics between class labels. By assuming disjoint set of hidden state variables, parameter estimation and inference is done efficiently using belief propagation [Pearl 1982]. They showed that the model is capable of learning both internal and external dynamics of class structure, demonstrating it on the task of spotting head or eye gestures from unsegmented video streams.

However, both DCRFs and LDCRFs still require an input sequence to be bounded, i.e., label prediction is done not successively at each time step, but once the whole sequence is given. These approaches are thus of limited use for real-time gesture understanding. In this work, we develop a two-layered heuristic approach that performs sequence segmentation and predicts gesture labels successively.

## 3. SYSTEM OVERVIEW

Fig. 1 shows an overview of our gesture recognition system. In the first part of the pipeline, image pre-processing, depth maps are calculated using images captured from a stereo camera, and the images are background subtracted using a combination of a codebook background model [Kim et al. 2005] and a "depth-cut" method.

For the second part, 3D body pose estimation, we construct a generative model of the human upper body and compare various features extracted from the model to corresponding features extracted from input image. In order to deal with body pose ambiguities that arise from self-occlusion, we examine both static and dynamic attributes of motion. Poses are then estimated in a multi-hypothesis Bayesian inference framework with a particle filter [Isard and Blake 1998].

For the third part, hand pose classification, we use information from body pose estimation to make the hand tracking task efficient: two small search regions are defined around estimated wrist joints, and our system searches for hands over these regions using a sliding window. A multi-class SVM classifier [Vapnik 1995] is trained off-line using manually-segmented images of hands, extracting HOG features [Dalal

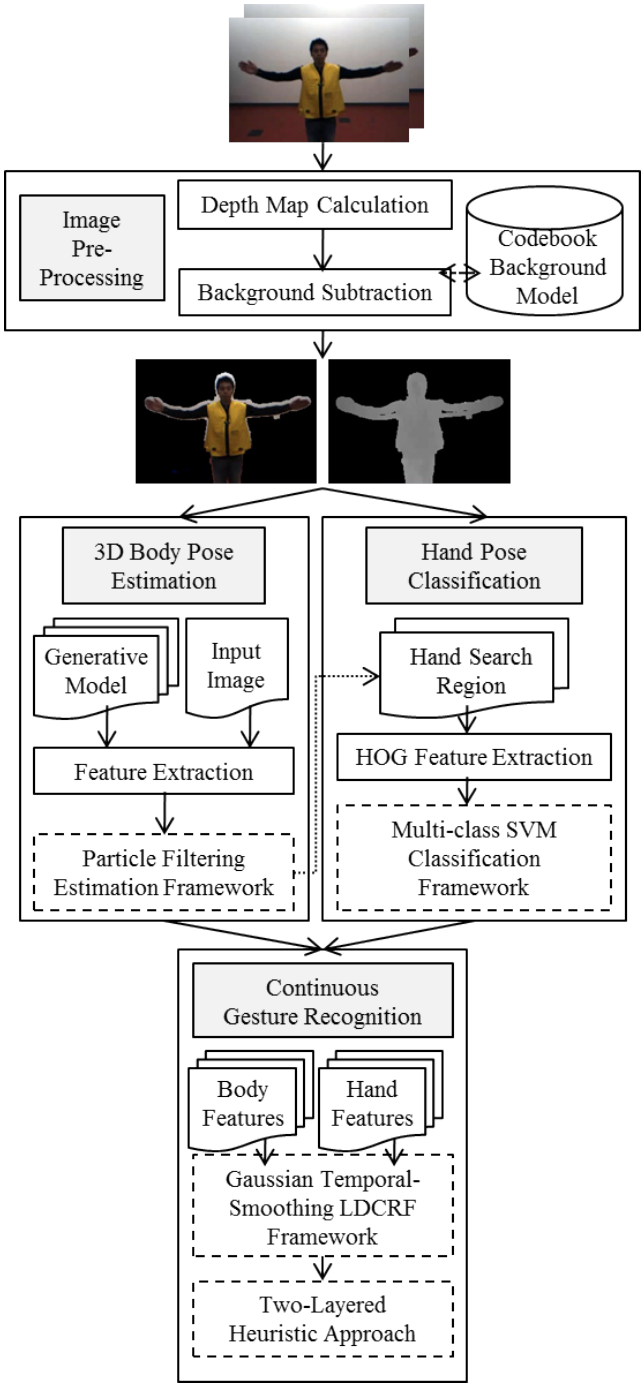


Fig. 1. A pipeline view of our continuous body and hand gesture recognition framework.

and Triggs 2005] from the images, and is then used to classify hand poses.

In the last part, continuous gesture recognition, we perform recognition by combining body and hand pose information. An LDCRF with a Gaussian temporal-smoothing kernel is trained off-line using a supervised body-and-hand gesture dataset, and is used with our two-layered heuristic approach to perform sequence segmentation and successive labeling, given unsegmented and unbounded input.

## 4. OBTAINING BODY AND HAND SIGNALS

In this section, we describe body and hand pose tracking, which receives input images from a stereo camera and produces body and hand signals by performing 3D body pose estimation and hand pose classification.

We describe image pre-processing part in Section 4.1, which produces depth maps and mask images. Then we describe 3D body pose estimation in Section 4.2 and hand pose classification in Section 4.3. We conclude this section with an evaluation of our pose tracking and the results from quantitative and qualitative analysis in Section 4.4.

### 4.1 Image Pre-Processing

The system starts by receiving pairs of time-synchronized images recorded from a Bumblebee2 stereo camera, producing 320 x 240 pixel resolution images at 20 FPS. While recording videos, the system produces depth maps and mask images in real-time. Depth maps allow us to reconstruct body poses in 3D space and resolve some of pose ambiguities arising from self-occlusion; mask images allows us to concentrate on the objects of interest and ignore the background, optimizing the use of available computational resources. We obtain depth maps using a manufacture-provided SDK. [1]

We obtain mask images by performing background subtraction. Ideally, background subtraction could be done using depth information alone by the "depth-cut" method: filter out pixels whose distance is further from camera than a foreground object, assuming there is no object inbetween the camera and the subject. However, as shown in Fig. 2, depth maps typically have lower resolution than color images, meaning the resolution of mask images produced are equally low resolution. This motivates our approach of performing background subtraction using a codebook approach [Kim et al. 2005], then refining the result with the depth-cut method.

The codebook approach works by learning a background model from a history of 2D color images of the background sampled over a period of time, then segmenting out the "outlier" pixels in an input image as foreground. For each pixel, the background model is learned by constructing a set of disjoint RGB intensity bounds, where each bound is determined by a newly observed value of the pixel. If the new value falls into or is close to one of the existing bounds, it is modeled as a perturbation on that bound, making the bound grow to cover the perturbation of values seen over time; otherwise, a new intensity bound is created and added to the set. The set of disjoint intensity bounds is called a codebook, and can be envisioned as several boxes located in RGB space, each box capturing a particular intensity range considered likely to be background.

[1] http://www.ptgrey.com

One weakness of the codebook approach is its sensitivity to shadows, arising because the codebook defines a foreground object as any set of pixels whose color values are noticeably different from the background. To remedy this, after input images are background subtracted using the codebook approach, we refine the result using the depth-cut method described above, which helps remove shadows created by a foreground object. Sample images of the videos are shown in Fig 2.

Fig. 2. Example images of input image (left), depth map (middle), and mask image (right). The "T-pose" shown in the figures is used for body tracking initialization.

### 4.2 3D Body Pose Estimation

The goal here is to reconstruct upper body pose in 3D space given the input images. We formulate this as a Bayesian inference problem, i.e., making an inference about a posterior state density $p(\mathbf{x} \mid \mathbf{z})$, having observed an input image $\mathbf{z}$ and knowing the prior density $p(\mathbf{x})$, where $\mathbf{x} = (x_1 \cdots x_k)^T$ is a vector representing the body pose we are estimating.

4.2.1 *Generative Model.* Our generative model of the human upper body is constructed in 3D space, using a skeletal model represented as a kinematic chain and a volumetric model described by superellipsoids [Barr 1981] (Fig. 3). The model includes 6 body parts (trunk, head, upper and lower arms for both sides) and 9 joints (chest, head, navel, left/right shoulder, elbow, and wrist). A shoulder is modeled as a 3 DOF ball-and-socket joint, an elbow is modeled as a 1 DOF revolute joint. Coordinates of each joint are obtained by solving the forward kinematics problem following the Denavit-Hartenberg convention [Denavit and Hartenberg 1955], a compact way of representing $n$-link kinematic structures. We prevent the model from generating anatomically implausible body poses by constraining joint angles to known physiological limits [NASA 1995].

The human shoulder has historically been the most challenging part for human body modeling [Engin 1980]. It has a complicated anatomical structure, with bones, muscles, skin, and ligaments intertwined, making modeling of the shoulder movement difficult.

Although having a high fidelity shoulder model is the basis for a successful body pose tracking, many approaches in the generative model-based body pose estimation sacrifice some accuracy for simplicity, usually modeling the shoulder as a single ball-and-socket joint. In the biomechanics community, there have been many approaches to more sophisticated shoulder models (see [Feng et al. 2008] for a survey), where most approaches have used a model with 5 to 9 DOF joints to model the shoulder

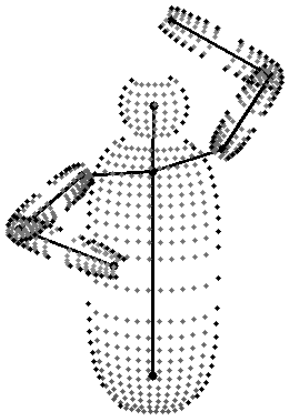

Fig. 3. Generative model of the human upper body with improved shoulder model.

accurately. Although these models offer high fidelity, a higher DOF model makes the body pose estimation problem more difficult.

We improve on our basic model of human upper body by building a more precise model of the shoulder, while not including additional DOFs. To capture arm movement more accurately, the shoulder model is approximated analytically by computing the angle $\varphi$ between the line from the mid-chest to the shoulder and the line from mid-chest to the elbow. The chest-to-shoulder angle $\theta^{CS}$ is then updated as

$$\theta^{CS'} = \begin{cases} \theta^{CS} + \frac{\varphi}{\theta^{CS}_{MAX}} & \text{if elbow is higher than shoulder} \\ \theta^{CS} - \frac{\varphi}{\theta^{CS}_{MIN}} & \text{otherwise} \end{cases} \tag{1}$$

where $\theta^{CS}_{min}$ and $\theta^{CS}_{max}$ are minimum and maximum joint angle limits for chest-to-shoulder joints [NASA 1995]. This simplified model only mimics shoulder movement in one-dimension, up and down, but works quite well if the subject is facing the camera, as is commonly true for human-computer interaction.

With these settings, an upper body pose is parameterized as

$$\mathbf{x} = (G\ R)^T \tag{2}$$

where $G$ is a 6 DOF global translation and rotation vector, and $R$ is a 8 DOF joint angle vector (3 for shoulder and 1 for elbow, for each arm).

4.2.2 *Particle Filter.* Human body movements can be highly unpredictable, so an inference framework that assumes its random variables form a single Gaussian distribution can fall into a local minima or completely loose track. A particle filter [Isard and Blake 1998] is particularly well suited to this type of task for its ability to represent the posterior state density $p(\mathbf{x} \mid \mathbf{z})$ as a multimodal non-Gaussian distribution. It maintains multiple hypotheses during inference, discarding less likely hypotheses only slowly.

We briefly review the particle filter to set the context for our work. Inference on $p(\mathbf{x}_t \mid \mathbf{Z}_t)$ (i.e., the probability of a pose $\mathbf{x}$ at time $t$, given a history of images $\mathbf{Z}_t =$

$\{\mathbf{z}_1, \cdots, \mathbf{z}_t\}$ up to that time) over discrete time steps is made with the following probability density propagation rule:

$$p(\mathbf{x}_t \mid \mathbf{Z}_t) = k_t \, p(\mathbf{z}_t \mid \mathbf{x}_t) \, p(\mathbf{x}_t \mid \mathbf{Z}_{t-1}) \tag{3}$$

where

$$p(\mathbf{x}_t \mid \mathbf{Z}_{t-1}) = \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \, p(\mathbf{x}_{t-1} \mid \mathbf{Z}_{t-1}) \tag{4}$$

and $k_t$ is a normalization constant that does not depend on $\mathbf{x}_t$. The conditional state density $p(\mathbf{x}_t \mid \mathbf{Z}_t)$ is approximated by a set of $N$ weighted particles: $\{\left(s_t^{(1)}, \pi_t^{(1)}\right), \cdots, \left(s_t^{(N)}, \pi_t^{(N)}\right)\}$, where each particle $s_t$ represents a pose configuration, and the weights $\pi_t^{(n)} = p(\mathbf{z}_t \mid \mathbf{x}_t = s_t^{(n)})$ are normalized so that $\sum_N \pi_t^{(n)} = 1$.

The dynamic model of joint angles is constructed as a Gaussian process:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + e, \quad e \sim \mathcal{N}(0, \sigma^2) \tag{5}$$

Once $N$ samples $s_t^{(1)}, \cdots, s_t^{(N)}$ are generated, we calculate an estimation result as the weighted mean of samples $s_t^{(n)}$:

$$\mathbb{E}\left[f(\mathbf{x}_t)\right] = \sum_{n=1}^{N} \pi_t^{(n)} \, f(s_t^{(n)}). \tag{6}$$

Iterative methods need a good initialization. We initialize the generative model at the first frame: The initial body pose configurations (i.e., joint angles and limb lengths) are obtained by having the subject assume a static "T-pose" (as shown in Fig. 2), and fitting the model to the image with exhaustive search.

4.2.3 *Likelihood Function.* The likelihood function $p(\mathbf{z}_t \mid \mathbf{x}_t = s_t^{(n)})$ is defined as an inverse of an exponentiated fitting error $\varepsilon(\mathbf{z}_t, s_t^{(n)})$:

$$p(\mathbf{z}_t \mid \mathbf{x}_t = s_t^{(n)}) = \frac{1}{\exp\left\{\varepsilon\left(\mathbf{z}_t, s_t^{(n)}\right)\right\}} \tag{7}$$

where the fitting error is computed by comparing three features extracted from the generative model to the corresponding ones extracted from input images: a 3D visible-surface point cloud, a 3D contour point cloud, and a motion history image (MHI) [Bobick and Davis 1996]. The first two features capture discrepancies in static poses; the third captures discrepancies in the dynamics of motion.

The first two features, 3D visible-surface and contour point clouds, are used frequently in body motion tracking (e.g., [Deutscher et al. 2000]), for their ability to evaluate how well the generated body pose fits the actual pose observed in image. We measure the fitting errors by computing the sum-of-squared Euclidean distance errors between the point cloud of the model and the point cloud of the input image (i.e., the 3D data supplied by the image pre-processing step described above).

The third feature, an MHI, is an image where each pixel value is a function of the recency of motion in a sequence of images (Fig. 4). This often gives us useful

information about dynamics of motion, as it indicates where and how the motion has occurred. We measure discrepancies in the dynamics of motion by comparing the MHI of the model and the MHI of the input image.

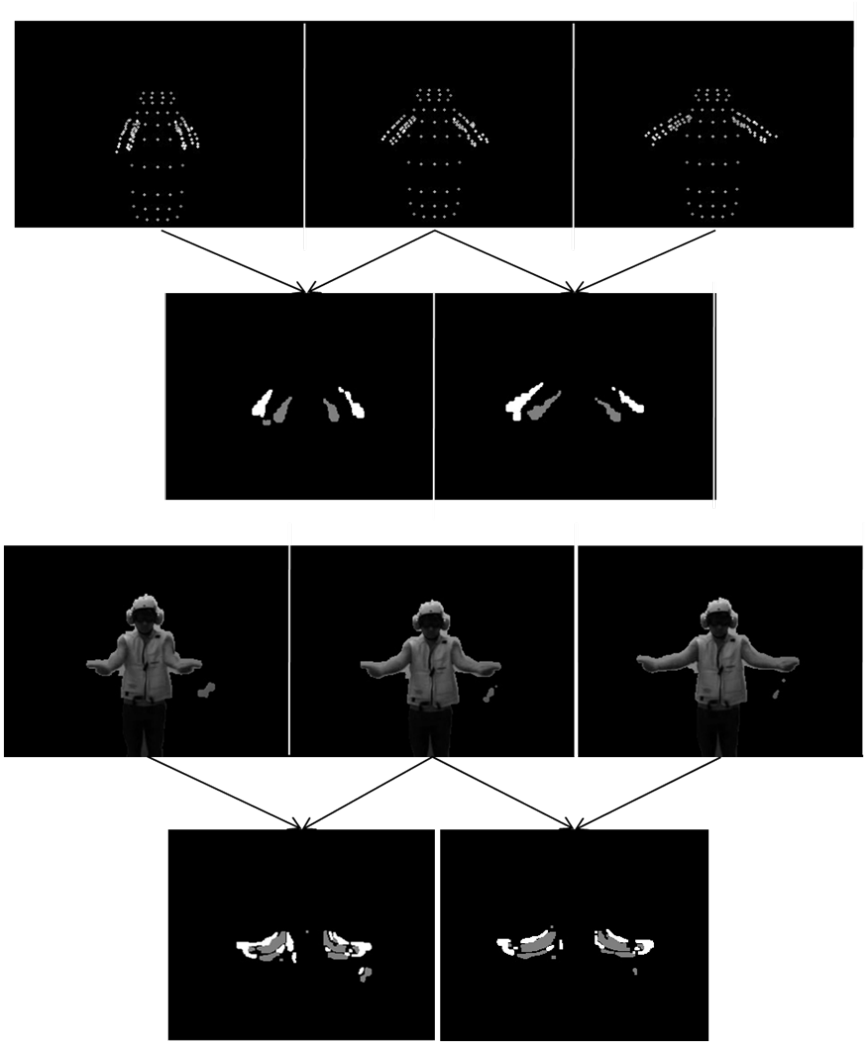

Fig. 4. MHIs of the model (top) and the observation (bottom).

An MHI is computed from $I_{t-1}$ and $I_t$, two time-consecutive 8-bit unsigned integer images whose pixel values span 0 to 255. For the generative model, $I_t$ is obtained by rendering the model generated by a particle $s_t^{(n)}$ (i.e., rendering an image of what body pose $s_t^{(n)}$ would look), and $I_{t-1}$ is obtained by rendering the model generated by the previous step's estimation result $\mathbb{E}[f(\mathbf{x}_t)]$ (Eq. 6). For the input images, $I_t$ is obtained by converting an RGB input image to YCrCb color

space and extracting the brightness channel (Y)[2], and this is stored to be used as $I_{t-1}$ for the next time step. Then an MHI is computed as

$$I_{MHI} = \text{thresh}(I_{t-1} - I_t, 0, 127) + \text{thresh}(I_t - I_{t-1}, 0, 255) \tag{8}$$

where $\text{thresh}(I, \alpha, \beta)$ is a binary threshold operator that sets each pixel value to $\beta$ if $I(x, y) > \alpha$, and set to zero otherwise. The first term captures pixels that were occupied at the previous time step but not in the current time step. The second term captures pixels that are newly occupied in the current time step. The values 127 and 255 are chosen to indicate the time information of those pixels: 0 means there has been no change in the pixel, regardless of whether or not there was an object; 127 means there was an object in the pixel but it has moved; while 255 means an object has appeared in the pixel. This allows us to construct an image that concentrates on only the moved regions (e.g., arms), while ignoring the unmoved parts (e.g., trunk, background). The computed MHI images are visualized in Fig. 4.

Finally, an MHI error is computed as

$$\varepsilon_{MHI} = \mathbb{C}\text{ount}\,[\,\text{thresh}(I', 127, 255)\,] \tag{9}$$

where

$$I' = \text{abs}\left(I_{MHI}(\mathbf{z}_t, \mathbf{z}_{t-1}) - I_{MHI}(s_t^{(n)}, \mathbb{E}\,[f(\mathbf{x}_t)])\right) \tag{10}$$

This error function first subtracts an MHI of the model $I_{MHI}(s_t^{(n)}, \mathbb{E}[f(\mathbf{x}_t)])$ from an MHI of the observation $I_{MHI}(\mathbf{z}_t, \mathbf{z}_{t-1})$, and computes an absolute-valued image of it (Eq. 10). Then it applies the binary threshold operator with the cutoff value and result value noted above, and counts non-zero pixels with $\mathbb{C}\text{ount}\,[\cdot]$ (Eq. 9). The intuition behind setting the cutoff value to 127 can be found in Table I: the shaded conditions ($I'(x, y) > 127$) represent errors at the current time-step, where by error we mean that the pixel values of two MHIs do not agree. Note that we want to penalize the conditions in which two MHIs do not match at the current time-step, independent of the situation at the previous time-step. The cutoff value 127 does this, efficiently capturing the motion errors.

4.2.4 *Output Feature Types.* We get four types of features from body pose estimation: joint angles and joint angular velocities, and uniform-length joint coordinates and velocities. Joint angles are 8 DOF vectors (3 for shoulder and 1 for elbow, for each arm) obtained directly from the estimation. To obtain uniform-length relative joint coordinates, we first generate a model with the estimated joint angles and fixed-length limbs, so that all generated models have the same set of limb lengths across subjects. This results in 12 DOF vectors (3D coordinates of elbows and wrists for both arms) obtained by logging global joint coordinates relative to the chest joint. The uniform length model allows us to reduce cross-subject variances. Joint angular velocities and uniform joint velocities are calculated by taking derivatives of joint angles and uniform-length relative joint coordinates.

[2]Empirically, most of the variation in images is better represented along the brightness axis, not the color axis [Bradski and Kaehler 2008].

| $I_{MHI}(\mathbf{z}_t, \mathbf{z}_{t-1})$ | $I_{MHI}\left(s_t^{(n)}, \mathbb{E}[f(\mathbf{x}_{t-1})]\right)$ | $I'$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 127 | 127 |
| 0 | 255 | 255 |
| 127 | 0 | 127 |
| 127 | 127 | 0 |
| 127 | 255 | 128 |
| 255 | 0 | 255 |
| 255 | 127 | 128 |
| 255 | 255 | 0 |

Table I. Possible conditions for computing $\varepsilon_{MHI}\left(I_{MHI}(\mathbf{z}_t, \mathbf{z}_{t-1}),\ I_{MHI}\left(s_t^{(n)}, \mathbb{E}[f(x_{t-1})]\right)\right)$. Note that, for the first two columns, the value 0 means there has been no object in the pixel, the value 127 means there was an object in the pixel but it has moved, and the value 255 means there is an object. Therefore, by thresholding the absolute subtracted values with the cutoff value 127, we can ignore the mistakes happened at $t-1$ and concentrate on the mistakes that happened at time $t$.

### 4.3 Hand Pose Classification

The goal of hand pose classification is to classify hand poses made contemporaneously with gestures. We selected four canonical hand poses (thumb up and down, opened and closed hand) that are often used in hand signals (e.g., NATOPS gestures [Song et al. 2011b]) (Fig. 5). As searching for hands in an entire image can be time-consuming, we use the body pose estimation result to narrow down the search region, dramatically reducing processing time.

Fig. 5. Four canonical hand poses defined in this work (thumb up and down, opened and closed hand), selected from the NATOPS database [Song et al. 2011b].

4.3.1 *Training Dataset.* A training dataset was collected from the NATOPS database we created [Song et al. 2011b], choosing the recorded video clips of the first 10 participants (out of 20). Positive samples were collected by manually selecting 32 x 32 pixel images that contained hands and labeling them; negative samples were collected automatically after collecting positive samples, by choosing two random foreground locations and cropping the same-sized images. We applied affine

transformations to the positive samples, to make the classifier more robust to scaling and rotational variations, and to increase and balance the number of samples across hand pose classes. After applying the transformations, the size of each class was balanced at about 12,000 samples.

4.3.2 *HOG Features.* HOG features [Dalal and Triggs 2005] are image descriptors based on dense and overlapping encoding of image regions. The central assumption of the method is that the appearance of an object is rather well characterized by locally collected distributions of intensity gradients or edge orientations, even without having the knowledge about the corresponding gradient or edge positions that are globally collected over the image.

HOG features are computed by dividing an image window into a grid of small regions (cells), then producing a histogram of the gradients in each cell. To make the features less sensitive to illumination and shadowing effects, the same image window is also divided into a grid of larger regions (blocks), and all the cell histograms within a block are accumulated for normalization. The histograms over the normalized blocks are referred to as HOG features. We used a cell size of 4 x 4 pixels, block size of 2 x 2 cells, window size of 32 x 32 pixels, with 9 orientation bins. Fig. 6 shows a visualization of the computed HOG features.

Fig. 6. Four hand poses and a visualization of their HOG features. Bright spots in the visualization indicate places in the image that have sharp gradients at a particular orientation, e.g., the four vertical orientation in the first visualization.

4.3.3 *Multi-Class SVM Classifier.* To classify the HOG features, we trained a multi-class SVM classifier [Vapnik 1995] using an existing library (LIBSVM [Chang and Lin 2001]). Since HOG features are high dimensional, we used an RBF kernel to transform input data to the high-dimensional feature space. We trained a multi-class SVM following the one-against-one method [Knerr et al. 1990] for fast training, while obtaining comparable accuracy to one-against-all method [Hsu and Lin 2002]. We performed grid search and 10-fold cross validation for parameter selection.

4.3.4 *Search Region.* We use the information about wrist position computed in body pose estimation to constrain the search for hands in the image. We create a small search region around each of the estimated wrist positions, slightly larger than the average size of an actual hand image, and compute the HOG features in that region using a sliding window. Estimated wrist positions are of course not

always accurate, so a search region might not contain a hand. We compensate for this by including information on hand location from the previous step's hand pose classification result. If a hand is found at time $t-1$, for time $t$ we center the search region at the geometric mean of the estimated wrist position and the hand position at time $t-1$. Our search region was 56 x 56 pixels; the sliding window was 32 x 32 pixels.

4.3.5 *Clustering.* Each time a sliding window moves to a new position within a search region, the HOG features are computed, and the SVM classifier examines them, returning a vector of $k+1$ probability estimates ($k$ hand classes plus one negative class). We thus get multiple classification results per search region, with one from each window position. To get a single classification per search region, we cluster all classification results within the region, averaging positions and probability estimates of the classification results (Fig. 7).

Fig. 7. Search regions around estimated wrist positions (black rectangles). Colored rectangles are clustered results (blue/red: palm open/close), and small circles are individual classification results.

4.3.6 *Output Feature Type.* We get two types of features from hand pose classification: a soft decision and a hard decision. The soft decision is an 8 DOF vector of probability estimates obtained from the SVM (four hand classes for each hand); the hard decision is a 2 DOF vector of hand labels obtained from the soft decision, selecting a label with the highest probability for each hand.

## 4.4 Evaluation

In evaluating any system we need to consider the system's task. For example, pose tracking used for controlling a 3D avatar, as in computer animation film making, requires tracking that matches the input at the pixel level. However, if tracking results are used for other higher-level tasks, such as gesture understanding, as in this work, a coarser-grained comparison will also be useful.

In keeping with this, we evaluate our system's body pose estimation and hand pose classification in two ways :using a quantitative analysis that evaluates accuracy at the pixel level, and a qualitative analysis done by visually comparing tracking results to the input. We selected 10 body-and-hand gestures (Fig. 14) from the NATOPS database [Song et al. 2011b] that we believe well represent the challenges

posed by the 24 gesture set. [3]

4.4.1 *Body Pose Estimation.* Body pose estimation was performed with 500 particles, taking about 0.7 seconds to estimate each frame on an Intel Xeon Dual Core 2.66 GHz machine with 3.25GB of RAM.

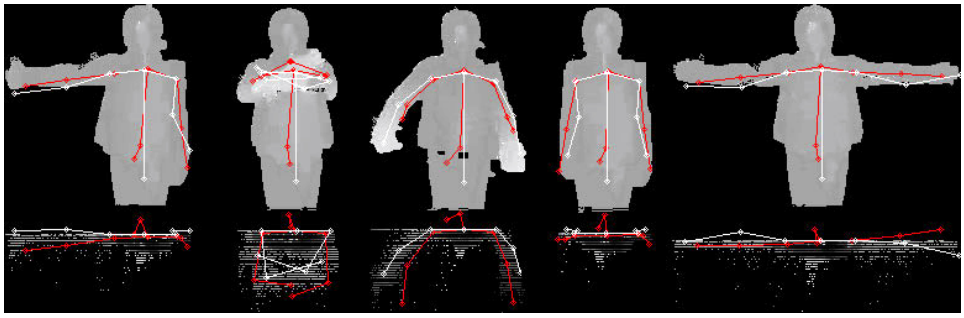

Fig. 8. Vicon ground-truth data (red lines) superimposed onto depth maps with estimation results (white lines).

We evaluated our system's accuracy quantitatively using ground-truth data collected using the Vicon motion capture system.[4] To get the ground-truth data, one participant was selected and recorded using both a stereo camera and the Vicon system simultaneously. The Vicon ground-truth body poses were superimposed onto the input images, scaled and translated properly so that they aligned with the coordinate system of the estimated body pose (Fig. 8). Pixel displacement errors were then calculated and accumulated for each joint, providing a measure of total pixel error. The result is shown in Fig. 9. In a 320 x 240 pixel frame, the average pixel error per frame was 29.27, with a lower error for 2D gestures (where arms were in the same plane as the body, mean = 24.32 pixels) and higher for 3D gestures (mean = 34.20 pixels).

We also performed a qualitative analysis, visually comparing the estimation results to the actual body poses in input images, frame-by-frame, counting the number of erroneous frames. We randomly selected one trial sequence (out of 20) for each gesture and checked even-numbered frames in the sequences. An estimation result was regarded as an error if it looked visually different from the actual body pose (i.e., an arm pointed to a position that is more than 45 degree off from the original one) or was anatomically unreasonable (i.e., an arm twisted abnormally). Displacements due to noise (i.e., an arm slightly shaking frame to frame) or near-misses (i.e., the estimated arm position did not overlap perfectly to the actual position although body configuration was highly similar visually) were not counted as an error. The result is shown in Table II. The overall accuracy was 92.28%.

[3]We give detailed description of the 10 gestures in Section 6.1.

[4]We used the Vicon motion capture system from the MIT CSAIL holodeck room (16 cameras, 120 Hz, 1mm precision).

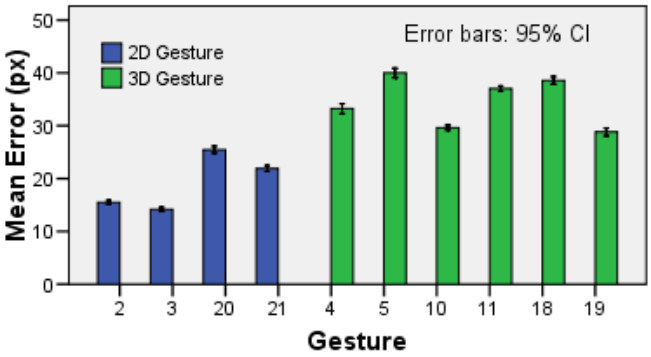


Fig. 9. Measures of total pixel errors for body pose estimation.

| Gesture Index | Description | Accuracy |
|---|---|---|
| #2 and #3 | All/not clear | 92.68% |
| #4 and #5 | Spread/fold wings | 90.19% |
| #10 and #11 | Remove/insert chocks | 97.71% |
| #18 and #19 | Engage nosegear steering / Hot brakes | 93.22% |
| #20 and #21 | Brakes on/off | 87.64% |

Table II. A qualitative analysis result for 3D body pose estimation.

In general, more errors occurred on the gestures that included self-occluded body poses. Gesture #4 (spread wings) (Fig. 10), for example, contained body poses where both arms were located in front of the body at a close distance: it started with making a shrugging gesture, with both arms kept close to the body and moving to the chest. A close look at the estimation result revealed that most errors on this gesture occurred during tracking the shrugging part of the gesture. Note that two of the features we extracted (3D visible surface and contour point cloud) were highly related to the depth information. Therefore, we can expect that the estimation accuracy may be degraded if the depth of an arm does not differ noticeably from the rest of the body (as in the case of shrugging gesture).

Also, error rates of gesture #20 (brakes on) and #21 (brakes off) (Fig. 11) were, although still low, relatively higher than others. Note that, these two gestures included raising both arms outwards (making the T-pose) and bending both elbows so that both hands point upward. A majority of errors on these two gestures were that the estimated bending direction of elbows were opposite to the actual one.

4.4.2 *Hand Pose Classification.* When tested with a 10-fold cross validation on pre-segmented images of hands, the trained SVM hand pose classifier gave near-perfect accuracy (99.94%). However, what matters more is how well the classifier performs on unsegmented video input. To explore this, we randomly selected a subset of full image frames from four gestures (i.e., #2, #3, #20, and #21) that

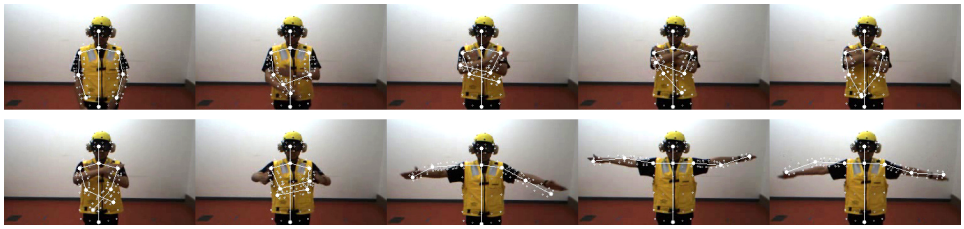

Fig. 10. Gesture #4 (spread wings) contains a shrugging gesture, which causes a major portion of estimation errors for this gesture. The pose estimator tracks the shrugging gesture correctly for a few frames at the beginning, but fails when arms get too close to the body. Note that it quickly finds the correct pose when there is no more self-occlusion.

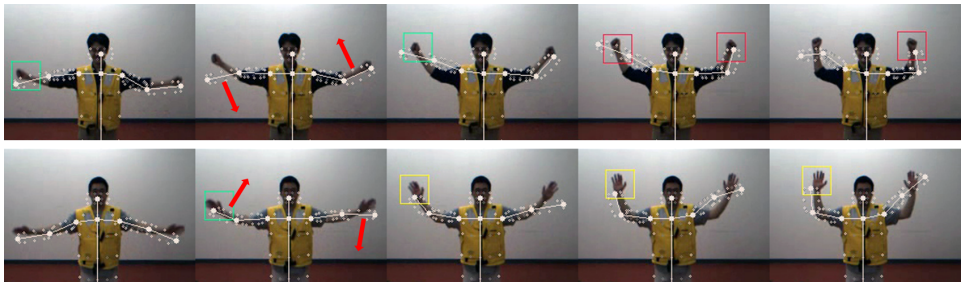

Fig. 11. Estimation for Gesture #21 (brakes off) can fail when the bending direction of an elbow (red arrows) is incorrect. (Rectangles around hands are hand pose estimation result).

| Gesture Index | Missed | Misclassified | Total | Accuracy |
|---|---|---|---|---|
| #2 and #3 | 23 | 11 | 747 | 95.45% |
| #20 and #21 | 394 | 0 | 1591 | 75.24% |

Table III. A qualitative analysis result for hand pose classification.

contained the canonical hand poses. After classification was performed, the results were overlaid on the original images, allowing us to visually compare the classification result to the ground-truth labels (i.e., actual hand poses in the images). We counted the number of misses (no classified result although there was a hand pose in an image) and misclassifications (classified result did not match the actual one), and combined them to obtain the number of erroneous classification results. For the simplicity, we used hard decision values.

The result is shown in Table III. Not surprisingly, gesture #20 and #21 were the most difficult. A majority of the errors here were due to rotational variations and imperfect body pose estimation results, i.e., search regions that did not include hand images. This indicates that the significant speed advantage of using estimated wrist position, in some cases, decreases hand detection accuracy.

## 5. GESTURE RECOGNITION

The goal here is to predict gesture labels given an unsegmented and unbounded temporal input sequence that consists of body and hand poses, obtained from 3D video data. For each image $\mathbf{z}_t$, we extract body pose features $\phi(\mathbf{x}_t^1) \in \mathbb{R}^{N_1}$ (Section 4.2) and hand pose features $\phi(\mathbf{x}_t^2) \in \mathbb{R}^{N_2}$ (Section 4.3); that is, each $\mathbf{x}_t$ is represented as a dual-signal feature-vector

$$\phi(\mathbf{x}_t) = \left(\phi(\mathbf{x}_t^1)\ \phi(\mathbf{x}_t^2)\right)^T. \tag{11}$$

We use a discriminative hidden-state approach to learn patterns of body-and-hand gestures. The approach has recently shown promising results in many pattern recognition tasks [Lafferty et al. 2001; Quattoni et al. 2005; Sutton et al. 2004; Gunawardana et al. 2005; Wang et al. 2006; Morency et al. 2007].

As discussed earlier, the main advantage of the discriminative approach compared to the generative approach is that they do not make the conditional independence assumption, which is often both too restrictive and unrealistic. It has been shown that when conditional independence does not hold, the asymptotic accuracy of discriminative model is higher than generative models [Mitchell 1997]. In our task, the input signal patterns tend to exhibit long-range temporal-dependencies (e.g., body parts move coherently as time proceeds, hand poses are articulated in relation to body poses in a time-sequence, etc.). Thus, although a gesture label is given, individual observations may not be independent of each other; observations rather seem to be important clues to distinguish similar patterns of gestures.

The task of recognizing gestures from continuous input is more challenging. Therefore, we first developed isolated gesture recognition and explored various issues that arise from body and hand gesture recognition (Section 5.1). We then developed continuous gesture recognition to extend the framework for unsegmented and unbounded input (Section 5.2).

In the following sections, we describe the two versions of gesture recognition, isolated and continuous, with the focus on explaining the use of a Gaussian temporal-smoothing kernel and our two-layered heuristic approach to continuous gesture recognition.

### 5.1 Isolated Gesture Recognition

This section describes isolated gesture recognition using a Gaussian temporal-smoothing HCRF model.

5.1.1 *HCRFs: A review.* An HCRF [Quattoni et al. 2005] is a discriminative framework for building probabilistic models to label segmented sequential data (i.e., data that has been divided at signal boundaries, such as gesture start and end). The framework extends CRF models [Lafferty et al. 2001], which assumes a tree-structured undirected graph $G$, by incorporating hidden state variables into the graphical structure. The framework is designed to capture complex dependencies in observations efficiently, without attempting to specify exact conditional dependencies. The goal is to learn a mapping function of observations $\mathbf{x}$ to class labels $y \in \mathcal{Y}$, by introducing hidden state variables $h \in \mathcal{H}$ to compactly represent the distribution of observations. The conditional probability distribution $p(y \mid \mathbf{x};\, \theta)$ of

a class label $y$ given a set of observation $\mathbf{x}$ with parameter vector $\theta$ is constructed as

$$p(y \mid \mathbf{x};\ \theta) = \sum_{\mathbf{h}} p(y, \mathbf{h} \mid \mathbf{x};\ \theta) = \frac{1}{Z} \sum_{\mathbf{h}} e^{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)} \tag{12}$$

where $Z$ is a *partition function* defined as

$$Z = \sum_{y \in \mathcal{Y}} \sum_{\mathbf{h}} p(y, \mathbf{h} \mid \mathbf{x};\ \theta) \tag{13}$$

and $\Psi(y, \mathbf{h}, \mathbf{x}; \theta)$ is a *potential function* defined as

$$\Psi(y, \mathbf{h}, \mathbf{x}; \theta) = \sum_{v \in V} \theta_V \cdot f\left(v, \mathbf{h}|_v, y, \mathbf{x}\right) + \sum_{(i,j) \in E} \theta_E \cdot f\left((i,j), \mathbf{h}|_{(i,j)}, y, \mathbf{x}\right). \tag{14}$$

The potential function models dependencies in the graphical structure, where $\theta_V$ and $\theta_E$ are parameters that determine dependencies within $\mathbf{h}|_S$, a set of components of $\mathbf{h}$ associated with the vertices and edges in subgraph $S$ of $G$.

Following previous work on CRFs [Lafferty et al. 2001], parameter optimization is performed using:

$$L(\theta) = \sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i;\ \theta)) - \frac{1}{2\sigma^2} ||\theta||^2 \tag{15}$$

where the second term, the *regularization factor*, is introduced to prevent overfitting of the training data. The optimal parameter values are obtained by solving the maximum log-likelihood function $\theta^* = \arg\max_\theta L(\theta)$ using belief propagation [Pearl 1982]. Finally, a class label for a new observation is determined as

$$y^* = \arg\max_{y \in \mathcal{Y}} p(y \mid \mathbf{x};\ \theta). \tag{16}$$

5.1.2 *Gaussian Temporal-Smoothing Kernel.* Because we obtain body and hand pose signals from statistical estimation and classification, the signals often exhibit high-frequency fluctuations due to noisy data and statistical fluctuations. Also, in our task the signals exhibit long-range temporal-dependencies, as gestures are articulated over a long sequence of frames. [5]

In order to capture the long-range dependencies as well as increase the SNR, we perform temporal smoothing over the signals. A variety of temporal smoothing techniques have been used to increase the SNR; in this work, we turn to Gaussian temporal-smoothing.

We incorporated a Gaussian temporal-smoothing kernel into the potential function in the HCRF formulation, defining the potential function as

$$\Psi(y, \mathbf{h}, \mathbf{x}; \theta) = \sum_{t} K\left(\phi(\mathbf{x}), g(\omega), t\right) \cdot \theta(h_t) + \sum_{t} \theta(y, h_t) + \sum_{t-1,t} \theta(y, h_{t-1}, h_t) \tag{17}$$

where $K\left(\phi(\mathbf{x}), g(\omega), t\right)$ is a Gaussian temporal-smoothing kernel; $\phi(\mathbf{x})$ is a feature vector obtained as Eq. 11; $g(\omega)$ is a Gaussian window with the size $\omega$.

[5]Gestures in the NATOPS database [Song et al. 2011b] lasted 2.3 seconds on average (46 frames with images recorded at 20 FPS).

The first term in Eq. 17 captures dependencies between the temporally smoothed input feature vectors and hidden state variables; the second term captures dependencies between class labels and hidden states variables; and the last term captures dependencies among class labels and two time-consecutive hidden state variables.

The Gaussian kernel performs a convolution of the input feature vector with a normalized $\omega$-point Gaussian window vector, computed from

$$g(\omega)[n] = e^{-\frac{1}{2}\left(\alpha \frac{n}{\omega/2}\right)^2} \tag{18}$$

where $-\frac{\omega-1}{2} \leq n \leq \frac{\omega-1}{2}$, and $\alpha$ is inversely proportional to the standard deviation of a Gaussian random variable. [6]

Intuitively, the kernel computes for each time frame a weighted mean of $\omega$ neighboring feature vectors with a Gaussian filter; thus the computed feature vector at each time frame incorporates long-range observations as well as reduces signal noise.

### 5.2 Continuous Gesture Recognition

In this section we describe continuous gesture recognition for unsegmented and unbounded input using an LDCRF model, using the Gaussian temporal-smoothing kernel described above and our two-layered heuristic approach for successive gesture labeling.

5.2.1 *LDCRFs: A Review.* An LDCRF [Morency et al. 2007] is a discriminative framework for simultaneous sequence segmentation and labeling. The conditional probability distribution $p(\mathbf{y} \mid \mathbf{x};\, \theta)$ of a label sequence $\mathbf{y} = \{y_1, \cdots, y_t\}$ given an observation sequence $\mathbf{x} = \{\mathsf{x}_1, \cdots, \mathsf{x}_t\}$ with parameter vector $\theta$ is constructed as

$$p(\mathbf{y} \mid \mathbf{x};\, \theta) = \sum_{\mathbf{h}} p(\mathbf{y} \mid \mathbf{h}, \mathbf{x};\, \theta)\, p(\mathbf{h} \mid \mathbf{x};\, \theta). \tag{19}$$

In order to make the computation tractable, LDCRFs assume a disjoint set of hidden state variables $h_j \in \mathcal{H}_{y_j}$ per class label $y_j$, which makes $p(\mathbf{y} \mid \mathbf{h}, \mathbf{x};\, \theta) = 0$ for $h_j \notin \mathcal{H}_{y_j}$. Therefore, Eq. 19 becomes

$$p(\mathbf{y} \mid \mathbf{x};\, \theta) = \sum_{\mathbf{h}:\forall h_j \in \mathcal{H}_{y_j}} p(\mathbf{h} \mid \mathbf{x};\, \theta). \tag{20}$$

Inference is performed on a per-frame basis by finding the most probable label $y_t^*$ for each frame, given an input sequence $\mathbf{x}$,

$$y_t^* = \arg\max_{y_j \in \mathcal{Y}} \sum_{a \in \mathcal{H}_{y_j}} p(h_t = a \mid \mathbf{x};\, \theta^*). \tag{21}$$

5.2.2 *The Heuristic Approach.* As discussed earlier, continuous gesture recognition from unsegmented and unbounded input sequence is a challenging task, because we do not know where are the boundaries of each gesture (i.e., unsegmented) as well as when the whole sequence ends (i.e., unbounded).

[6] Following [Harris 1978], we set $\alpha$=2.5.

LDCRFs learn both internal sub-structures and external dynamics of class labels, and thus are able to predict class labels on a per-frame basis given an unsegmented temporal input sequence. The model then performs sequence segmentation based on the predicted labels, by looking at label discontinuities within the sequence.

Regardless of whether the sequence is segmented or unsegmented, however, LDCRFs presume that a sequence to evaluate is bounded (i.e, the whole sequence is already given), which is a somewhat restricting assumption limiting its use in real-time applications. Ideally, the model should be able to predict labels successively as each new observation is made. Also, since sequence segmentation in LDCRFs is done based on prediction results, sequence labeling errors can lead to incorrect segmentation, making it necessary to use additional mechanisms to reduce the noise that labeling inaccuracy introduces.

In order to make LDCRFs predict labels successively and perform segmentation more accurately, we developed a two-layered heuristic approach that works for unsegmented and unbounded input. The method sets a fixed-sized sliding window, predicting a label for each successive individual frame, using information about all previous prediction results.

We use the terms "local" prediction and "global" prediction to introduce the concept of using two layers: the local prediction is made at the first layer using weighted averaging, while the global prediction is made at the second layer, based on the local prediction results, using exponential smoothing. Our two-layered heuristic approach is illustrated in Fig. 12 and described below.

5.2.3 *Successive Sequence Labeling.* At each time $t$, a $k$-point window slides forward, and an LDCRF evaluates a sequence of $k$ frames $\mathbf{x}_t = \{\mathsf{x}_{j=t-k+1}, \cdots, \mathsf{x}_t\}$ to predict a label sequence $\mathbf{y}_t = \{y_j, \cdots, y_t\}$, by computing $p_t(\mathbf{y}_t \mid \mathbf{x}_t; \theta^*)$ using Eq. 20. The prediction result can be viewed as a $|\mathcal{Y}|$-by-$k$ matrix, where each column vector $p_t(y_i \mid \mathbf{x}_t; \theta^*)$ is a probability estimate of all class labels for the $i$-th frame ($t - k + 1 \leq i \leq t$).

At the first layer, a local prediction $\bar{p}_t(y_j)$ is made for the first frame $\mathsf{x}_j$ in the window (i.e., the tail edge) by computing a weighted average of $k$ previous LDCRF prediction results for that frame $\mathsf{x}_j$ using a weight vector $\gamma$, [7]

$$\bar{p}_t(y_j) = \sum_{i=1}^{k} \gamma_i \cdot \, p_{t-i+1}(y_j \mid \mathbf{x}_{t-i+1}; \theta^*). \tag{22}$$

We have experimented with two weight functions: a uniform weight function ($\gamma_i = 1$) and a Gaussian weight function obtained using Eq. 18, where the weights are normalized so that $\sum_{i=1}^{k} \gamma_i = 1$. The uniform weight function performed slightly better than the Gaussian weight function, although the difference was negligible.

At the second layer, a global prediction $q_t(y_j)$ is made based on the local prediction results using exponential smoothing. This is based on the intuition that gestures change fairly slowly, i.e., over dozens of frames rather than frame to frame. Exponential smoothing is a technique that can be applied to time series data, often to produce smoothed data from noisy input signals. We compute the global

[7] Since the size of the window is $k$, each frame is evaluated $k$ times.

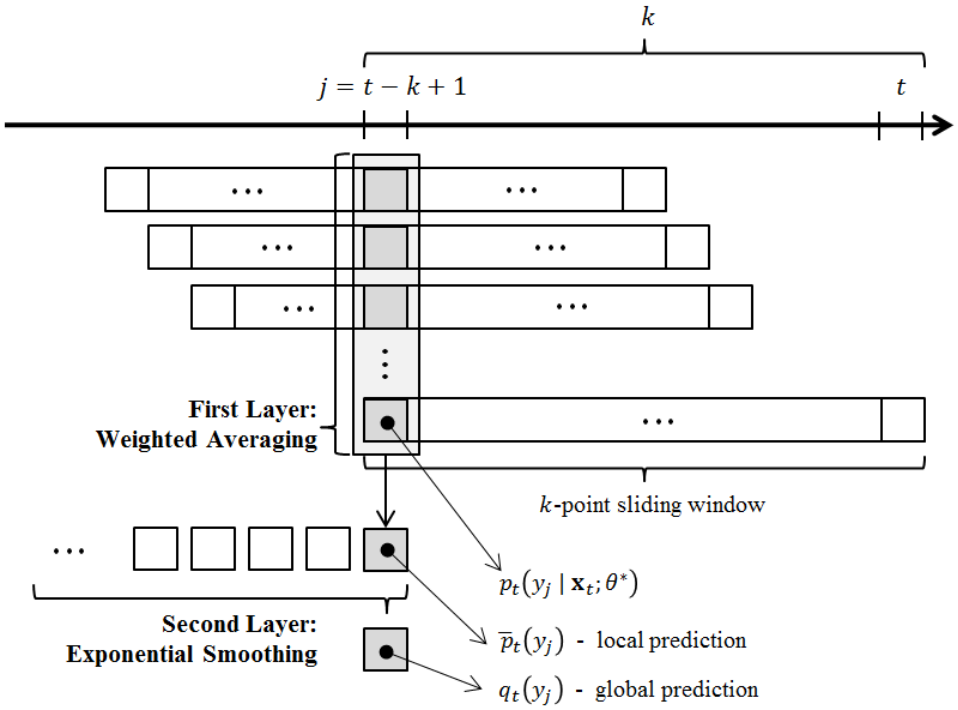


Fig. 12. A graphical illustration of the two-layered heuristic approach for continuous gesture recognition. As a $k$-point window slides forward, each individual frame is evaluated $k$ times using an LDCRF model. At each time $t$, a label for the first frame in the window $\mathbf{x}_{j=t-k+1}$ is predicted based on the $k$ previous LDCRF prediction results, using weighted averaging and exponential smoothing.

prediction $q_t(y_j)$ as

$$q_t(y_j) = \alpha \cdot \bar{p}_t(y_j) \; + \; (1-\alpha) \cdot q_{t-1}(y_{j-1}) \tag{23}$$

where $\alpha$ is a smoothing factor, which determines the level of smoothing (larger values of $\alpha$ reduce the level of smoothing). We set the smoothing factor adaptively, to the highest probability value in the local prediction result

$$\alpha = \max \bar{p}_t(y_j), \tag{24}$$

so that the smoothing takes into account how confident the current local prediction is (the more confident a local prediction is, the less smoothing performed).

Finally, label prediction is done by selecting a label with the highest probability

$$y_j^* = \arg\max_{y' \in \mathcal{Y}} q_t(y_j = y'). \tag{25}$$

5.2.4 *Sequence Segmentation.* Sequence segmentation is performed using the same method an LDCRF uses (i.e., a segment is determined when the predicted label changes over time). One difference is that our segmentation is performed using

a history of global prediction results $q_t(y_j)$ instead of using an LDCRF's prediction result.

Fig. 13 shows a comparison of two segmentation results using our method and an LDCRF. As seen in the figure, an LDCRF's prediction result fluctuates over time, while our method changes slowly, yielding a more robust sequence segmentation.

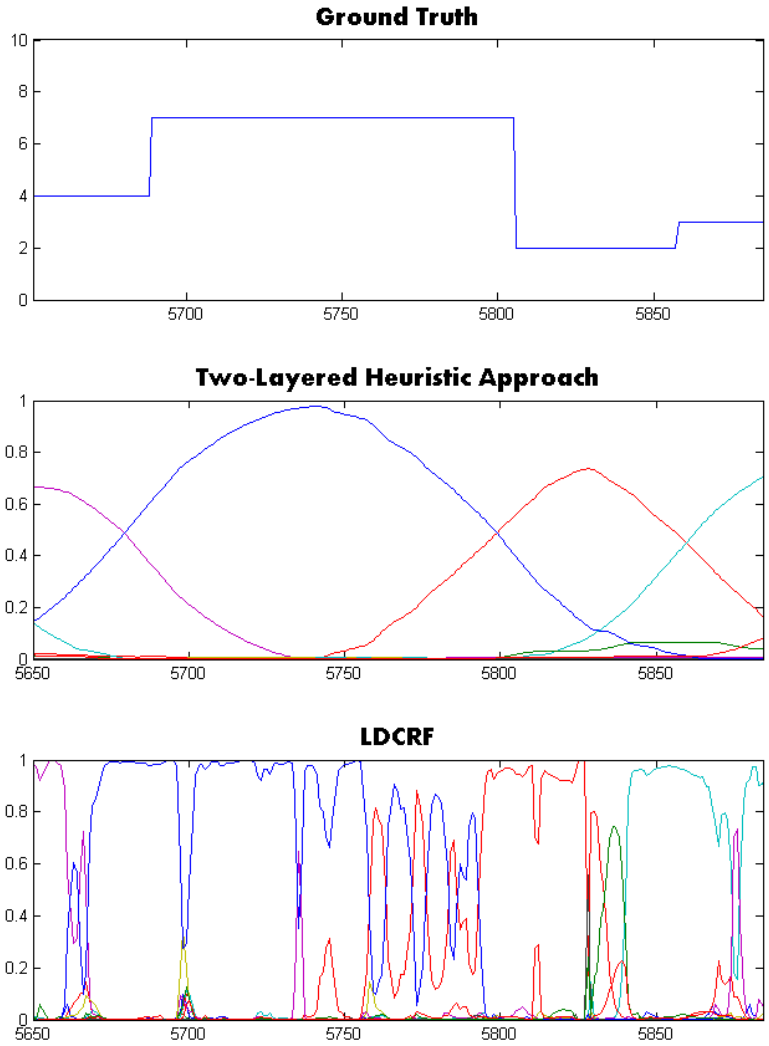


Fig. 13. A sequence of ground truth labels (top), and probability distributions obtained from our two-layered heuristic approach (middle) and from an LDCRF (bottom). As seen in the bottom graph, sequence segmentation is not accurate when done with LDCRF prediction results (especially from frame 5750 to frame 5800) when compared to the segmentation result with our two-layered heuristic approach.

## 6. EXPERIMENT

We conducted a variety of experiments to evaluate the performance of the Gaussian temporal smoothing and the two-layered heuristic approach, as well as exploring various issues arising in continuous body and hand gesture recognition. We focused in particular on the following issues: (1) whether combining body and hand signals improves accuracy; (2) which body and hand features are most informative; (3) whether performing Gaussian temporal-smoothing improves recognition performance; and (4) whether our two-layered heuristic approach improves recognition performance. The experiments were performed with both isolated gestures and continuous gestures, using models trained with an HCRF and an LDCRF, respectively.

In this section we first describe the dataset used in our experiments, then discuss results from the experiments we performed.

### 6.1 NATOPS Aircraft Handling Signal Dataset

We used the NATOPS dataset [Song et al. 2011b], a body-and-hand gesture dataset containing an official gesture vocabulary used for communication between carrier deck personnel and Navy pilots (e.g., yes or no signs, taxing signs, fueling signs, etc.). The dataset contains 24 gestures, with each gesture performed by 20 participants 20 times, resulting in 400 samples per gesture.

We selected five pairs of gestures (see Fig. 14) that are particularly interesting because the gestures in each pair are very similar. Two pairs (#2 & #3 and #20 & #21) are in fact indistinguishable in the absence of knowledge of hand pose. Gestures #20 ("brakes on") and #21 ("brakes off") are performed by raising both hands, with either open palms that are closed ("brakes off"), or vice versa ("brakes on"). Here, the role of hand pose is crucial to distinguishing two very similar gestures with opposite meanings. As a more subtle case, gestures #10 ("insert chocks") and #11 ("remove chocks") are performed with both arms down and waving them in/outward. The only difference is the position of thumbs: inward ("insert chocks") and outward ("remove chocks").

Experiments were conducted using combinations of body and hand features extracted using methods described in this paper (Section 4.2.4 and Section 4.3.6). There are 4 body features and 2 hand features.

The four body features are joint angles (T), joint velocities (dT), uniform-length relative joint coordinates (P), and the corresponding velocities (dP). The joint angle features (T and dT) are 8 DOF vectors (3 for shoulder and 1 for elbow, for both arms), and the joint coordinate features (P and dP) are 12 DOF vectors (3D coordinates of elbows and wrists for both arms). The uniform-length relative joints are obtained by configuring a generative model with the estimated joint angles with uniform limb lengths (so that their joint coordinates have less variance), and recording joint coordinates relative to the chest point.

The two hand features are "soft decision" and "hard decision." As noted earlier, the soft decision (S) is an 8 DOF vector with probability estimates obtained from the SVM (four hand poses for each hand), while the hard decision (H) is a 2 DOF vector obtained by selecting the highest probability estimate for each hand. Intuitively, S has richer information about the shape of hands, while H has a lower

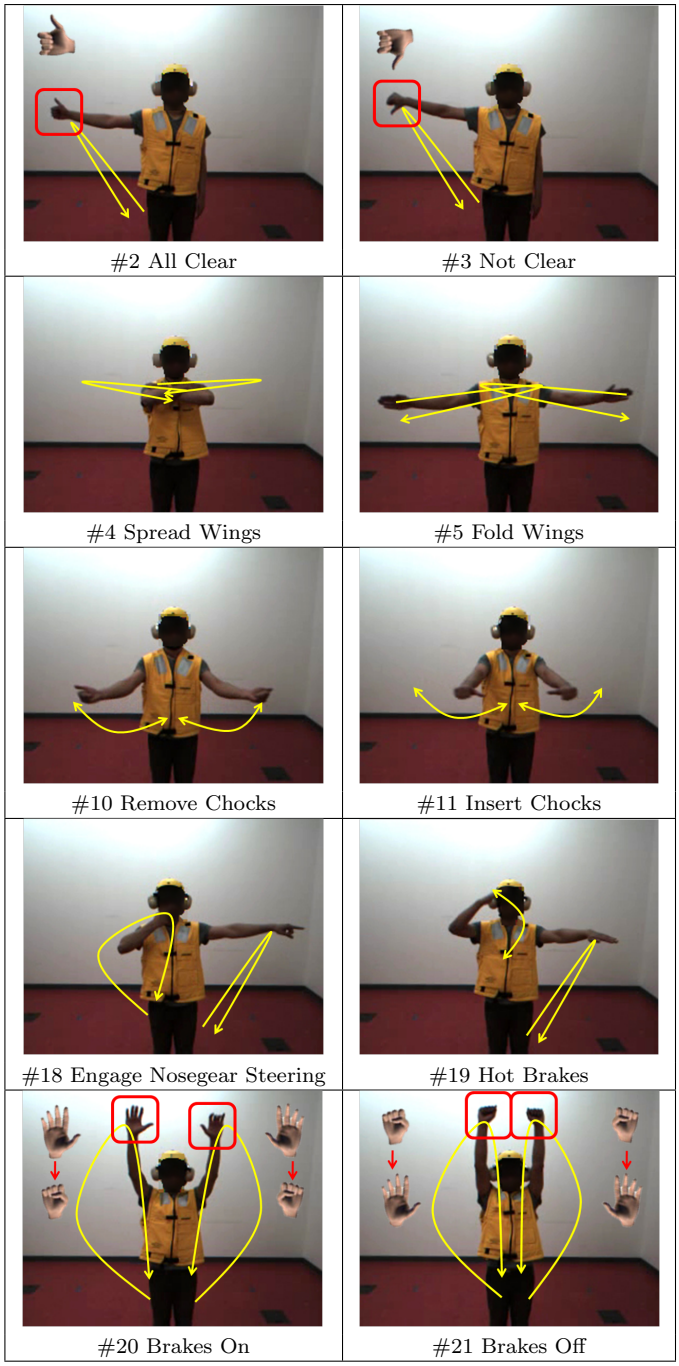


Fig. 14. Ten NATOPS aircraft handling signal gestures. Body movements are illustrated in yellow arrows, and hand poses are illustrated with synthesized images of hands. Red rectangles indicate hand poses are important in distinguishing the gesture pair.

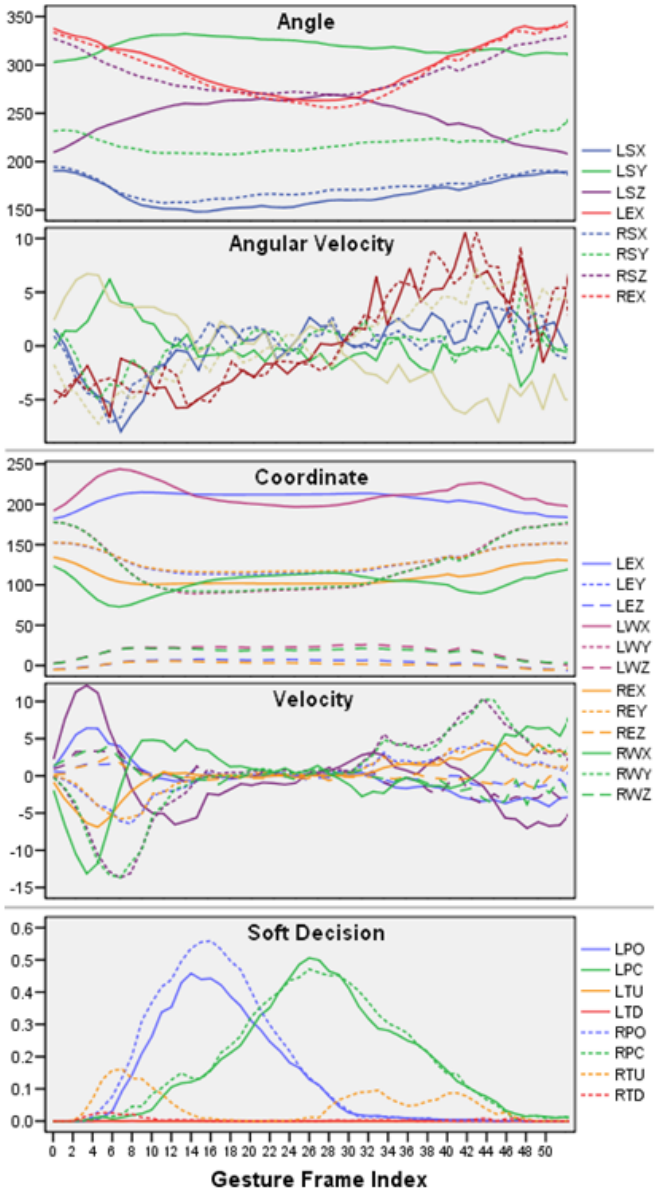


Fig. 15. An example feature data sequence for the gesture #20 ("brakes on") averaged over all individual trials of 20 participants. From the top: two body joint angle features, two body joint coordinate features, and one hand feature. Body labels are coded as: L/R-left/right; S/E/W-shoulder, elbow, wrist; X/Y/Z-axis. Hand labels are coded as: L/R-left/right; PO/PC-palm opened/closed; TU/TD-thumb up/down.

| Body Feature Type | Condition | Mean | Std. Dev | Independent Samples T-test |
|---|---|---|---|---|
| T | BodyOnly | 20.09 | 3.57 | $t(22)$=1.00, $p$=.326 |
| | BodyHand | 27.02 | 3.83 | |
| P | BodyOnly | 23.26 | 11.07 | $t(22)$=1.21, $p$=.24 |
| | BodyHand | 32.73 | 20.57 | |
| dT | BodyOnly | 62.47 | 7.21 | $t(22)$=4.06, $p$=.001 |
| | BodyHand | 76.23 | 8.10 | |
| dP | BodyOnly | 70.94 | 6.73 | $t(22)$=3.82, $p$=.001 |
| | BodyHand | 80.65 | 5.30 | |

Table IV. Statistics for the recognition accuracies comparing two conditions, body pose only (BodyOnly) and body and hand pose combined (BodyHand), under the four different body pose features.

degree of freedom, which can reduce the computational cost in an estimation step.

Fig. 15 illustrates example sequences of features for gesture #20 ("brakes on"), where we averaged all individual trials over 20 participants.

All experiments were conducted with $n$-fold cross validation ($n$-CV), allowing us to perform a cross-subject analysis, i.e., train the model with a dataset that does not include gesture examples performed by participants in a test dataset, resulting in more accurate measurement of performances. We measured accuracy with an F1 score ($\text{F1}=2 * \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$). In all tests, we set the regularization factor in Eq. 15 at 1,000 which, based on our preliminary experiments, helps prevent overfitting.

### 6.2 Isolated Gesture Recognition

We first performed three experiments for isolated gestures, to investigate: (1) whether combining body and hand signals improves accuracy; (2) which body and hand features are most informative; and (3) whether performing Gaussian temporal-smoothing improves recognition performance.

6.2.1 *Does Combining Body and Hand Pose Improve Accuracy?.* To determine whether combining body and hand poses helps improve recognition performance, we compared recognition performance under two conditions: body feature only (BodyOnly) and body and hand feature combination (BodyHand). Test result for BodyHand was obtained by averaging the test results of two conditions, using S and H.

For each test, we performed 4-CV analysis, varying the number of hidden states from 3 to 4 and taking an average. Since a 4-CV analysis performs four repetitive tests, we get variances in the results; we performed independent samples T-tests to see if the differences between two conditions were statistically significant.

Table IV shows means and standard deviations for the overall recognition accuracy rates averaged over 10 gestures, as well as the results from independent samples T-tests. In all our test cases, using body and hand pose together resulted

in higher recognition accuracy rates. For two of the body pose features (dT and dP) the differences were statistically significant ($p$=.001).

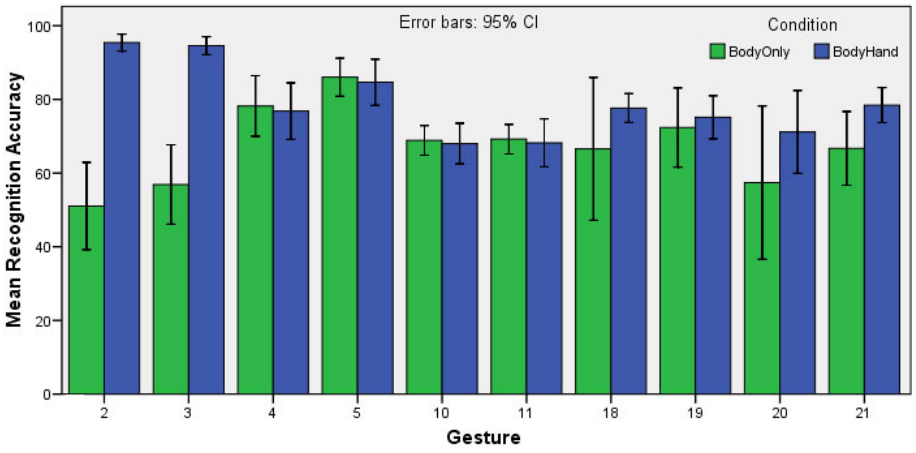


Fig. 16. Per-gesture comparisons of BodyOnly and BodyHand.

Fig. 16 shows per gesture comparisons of the two conditions (BodyOnly and BodyHand). Note that the graph used only the higher performing body features dT and dP. As expected, the performance difference was significant for the 4 gestures (#2, #3, #20, and #21) where the hand pose plays an important role in defining the gesture (see Fig. 14). The difference between BodyOnly and BodyHand is especially obvious for gesture pair #2 and #3, where recognition without knowing hand pose (BodyOnly) was no better than random. Our result indicated that using body and hand pose together on these 4 gestures achieved on average 27.5% higher accuracy; for the other 6 gestures there were slight differences, but none were significant.

6.2.2 *Which Features Are Most Informative?.* Various types of body or hand features have been explored in gesture recognition research, but there is no clear sense as to which features are most informative. In response, we compared the system's recognition accuracy using various combinations of three body features (dT, dP, and dTdP) and two hand features (S and H). Two body features (T and P) were omitted because our previous experiment showed that they resulted in inferior recognition performance. For each test case we performed 10-CV analysis, varying the number of hidden states from 3 to 5 and taking an average.

Table V shows comparisons of the resulting performance. Overall, the hand feature S performed significantly better than the feature H ($t(178)$=2.24, $p$=.018), achieving on average 3.44% higher accuracy rate. This indicates that considering probabilities for all class labels provides richer information, improving recognition accuracy. For body pose, dP performed the best, while the performances obtained using dT and dTdP were similar. We found no statistical significant in body feature differences. This indicates that the derivatives of joint coordinates are more informative than the derivative of joint angles.

| Body Feature Type | Hand Feature Type | | |
|---|---|---|---|
| | H, $\mu(\sigma^2)$ | S, $\mu(\sigma^2)$ | Average |
| dT | 78.02 (10.97) | 82.27 (10.42) | 80.15 (10.82) |
| dP | 80.72 (9.85) | 86.02 (8.32) | 83.37 (9.37) |
| dTdP | 80.08 (8.21) | 80.86 (9.51) | 90.47 (8.82) |
| Average | 79.61 (9.67) | 83.05 (9.60 ) | · |

Table V. Statistics for the recognition accuracies comparing three types of body features.

All in all, for the features we used, a combination of dP (uniform-length relative body joint velocity) and S (probability estimates of a hand pose) was the most informative feature for this task.

6.2.3 *Does Gaussian Temporal-Smoothing Help?.* The third experiment aimed to measure the advantage of the Gaussian temporal smoothing HCRF. Based on the previous results, we selected dPS as a feature combination (joint velocities for body and soft decision for hands). All tests were performed with 10-CV analysis, fixing the number of hidden states at 5, and varying the Gaussian window size from 1 to 21 (using only odd numbers).

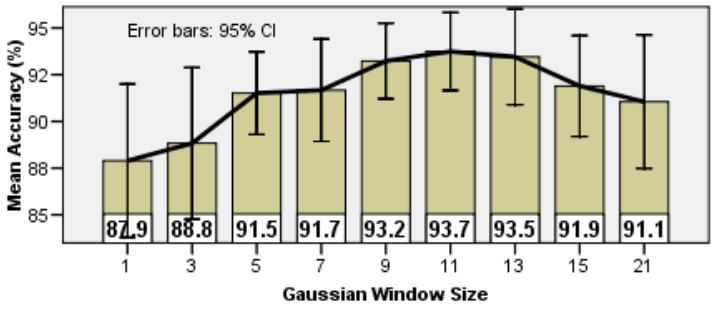


Fig. 17. Recognition accuracy for different window sizes.

As can be seen in Fig. 17, Gaussian temporal-smoothing significantly improved the performance: when compared to non-smoothing ($\omega$=1, 12.1% error), a half-second sized Gaussian window ($\omega$=11, 6.3% error) was able to reduce 48% of remaining errors. The performance dropped as the window size increased more than $\omega$=11, indicating that it started losing some important gesture information when the Gaussian window size is larger than half a second.

Table. VI shows confusion matrices comparing $\omega$=1 (no temporal smoothing) and $\omega$=11 (the best performing setting). We can see that both false positives and false negatives were decreased for all individual classes, with the highest gain achieved for gesture #10 (22% improvement).

No Temporal-Smoothing ($|\mathcal{H}|$=5, $\omega$=1)

| | #2 | #3 | #4 | #5 | #10 | #11 | #18 | #19 | #20 | #21 |
|---|---|---|---|---|---|---|---|---|---|---|
| #2 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| #3 | 0.00 | **0.98** | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| #4 | 0.00 | 0.00 | **0.79** | 0.03 | 0.08 | 0.01 | 0.01 | 0.01 | 0.00 | 0.01 |
| #5 | 0.00 | 0.00 | 0.06 | **0.92** | 0.01 | 0.01 | 0.01 | 0.01 | 0.00 | 0.00 |
| #10 | 0.00 | 0.00 | 0.06 | 0.01 | **0.73** | 0.11 | 0.00 | 0.00 | 0.00 | 0.00 |
| #11 | 0.00 | 0.01 | 0.03 | 0.02 | 0.14 | **0.86** | 0.01 | 0.00 | 0.00 | 0.01 |
| #18 | 0.00 | 0.01 | 0.01 | 0.00 | 0.01 | 0.00 | **0.90** | 0.08 | 0.01 | 0.04 |
| #19 | 0.00 | 0.00 | 0.02 | 0.01 | 0.00 | 0.01 | 0.07 | **0.88** | 0.03 | 0.03 |
| #20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | **0.87** | 0.06 |
| #21 | 0.00 | 0.00 | 0.03 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.09 | **0.85** |

Gaussian Temporal-Smoothing ($|\mathcal{H}|$=5, $\omega$=11)

| | #2 | #3 | #4 | #5 | #10 | #11 | #18 | #19 | #20 | #21 |
|---|---|---|---|---|---|---|---|---|---|---|
| #2 | **1.00** | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| #3 | 0.00 | **0.99** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 |
| #4 | 0.00 | 0.00 | **0.87** | 0.01 | 0.01 | 0.00 | 0.02 | 0.00 | 0.00 | 0.01 |
| #5 | 0.00 | 0.00 | 0.03 | **0.98** | 0.00 | 0.01 | 0.00 | 0.00 | 0.01 | 0.00 |
| #10 | 0.00 | 0.00 | 0.03 | 0.00 | **0.95** | 0.09 | 0.00 | 0.00 | 0.00 | 0.00 |
| #11 | 0.00 | 0.00 | 0.01 | 0.01 | 0.03 | **0.89** | 0.00 | 0.00 | 0.01 | 0.01 |
| #18 | 0.00 | 0.00 | 0.02 | 0.00 | 0.01 | 0.01 | **0.95** | 0.07 | 0.00 | 0.02 |
| #19 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | **0.93** | 0.01 | 0.00 |
| #20 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.92** | 0.07 |
| #21 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 | **0.88** |

Table VI. Confusion matrices comparing no temporal-smoothing ($|\mathcal{H}|$=5, $\omega$=1) and Gaussian temporal-smoothing ($|\mathcal{H}|$=5, $\omega$=11).

Fig. 18 shows distributions of hidden states for each gesture class, obtained using the dPS feature combination with $|\mathcal{H}|$=5 and $\omega$=11. Here we can see that the hidden states are roughly evenly distributed over the gesture classes, suggesting that the number of hidden states was appropriate.

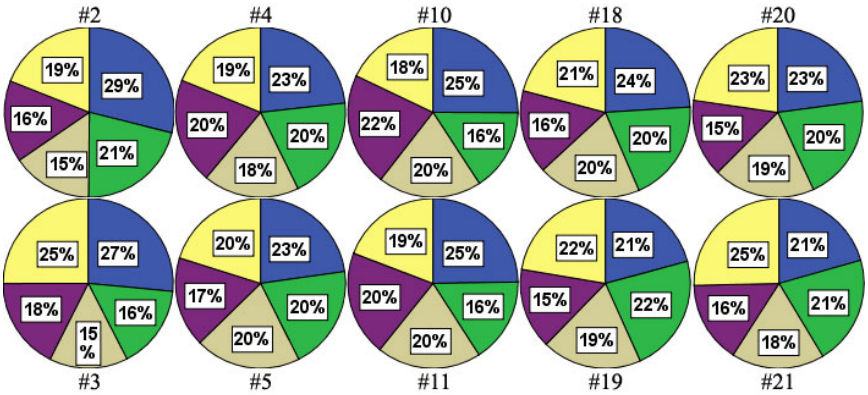


Fig. 18. Distributions of assigned hidden states ($|\mathcal{H}|$=5, $\omega$=11). The numbers enclosed in each area indicates the hidden state assignments.

| $\|\mathcal{H}\|$ | $\omega$ | LDCRF | Two-Layered Heuristic |
|---|---|---|---|
| 3 | 1 | 65.68% (4.87) | 76.84% (5.12) |
| | 11 | 78.06% (5.68) | 86.85% (4.68) |
| 4 | 1 | 71.60% (6.58) | 82.70% (7.66) |
| | 11 | 76.17% (4.61) | 88.04% (5.71) |
| 5 | 1 | 71.53% (5.31) | 81.94% (6.08) |
| | 11 | 78.70% (5.49) | 88.37% (5.33) |
| Average | | 73.62% (6.95) | 84.12% (6.95) |

Table VII. Statistics for continuous gesture recognition accuracies comparing two conditions, baseline approach using an LDCRF and our two-layered heuristic approach. The difference between the overall accuracy rates of the baseline (73.62%) and our two-layered heuristic approach (84.12%) was statistically significant ($t(118)$=8.28, $p \leq$0.001).

One important thing to note is that temporal-smoothing improves recognition accuracy significantly (by considering long-range input features and increasing the SNR), but does not increase the computational complexity of inference. Previous work on HCRF for gesture recognition [Wang et al. 2006] defined a window to concatenate neighboring input features, thus increasing the dimensionality. Our approach computes a weighted mean of neighboring input features, which does not increase the dimensionality or complexity compared to the original HCRF model [Quattoni et al. 2005] (additions and multiplications in the kernel operation can be negligible compared to the complexity of the inference algorithm).

### 6.3 Two-Layered Heuristic Approach for Continuous Gesture Recognition

In this last experiment we evaluated the performance of our two-layered heuristic approach for continuous gesture recognition. As in the previous experiment, we used the dPS feature combination to train LDCRF models, varying the number of hidden states from 3 to 5 as well as the size of a Gaussian window from 1 (no smoothing) to 11 (a half-second sized window). For each test case we performed 10-CV analysis, varying the number of hidden states from 3 to 5 and taking an average.

As a baseline, we used an LDCRF model to predict a label for each successive individual frame, choosing a label by taking the majority vote within each window.

We set the size of the sliding window to 71 (3.5 seconds) for both baseline and our heuristic approaches, which was chosen empirically as the best performing parameter. In all tests, a uniform weight function was used to make the local prediction (Eq. 22).

Table VII shows means and standard deviations for all recognition accuracy rates over 10 gestures, varying the number of hidden states and the size of the Gaussian window. In all our test cases, our two-layered heuristic approach outperformed the baseline approach, which indicates that our method improves recognition accuracy. An independent samples T-test indicated this accuracy difference was statistically significant ($t(118)$=8.28, $p \leq$0.001).

LDCRF ($|\mathcal{H}|$=5, $\omega$=11, $k$=71)

| | #2 | #3 | #4 | #5 | #10 | #11 | #18 | #19 | #20 | #21 |
|---|---|---|---|---|---|---|---|---|---|---|
| #2 | **0.91** | 0.11 | 0.02 | 0.01 | 0.00 | 0.00 | 0.01 | 0.01 | 0.01 | 0.00 |
| #3 | 0.01 | **0.79** | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| #4 | 0.01 | 0.02 | **0.82** | 0.07 | 0.08 | 0.06 | 0.04 | 0.01 | 0.07 | 0.04 |
| #5 | 0.01 | 0.01 | 0.05 | **0.85** | 0.04 | 0.05 | 0.02 | 0.01 | 0.02 | 0.01 |
| #10 | 0.01 | 0.01 | 0.03 | 0.02 | **0.78** | 0.26 | 0.01 | 0.01 | 0.01 | 0.01 |
| #11 | 0.00 | 0.01 | 0.02 | 0.01 | 0.07 | **0.60** | 0.01 | 0.00 | 0.00 | 0.01 |
| #18 | 0.01 | 0.01 | 0.01 | 0.00 | 0.01 | 0.01 | **0.58** | 0.02 | 0.00 | 0.00 |
| #19 | 0.02 | 0.02 | 0.01 | 0.01 | 0.01 | 0.00 | 0.31 | **0.92** | 0.03 | 0.02 |
| #20 | 0.01 | 0.01 | 0.01 | 0.01 | 0.00 | 0.01 | 0.01 | 0.01 | **0.79** | 0.05 |
| #21 | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.07 | **0.86** |

Two-Layered Heuristic Approach ($|\mathcal{H}|$=5, $\omega$=11, $k$=71)

| | #2 | #3 | #4 | #5 | #10 | #11 | #18 | #19 | #20 | #21 |
|---|---|---|---|---|---|---|---|---|---|---|
| #2 | **0.99** | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| #3 | 0.00 | **0.97** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| #4 | 0.01 | 0.00 | **0.96** | 0.06 | 0.05 | 0.03 | 0.03 | 0.01 | 0.06 | 0.03 |
| #5 | 0.00 | 0.00 | 0.02 | **0.94** | 0.04 | 0.03 | 0.01 | 0.00 | 0.01 | 0.00 |
| #10 | 0.00 | 0.00 | 0.01 | 0.00 | **0.86** | 0.24 | 0.01 | 0.00 | 0.00 | 0.00 |
| #11 | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 | **0.69** | 0.00 | 0.00 | 0.00 | 0.00 |
| #18 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | **0.64** | 0.01 | 0.00 | 0.00 |
| #19 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.31 | **0.98** | 0.01 | 0.00 |
| #20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.89** | 0.01 |
| #21 | 0.00 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | **0.96** |

Table VIII. Confusion matrices comparing the baseline approach (top) and two-layered heuristic approach (bottom) ($|\mathcal{H}|$=5, $\omega$=11, $k$=71 for both cases).

Table VIII shows confusion matrices comparing the baseline approach to our two-layered heuristic approach (in both cases, $|\mathcal{H}|$=5 and $\omega$=11 were the best performing setting). We can see that both false positives and false negatives decreased for all individual classes, with the higest gain achieved for gesture #3 (18% improvement).

Fig. 19 shows an example of sequence segmentation results using the baseline approach and our two-layered heuristic approach. This figure visually confirms that our approach effectively reduces prediction noise obtained from an LDCRF model and makes sequence segmentation and labeling task more robust.

## 7. CONCLUSION AND FUTURE WORK

We presented a new approach to vision-based continuous gesture recognition that combines 3D body pose estimation and hand pose classification. A stereo camera was used to obtain 3D images, and the images were background subtracted using a combination of the codebook approach and depth information.

For 3D body pose estimation, we constructed a parametric model of the human upper body to generate 3D body poses. This model was then fitted to input images by comparing both static and dynamic attributes of motion, where static features were computed from 3D visible-surface point clouds and contour point clouds, and dynamic features were computed from MHIs. We proposed an error function using MHIs, which allows us to compute dynamic motion error efficiently. The pose

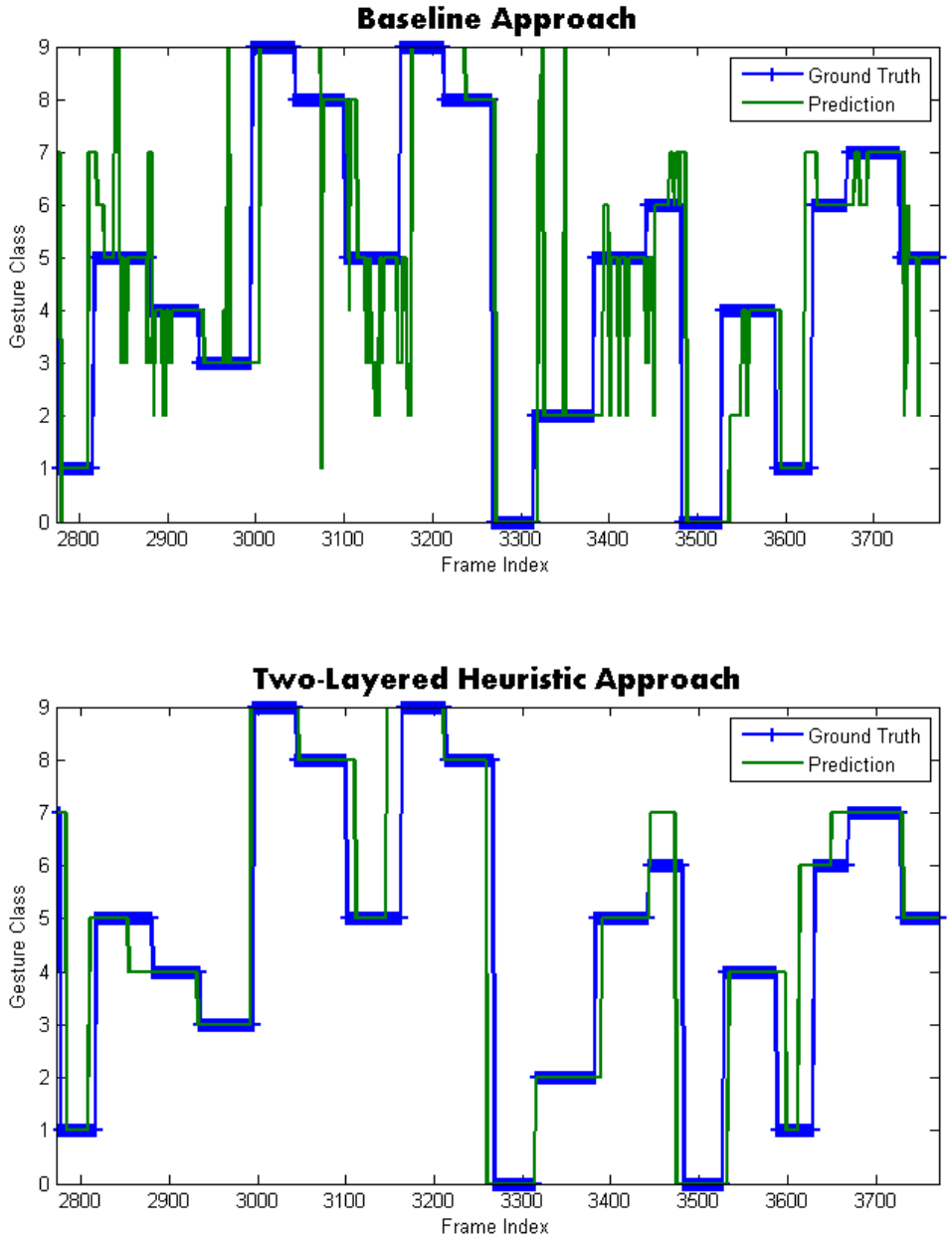


Fig. 19. Sequence segmentation results comparing LDCRF and two-layered heuristic approach ($|\mathcal{H}|$=5, $\omega$=11, $k$=71 for both cases).

estimation was performed using a particle filter.

For hand pose classification, we first defined a vocabulary of four canonical hand poses that included opened and closed hand, and thumb up and down. A multi-class SVM classifier was trained on a dataset containing HOG features extracted from manually segmented images of hands. Hand pose classification was performed by searching for hands in the image around the wrist positions obtained from body pose estimation, then classifying them using the SVM classifier.

Finally, for continuous gesture recognition, we combined body and hand features to train an LDCRF model that is capable of performing simultaneous sequence segmentation and labeling. In order to make the model to predict labels successively and perform segmentation more accurately as new observations are made, we developed a two-layered heuristic approach. Our approach sets a fixed-sized sliding window to evaluate chunks of frames successively. Then label prediction is done using information about all previous prediction results made repeatedly, and sequence segmentation is done based on the label prediction result.

The system was evaluated on a real-world human-computer interaction scenario: we tested the performance of our continuous gesture recognition system with a subset of the NATOPS aircraft handling signals, a challenging gesture vocabulary that involves both body and hand poses articulations. We showed that combining body and hand pose signals significantly improved the gesture recognition accuracy. We also showed what types of body and hand pose features performed the best: for the body pose, the derivatives of joint coordinate was the most informative; for the hand pose, a vector of probability estimates for all classes was the most informative. Lastly, we showed that a two-layered heuristic approach is able to recognize gesture labels successively, achieving the recognition accuracy rate of 88.37%.

Our current system can be improved in a number of ways. We performed body pose estimation and hand pose classification serially, using estimated wrist positions to search for hands. However, once the hands are detected, they could be used to refine the body pose estimation (e.g., by inverse kinematics). Context-sensitive pose estimation may also improve performance. There is a kind of grammar to gestures in practice: for the NATOPS scenario as an example, once the "brakes on" gesture is performed, a number of other gestures are effectively ruled out (e.g., "move ahead"). Incorporating this sort of context information might significantly improve estimation performance.

Lastly, in order for our system to be interactive, it is necessary to allow a two-way communication between the users and the system. Although it is crucial for a system to be able to understand humans gestures, it is also necessary for the system to have an appropriate feedback mechanism, that is, the system have to be able to gesture back, just as a human would do in the same situation. There are many questions to be answered: what does it mean for a system to gesture?; how can we define a natural feedback mechanism, or how natural a system's feedback should be? We look forward to exploring these questions in future work.

ACKNOWLEDGMENTS

REFERENCES

AGGARWAL, J. K. AND CAI, Q. 1999. Human motion analysis: a review. *Comput. Vis. Image Underst. 73*, 428–440.

BARR, A. 1981. Superquadrics and angle-preserving transformations. *Computer Graphics and Applications, IEEE 1,* 1, 11 –23.

BOBICK, A. AND DAVIS, J. 1996. Real-time recognition of activity using temporal templates. In *Applications of Computer Vision, 1996. WACV '96., Proceedings 3rd IEEE Workshop on*. 39 –42.

BRADSKI, G. AND KAEHLER, A. 2008. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA.

BRAND, M. 1999. Shadow puppetry. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*.

BUEHLER, P., ZISSERMAN, A., AND EVERINGHAM, M. 2009. Learning sign language by watching tv (using weakly aligned subtitles). *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 0*, 2961–2968.

CHANG, C.-C. AND LIN, C.-J. 2001. LIBSVM: a Library for Support Vector Machines.

DALAL, N. AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. 886 –893 vol. 1.

DEMIRDJIAN, D. AND DARRELL, T. 2002. 3-d articulated pose tracking for untethered diectic reference. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*.

DENAVIT, J. AND HARTENBERG, R. S. 1955. A kinematic notation for lower-pair mechanisms based on matrices. *Trans ASME J. Appl. Mech 23*, 215–221.

DEUTSCHER, J., BLAKE, A., AND REID, I. 2000. Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*.

ENGIN, A. 1980. On the biomechanics of the shoulder complex. *Journal of Biomechanics 13,* 7, 575 – 581, 583–590.

EROL, A., BEBIS, G., NICOLESCU, M., BOYLE, R. D., AND TWOMBLY, X. 2007. Vision-based hand pose estimation: A review. *Comput. Vis. Image Underst. 108*, 52–73.

FENG, X., YANG, J., AND ABDEL-MALEK, K. 2008. Survey of biomechanical models for the human shoulder complex. Tech. rep., SAE International.

GAVRILA, D. M. 1999. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding 73,* 1, 82 – 98.

GUNAWARDANA, A., MAHAJAN, M., ACERO, A., AND PLATT, J. C. 2005. Hidden conditional random fields for phone classification.

HARRIS, F. 1978. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE 66,* 1, 51 – 83.

HSU, C.-W. AND LIN, C.-J. 2002. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on 13,* 2 (Mar.), 415 –425.

ISARD, M. AND BLAKE, A. 1998. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision 29,* 1, 5–28.

KIM, K., CHALIDABHONGSE, T. H., HARWOOD, D., AND DAVIS, L. S. 2005. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging 11,* 3, 172–185.

KNERR, S., PERSONNAZ, L., AND DREYFUS, G. 1990. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In *Neurocomputing: Algorithms, Architectures and Applications*, J. Fogelman, Ed. Springer-Verlag.

LAFFERTY, J. D., MCCALLUM, A., AND PEREIRA, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 282–289.

Lee, M. W. and Cohen, I. 2006. A model-based approach for estimating human 3d poses in static images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 905–916.

Mitchell, T. M. 1997. *Machine Learning*, 1 ed. McGraw-Hill Science/Engineering/Math.

Mitra, S. and Acharya, T. 2007. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37,* 3 (May), 311 –324.

Moeslund, T. B. and Granum, E. 2001. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding 81,* 3, 231–268.

Moeslund, T. B., Hilton, A., and Kruger, V. 2006. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding 104,* 2-3, 90–126.

Morency, L.-P., Quattoni, A., and Darrell, T. 2007. Latent-dynamic discriminative models for continuous gesture recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*. 1 –8.

Mori, G. and Malik, J. 2006. Recovering 3d human body configurations using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28,* 7, 1052 –1062.

NASA. 1995. *Man-Systems Integration Standards: Volume 1. Section 3. Anthropometry And Biomechanics*.

Pearl, J. 1982. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference on AI*. Pittsburgh, PA, 133–136.

Poppe, R. 2007. Vision-based human motion analysis: An overview. *Comput. Vis. Image Underst. 108*, 4–18.

Quattoni, A., Collins, M., and Darrell, T. 2005. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, Cambridge, MA, 1097–1104.

Shakhnarovich, G., Viola, P., and Darrell, T. 2003. Fast pose estimation with parameter-sensitive hashing. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. 750 –757 vol.2.

Sminchisescu, C. and Triggs, B. 2003. Kinematic jump processes for monocular 3d human tracking. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 1. I–69 – I–76 vol.1.

Song, Y., Demirdjian, D., and Davis, R. 2011a. Multi-signal gesture recognition using temporal smoothing hidden conditional random fields. In *9th IEEE Conference on Automatic Face and Gesture Recognition (FG 2011) (in submission)*. Santa Barbara, CA.

Song, Y., Demirdjian, D., and Davis, R. 2011b. Unified framework for body and hand tracking: Natops aircraft handling signals. In *9th IEEE Conference on Automatic Face and Gesture Recognition (FG 2011) (in submission)*. Santa Barbara, CA.

Sutton, C., Rohanimanesh, K., and McCallum, A. 2004. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of the twenty-first international conference on Machine learning*. ICML '04. ACM, New York, NY, USA, 99–.

Vapnik, V. N. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Wang, R. Y. and Popović, J. 2009. Real-time hand-tracking with a color glove. In *ACM SIGGRAPH 2009 papers*. SIGGRAPH '09. ACM, New York, NY, USA, 63:1–63:8.

Wang, S. B., Quattoni, A., Morency, L.-P., Demirdjian, D., and Darrell, T. 2006. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*.

Yin, Y. and Davis, R. 2010. Toward natural interaction in the real world: Real-time gesture recognition. In *Proceedings of the International Conference on Multimodal Interfaces*. Beijing, China.

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. 1986. A hand gesture interface device. *SIGCHI Bull. 18*, 189–192.

Received December 2010; revised ; accepted