MIT Open Access Articles

Tracking body and hands for gesture recognition: NATOPS aircraft handling signals database

Tracking Body and Hands for Gesture Recognition: NATOPS Aircraft Handling Signals Database

Yale Song, David Demirdjian, and Randall Davis
MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, MA 02139
{yalesong,demirdj,davis}@csail.mit.edu

Abstract— We present a unified framework for body and hand tracking, the output of which can be used for understanding simultaneously performed body-and-hand gestures. The framework uses a stereo camera to collect 3D images, and tracks body and hand together, combining various existing techniques to make tracking tasks efficient. In addition, we introduce a multi-signal gesture database: the NATOPS aircraft handling signals. Unlike previous gesture databases, this data requires knowledge about both body and hand in order to distinguish gestures. It is also focused on a clearly defined gesture vocabulary from a real-world scenario that has been refined over many years. The database includes 24 body-and-hand gestures, and provides both gesture video clips and the body and hand features we extracted.

I. INTRODUCTION

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.

We have developed a multi-signal gesture recognition system that attends to both bodies and hands, allowing a richer gesture vocabulary and more natural human-computer interaction. In this paper, we present the *signal processing* part of the system, a unified framework for tracking bodies and hands to obtain signals. The *signal understanding* part (i.e., learning to recognize patterns of multi-signal gestures) is described in a companion paper [16].

There has been extensive work in human pose tracking, including upper or full body, hand, head, and eye gaze. In [3], for example, Buehler *et al.* presented an arm-and-hand tracking system that enabled the extracted signals to be used in sign language recognition. Hand poses were estimated using histograms of oriented gradients (HOG) [5] features, but not classified explicitly. Also, body poses were reconstructed in 2D space, losing some of the important features in gesture recognition (e.g., pointing direction). In [15], Nickel *et al.* developed a head-and-hand tracking system for recognizing pointing gestures. The system tracked 3D positions of head and hands based on skin-color distribution. The extracted signals were used for recognizing pointing gestures using an HMM. However, their application scenario included static pointing gestures only, a task too simple to explore the complex nature of multi-signal gestures.

Our system performs 3D upper body pose estimation and hand pose classification together. Upper body poses are estimated in a multi-hypothesis Bayesian inference framework [10] following a generative model-based approach. Similar to [3], the estimated body poses are used to guide the search for hands and left/right hand assignment. Hand poses are classified into one of a set of predefined poses using a multiclass Support Vector Machine (SVM) [18] that has been trained offline using HOG features.

Ideally, depth maps will have highly accurate 3D information, in which case examining static poses would suffice to track body pose successfully. However, current depth sensor technology is limited in resolution (i.e., depth accuracy decreases exponentially as the distance gets further). In our scenario, the subject is assumed to stand 50 feet away from the camera[1], so relying solely on the static 3D point cloud returned from the sensor will lead to an unsatisfactory result. Instead we also want to exploit dynamic features of body motion, and we do this by introducing an error function based on motion history images (MHIs), in which each pixel value is a function of the recency of motion in that location in the image. This often gives us useful information about dynamics of motion, indicating where and how the motion has occurred.

Publicly available gesture databases allow researchers to build and evaluate their ideas quickly and conveniently. Currently, there are many such gesture databases (e.g., [9], [13]), but most current gesture vocabularies are characterized by a single signal only, e.g., body pose alone. In current databases this is sufficient to distinguish gestures, and this in turn limits an opportunity to test and evaluate multi-signal gesture recognition.

In [9], for example, Hwang *et al.* presented a full-body gesture database, containing 2D video clips and 3D motion data of gestures recorded and extracted from 20 subjects. Although the database contained 54 distinct gestures, it contained a single signal only, body pose. In [13], Martinez *et al.* presented a database of American Sign Language (ASL) that included body motions, hand shapes, words, and sentences. A comprehensive set of gestures were provided:

---

[1]In general, carrier deck personnel must keep at least 50 feet from the aircrafts to ensure their safety [17].

39 body motions and 62 hand shapes. However, the gestures were performed with either body or hands but not both simultaneously.

We have created a multi-signal gesture database: the Naval Air Training and Operating Procedures Standardization (NATOPS) aircraft handling signals database. It uses the official gesture vocabulary for the U.S. Navy aircraft carrier environment [17], which defines a variety of body-and-hand signals that carrier flight deck personnel use to communicate with the U.S. Navy pilots. The database consists of two parts: gesture video clips and extracted features of body and hand poses. To our knowledge, our database is the first to contain simultaneous body-and-hand gestures.

Several things make the database interesting for gesture recognition. First, it contains a multi-signal gesture vocabulary of body-and-hand gestures; thus, various issues in multi-signal pattern recognition (e.g., modeling information fusion, capturing dependencies within data, etc.) can be explored. Second, there are many similar gesture pairs with subtle differences in either body or hand pose; the gestures thus pose a challenging recognition task. Third, the gesture vocabulary is designed to handle a wide range of complex deck situations, so the gestures have been extensively refined and optimized over the years, suggesting it is a clearly defined vocabulary from a real-world scenario. Finally, successful gesture recognition on this database can help solve a real world problem: DARPA (the Defense Advanced Research Projects Agency) and the U.S. Navy are investigating the feasibility of deploying unmanned combat aerial vehicles (UCAVs) onto the aircraft carriers [6]. It would clearly be beneficial to allow deck personnel to communicate with UCAVs with the same gestures they use with a human pilot.

Section II describes the unified framework for body and hand tracking, Section III describes the NATOPS aircraft handling signals database, and Section IV shows evaluation results and discusses the accuracy of the extracted body and hand features. Section V concludes with listing contributions and suggesting directions for future work.

II. BODY AND HAND TRACKING FRAMEWORK

A. Input Data

Input to our system is video recorded using a Bumblebee 2 stereo camera[2], producing 320 x 240 pixel resolution images at 20 FPS. While recording videos, we produce depth maps and mask images in real-time as the video is being recorded. Depth maps are calculated using the manufacturer-provided SDK. Mask images are obtained by performing background subtraction with a combination of a codebook approach [11] and a "depth-cut" method: after performing background subtraction using the codebook approach, we filter out pixels where the distance is further from camera than a foreground object. This helped to remove shadows created by a foreground object. Sample images from the videos are shown in Fig 1.

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

Fig. 1. Example images of (a) input image, (b) depth map, and (c) mask image. The "T-pose" shown in the figures is used for body tracking initialization
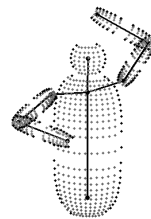

Fig. 2. Generative model of the human upper body.

B. 3D Upper 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., we make 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 of random variables representing a body pose that we are estimating.

1) Generative Model: A generative model of the human upper body is constructed in 3D space, representing a skeletal model as a kinematic chain and a volumetric model described by superellipsoids [1] (Fig. 2). The basic model includes 6 body parts (trunk, head, upper/lower arms for left/right) and 9 joints (chest, head, navel, left/right shoulder, elbow, wrist); of the 9 joints, 4 are articulated (shoulder and elbow) while others remain fixed once initialized. We prevent the model from generating anatomically implausible body poses by constraining joint angles to known physiological limits [14].

We improve on this basic model by building a more precise model of the shoulder, but do so in a way that does not add additional DOFs. To capture arm movement more accurately, the shoulder model is approximated analytically by examining relative positions of shoulder and elbow: we compute the angle φ between the line from the mid-chest to the shoulder and the line from mid-chest to the elbow. The chest-to-shoulder joint angle θ^{CS} is then updated as

$$\theta^{CS'} = \begin{cases} \theta^{CS} + \frac{\varphi}{\theta^{CS}_{max}} & \text{if elbow is higher than shoulder} \\ \theta^{CS} - \frac{\varphi}{\theta^{CS}_{min}} & \text{otherwise} \end{cases} \quad (1)$$

where θ^{CS}_{min} and θ^{CS}_{max} are minimum and maximum joint angle limits for chest-to-shoulder joints [14]. This simplified model only mimics shoulder movement in one-dimension, up and down, but works quite well in practice, as most variation in arm position comes from up and down motion.

With this model, an upper body pose is parameterized as

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

where G is a 4 DOF global translation and rotation vector (rotation around the vertical axis only), and R is an 8 DOF joint angle vector (3 for shoulder and 1 for elbow, for each arm). Since the positions of the camera and subject are assumed to be fixed, we estimate only the R vector during inference; the others are set during model initialization.

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 [10] is particularly well suited to this type of inference problem, for its ability to keep multiple hypotheses during inference while discarding less likely hypotheses only slowly.

A particle filter assumes the posterior state density $p(\mathbf{x} \mid \mathbf{z})$ to be a multimodal non-Gaussian distribution, approximating it by a set of N weighted samples: $\left\{ \left(s_t^{(1)}, \pi_t^{(1)}\right), \cdots, \left(s_t^{(N)}, \pi_t^{(N)}\right) \right\}$, where each sample 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 initial body pose configurations (i.e., joint angles and limb lengths) are obtained by having the subject assume a static "T-pose" (shown in Fig. 1), and fitting the model to the image with exhaustive search. The dynamic model of joint angles is constructed as a Gaussian process:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}(0, \sigma^2). \qquad (3)$$

We calculate an estimation result as the weighted mean of all samples:

$$\mathbb{E}\left[f(\mathbf{x}_t)\right] = \sum_{n=1}^{N} \pi_t^{(n)}\, f(s_t^{(n)}). \qquad (4)$$

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 $\varepsilon(\mathbf{z}_t, \mathbf{z}_{t-1}, s_t^{(n)}, \mathbb{E}\left[f(\mathbf{x}_{t-1})\right])$:

$$p(\mathbf{z}_t \mid \mathbf{x}_t = s_t^{(n)}) = \frac{1}{\exp\left\{\varepsilon(\cdot)\right\}} \qquad (5)$$

where the fitting error $\varepsilon(\cdot)$ 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) [2]. The first two features capture discrepancies in static poses; the third captures discrepancies in the dynamics of motion. We set the weight of each error term empirically.

The first two error terms, computed from 3D visible-surface and contour point clouds, are used frequently in body motion tracking (e.g., [7]), for their ability to evaluate how well the generated body pose fits the actual pose observed in the 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.
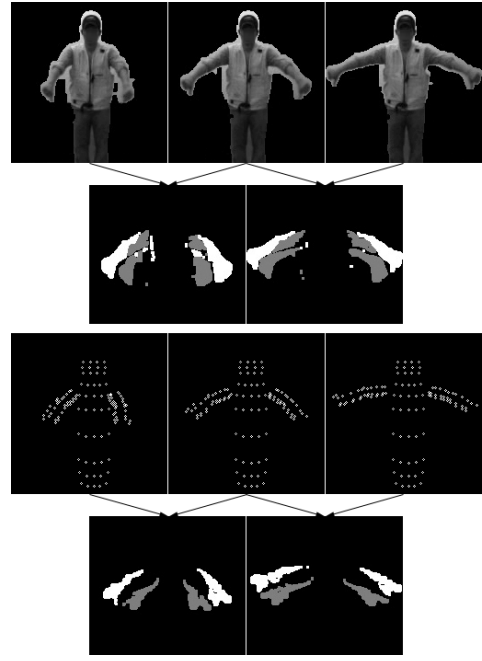

Fig. 3. MHIs of the input image (top) and the model (bottom).

The third error term, an MHI error, measures discrepancies in the dynamics of motion by comparing an MHI of the model and an MHI of the input image. We compute an MHI using I_{t-1} and I_t, two time-consecutive 8-bit unsigned integer images. For the generative model, I_t is obtained by rendering an image of the model generated by a particle $s_t^{(n)}$, and I_{t-1} is obtained by rendering the model generated by $\mathbb{E}\left[f(\mathbf{x}_{t-1})\right]$ (Eq. 4). For the input images, I_t is obtained by converting an RGB input image to YCrCb color space and extracting the brightness channel (Y); this is stored to be used as I_{t-1} for the next time step. Then an MHI is computed as

$$I_{MHI} = \lambda(I_{t-1} - I_t, 0, 127) + \lambda(I_t - I_{t-1}, 0, 255) \qquad (6)$$

where $\lambda(I, \alpha, \beta)$ is a binary threshold operator that sets each pixel value to β if $I(x,y) > \alpha$, and to zero otherwise. The values 127 and 255 are chosen to indicate the time information of those pixels. 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. 3.

Finally, an MHI error is computed using an MHI of the model $I_{MHI}(s_t^{(n)}, \mathbb{E}[f(\mathbf{x}_{t-1})])$ and an MHI of the input image $I_{MHI}(\mathbf{z}_t, \mathbf{z}_{t-1})$ as

$$\varepsilon_{MHI} = \mathbb{C}\text{ount}\left[\, \lambda(I', 127, 255) \,\right] \qquad (7)$$

where

$$I' = \text{abs}\left(I_{MHI}(\mathbf{z}_t, \mathbf{z}_{t-1}) - I_{MHI}(s_t^{(n)}, \mathbb{E}\left[f(\mathbf{x}_{t-1})\right])\right). \qquad (8)$$

The reason for setting the cutoff value to 127 in Eq. 7 is to penalize the conditions in which two MHIs do not match at the current time-step only, independent of the situation at

Fig. 4. 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.

Fig. 5. 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.

the previous time-step, where by "not match" we mean that the pixel values of two MHIs do not agree.

4) Output Feature Types: We get four types of features from body pose estimation: joint angles, joint angular velocities, joint coordinates, and joint coordinate velocities. Joint angles are 8 DOF vectors (3 for shoulder and 1 for elbow, for each arm) obtained directly from the estimation. To obtain joint coordinates, we first generate a model with the estimated joint angles and uniform-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 coordinate velocities are calculated by taking the first derivatives of joint angles and coordinates.

C. Hand Pose Classification

Hand poses used in NATOPS gestures are relatively discrete and few in number, likely because of the long distance (50~ft.) between deck personnel and pilots [17]. For our experiments we selected four hand poses that are crucial to distinguishing the NATOPS gestures (Fig. 4).

1) HOG Features: HOG features [5] 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 well characterized by locally collected distributions of intensity gradients or edge orientations, and does not require 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 8 x 8 pixels, window size of 32 x 32 pixels, with 9 orientation bins. Fig. 4 shows a visualization of the computed HOG features.

2) Multi-Class SVM Classifier: To classify the HOG features, we trained a multi-class SVM classifier [18] using LIBSVM [4]. 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 [12] for fast training, while obtaining comparable accuracy to *one-against-all* method [8]. We performed grid search and 10-fold cross validation for parameter selection.

A training dataset was collected from the recorded video clips. Due to the difficulty of manual labeling, we collected samples from the first 10 subjects only (out of 20). Positive samples were collected by manually cropping 32 x 32 pixel images and labeling them; negative samples were collected automatically at random location after collecting the positive samples. We scaled and rotated 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.

3) Tracking: We use estimated wrist positions to constrain the search for hands in the image as well as to decide left/right hand assignment. We create a 56 x 56 pixel search region around each of the estimated wrist positions (see Fig. 5). Estimated wrist positions are of course not always accurate, while current hand classification often provides a useful prediction of subsequent hand location. Therefore, when a hand is found at the previous time step, we center the search region at the geometric mean of the estimated wrist position at time t and the found hand position at time $t - 1$.

Within the 56 x 56 pixel search region, we use a 32 x 32 pixel sliding window to examine the region, moving with 8 pixel steps (i.e., examining 16 times for each search region). Each time a sliding window moves to a new position, 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). To get a single classification result per search region, we cluster all positive classification results within the region, averaging positions and probability estimates of all positive classification results (i.e., classified into one of the k positive classes). Fig. 5 illustrates this clustering process.

4) 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 classifier (4 classes for each hand); the hard decision is a 2 DOF vector of hand labels.

III. NATOPS Body-and-Hand Gesture Database

We selected 24 NATOPS aircraft handling signals, the gestures most often used in routine practice on the deck environment.[3] The gestures have many similar looking pairs with subtle differences in either body or hand pose (Fig. 9). For example, gestures #4 and #5, gestures #10 and #11, and gestures #18 and #19 have the same hand poses but similar body gestures (e.g., one performed in forward and the other one in backward, etc.). In contrast, gestures #2 and #3, gestures #7 and #8, and gestures #20 and #21 have the same body gesture with different hand poses (e.g., thumb up/down or palm opened/closed).

Twenty subjects repeated each of 24 gestures 20 times, resulting in 400 samples for each gesture class. Each sample had a unique duration; the average length of all samples was 2.34 sec (σ^2=0.62). Videos were recorded in a closed room environment with a constant illuminating condition, and with positions of cameras and subjects fixed throughout the recording. We use this controlled circumstance as our first step towards developing a proof-of-concept for NATOPS gesture recognition, and discovered that even this somewhat artificial environment still posed substantial challenges for our vocabulary.

The NATOPS database consists of two parts: gesture video clips and extracted features of body and hand poses. The first part includes stereo camera-recorded images, depth maps, and mask images. The second part includes the four types of body features and the two types of hand features we estimated. The database can be used for two purposes: pose estimation and gesture recognition. The gesture video clips can be used as a database for body-and-hand tracking, while the feature data can be used as a database for multi-signal gesture recognition. Fig. 6 illustrates example sequences of features for gesture #20 ("brakes on"), where we averaged all individual trials over 20 subjects (400 samples).

To collect ground-truth data for pose estimation, we selected one subject and recorded gestures using both a stereo camera and a Vicon system[4] simultaneously, producing body pose labels for that subject. Hand pose labels were created by selecting the same subject and visually checking each image frame, manually labeling hand poses. Lastly, the ground-truth data for gesture recognition was produced by manually segmenting and labeling sequences of the estimated features into individual trials.

IV. Evaluation

To evaluate the accuracy of body pose estimation and hand pose classification, we selected 10 gestures that we believe well represent the intricacy of the entire set, with each gesture paired with a corresponding similar gesture: #2 and #3; #4 and #5; #10 and #11; #18 and #19; #20 and #21.

[3]These gestures are being taught to all Aviation Boatswain's mate Handlers (ABHs) during their first week of classes at the technical training school in Naval Air Station Pensacola.

[4]The Vicon motion capture system included 16 cameras at 120 Hz frequency, 1 mm precision.
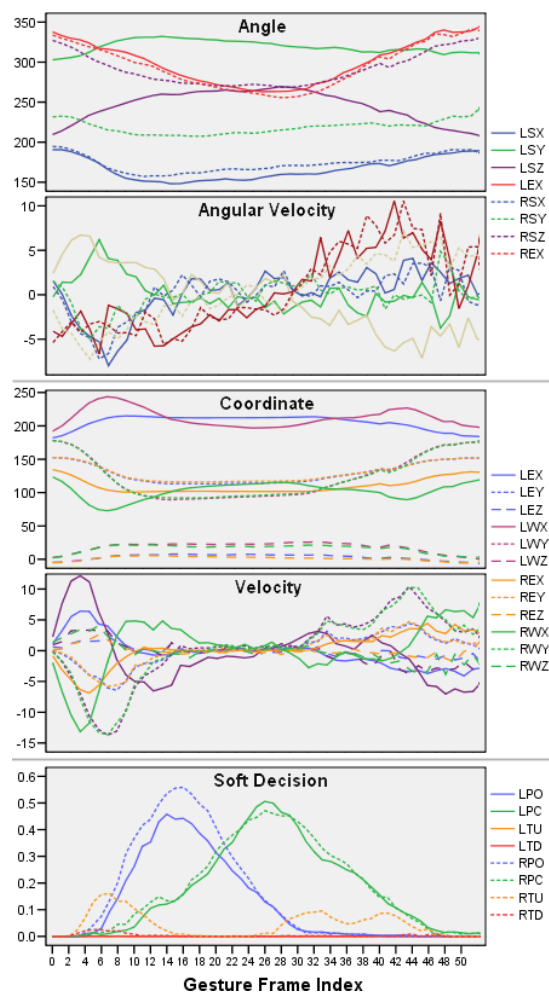

Fig. 6. Example sequences of features for the gesture #20 ("brakes on") averaged over all individual trials of 20 subjects. From the top: two joint angle features, two joint coordinate features, and one hand feature. Body labels are coded as: L/R-left/right; S/E/W-shoulder, elbow, wrist; X/Y/Z-axis. Hand labels are coded as: L/R-left/right; PO/PC-palm opened/closed; TU/TD-thumb up/down.

The estimation was performed with 500 particles, taking about 0.4 seconds to estimate each frame on an Intel Xeon Dual Core 2.66 GHz machine with 3.25GB of RAM.

A. Body Pose Estimation

The Vicon ground-truth body poses were superimposed onto the input images, scaled and translated properly so that they align with the coordinate system that the estimated body pose is in (Fig. 7). We calculated pixel displacement errors for each joint and accumulated, providing a total measure of pixel error. As shown in Fig. 8, in a 320 x 240 pixel frame, the average pixel error per frame was 29.27 pixels, with a lower error for 2D gestures (mean = 24.32 pixels) and higher for 3D gestures (mean = 34.20 pixels).

B. 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,
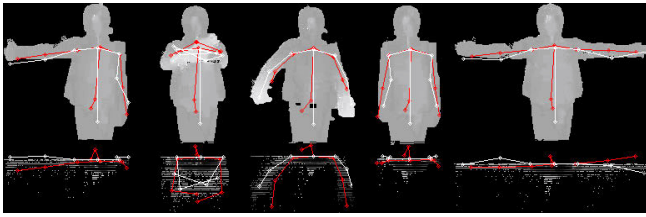
Fig. 7. Vicon ground-truth data (red lines) superimposed onto depth maps with estimation results (white lines).
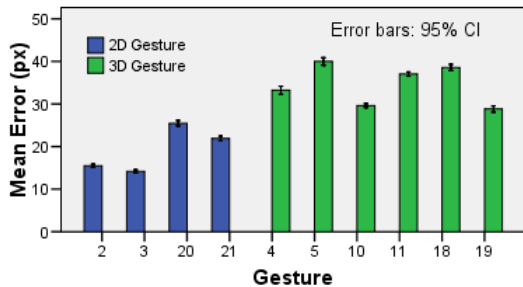

Fig. 8. Measures of total pixel errors for body pose estimation.

what matters more is how well the classifier performs on the video images, rather than on segmented images. To explore this, we randomly selected a subset of full image frames from four gestures that contained the canonical hand poses (i.e., #2 and #3; #20 and #21). 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). For simplicity, we used hard decision values. The result is shown in Table I. The slightly lower accuracy rates compared to the test result on pre-segmented samples indicates that using estimated wrist position can in some cases decrease hand detection accuracy, although it can reduce hand search time dramatically.

V. CONCLUSION AND FUTURE WORK

We presented a unified framework for body and hand tracking, and described the NATOPS body-and-hand gesture database. This work lays foundation for our multi-signal gesture recognition, described in a companion paper [16].

The goal of this pose tracking work was to provide high quality body and hand pose signals for reliable multi-signal gesture recognition; hence real-time tracking ability was not considered in this work. Faster processing could

TABLE I
HAND POSE CLASSIFICATION ACCURACY

Gesture	Precision	Recall	F1 Score
#2	0.97	0.91	0.94
#3	0.99	1.00	0.99
#20	1.00	0.90	0.94
#21	1.00	0.80	0.89

be achieved in a number of ways, including optimizing the number of particles in body pose estimation, tracking with a variable frame rate (e.g., using an MHI to quantify the extent of motion difference was made), or using GPUs for fast computation.

We performed body pose estimation and hand pose classification serially, using estimated wrist position 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 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 pose tracking performance.

VI. ACKNOWLEDGMENTS

REFERENCES

[1] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Comput. Graph. Appl.*, 1(1):11–23, 1981.
[2] A. F. Bobick and J. W. Davis. Real-time recognition of activity using temporal templates. In *Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV)*, pp.39–42, 1996.
[3] P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching TV (using weakly aligned subtitles). In *CVPR*, pp.2961-2968, 2009.
[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.
[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pp.8660-893, 2005.
[6] *Joint Unmanned Combat Air Systems, J-UCAS Overview.* http://www.darpa.mil/j-ucas/fact_sheet.htm
[7] J. Deutscher, A. Blake, and I. D. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR*, pp.2126–2133, 2000.
[8] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, Mar 2002.
[9] B.-W. Hwang, S. Kim, and S.-W. Lee. A full-body gesture database for automatic gesture recognition. In *FG*, pp.243–248, 2006.
[10] M. Isard and A. Blake. CONDENSATION-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
[11] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. S. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.
[12] S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited: A stepwise procedure for building and training a neural network. In *Neurocomputing: Algorithms, Architectures and Applications*. Vol F68 of NATO ASI Series, pp.41–50. Springer-Verlag, 1990.
[13] A. M. Martinez, R. B. Wilbur, R. Shay, and A. C. Kak. Purdue RVL-SLLL ASL database for automatic recognition of American Sign Language. In *ICMI*, pp.162–172, 2002.
[14] NASA. *Man-Systems Integration Standards: Volume 1. Section 3. Anthropometry and Biomechanics*, 1995
[15] K. Nickel, E. Seemann, and R. Stiefelhagen. 3D-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario. In *FG*, pp.565–570, 2004.
[16] Y. Song, D. Demirdjian, and R. Davis. Multi-signal gesture recognition using temporal smoothing hidden conditional random fields. In *FG*, 2011.
[17] U.S. Navy. *Aircraft Signals NATOPS Manual, NAVAIR 00-80T-113*. Washington, DC, 1997.
[18] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2nd edition, Nov 1999.
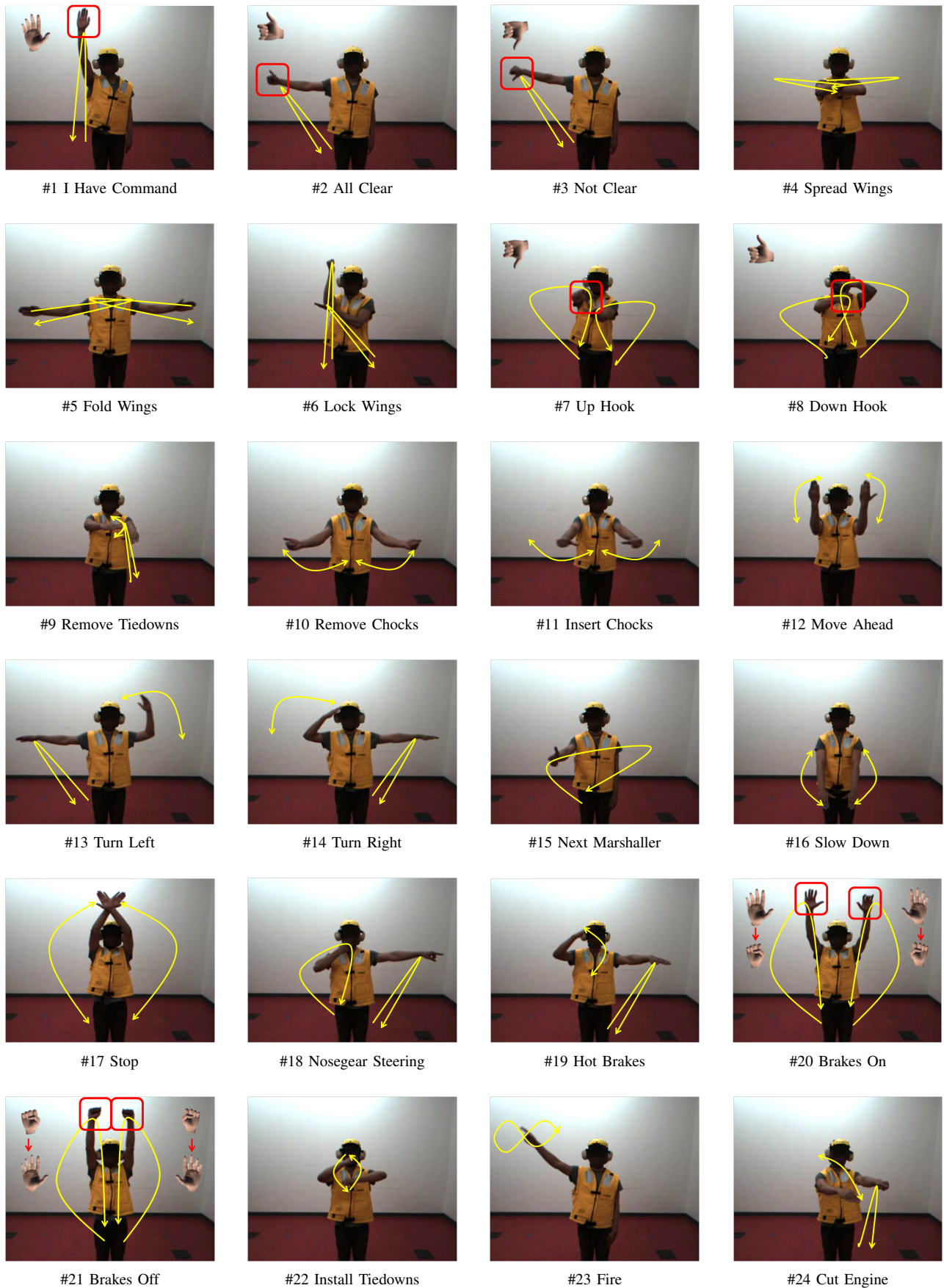
Fig. 9. Twenty-four NATOPS aircraft handling signals. Body movements are illustrated in yellow arrows, and hand poses are illustrated with synthesized images of hands. Red rectangles indicate hand poses are important in distinguishing the gesture with its corresponding similar gesture pair.