A Machine Vision-Based Gestural Interface for People With Upper Extremity Physical Impairments

Hairong Jiang, Bradley S. Duerstock, and Juan P. Wachs, *Member, IEEE*

***Abstract*—A machine vision-based gestural interface was developed to provide individuals with upper extremity physical impairments an alternative way to perform laboratory tasks that require physical manipulation of components. A color and depth based 3-D particle filter framework was constructed with unique descriptive features for face and hands representation. This framework was integrated into an interaction model utilizing spatial and motion information to deal efficiently with occlusions and its negative effects. More specifically, the suggested method proposed solves the false merging and false labeling problems characteristic in tracking through occlusion. The same feature encoding technique was subsequently used to detect, track and recognize users' hands. Experimental results demonstrated that the proposed approach was superior to other state-of-the-art tracking algorithms when interaction was present (97.52% accuracy). For gesture encoding, dynamic motion models were created employing the dynamic time warping method. The gestures were classified using a conditional density propagation-based trajectory recognition method. The hand trajectories were classified into different classes (commands) with a recognition accuracy of 95.9%. In addition, the new approach was validated with the "one shot learning" paradigm with comparable results to those reported in 2012. In a validation experiment, the gestures were used to control a mobile service robot and a robotic arm in a laboratory chemistry experiment. Effective control policies were selected to achieve optimal performance for the presented gestural control system through comparison of task completion time between different control modes.**

***Index Terms*—Condensation, dynamic time warping, gesture recognition, one shot learning, particle filter.**

I. Introduction

ASSISTIVE technologies is about finding new ways to engage cutting-edge technologies in support of individuals with physical and/or cognitive impairments. The development of technologies relying on high usability principles, exploited new communicational channels, such as eye blinking, voice, hand gestures, sip and puff, and electromyogram (EMG) as effective control modalities [1]. These channels have led to ingenious interfaces in support of the disabled [2], such as intelligent wheelchairs systems, home medical alert systems, and assistive robotic control, to mention a few [3]–[5]. These interfaces offer additional degrees of mobility and control which were not possible prior to these developments, leading to a higher quality of life and sense of independence.

Among all these interaction channels, hand gestures is a valuable alternative since it does not require the user to be tethered through cables or sensors, and it only requires learning a few customized gestures for a given task. In particular, upper extremity gesture control can serve as an important human computer interaction (HCI) modality for individuals with quadriplegia who lack hand fine motor skills. For instance, upper limb gesture control requires less targeting accuracy than joysticks, the mouse, and other continuous input devices. Likewise, the option to employ either continuous or discrete input control modes reduces the effort required for individuals with quadriplegia to perform navigational operations [6]. Unlike voice control, gesture control is effective in noisy environments [7]. In addition, for most of the cases, individuals with quadriplegia can only use gross motor instead of fine motor function to perform certain tasks [8]. Apart from other common modalities, such as keyboard and joystick that require fine motor control to hit a key or move and twist a handle, upper extremity gesture control only requires gross motor function for targeting and navigational tasks [9]. Lastly, hand gesture-based HCI is unencumbered because it does not require the user to directly contact or wear sensors as sip-and-puff and EMG-based systems [10], [11]. While not every individual with upper extremity mobility impairments can use hand gesture control reliably, for those who are able to move their arms to some degree, gesture-based HCI can be seen as a promising alternative or complement to an existing control modality.

In our previous work [12], a prototype of gesture recognition-based interface was developed for people with upper extremity mobility impairments. In the current manuscript, the tracking algorithm was greatly improved and compared with five state-of-the-art algorithms to demonstrate a better tracking performance. Further, more experimental results were provided with subjects with upper extremity mobility

Manuscript received August 14, 2012; revised April 3, 2013; accepted May 6 2013. Date of publication August 8, 2013; date of current version April 11, 2014. This work is supported in part by the National Institutes of Health through the NIH Director's Pathfinder Award to Promote Diversity in the Scientific Workforce under Grant DP4-GM096842-01. This paper was recommended by Associate Editor Y. Xiao.

H. Jiang and J. P. Wachs are with the School of Industrial Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: jiang115@purdue.edu and jpwachs@purdue.edu).

B. S. Duerstock is with the School of Industrial Engineering and the Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: bsd@purdue.edu).

Color versions of one or more of the figures in this paper are available online at http://iee explore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2013.2270226

impairments and one shot learning was employed to allow instant customization of the gestural system. Face and hand tracking under frequent self-occlusion was modeled as a multiobject tracking (MOT) problem. This problem is challenging since hands are nonrigid objects and their form varies among individuals, while performing a certain candidate gesture. Additionally, since the appearance of the left and right hand are similar for the same individual, trackers can focus on one hand or exchange positions when the hands are too close to each other. In this paper, an integrated approach was proposed to tackle the challenging problem of tracking under self-occlusion.

A. Related Work

Often, hand gesture recognition involves segmentation of the hands, tracking them through occlusion, and the classification of hand's dynamic trajectories and static pose. For vision-based real-time gesture-based interfaces for assistive technologies, robustness is a critical requirement [13] for its adoption. For hand segmentation, a commonly used method is to back-project the prebuilt skin color histogram model into new video frames. These methods are likely to fail in true world conditions, where illumination is uncontrolled and the background is cluttered. Adding depth information can relax at some extent the problem, by utilizing stereo vision [14] or other depth commodity sensors, such as Kinect [15] or Leap Motion [16].

Face and hands tracking is a special case of MOT problem. If gestures in the lexicon only carry trajectory information, (the hand shape does not convey extra information), classical tracking approaches can be adopted. For example, CAMSHIFT [17] and conditional density propagation (CONDENSATION) [18] have been shown to successfully track the hands; however, they are susceptible to lose the tracked hands when occluded by new objects, or when the scene illumination changes. Another widely used technique for object tracking is particle filters [19]. Perez *et al.* [20] integrated color-based appearance models to a particle filter framework to enhance tracking under complex background and occlusion, and then applied the particle filter framework to multiple objects tracking. Okuma *et al.* [21] further extended particle filters by incorporating a boosting detector and enabling automatic initialization of potential multiple targets. One problem of these techniques is that the interaction between the tracked objects (and occlusion) was not considered part of the main framework. When the objects interact one with the other, occlusions occur frequently. Local motion information was incorporated into a color-based particle filter framework by Kristan *et al.* [22] to solve the self-occlusion problem through object tracking. Qu *et al.* [23] combined a joint state space representation with color-based particle filter and performed joint data association in a multi-object tracking scenario. All the discussed algorithms so far, attempted to solve the MOT problem; however, they presented limited performance when tracking multiple nonrigid similar objects.

With the advent of Kinect and other 3-D sensors, hand or body tracking techniques in real-time were exploited. Eichner *et al.* [24] presented a technique to estimate the body layout of humans by using still images. Their approach is capable of estimating upper body pose in highly uncontrolled environment. Further, Yang *et al.* [25] described a method to estimating human pose from static images using body part models. By using the depth information, Shotton *et al.* [26] proposed a method to predict 3-D positions of body joints from a single depth image. They solved the pose estimation problem through a simple per-pixel classification problem. A similar method is also used by OpenNI for human body skeleton tracking. One problem of these skeleton-based tracking methods is that they work well when users are standing with their extremities extended, but suffer performance degradation for seated users with contracted limbs, which normally occurs with individuals with quadriplegia.

Only color and depth information captured from Kinect were adopted for hand tracking by Oikonomidis *et al.* [27]. They presented a method to track the full articulation of two hands that interact with each other in an uncontrolled manner. This method is effective for static gesture recognition; however, the computation cost is excessive which affects its real-time extension for gesture tracking.

One of the most widely used techniques for gesture recognition is Hidden Markov Models (HMM) [28]–[30]. Common problems with HMM approach consist of finding the optimal parameters set (e.g. initial probabilities) and trajectory spotting for gesture temporal segmentation. Black and Jepson [31] proposed a CONDENSATION-based trajectory gesture recognition algorithm that can obtain less sensitive parameters set and achieve robust tracking, yet gesture temporal segmentation was not fully addressed. Alon, *et al.* [32] applied the dynamic time warping (DTW) approach to gesture recognition and look at subgestures composition to solve the temporal segmentation problem (also known as spotting). Interaction between hands was not specifically tackled.

Recently, a new type of challenge was attracted the attention of the gesture recognition community – the One Shot Learning Challenge [33]. The one shot learning [34] consists of learning a gesture category by only observing one instance of that gesture, similar to how humans learn. In this context, Wu *et al.* [35] adopted the extended-motion-histogram image for motion feature representation and applied it to segment and classify hand gestures. Yang *et al.* [36] proposed discovering high level subactions by clustering optical flow in four dimensions (RGB-D). In our work, one shot learning provides an interesting test-bed to demonstrate the robustness of our approach, compared with the state of the art [37].

In the current paper, we also extended our method to robotic control. One advantage of using hand gestures to control robots is that it provides a natural way for navigational tasks by sending navigational information (e.g., left, right, forward, and backward commands [38]).

B. Outline of Our Approach

In this paper, an interaction model was incorporated to the color histogram-based particle filter framework to track hands through interaction and occlusion. A procedure was proposed to create dynamic motion models by DTW method and classify input gesture trajectories using the CONDENSATION

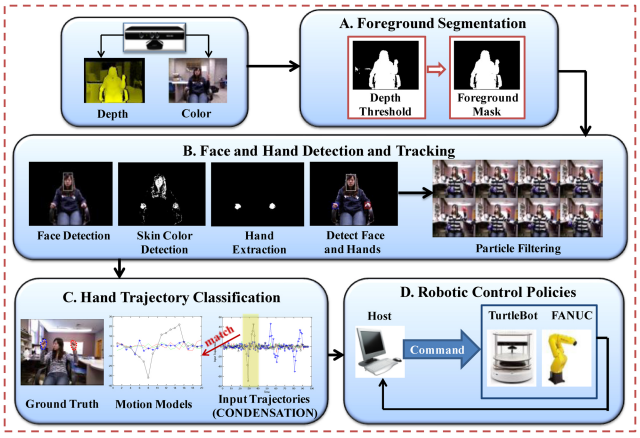


Fig. 1. System overview.

algorithm. The system was integrated in a simplistic yet robust fashion by combining CONDENSATION algorithm with an interaction model-based particle filter, which makes it suitable for human robot interaction in assistive technologies.

The contribution of this paper is three-fold:

1) prove the effectiveness of hand gestures as an alternative modality for individuals with mobility impairments by both subjective explanation and quantified results;
2) solve the frequently hand gesture interaction and occlusion problem through integration of color and 3-D spatial information as an interaction model;
3) new gestures can be created and learned through the one shot learning paradigm, leading to an almost effortless training process (a necessary attribute for subjects with severe spinal cord injuries).

The paper is organized as follows. In Section II, the architecture is presented for the gesture recognition system. In Section III, the approach suggested to track and recognize dynamic hand gestures is discussed in details. In Section IV, comparative tests and results are presented, Section V discusses and concludes the paper, and Section VI presents future work.

II. System Architecture

The architecture of the proposed system is illustrated in Fig. 1. Eight gestures were selected to constitute the gesture lexicon which in turn was used to control the robots. The machine vision-based gestural system included four parts: foreground segmentation, face and hand detection and tracking, hand trajectory classification, and robotic control policies. These parts are described in the following sections.

A. Foreground Segmentation

In foreground segmentation section, the background was ruled out from the captured frames and the whole human body was kept as the foreground.

B. Face and Hand Detection and Tracking

Face and hand detection was used to initialize the position of the face and hands for the tracking phase. After initialization, both face and hands were tracked through video sequences by the particle filter method.

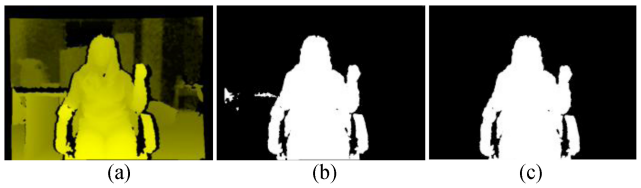


Fig. 2. Foreground Segmentation. (a) Depth image. (b) Depth threshold mask. (c) Foreground segmentation mask.

C. Hand Trajectory Classification

Hand tracking results were segmented as trajectories, compared with motion models, and decoded as commands for robotic control.

D. Robotic Control Policies

The commands decoded by gesture recognition results were sent to control the mobile robot and the robotic arm.

III. Gesture Recognition

A. Foreground Segmentation

Initially, the user's body was treated as a foreground object in order to detect the user's movements. Two steps were used to segment the foreground (refer to algorithm 1 in Table I). In the first step, the sensed image acquired by a Kinect [15] sensor was thresholded using depth information. The depth value of each pixel was defined as $D(i, j)$ with i and j indicating the horizontal and vertical coordinates of the pixel in each frame of the video sequence. An example of a depth image is shown by Fig. 2(a), where the distance between objects and the depth sensor was mapped to intensity levels. The nearer the object was to the sensor, the larger the intensity was. Two absolute depth thresholds (a low threshold T_{DL} and a high threshold T_{DH}) were custom set by the user according to their relative distance to the depth sensor. T_{DL} was set to no less than a constant which was the minimum distance that can be registered by the depth sensor (due to its physical limitations). T_{DH} was set to be the maximum distance that can be reached by the user while seated in a wheelchair.[1] In this paper, T_{DL} and T_{DH} were set to be 0.4 m and 2.0 m to achieve an optimal performance for segmentation. A mask image [Fig. 2(b)] was generated by keeping the pixels with a depth value between the two thresholds while discarding the others. In the second step, the region (blob) with the largest area (denoted as T_{SH}) was extracted from the mask image. All the remaining blobs with an area smaller than T_{SH} were discarded [Fig. 2(c)]. If the extracted region contained an object that was not part of the user's body, it would be discarded in a later stage since tracking was achieved based on both color and spatial information.

[1]These values are selected since they resulted in the best performance; other thresholds can be used and the impact on the overall performance is likely to be negligible.

TABLE I
FOREGROUND SEGMENTATION ALGORITHM

Algorithm 1: Foreground Segmentation

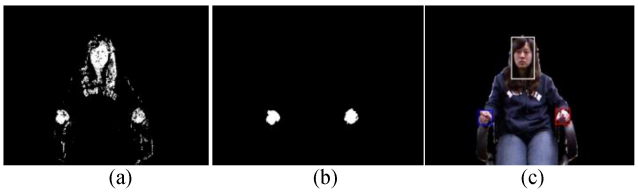
Input: Low depth threshold T_{DL}; High depth threshold T_{DH}; pixel value of depth Image $D(i, j)$;

Output: pixel value of mask image $D_1(i, j)$; pixel value of foreground mask image $D_2(i, j)$.

$$D_1(i,j) = \begin{cases} 1: & T_{DL} \le D(i,j) \le T_{DH} \\ 0: & otherwise \end{cases}$$

$T_{SH} = max(Area(B_i))$ //B_i is the ith blob in the mask image D_1

$$D_2(i,j) = \begin{cases} 1: & D_1(i,j) \in B_i \ \& \ Area(B_i) == T_{SH} \\ 0: & otherwise \end{cases}$$



(a) (b) (c)

Fig. 3. Face and hand detection. (a) Skin color detection. (b) Hand extraction. (c) Face and hand localization.

B. Face and Hand Detection

In this section, the centroids of the face and hand regions were extracted to initialize the tracking stage. Two 3-D histograms—a skin and a nonskin color histogram were created using the Compaq database [39] and the HSV color space to achieve higher robustness for skin color detection (referred to [12] for a detailed description). The mask image obtained from histogram back-projection is shown as in Fig. 3(a). To obtain the hand regions without the face, a face detector [40] was adopted [Fig. 3(c)] to remove the face region from the target image. Two largest blobs in the target image were then selected as hand regions [Fig. 3(b)]. The centroids of the hands were obtained by computing the first moment of the two blobs. This hand detection procedure was only used to provide automatic initialization to the particle filter tracking procedure. Afterwards the hands positions were continuously tracked by the particle filter.

C. Face and Hand Tracking

A 3-D particle filter framework based on color, depth, and spatial information was used to track the face and hands through video sequences. A detailed description of the particle filter algorithm was illustrated in [20], [41], and [42]. The equation of particle filtering is

$$p(X_t|Z_{1:t}) = k.p(Z_t|X_t) \int^p (X_t|X_{t-1})\, p(X_{t-1}|Z_{1:t-1})\, dx_{t-1} \tag{1}$$

where X_t is the process state at time t, $Z_{1:t} = \{Z_1, \ldots Z_t\}$ denotes the set of observations from time one to t, $p(X_t|Z_{1:t})$ and $p(Z_t|X_t)$ expresses the posterior and prior distribution at time t, $p(X_t|X_{t-1})$ is the transition probability of the system at state X_t given that the previous state was X_{t-1}, and k is a normalization factor to normalize the sum of all posterior probabilities to one. In the particle filter algorithm, N weighted particles can be used to approximated the posterior as: $p(X_{t-1}|Z_{1:t-1}) \approx \left\{X_{t-1}^r, \omega_{t-1}^r\right\}_{r=1}^N$, where ω_{t-1}^r denotes the weight of the particle r at time *t-1*. After propagation, the tracker output at time t can be approximated by the expectation of the process state: $\hat{X}_t \approx E[X_t|Z_{1:t}] = \sum_{r=1}^{N} \omega_t^r X_t^r$. Thus, (1) is converted to

$$p(X_t|Z_{1:t}) \approx k.p(Z_t|X_t) \sum_{r=1}^{N} \omega_t^r\, p\left(X_t^r|X_{t-1}^r\right). \tag{2}$$

The particles were initialized by using the centroids of face and hands calculated in Section III-B.

The particle filter tracking process consists of three main phases: predicting, measuring and resampling. In the proposed system, for the predicting phase, a second order autoregressive (AR) model [as in (3)] [20], [41] was selected to model the dynamic motion of each particle

$$X_t^r = A_1\left(X_{t-1}^r - X_0^r\right) + A_2\left(X_{t-2}^r - X_0^r\right) + X_0^r + B\nu_t \tag{3}$$

where $\nu_t \sim N(0, \Sigma)$ is a Gaussian distribution with zero mean and variance matrix Σ, X_0 is the original particle coordinate, A_1, A_2, and B are the optimal parameter matrices that can best match the real motion of the tracked object, X_t^r is the state of the particle r at time t. In this paper, a 3-D particle filter tracking was adopted. The state of particle r at time t is written as: $X_t^r = [x_t^r, y_t^r, z_t^r, s_t^r, x_{t-1}^r, y_{t-1}^r, z_{t-1}^r]$, where s_t is the scale of object at time t, x_t^r, y_t^r, z_t^r are the 3-D coordinates of particle r at time t, and $x_{t-1}^r, y_{t-1}^r, z_{t-1}^r$ are the 3-D coordinates of particle r at time *t-1*. For the measuring phase, the selection of the observation model determines the weight of the particles. Many appearance-based models, such as contour, edge, and piece-wise, were used in object tracking. Color-based preprocessing using HSV space can facilitate the extraction of the aforementioned features for face and hands tracking. As explained earlier, the initial phase of the face and hands were determined by the combination of depth-based thresholding and image processing techniques. The extracted face and hands regions were used to compute the reference HSV histogram models (H_f^*, H_{h1}^*, and H_{h2}^*) for tracking initialization. During the resampling phase, each particle, assigned in the predicting phase, was reweighted by the observation likelihood function. For every hypothesized face or hand location of a particle r, the candidate histograms were computed as H_f^r, H_{h1}^r, and H_{h2}^r. The Bhattacharyya distance [20] D was used to measure similarity between reference and candidate histograms as

$$D_i\left(H^*, H^r\right) = \left[1 - \sum \sqrt{H_i^* H_i^r}\right]^{\frac{1}{2}} \tag{4}$$

where $H_i^* = H_f^*, H_{h1}^*, or H_{h2}^*$ and $H_i^r = H_f^r, H_{h1}^r, or H_{h2}^r$. The observation likelihood function can be written as

$$p(Z_t|X_t) \propto \exp(-\lambda_1(\mathrm{D_i^r})^2) \tag{5}$$

where λ_1 measures the variance of the HSV histogram. Equation (5) can be rewritten by adding a normalization factor k to normalize the sum of all particles' weight to one, obtaining

$$p(Z_t|X_t) = k.\exp(-\lambda_1(D_i^r)^2). \tag{6}$$

TABLE II
HAND TRACKING THROUGH INTERACTION AND OCCLUSION

Algorithm 2: 3D Particle Filter tracking

Input: Reference HSV histogram models H_f^*, H_{h1}^*, and H_{h2}^*; Optimal parameter $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$.
Output: Centroids and the associated bounding box of the face and hands.
1. Initialize:
//Initialize particle states and weight for face and both hands as:
$x_0^i = x_0^*, \ \omega_0^i = \frac{1}{n}, \quad$ where $i = 1,\ldots,n$
2. Predict, Measure and Resample:
//Select k;
for i=1,2,3 //(1-face, 2-right hand, 3-left hand)
for r=1 to N
$x_{i,t}^r = A_1\left(x_{i,t-1}^r - x_0^i\right) + A_2\left(x_{i,t-2}^r - x_{i,0}^r\right) + x_{i,0}^r + B\nu_t$
//Compute candidate histograms H^r
$D_i(H^*, H^r) = \left[1 - \sum\sqrt{H^* H^r}\right]^{\frac{1}{2}}$
//Calculate the weight:
$\omega_{i,t}^r = k\ exp\ (-\lambda D_{i,t}^2)$
end for
Normalize the weights and resample the particles
Estimate $\hat{x}_{i,t} = \sum_{r=1}^{N} \omega_{i,t}^r x_{i,t}^r$
//Check interaction
if interactions happens for object *i* and *j*
for q=1,…,N
//computer interaction likelihood ψ1 and ψ2:
Compute $\psi_1{}_{i,t}^{q}\left(X_{i,t}^{q}, X_{j,t}^{q}\right)$ and $\psi_2{}_{i,t}^{q}\left(X_{i,t}^{q}, X_{j,t}^{q}\right)$ using (8) and (9)
//Calculate the weight:
$\omega_{i,t}^q = \omega_{i,t}^q \cdot \psi_1{}_{i,t}^{q} \cdot \psi_2{}_{i,t}^{q}$
end for
Normalize the weights and resample the particles.
Estimate $\hat{x}_{i,t} = \sum_{r=1}^{N} \omega_{i,t}^r x_{i,t}^r$
end if
end for

D. Hand Tracking Through Interaction and Occlusion

Color-based particle filter tracking was effective for multiple independent objects tracking when the objects did not interact or occlude each other. However, if interaction or occlusion occurs, multiple independent particle filters can be used. Standard MOT with interaction and occlusion suffers from the false merging and false labeling problems [23]. The false merging problem denotes the situation that the tracker shift from the object being tracked to a different object that has higher observation likelihood. Conversely, the false labeling problem denotes the situation that the objects being tracked exchange their labels after interaction or occlusion occurred. In the proposed system, the face and both hands were tracked.

In this paper, two models were constructed to solve false merging and false labeling problems separately. The first model was called the competition potential (CP) model. The idea of this model comes from the joint Markov random fields (MRF) theory [42]. The likelihood function for CP model is defined as $\psi_1\left(X_{i,t}, X_{j,t}\right)$, which represents the pairwise interaction potential of the MRF [43]. The second model is called motion consistency (MC) model. The likelihood function for MC model is defined as $\psi_2\left(X_{i,t}, X_{j,t}\right)$, which is based on the assumption that a particle region that has similar motion information to the previous state of that particle will have higher probability than a particle region that has distinct motion information.

For CP model, as in [43], we have $p\left(X_t|X_{t-1}\right) \propto \prod_{i,j\in E} \psi_1\left(X_{i,t}, X_{j,t}\right) \prod_{i,j\in E} \psi_1\left(X_{i,t}, X_{j,t}\right)$. The particle filter function (2) can be rewritten as

$$p\left(X_t|Z_{1:t}\right) = k.p\left(Z_t|X_t\right) \prod_{i,j\in E} \psi_1\left(X_{i,t}, X_{j,t}\right) \sum_r \omega_{t-1}^r \prod_i p\left(X_{i,t}|X_{i,t-1}^r\right). \tag{7}$$

The likelihood function for CP model is then defined as

$$\psi_{1_{i,t}}^r\left(X_{i,t}, X_{j,t}\right) = \beta_1 \exp\left(-\frac{\lambda_2}{d\left(X_{i,t}^r, X_{j,t}\right)^2}\right) \cdot \exp\left(-\lambda_3 d\left(X_{i,t}^r, X_{j,t-1}\right)^2\right) \cdot \exp\left(-\frac{\lambda_4}{d_z\left(X_{i,t}^r - X_{j,t}\right)^2}\right) \tag{8}$$

where $d\left(X_{i,t}^r, X_{j,t}\right)$ denotes the 2-D Euclidean distance metric between two objects, $d\left(X_{i,t}^r, X_{i,t-1}\right)$ represents a distance metric between the previous and current centroid of object i, $d_z\left(X_{i,t}^r - X_{j,t}\right)$ represents the difference of depth value between two objects, and β_1 is a normalization factor so the sum of all particles' weight is one. MC model was used to solve the false labeling problem. The 3-D motion information was incorporated into the original likelihood function to increase the robustness of the method. We adopted a compact expression of the likelihood function similar to [23], which integrates the magnitude and direction information of motion as (9). Instead of using 2-D, 3-D motion features are used to compute the motion information of the hand movement. The likelihood function for the MC model is defined as

$$\psi_{2_{i,t}}^r\left(X_{i,t}, X_{j,t}\right) = \beta_2 \cdot \exp\left(-\lambda_5(\theta_t^r)^2\right) \cdot \exp\left(-\lambda_6\left(A_t^r - A_{ref,t}\right)^2\right) \tag{9}$$

where A_t^r and $A_{ref,t}$ represent the norm of 3-D motion vector and reference motion vector (can be computed by the difference of the current and the previous 3-D position vector) of particle r at state t, respectively. θ_t^r is the angle between the 3-D motion vector of particle r and the reference vector and β_2 is a normalization factor to normalize the sum of all particles' weight to one. This likelihood function assumes that a particle region that has a similar motion to the previous state will have a larger weight than one with a different motion. When the objects' observations do not interact with each other, the approach suggested behaves as if multiple independent trackers were applied to the objects [Fig. 4(a)]. However, when the objects' observations interact (e.g. partial or complete occlusion occurs), the conventional particle filter framework is extended [Fig. 4(b) and (c)]. The decision of when the objects interact is made based on the interdistance between the hands. When this distance is below a certain threshold, the system switches to the interaction model (Fig. 4). To find the optimal threshold, a histogram of the number of tracking frame errors at each distance is obtained (Fig. 5) and the threshold is selected so the tracking errors are minimized when the interaction model is activated. Note, a dramatic decrease of error, at the distance around 50 pixels at which hand interaction frequently occurred. The threshold T was determined according to the distribution of errors in

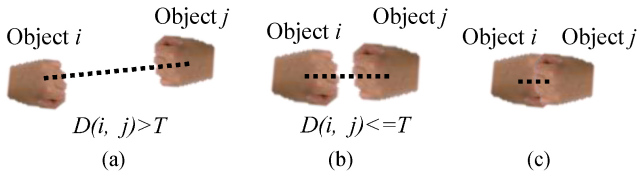


Fig. 4. Dynamic motion analysis. (a) Independent object tracking. (b) Interaction model added. (c) Objects occlusion occurs.

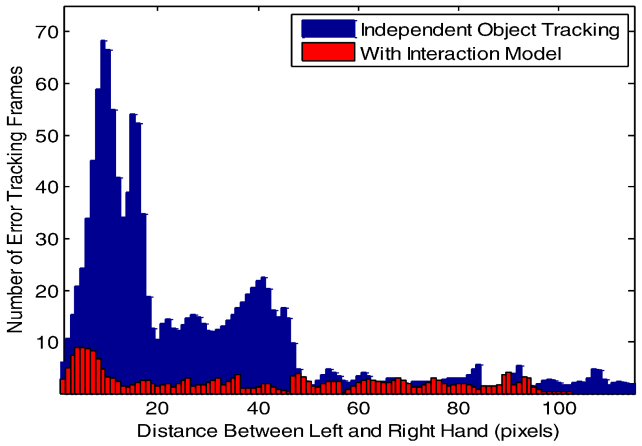


Fig. 5. Number of error tracking frames for each distance before and after the addition of interaction model.

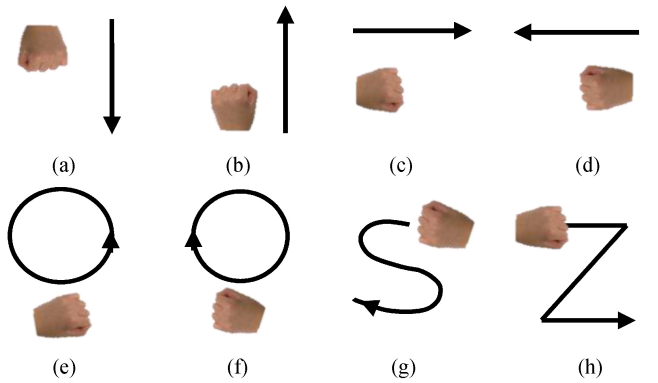


Fig. 6. Gesture lexicon. (a) Downward. (b) Upward. (c) Rightward. (d) Leftward. (e) Counter-clockwise circle. (f) Clockwise Circle. (g) S. (h) Z.

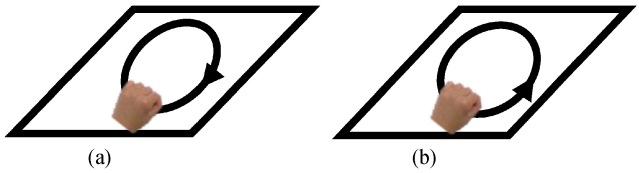


Fig. 7. Extended lexicon. (a) Clockwise circle in horizontal plane. (b) Counter-clockwise circle in horizontal plane.

the histogram. The extension models were given through (8) and (9). The parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$, and λ_6 in (6), (8), and (9) were optimized by utilizing a neighborhood search method [38]. The algorithm for hand tracking during interaction and occlusion is shown in Table II.

E. Gesture Lexicon

A gesture lexicon was designed such that users will physical impairments can perform the gestures with minimal effort. These gestures were found through a series of interviews conducted with subjects with upper mobility impairments. Borg Scale [44] was used to rank the physical stress required to perform a gesture by participants with upper mobility impairments. An eight-gesture lexicon (Fig. 6) was then constructed by analyzing the Borg Scale results collected from the subjective rankings and selecting those corresponding to the least required effort. For detailed description of the process for gesture lexicon construction refer to [45].

F. Hand Trajectory Classification

For each frame in the video sequence, the centroids of the face and hands were obtained from the tracking stage. The motion model for each gesture trajectory was created based on the data collected from gestures performed by ten subjects. Two of the pool of ten subjects were quadriplegic due to a cervical spinal cord injury.

Even though the trajectories for each gestures performed by different subjects or the same subject in different instances may look similar, the precise duration of each subtrajectory within the trajectory were different. To normalize the trajectories (temporal alignment), DTW was employed [46]. The velocities' components in horizontal, vertical and depth directions of both hands were selected as the feature components for each motion model [41]. The procedure to construct the motion models is described in our previous work [12].

The CONDENSATION algorithm [31] was employed to classify hand gesture trajectories in the lexicon (as in Fig. 6). It employs a set of weighted samples to fit the observed data. The original algorithm in [31] was extended to work for two hands. The original expression $S_t = (\mu, \phi, \alpha, \rho)$ (the state at time t) was extended to

$$\begin{aligned} S_t &= (\mu,\ \phi^i, \alpha^i,\ \rho^i) \\ &(\mu, \phi^{\text{right}}, \phi^{\text{left}}, \alpha^{\text{right}}, \alpha^{\text{left}}, \rho^{\text{right}}, \rho^{\text{left}}) \end{aligned} \tag{10}$$

where, μ is the index of the motion models, ϕ is the current phase in the model, α is an amplitude scaling factor, ρ is a time dimension scaling factor, and $i \in \{\text{right hand, left hand}\}$. The gestures in the lexicon (as in Fig. 6) were spotted using a rest position gesture [when the subjects put their hands on the arm rest (neutral position) with no hand movement]. A dynamic motion model was created for the rest position gesture. The segment between two recognized discontinuous rest position gestures is treated as a spotted gesture.

G. Gesture Customization (One Shot Learning)

One of the objectives of our prototype follows the "came as you are" paradigm [13], where new gestures can be learned by the system automatically or by observing only one instance of it. The reason for this is to reduce the level of effort involved in the training phase of the system. In this section, we validated our approach in the context of one shot learning to assess the ability of the system to generalize learning from very few observations. A Savitzky–Golay smoothing filter [47] was added to smooth the 3-D trajectories during the creation of the

TABLE III
HAND TRACKING PERFORMANCE

Method	False Merging (6080 Frames)	False Labeling (157 Interactions)	Tracking Accuracy (%)	Particle Number	Number of Body Parts
MCMC [41]	568	33	74.87	100	--
MI [23]	323	20	82.58	100	--
ETH (color) [24]	11	4	74.80	--	6
ETH (depth) [24]	0	3	58.25	--	6
Body Part (color) [25]	72	33	64.80	--	26
Body Part (depth) [25]	220	14	27.58	--	26
Kinect Skeleton [48]	0	3	73.87	--	16
CPMC(Proposed)	1	4	97.52	100	--

motion models. Two new gestures (Fig. 7) were added to the lexicon to offer another degree of navigational control.

IV. EXPERIMENTS AND RESULTS

A. Experiment 1: Hand Tracking Performance

A dataset of 16 videos (4 subjects x 4 activities) was used to evaluate the proposed tracking algorithm. The videos were captured with a Kinect™; camera at 30 Hz using an image size of 640×480. Among the four subjects, three were able-bodied individuals and one was an individual with Cervical-6 level quadriplegia. The four activities performed by the subjects were: 1) holding a cup; 2) clapping hands; 3) moving one hand up and down (to occlude the other hand); and 4) rotating two hands forward and backward (to occlude each other). The total number of frames for all the videos was 6080, while the total number of interactions between the two hands was 157. The total number of frames of each video corresponding to each of the four activities was: 930, 2230, 1320, and 1600, respectively (two sample sequences are shown in Figs. 17 and 18 in appendix). The ground truth position of the left and right hands in each video was provided by manually hand labeling.

The local likelihood $p(z_t^i|x_t^i)$ was calculated using the 3-D color histograms and two interaction models as the algorithm mentioned in Table II. The performance of the proposed method—competition potential and motion consistency (CPMC)–was compared to other existing methods, such as 1) Markov Chain Monte Carlo (MCMC)-based particle filter tracking [43]; 2) magnetic-inertial-based particle filter tracking [23]; 3) ETH skeleton tracking based on color or depth frames [24]; 4) body-parts tracking based on color or depth frames [25]; and 5) Kinect OpenNI SDK skeleton tracking [48]. For methods 1, 2, and the proposed method, 100 particles were used for face and each hand tracking. For methods 3, 4, and 5, the number of body parts (segments of a human body, i.e., hand, head, leg, and part of the arms) being tracked were: six, 26, and 16, respectively. For method 3, only the upper body parts were tracked. Six parts were used. Since the focus was on hand tracking, the results of two hands interaction for all the activities are shown as in Table III.

The tracking performance of these algorithms was evaluated by employing three metrics: false merging, false labeling and tracking accuracy. The false merging is defined as the situation where the tracker of one hand occupies 80% of the area of the other hand. The false labeling is defined as the situation where the trackers of both hands change positions during/after interaction or occlusion. The tracking accuracy is defined by

$$\text{Tracking Accuracy} = \frac{\text{total number of, (true positives +true negative)}}{\text{total number of tracked frames}} \quad (11)$$

where a true positive is defined as the situation whereas a target object is present and the tracker was able to find it. True negatives are instances where the target object is not present and the tracker also agreed that the object was absent [47]. Table III shows that the proposed algorithm (CPMC) exhibits the best performance for the interaction and occlusion conditions among the three particle-based methods. There is a marginal decrease in the algorithm speed. When there is no interaction or occlusion occurring, CPMC has the same speed as MCMC and MI approaches. Comparing to the skeleton tracking [24], [48] and body part tracking method proposed by [25], the proposed method obtained higher tracking accuracy. Since our targeting user group is individuals with upper mobility impairments, two challenges exist for hand tracking in the proposed system that could not be tackled very well by skeleton-based or other articulated pose tracking method. One challenge is that the users with upper extremity mobility impairments need to sit most of the time on a wheelchair and perform limited space hand gestures. Their hands could be very close to the body when they perform the gestures. The tracking method based purely on depth information could easily lose track when the hands are so close to the torso [24], [25]. The second challenge is that most of the wheelchairs have armrest, which can be easily confused with human arms and hands. This can explain why the skeleton-based tracking (method 3 and 5), and the articulated pose tracking (method 4) does not work well for our dataset. The values of the performance metrics "false merging" and tracking accuracy versus the distance between left and right hands are shown in Figs. 8 and 9. From Fig. 8, we can see that the proposed approach outperforms method 1, 2, 4, and

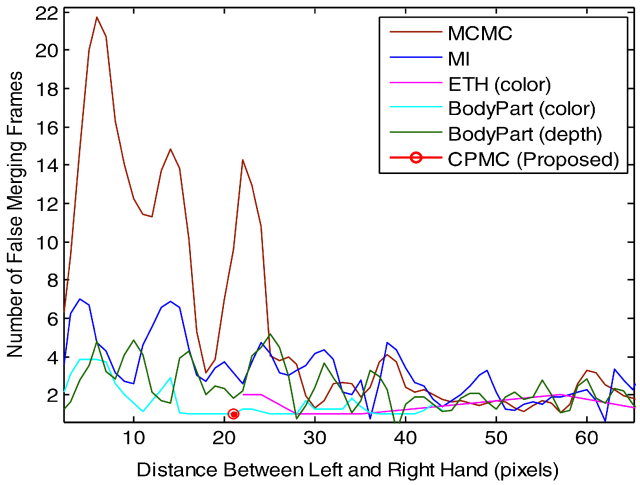


Fig. 8. Number of false merging occurred versus hand distance.

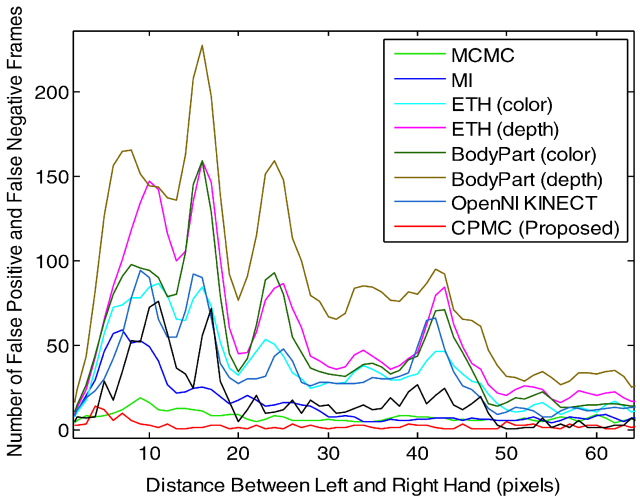


Fig. 9. Number of false positive and false negative versus hand distance.

method 3 (with color images). Additionally, the result of the proposed approach (one false merging frame) was very close to the results of method 3 (with depth images) and method 5 (no false merging occurred). In Fig. 9, the total number of false positive and false negative frames versus the hands' interdistance was presented for each method. This figure shows that the proposed approach outperformed all the other state of art algorithms, since it displayed the fewest number of false positive and false negative frames among all the algorithms for nearly all distances.

B. Experiment 2: Gesture Recognition Performance

The motion models were constructed using the DTW algorithm. The velocities (3-D directions) of right and left hands were used as the main feature components. The gesture lexicon in Fig. 6 was adopted, and those gestures were used to create spatio-temporal trajectories that later were classified by the gesture-based recognition system.

The system was validated by eight able-bodied subjects and two subjects with quadriplegia due to cervical spinal cord injuries aged around 24–40. The ten subjects performed

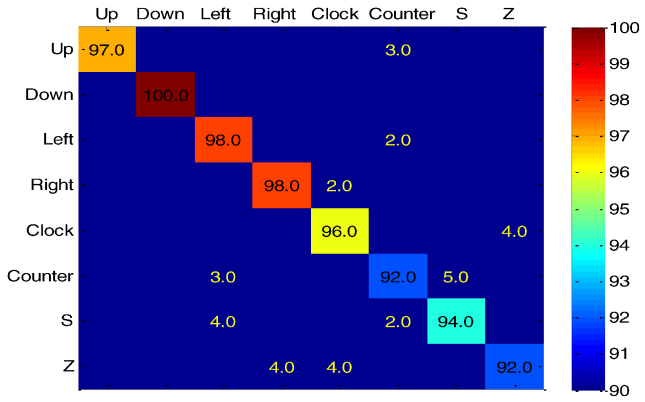


Fig. 10. Confusion matrix with window size of w = 19.

TABLE IV
GESTURE RECOGNITION PERFORMANCE

Method	Basic	PCA	DTW	HMM	CONDENSE
Accuracy (%)	54.6	66.0	67.4	94.2	95.9

all the gestures in the lexicon each ten times (8 gestures x 10 subjects x 10 repetitions). Ten sessions were used for cross validation for each gesture (k-fold with k = 10). In each session, 720 observations (8 gestures x 9 subjects x 10 repetitions) were used for training and 80 gestures (8 gestures x 1 subject x 10 repetitions) were used for testing. This cross validation resulted in an average accuracy of 95.9%. A confusion matrix was computed and shown in Fig. 10 (with a temporal window size of w = 19). Confusions occurred when the subjects performed a gesture mistakenly in a single direction or not enough motion was exhibited as expected in other directions. Other cases of misclassification occurred when two gestures shared similar subtrajectories (i.e, counter clock and S gestures).

The recognition performance for the CONDENSATION algorithm with our training procedures (CONDENSE) was compared to four other existing state-of-the-art recognition algorithms: 1) Basic motion [50]; 2) Motion-based PCA [51]; 3) DTW [52]; and 4) HMM [28]. After applying each gesture recognition method to our data set, the results shown in Table IV were obtained. The confusion matrices for the different methods are shown in Figs. 11, 12, 13, and 14, respectively. Method 1, 2, and 3 used motion information to recognize hand gestures, while 4 and the CONDENSATION method recognized hand gestures by extracting and classifying hand trajectories. The comparison results demonstrate a high recognition accuracy for the trajectories classification-based method. HMM-based recognition method can get comparable results as the method used in our paper.

C. Experiment 3: One Shot Learning Performance

One instance (one repetition by a subject) for each gesture in the lexicon was used for training and remaining observations were used for testing. Ten sessions were used for cross validation for each gesture (k-fold with k = 10). In each session, ten observations (10 gestures x 1 subjects x 1 repetitions) were

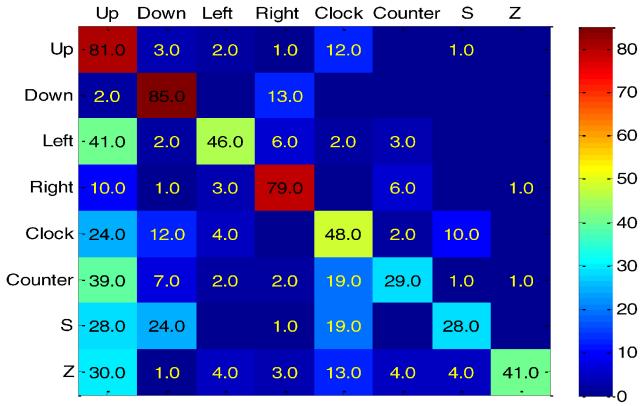


Fig. 11. Confusion matrix for basic motion method.

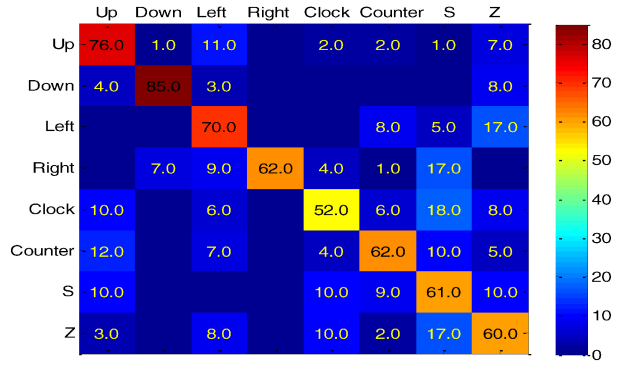


Fig. 12. Confusion matrix for PCA method (with 12 principal components)

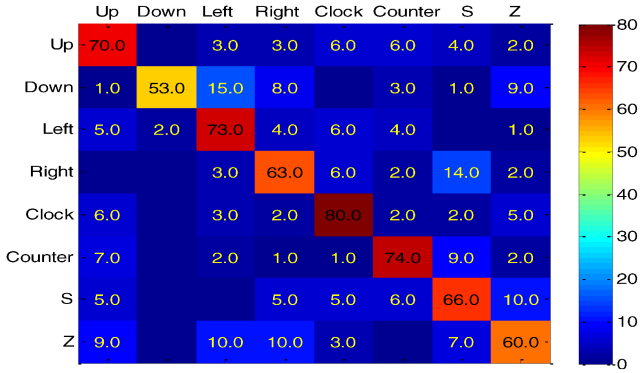


Fig. 13. Confusion matrix for DTW method.

used for training and 740 gestures (8 gestures x 9 subject x 10 repetitions and 2 gestures x 1 subjects x 10 repetitions) were used for testing. This cross validation resulted in an average accuracy of 82.78%. A confusion matrix was computed and is shown in Fig. 15 (with a temporal window size of w = 19). The recognition accuracy found is comparable to those reported in the ChaLearn Competition [33] in 2012 (fourth place in the competition).

D. Experiment 4: Robotic Control Performance

Since the first remotely driven robotic arm developed by Goldberg *et al.* [53] for gardening tasks (Telegarden), there has been an extensive wave of remote labs enabling users to perform lab experiments without the need to physically

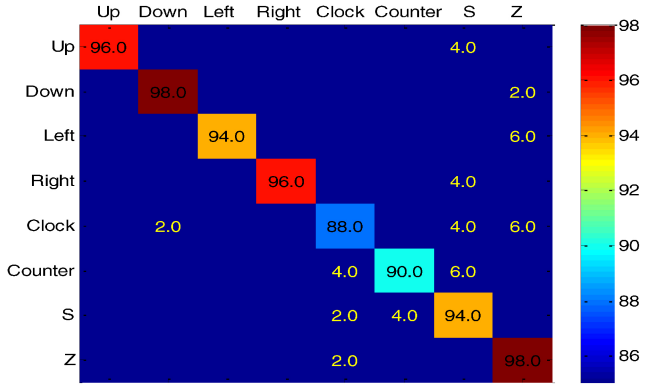


Fig. 14. Confusion matrix for HMM method.

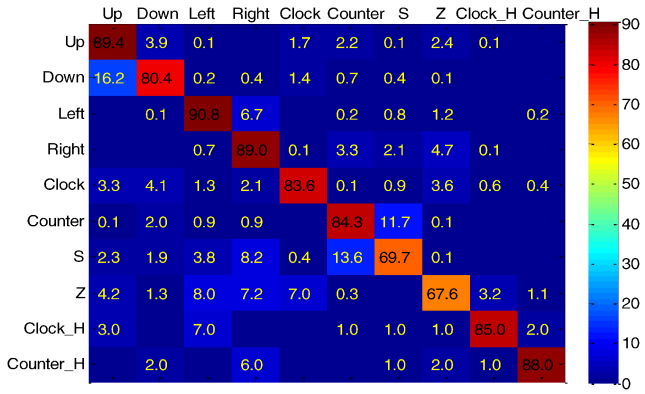


Fig. 15. Confusion matrix for one shot learning (window size w = 19).

attend them. Some examples include a work-cell with a 6 axes robot for conducting experiments remotely [54]; a LEGO mobile robotic platform for experimenting with autonomous navigation [55]; and a remote laboratory on robotics was developed at University of Siena called TeleTab [56].

A chemistry laboratory-based experiment was performed by five subjects including two individuals with quadriplegia due to a cervical spinal cord injury and three able-bodied individuals. In the laboratory case study experiment, a mobile robot was controlled by the gesture algorithm to transport a beaker to a position near a robotic arm. The robotic arm was activated by the operator to add a reagent to the beaker and then, the mobile robot was brought back to its original position. The gestures (a)–(h) (from the lexicon in Fig. 6) were used and mapped to the commands: change mode, robotic arm action, go forward, go backward, turn left, turn right, stop, and enable robotic arm. The two robots were controlled by three modes—discrete, continuous, and hybrid mode (discrete plus continuous mode). In discrete mode, for each issued command, the mobile robot moved a fixed increment of distance. While in continuous mode, the mobile robot responded to a given command, until the stop command was issued. To switch between the discrete and continuous mode one distinctive gesture (upward) was used. In the experiment, the discrete, continuous and hybrid (continuous plus discrete) control modes were each tested five times by all subjects. The resulting average task completion times were 241.8, 134.7, and 169.6 seconds, for the discrete,

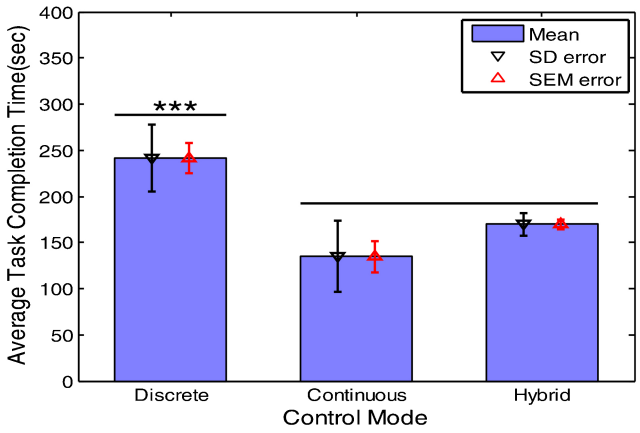


Fig. 16. Average task completion time, unpaired t-test, p<0.001.

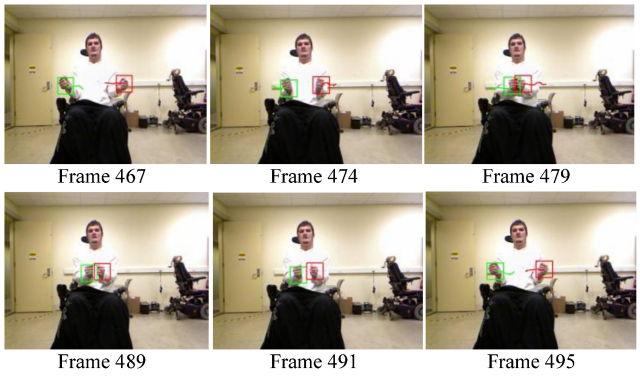


Fig. 17. Hand tracking sequence for clapping hands activity.

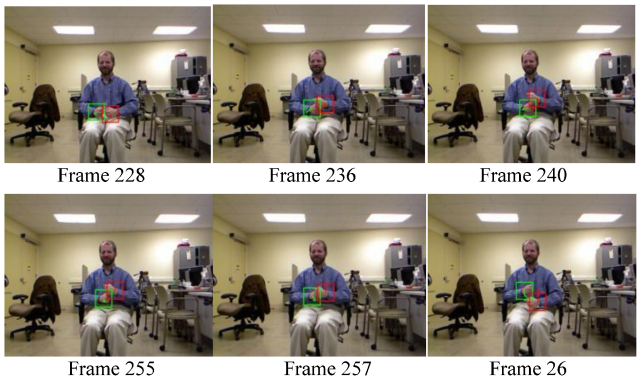


Fig. 18. Hand tracking sequence for moving one hand up and down activity.

continuous and hybrid mode, respectively (Fig. 16). From the results, the completion time of discrete mode took longer time than the continuous. Continuous and hybrid modes require commands to be issued only when the robot needs to change directions or stop, therefore fewer operations were required for continuous and hybrid modes than for discrete mode for the task observed.

V. Conclusion

A machine vision-based gestural interface was developed for individuals with upper extremity physical impairments. Since skin and nonskin color histogram models were used to initialize the face and hands' centroid, the performance of the system may be affected when the users wear short sleeves. In addition, it was expected that users will be seated within the working distance to range specified by the Kinect sensor. An interaction model was incorporated into the color-based particle filter framework for hand tracking. When there was no interaction between the face and hands, multiple independent particle filters tracked the users' movements. When interaction was present, the multiple independent particle filter trackers were combined with an interaction model to solve false merging and false labeling problems. A comparison between our proposed approach (CPMC) and five state-of-the-art algorithms demonstrated that our approach can achieve robust performance for hand tracking through interaction and occlusion conditions. The proposed tracking strategy can obtain significantly better performance than the other three methods for both false merge and false labeling problems in hand tracking through interaction and occlusion. Yet, improvements are still needed for false labeling solving. A training procedure was proposed to obtain motion models for each gesture in the lexicon. The CONDENSATION algorithm with the proposed training procedure was used and compared with four other recognition algorithms to classify bimanual gestures. Results showed that HMM-based recognition methods may deliver comparable results to our method. Thus, higher recognition could be achieved by using trajectories classification-based method. The gesture recognition algorithm designed was found to reach a recognition accuracy of 95.8%. One shot learning was applied in this paper to customize gestures and reduce the number of repetitions required to train/teach the system to a minimum (one observation). The results obtained were comparable to the state-of-the-art one shot gesture recognition algorithms presented in the ChaLearn Challenge [33].

A simulated laboratory task experiment was conducted, a typical biomedical lab procedure with the help of two robots, which were controlled through a gestural interface. Subjects with upper extremity physical impairments can successfully use the machine vision-based gestural interface to control the two robots. It was found that the proposed gestural interface was robust enough to support the completion of this task for subjects with upper extremity mobility impairments. In addition, three modes of operation were compared: discrete, continuous, and hybrid. Results showed that the continuous mode required the least average task completion time, while the discrete control mode requires the most. Therefore, the authors recommend to use continuous control mode in general, and to use discrete mode only when the robot is very near to the target, for precise location and manipulation.

VI. Future Work

Future work for this paper may include: 1) develop more effective and robust algorithms to solve false merge and false labeling problems of hand tracking through interaction and occlusion; 2) extend the laboratory task to increase the pool of participating users. Ideally, users with physical impairments

can participate and provide feedback about the usability, learning and adaptability to the interface suggested.

APPENDIX

The video sequences of two activities for hand tracking through interaction are shown in Figs. 17 and 18.

REFERENCES

[1] M. R. Ahsan, "EMG signal classification for human computer interaction: A review," *Eur. J. Sci. Res.*, vol. 33, no. 3, pp. 480–501, 2009.

[2] J. A. Jacko, "Human–computer interaction design and development approaches," in *Proc. 14th HCI Int. Conf.*, 2011, pp. 169–180.

[3] I. H. Moon, M. Lee, J. C. Ryu, and M. Mun, "Intelligent robotic wheelchair with EMG-, gesture-, and voice-based interface," *Intell. Robots Syst.*, vol. 4, pp. 3453–3458, 2003.

[4] M. Walters, S. Marcos, D. S. Syrdal, and K. Dautenhahn, "An interactive game with a robot: People's perceptions of robot faces and a gesture-based user interface," in *Proc. 6th Int. Conf. Adv. Computer–Human Interactions*, 2013, pp. 123–128.

[5] O. Brdiczka, M. Langet, J. Maisonnasse, and J. L. Crowley, "Detection human behavior models from multimodal observation in a smart home," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 4, pp. 588–597, Oct. 2009.

[6] M. A. Cook and J. M. Polgar, *Cook & Hussey's Assistive Technologies: Principles and Practice,* 3rd ed. Maryland Heights, MO, USA: Mosby Elsevier, 2008, pp. 3–33.

[7] G. R. S. Murthy, and R. S. Jadon, "A review of vision based hand gesture recognition," *Int. J. Inform. Technol. Knowl. Manage*., vol. 2, no. 2, pp. 405–410, 2009.

[8] D. Debuse, C. Gibb, and C. Chandler, "Effects of hippotherapy on people with cerebral palsy from the users' perspective: A qualitative study," *Physiotherapy Theory Practice*, vol. 25, no. 3, pp. 174–192, 2009.

[9] J. A. Sterba, B. T. Rogers, A. P. France, and D. A. Vokes, "Horseback riding in children with cerebral palsy: Effect on gross motor function," *Develop. Med. Child Neurology*, vol. 44, no. 5, pp. 301–308, 2002.

[10] K. L. Kitto, "Development of a low-cost sip and puff mouse," in *Proc. 16th Annu Conf. RESNA,* 1993, pp. 452–454.

[11] Y. H. Yin, Y. J. Fan, and L. D. Xu, "EMG and EPP-integrated human–machine interface between the paralyzed and rehabilitation exoskeleton," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 4, pp. 542–549, Jul. 2012.

[12] H. Jiang, J. P. Wachs, and B. S. Duerstock, "Facilitated gesture recognition based interfaces for people with upper extremity physical impairments," in *Proc. Pattern Recogn., Image Anal., Comput. Vision, Applicat.*, 2012, pp. 228–235.

[13] J. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand gesture applications: Challenges and innovations," *Commun. ACM, Cover Article,* vol. 54, no. 2, pp. 60–71, 2011.

[14] Z. Li and R. Jarvis, "A multimodal gesture recognition system in a human–robot interaction scenario," in *Proc. IEEE Int. Workshop Robotic Sensors Environments*, 2009, pp. 41–46.

[15] E. A. Suma, B. Lange, A. Rizzo, D. M. Krum, and M. Bolas, "FAAST: The flexible action and articulated skeleton toolkit," in *Proc. IEEE Virtual Reality Conf.*, Mar. 2011, pp. 247–248.

[16] Leap Motion [Online]. Available: https://www.leapmotion.com/

[17] G. R. Bradski, "Computer vision face tracking as a component of a perceptual user interface," in *Proc. Workshop Applicat. Comput. Vision,* 1998, pp. 214–219.

[18] M. Isard and A. Black, "Condensation: Conditional density propagation for visual tracking," *J. Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[19] S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process*., vol. 50, no. 2, pp. 174–188, Feb. 2002.

[20] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, *Color-Based Probabilistic Tracking,* vol. 2350. Heidelberg, Germany: Springer, pp. 661–675, 2002.

[21] K. Okuma, A. Taleghani, N. Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. Eur. Conf. Comput. Vision*, 2004.

[22] M. Kristan, J. Pers, S. Kovacic, and A. Leonardis, "A local-motion-based probabilistic model for visual tracking," *Pattern Recogn.,* vol. 42, no. 9, pp. 2160–2168, 2009.

[23] W. Qu, D. Schonfeld, and M. Mohamed, "Real-time distributed multi-object tracking using multiple interactive trackers and a magnetic-inertia potential model," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 511–519, Apr. 2007.

[24] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, "Articulated human pose estimation and search in (almost) unconstrained still," ETH Zurich Tech. Rep. 272, Sep. 2010.

[25] Y. Yang, and D. Ramanan, "Articulated pose estimation with flexible mixture of parts," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2011, pp. 1385–1392.

[26] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2011, pp. 1297–1304.

[27] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.,* Jun. 2012, pp. 1862–1869.

[28] S. Bilal, R. Akmeliawati, A. A. Shafie, and M. J. E. Salami, "Hidden Markov model for human to computer interaction: A study on human hand gesture recognition," *Artificial Intell.,* 2011, pp. 1–22.

[29] B. W. Miners, O. A. Basir, and M. S. Kamel, "Understanding hand gestures using approximate graph matching," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 2, pp. 239–248, Mar. 2005.

[30] Z. Xu, C. Xiang, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 6, pp. 1064–1076, Nov. 2011.

[31] M. J. Black and A. D. Jepson, "A probabilistic framework for matching temporal trajectories: CONDENSATION-based recognition of gesture and expressions," in *Proc. Eur. Conf. Comput. Vision,* 1998, pp. 909–924.

[32] J. Alon, V. Athitsos, and W. Yuan, "A unified framework for gesture recognition and spatiotemproal gesture segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1685–1699, Sep. 2009.

[33] I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner, *Results and Analysis of the ChaLearn Gesture Challenge*. 2012.

[34] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[35] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from RGBD images," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn. Workshop Gesture Recogn.*, Jun. 2012, pp. 7–12.

[36] Y. Yang, I. Saleemi, and M. Shah, "Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1635–1648, Jul. 2013.

[37] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. J. Escalante, "ChaLearn gesture challenge: Design and first results," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn. Workshops,* Jun. 2012, pp. 1–6.

[38] J. Wachs, H. Stern, and Y. Edan, "Cluster labeling and parameter estimation for the automated setup of a hand-gesture recogntion system," *IEEE Trans. Syst., Man Cybern.*, vol. 35, no. 6, pp. 932–944, Nov. 2005.

[39] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn.,* vol. 1. Jun. 1999, pp. 81–96.

[40] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. Int. Conf. Comput. Vision Pattern Recogn.,* vol. 1. 2001, pp. 511–518.

[41] R. Hess and A. Fern, "Discriminatively trained particle filters for complex multiobject tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn.,* Jun. 2009, pp. 240–247.

[42] T. Yu, and Y. Wu, "Collaborative tracking of multiple targets," in *Proc. Comput. Soc. Conf. Comput. Vision Pattern Recogn.* Jun.–Jul. 2004, pp. 834–841.

[43] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets," in *Computer Vision-ECCV*. Berlin, Heidelberg, Germany: Springer, 2004, pp. 279–290.

[44] G. Borg, "Psychophysical bases of perceived exertion," *Med. Sci. Sports Exercise*, vol. 14, no. 5, pp. 377–383, 1982.

[45] H. Jiang, B. S. Duerstock, and J. P. Wachs, "Integrated gesture recognition based interface for people with upper extremity mobility impairments," in *Proc. 4th Int. Conf. Appl. Human Factors Ergonomics*, 2012, pp. 546–555.

[46] J. Aach and G. M. Church, "Alignment gene expression time series with time warping algorithms," *J. Bioinform.*, vol. 17, no. 6, pp. 495–508, 2001.
[47] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.,* vol. 36, no. 8, pp. 1627–1639, 1964.
[48] OpenNI [Online]. Available: http://www.openni.org/
[49] G. Pingali and J. Segen, "Performance evaluation of people tracking systems," in *Proc. IEEE Workshop Applicat. Comput. Vision,* Dec. 1996, pp. 33–38.
[50] I. Guyon, V. Athitsos, P. Jangyodsuk, and H. J. Escalante, *The ChaLearn Gesture Dataset,* 2011.
[51] H. J. Escalante and I. Guyon, "Principal motion: PCA-based reconstruction of motion histograms," Tech. Rep., ChaLearn Technical Memo., 2012.
[52] M. K. Sohn, S. H. Lee, D. J. Kim, B. Kim, and H. Kim, "A comparison of 3-D hand gesture recognition using dynamic time warping," in *Proc. 27th Conf. Image Vison Comput.,* 2012, pp. 418–422.
[53] K. Goldberg, *The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet.* Cambridge, MA, USA: MIT Press, 2000.
[54] S. M. Truntic, D. Hercog, and G. Pacnik, "Control and robotics remote laboratory for engineering education," *Int. J. Online Eng.*, vol. 1, no. 1, 2005.
[55] F. Carusi, M. Casini, D. Prattichizzo, and A. Vicino, "Distance learning in robotics and automation by remote control of LEGO mobile robots," in *Proc. Int. Conf. Robotics Autom.,* New Orleans, USA, Apr. 2004, pp. 1820–1825.
[56] M. Casini, F. Chinello, D. Prattichizzo, and A. Vicino, "RACT: A remote lab for robotics experiments," in *Proc. 17th IFAC World Congr., Seoul, Korea,* Jul. 2008, pp. 8153–8158.

Hairong Jiang received the B.S. and M.S. degrees in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2008 and 2010. She is currently pursuing the Ph.D. degree at the School of Industrial Engineering, Purdue University, IN, USA.

Her current research interests include gesture recognition and assistive technology.

Bradley S. Duerstock received the B.S. degree in biomedical engineering from Purdue University, West Lafayette, IN, USA, in 1994, and the Ph.D. degree in neurobiology from the College of Veterinary Medicine, Purdue University, 1999.

He was a Post-Doctoral Research Associate with the Center for Paralysis Research, Purdue University. He is currently an Associate Professor of Engineering Practice with the Weldon School of Biomedical Engineering and School of Industrial Engineering, Purdue University. He is the Director of Institute for Accessible Science, Purdue University. His research focuses on enhancing functionality of persons with disabilities through the development of assistive technologies and accessible design and through restoration of the biomedical system.

Juan P. Wachs (M'03) received the M. Sc. and Ph.D. degrees in industrial engineering and management from the Ben-Gurion University of the Negev, Beer-Sheva, Israel.

He is currently an Assistant Professor with the School of Industrial Engineering, Purdue University, West Lafayette, IN, USA. He is the director of the Intelligent Systems and Assistive Technologies Laboratory, Toronto, ON, Canada, and is affiliated with the Regenstrief Center for Healthcare Engineering, Purdue University. He completed post-doctoral training at the Naval Postgraduate School's MOVES Institute in the area of computer vision, under a National Research Council Fellowship from the National Academics of Sciences, and he was a recipient of the Air Force Young Investigator Award 2013. His current research interests include machine and computer vision, robotics, teleoperations, human robot interaction, and assistive technologies.

Dr. Wachs is a member of the Operation Research Society of Israel. He has published in journals including the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, *Journal of American Medical Informatics, Communications of the ACM*, and the *Journal of Robotic Surgery*.