Facilitated gesture recognition based interfaces for people with upper extremity physical impairments

Abstract. A gesture recognition based interface was developed to facilitate people with upper extremity physical impairments as an alternative way to perform laboratory experiments that require 'physical' manipulation of components. A color, depth and spatial information based particle filter framework was constructed with unique descriptive features for face and hands representation. The same feature encoding policy was subsequently used to detect, track and recognize users' hands. Motion models were created employing dynamic time warping (DTW) method for better observation encoding. Finally, the hand trajectories were classified into different classes (commands) by applying the CONDESATION method and, in turn, an interface was designed for robot control, with a recognition accuracy of 97.5%. To assess the gesture recognition and control policies, a validation experiment consisting in controlling a mobile service robot and a robotic arm in a laboratory environment was conducted.

Keywords: Gesture recognition, particle filter, dynamic time warping (DTW), CONDENSATION.

1 Introduction

Effective, natural and intuitive human computer interfaces (HCI) are critical aspects in the development of assistive technologies [1]. Voice, facial expressions, gaze and hand gestures have been widely used as communication channels for unimodal or multimodal interfaces for people with upper mobility impairments. Those interfaces were used for intelligent wheelchairs control, wellness monitoring and home medical alert systems [2-3], to mention a few. Hand gestures are of particular interest in the physically impaired community, since people already use gestures and thus relearning is avoided.

Hand gesture recognition algorithms involve the segmentation of the hands, tracking, and trajectories recognition. A common method for hand segmentation involves modeling the user skin color [4]. Depending purely on color information is unreliable, brightness, unstructured backgrounds, and clutter affects-object segmentation. If the focus is on the gestures' trajectories, instead of the hand shape, classic tracking approaches can be adopted. For example, *Camshift* is a well-established and basic algorithm for object tracking and was previously used for hand tracking [5]. Other more complex and robust approaches include the CONDENSATION algorithm developed by Isard and Black [6]. Particle filter is a common stochastic based technique for object tracking that can be easily parallelized. Perez applied the color-based appearance model to a particle filter framework to enhance tracking under complex backgrounds and occlusions [7]. In terms of gestures classification, the predominant approach is still Hidden Markov Models (HMM) (see Bilal for an extensive review of HMM ap-

plied to hand posture and gesture recognition [8]). Common problems with HMM involve finding good set of parameters (e. g. initial probabilities) and trajectory spotting for gesture temporal segmentation. Black and Jepson proposed a CONDENSATION-based trajectory gesture recognition algorithm [9] to this end. Nevertheless the gestures segmentation was not addressed in that work.

In this paper, both particle filters and the CONDESATION algorithm are combined for hand tracking and gesture classification in a simplistic yet robust fashion, which makes it suitable for HRI in assistive technologies.

2 System Architecture

The architecture of this system is illustrated in Fig. 1. The hand gesture based recognition system includes foreground segmentation, color-based detection, tracking, and trajectories recognition. A detailed description of the system is given in Section 3.

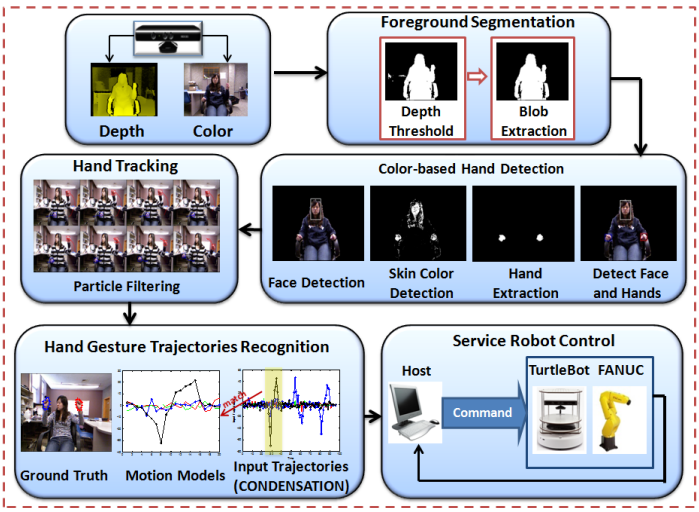


Fig. 1. System Overview

3 Gesture Recognition

3.1 Foreground Segmentation

To detect the user's movements, the user's body is treated as the foreground object. Two steps were employed to segment the foreground (refer to algorithm 1). The first step is to exclude pixels based on their distance to the camera (depth thresholding). The second step requires ruling out small areas and keep the largest blob in the remaining image as the foreground (blob cleaning). In the first step, the depth information was assessed through a Kinect sensor (fig. 2(a)). Two absolute depth thresholds (a low threshold T_{DL} and a high threshold T_{DH}) were manually set by the user

according to their relative distance to the sensor. Only those pixels with a depth value between the two thresholds were kept in a mask image (fig. 2(b)). The mask image was used to compute the area of the biggest region (blob), denoted as (B_{SH}). All the remaining blobs with a smaller area than B_{SH} were deleted (fig. 2(c)).

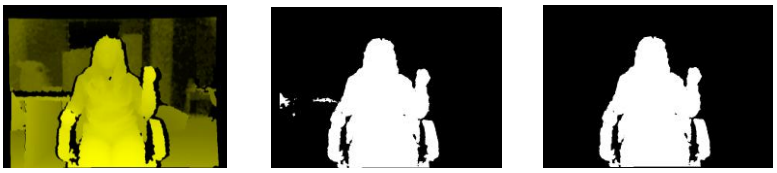
Algorithm 1: Foreground Segmentation

Input: T_{DL}; T_{DH}; depth Image D(i, j);

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

T_{SH} = max(Area(B_i)) //B_i is the i^{th} blob in the mask image D_1

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

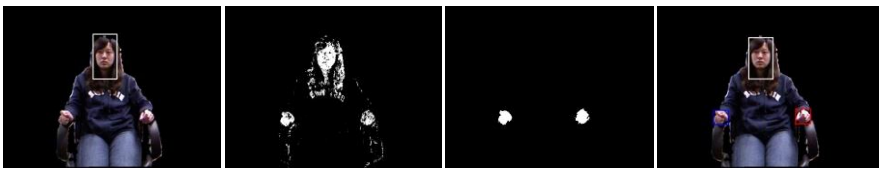


(a)Depth Image (b) Depth Threshold Mask (c) Foreground Mask

Fig. 2. Foreground Segmentation

3.2 Hand Detection

Before detecting the hands, a face detection method [10] was used to obtain the initial face region (as shown in fig. 3(a)). The result was used to remove the face region from the target image. Skin and non-skin color histogram models were constructed by using the Compaq database [11]. The probability of a pixel to be part of the hand was calculated as the division of the two histograms (which is a proxy of the distinctiveness- the higher the ratio, the more likely the two pixels belong to different color distributions). The mask image was obtained by applying the histogram ratio and back-projecting the probabilities of each pixel back in the image (as show in fig. 3(b)). To obtain the hand regions without the face, the region detected by the face detector was removed from the target image. After this, the two largest blobs were selected as hand regions (fig. 3(c), (d)). This hand detection procedure is only used to provide automatic initialization to particle filter tracking. Afterwards the hands positions are assessed through continuous tracking done by the particle filter.



(a)Face Detection (b) Skin Color Detection (c) Hand Extraction (d) Localization

Fig. 3. Face and Hand Detection

3.3 Hand Tracking

A color and spatial information based Sequential Importance Resampling (SIR) particle filter framework was proposed to track both face and hands through frames in video sequences [7, 12]. The Particle filter algorithm consists of three main phases: predicting, measuring and resampling. In the prediction phase, a second order auto-regressive (AR) was selected as the dynamic motion model as in equation (1):

$$x_{t+1} = A_1(x_t - x_0) + A_2(x_{t-1} - x_0) + x_0 + B\nu_t,\ \nu_t \sim \mathcal{N}(0, \Sigma) \quad (1)$$

where A_1, A_2, B and Σ were the parameter matrices that best matched the real motion of the tracked object; x was the state of particles. In the measuring stage, both color and spatial information were incorporated in the particle filter framework to calculate the likelihood function. The method in [8] was used to calculate the color likelihood function:

$$\omega^i \propto \exp(-\lambda D_i^2) \quad (2)$$

where λ is the Bhattacharyya similarity coefficient (set to 20). The spatial likelihood function included three parts: the distance between the face and hands (defined as D_{fh}), the distance between the two hands (defined as D_{hh}) and the distance between each particle and the center of the segmented hand in the previous video frame, (defined as D_{pc}). The spatial part of likelihood is then:

$$\omega^i \propto \exp\ (k_1 D_{fh} + k_2 D_{hh} + k_3(1/D_{pc})) \quad (3)$$

Combining equation (2) and (3), the likelihood function is:

$$\omega^i = \beta * \exp\ (-\lambda D_i^2 + k_1 D_{fh} + k_2 D_{hh} + k_3(1/D_{pc})) \quad (4)$$

where β is a normalization factor; k_1, k_2, k_3 are parameters that were set to change the weight of each feature for optimal tracking.

3.4 Hand Trajectory Classification

The positions of the hands in each frame of the video sequence were acquired from the tracking stage. The motion model for each gesture trajectory was created based on the data collected from eight subjects. Although gestures performed by each subject may have similar trajectories, the precise duration of each sub-trajectory within the trajectory were different. To normalize the trajectories, temporal alignment was conducted. The dynamic time warping (DTW) method was employed to accommodate differences in timing between different trajectories to the construct motion models [13]. The following procedure was proposed to obtain the motion models (Algorithm 2).

Algorithm 2: Procedures to Construct Motion Models

Input: Number of gestures G; number of subjects S; number of sampling trajectories from each subject T; horizontal and vertical velocity for left and right hand V_p^m, m=1,…,S*T.

for k = 1: G
  for j=1:S
    for i=1:T-1
      Align V_p^i with V_p^{i+1} to obtain V_{ap}^i
    end for
    Align V_p^T with V_p^1 to obtain V_{ap}^T
    $V_p^j = \sum_{i=1}^{T} V_{ap}^i$ / T
  end for
  for j=1:S-1
    Align V_p^j with V_p^{j+1} to obtain V_{ap}^j
  end for
  Align V_p^S with V_p^1 to obtain V_{ap}^S
  $V_p^k = \sum_{i=1}^{S} V_{ap}^i$ / S
end for

The CONDENSATION algorithm [9] was used to recognize the hand gesture trajectories. The original algorithm was extended to work for two hands. A state at time *t* is described as a parameter vector:

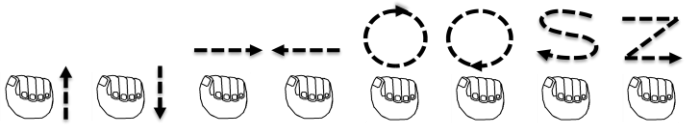
$$s_t = (\mu, \Phi^x, \alpha^x, \rho^x) \tag{5}$$

Where, μ was the index of the motion models, ϕ was the current phase in the model, α was an amplitude scaling factor, ρ was a time dimension scaling factor, x denoted the hand used x∈ {left hand, right hand}.

4 Experimental Results

4.1 Recognition Accuracy

An eight-gesture lexicon (as shown in fig. 4) was tested by eight users and resulted in an average cross validation accuracy of 97.5%. Ten sessions were used for cross validation of each gesture (k-fold with k=10). In each session, 72 gestures were used for training and 8 gestures were used for testing. A confusion matrix was computed using a temporal window size of w=20 (fig. 5). The ROC curve to demonstrate the system performance was obtained by changing the size of the window from 10 to 24 to different values (fig. 6).



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4. Gesture Lexicon

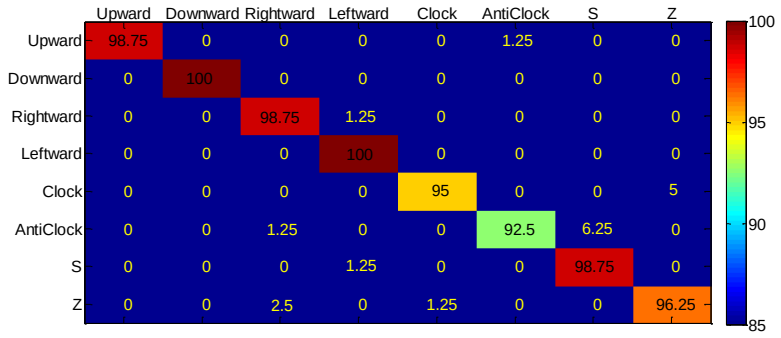


Fig. 5. Confusion Matrix with the window size- w =20

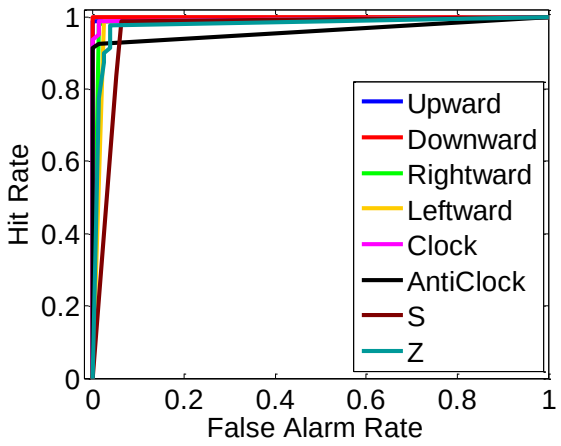


Fig. 6. ROC Curve showing recognition performance for each gesture

4.2 Laboratory Experiment

A laboratory case study was performed by using this gesture recognition based interface to test whether this system could be used by scientists with physical impairments to conduct chemistry experiments. In this experiment, the mobile robot carrying a beaker was controlled to the position where a robotic arm was located. Then the robotic arm added reagent to the beaker and the mobile robot came back to the user. To control the mobile robot and the robotic arm, the gestures in the lexicon from (a)-(h) were mapped to the following commands: ‘change mode’, ‘robotic arm action’, ‘go forward’, ‘go backward’, ‘turn left’, ‘turn right’, ‘stop’ and ‘enable robotic arm’. Two modes were used to control the mobile robot: *discrete* and *continuous* mode. In *discrete* mode, the robot moved an increment of distance, every time that a command was issued. While the *continuous* mode, the robot responded to the given command,

until the ‘stop’ command was issued. To switch between these two modes one distinctive gesture (‘upward’) was used. In the experiment, the discrete, continuous and combined (continuous plus discrete) control modes were tested. The resulting average task completion times were of 205, 143.2 and 109.8 seconds, respectively (fig. 7). Each recognition process required 47ms for face and both hands. The map for the lab and the trajectories of the robot for discrete (red star line), continuous (blue solid line), and combined (black dash line) control modes were recorded to test for correlation between average completion time for a task with fixed distance and the used control mode (fig. 8).

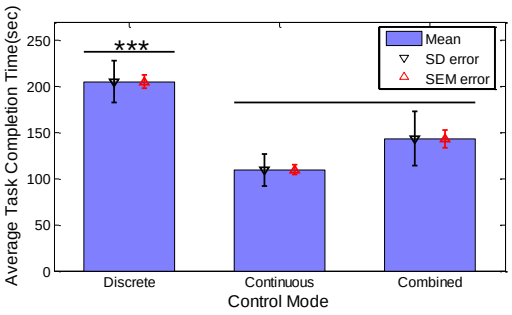


Fig. 7. Average Task Completion Time, Unpaired t-test, $p<0.001$

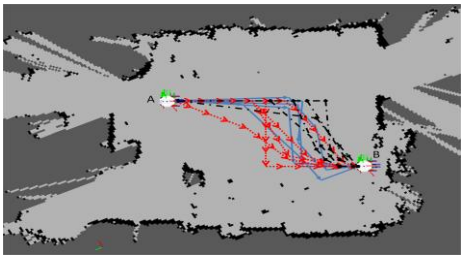


Fig. 8. Robot trajectories for different control modes

5 Conclusions and Future Work

A hand gesture recognition based interfaces was developed for people with upper extremity mobility impairments. The depth information was used to segment the human body from a non-static background. An automatic initialization procedure for the particle filter method was engineered by combining blob extraction, face detection, image dilation, erosion and color histograms techniques. Both color and spatial information were considered when applying the particle filter framework. A training procedure was proposed to obtain motion models for each gesture in the lexicon. The CONDENSATION algorithm was used to classify the bimanual gestures. The gesture recognition algorithm designed was found to reach a recognition accuracy of 97.5%. A laboratory task experiment was conducted to validate real time performance of the

gesture interface to assist in conducting a chemistry lab with the help of two robots. From the results, the continuous mode required the least average task completion time, while the discrete control mode required the most. Therefore, the authors recommend to use continuous control mode is used most of the time and the discrete is used only when the robot is very near to the target. Future work includes studying additional robust techniques for hand tracking to tackle the problem of resilience to occlusions (when one hand occludes the other).

Acknowledgement. This work was partially funded by the National Institutes of Health through the NIH Director's Pathfinder Award to Promote Diversity in the Scientific Workforce, grant number DP4-GM096842-01.

Reference

1. Jacko, J.A.: Human-Computer Interaction Design and Development Approaches. In: 14th HCI International Conference, pp. 169-180 (2011).
2. Moon, I., Lee, M. and Ryu, J., Mun, M.:. Intelligent Robotic Wheelchair with EMG-, Gesture-, and Voice-based Interfaces. In: International Conference on Intelligent Robots and Systems, IEEE Press, pp. 3453-3458 (2003).
3. Reale, M., Liu, P. and Yin. L.J.: Using eye gaze, head pose and facial expression for personalized non-player character interaction. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE Press, pp. 13-18 (2011).
4. Soriano, M., Martinkauppi, B., Huovinen, S., Laaksonen, M.: Skin detection in video under changing illumination conditions. In: 15th International Conference on Pattern Recognition, vol. 1, pp. 839-842. (2000).
5. Bradski, G.R.: Computer vision face tracking as a component of a perceptual user interface. In: Workshop on applications of computer vision, pp. 214-219. Princeton, NJ, (1998).
6. Isard, M., Black, A.: CONDENSATION: Conditional density propagation for visual tracking. J. International Journal of Computer Vision, Vol. 29, pp. 5-28 (1998).
7. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking, LNCS, Vol. 2350, pp. 661-675. Springer, Heidelberg (2002).
8. Bilal, S., Akmeliawati, R., Shafie, A.A., Salami, M.J.E.: Hidden Markov Model for human to computer interaction: a study on human hand gesture recognition. Artificial Intelligence (2011).
9. Black, M.J., Jepson, A.D.: A Probabilistic Framework for Matching Temporal Trajectories: CONDENSATION-Based Recognition of Gestures and Expressions. In: Burkhardt, H., Neumann, B. (eds) ECCV 1998. LNCS, vol I. pp. 909-924. Springer, Heidelberg (1998)
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: International Conference on Computer Vision and Pattern Recognition, pp. 511-518 (2001).
11. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 46, pp. 81-96 (2002).
12. Hess, R., Fern, A.: Discriminatively Trained Particle Filters for Complex Multi-Object Tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 240-247 (2009).
13. Aach, J., Church, G.M.: Alignment gene expression time series with time warping algorithms, J: Bioinformatics, vol. 17, no. 6, pp. 495-508, Oxford University Press (2001).