Intention, Context and Gesture Recognition for Sterile MRI Navigation in the Operating Room

Mithun Jacob, Christopher Cange, Rebecca Packer, and Juan P. Wachs

Purdue University, West Lafayette, IN 47906, USA
{mithunjacob,ccange,rpacker,jpwachs}@purdue.edu

Abstract. Human-Computer Interaction (HCI) devices such as the keyboard and the mouse are among the most contaminated regions in an operating room (OR). This paper proposes a sterile, intuitive HCI to navigate MRI images using freehand gestures. The system incorporates contextual cues and intent of the user to strengthen the gesture recognition process. Experimental results showed that while performing an image navigation task, mean intent recognition accuracy was 98.7% and that the false positive rate of gesture recognition dropped from 20.76% to 2.33% with context integration at similar recognition rates.

Keywords: Gesture recognition, operating room, human computer interaction.

1 Introduction

Recent advances in computer-assisted surgery are taking user centered interfaces to the operating room (OR) in more and more hospitals and outpatient clinics. Since HCI devices are possible sources of contamination due to the difficulty in sterilization, clinical protocols have been devised to delegate control of the terminal to a sterile human assistant [1], [2]. Nevertheless, this mode of communication has been shown to be cumbersome [3], prone to errors [1] and overall, inefficient. This paper proposes a sterile method for the surgeon to naturally, and efficiently manipulate MRI images through touchless, freehand gestures. Image manipulation through gestural devices has been shown to be natural and intuitive [4] and does not compromise the sterility of the surgeon. The system extends a system previously developed by the authors [5] with the use of dynamic two-handed gestures and contextual knowledge.

2 System Overview

2.1 MRI Image Browser

Users interact with an image browser developed to navigate and manipulate MRI images. The browser (developed with OpenGL and OpenCV libraries) displays several sequences on the left side of the screen for selection and a single slice from the selected sequence on the right side of the screen (see Fig. 1(a)). The user then selects an image representing an anatomical structure of interest. This image is then manipulated through several actions such as increasing/decreasing image brightness, and rotating the image in the clockwise or counter-clockwise directions.

L. Alvarez et al. (Eds.): CIARP 2012, LNCS 7441, pp. 220–227, 2012.

The images are accessed entirely through gestural commands (one gesture for each command). The lexicon consists of ten gestures (see Fig. 2(a)) which were selected from interviews with nine veterinary surgeons. The gestures encompass image navigation and manipulation tasks such as browsing (*up*, *down*, *left*, and *right*), zooming (*zoom-in*, and *zoom-out*), rotation (*clockwise*, and *counter-clockwise*) and brightness change (*brightness-up*, and *brightness-down*). The system can also be used independently of a fixed display such as a television or monitor; Fig. 1(b) shows the system being used with a pico-projector hanging around the user.

Fig. 1. (a) MRI Image Browser (b) Browser with the pico-projector

2.2 Gesture Recognition

A Microsoft Kinect using the OpenNI SDK was used to capture the user's skeleton thus providing the positions of various landmark positions on the human body from depth data (see Fig. 2(b)). The positions of the left and right shoulders, and the head were quantized and delivered to a decision tree, as visual cues to gauge the user's intention. If the user intends to use the system, the position of the left and right hands are tracked and the trajectories are classified with a set of 10 Hidden Markov Models (HMMs). Additionally, non-visual contextual cues such as the sequence of commands issued by the user was modeled as a Markov chain and the time between commands were also used as contextual cues to aid in gesture recognition. The recognized command is then sent to the MRI image browser.

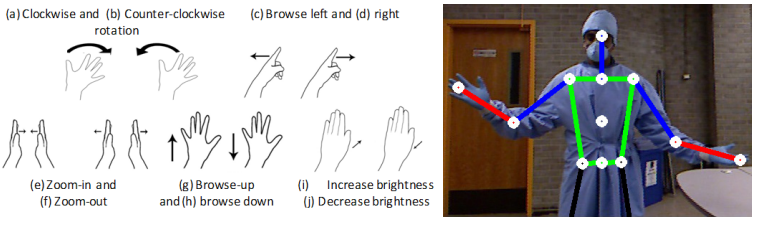

Fig. 2. (a) Gesture Lexicon (b) Skeleton model and tracked marker-less points

3 Gesture Recognition with Contextual Cues

Gaze has been established as a critical cue in establishing attention which stems from the intention of a user to interact with a person or a device [6]. Other fundamental cues are head orientation, body posture and the position of the hands w.r.t the body [7]. The following sections describe how visual and non-visual contextual cues are used to recognize the performed gestures.

3.1 Intention Recognition

The intention recognition module decides whether a performed gesture is intentional or not based on anthropometric and kinematic features of the human body. The torso orientation (T_θ), head orientation confidence (H_θ), hand orientation w.r.t. the torso (L_θ, R_θ) are combined to form the visual context feature vector,

$$V = [T_\theta, \quad H_\theta, \quad L_\theta, \quad R_\theta]^T \tag{1}$$

The information encapsulated in this feature vector allows us to successfully determine whether the user "intends" to perform a gesture or not. The cues are explained in detail below:

- **Torso orientation:** The orientation of the torso helps determine whether the user is facing the system and is thus intending to use it. The 3D position of the left and right shoulder ($\vec{L}_s, \vec{R}_s$) is used to compute the azimuth orientation of the torso w.r.t. the X-axis, i.e. $T_\theta = \cos^{-1}\left(\frac{(\vec{L}_s - \vec{R}_s)}{|\vec{L}_s - \vec{R}_s|} \cdot [1 \quad 0 \quad 0]^T\right)$.
- **Head orientation:** The location of the head is obtained from the skeleton and is used to reduce the search space of the Viola-Jones frontal face detector [8]. A continuous estimate H_θ of the confidence that the head of the user is forward-facing is computed by sliding a 10×1 mean filter over the output of the frontal face detector per frame. This cue provides information regarding the gaze of the user.
- **Hands position:** Arm orientation with respect to the torso is an indication of whether the user wants to gesticulate towards the camera, or instead, is engaged in a surgical task. The 3D position of each hand is used to compute its orientation with the waistline providing the inclination of each hand (i.e.L_θ, R_θ) with respect to the zenith angle.

Integration of Visual Contextual Cues
A dataset of 2100 sample sequences of "intentional" behavior, and 2650 samples of "unintentional" behavior were captured from users and manually annotated (I and U, respectively). Then, each sequence was quantized into a feature vector (see Equation 1). The data set was used to train a decision tree which was pruned to produce a minimum-cost tree of 63 nodes.

3.2 Gesture Spotting

Once intent has been determined, the gesture was segmented from the trajectory of the user's hands. Gesture spotting [7] is the process of automatically determining the start and the end of a gesture. Low-level features such as gesture acceleration [9] serve as a proxy to segment each gesture (gestures are preceded and succeeded by sudden acceleration and deceleration respectively). The segmented observations were used as inputs to the discrete HMMs.

Let the velocity of hand h at time t be $V_h(t)$, and t_1 and t_N the start and end times of a sliding window, respectively ($t \in \{t_1, \ldots t_k, \ldots, t_N\}$). If the variance $\sigma^2(V_h(t_k))$ exceeds the threshold α (an empirically determined), t_k is set as the

start point of the gesture. The end-point is similarly determined. If the length of the segmented gesture exceeds a threshold, a gesture has been spotted.

3.3 Pre-selection of Gesture Classes

The contextual information used so far is a good proxy for gestural intent; nevertheless it does provide much information about the likelihood of a given gestural occurrence. A combined measure of gesture likelihood is obtained using two non-visual contextual cues. These cues are learned independently of the gesture interface, since they are intrinsic to the task alone. All the non-visual cues were gathered after observing a large number of MRI browsing tasks completed by the users of the system.

- **Delay between commands:** The time between commands t_D provides predictive information. For example, navigational commands exhibit shorter delays between commands whereas image manipulation commands image have a longer delay. A normal distribution was fitted to the observed delays between commands, so each gesture class k (mapped to a command) has a normal distribution $\mathcal{N}(\mu_k, \sigma_k)$ assigned to it.
- **Command history:** The command history provides information regarding which commands are more probable to occur given the previous command. Since all commands are not equiprobable, the command history helps in reducing the possible set of gestures by using the knowledge of the previously evoked command C_{t-1}. The sequence of commands is modeled as a first-order Markov chain and a transition matrix A is learned from user-interactions.

Integration of Non-visual Contextual Cues

Given a command delay, and a current command, the probability of the next gesture k from the gesture lexicon Γ was computed by finding the joint probability between the gesture class given the delay time t_D, and the probability of transition to the current command. Formally,

$$P_k = P(k|t_D)P(k|C_{t-1}) \;\; \forall k \in \Gamma \tag{2}$$

All hand trajectories corresponding to gestures where $P_k > \epsilon$, were classified using chains of HMM detailed in the next subsection.

3.4 Post-selection of Gesture Classes

The motion of the centroids of the hands from the skeleton model is fed to the gesture recognition algorithm. This algorithm attempts to classify the spotted trajectory (see section 3.2) as belonging to one of the ten gestures in the surgical gesture lexicon Γ. The input to the gesture recognition algorithm is the feature vector u which encodes the velocity of each hand along each axis in $\mathbb{R}^3$. Given that the centroids of the first and second hand in the n^{th} frame are given by $(x_{f;n}, y_{f,n}, z_{f;n})$ and $(x_{s;n}, y_{s,n}, z_{s;n})$ respectively, then the feature vector for a frame n is computed as u,

$$[x_{f;n} - x_{f;n-1}, \quad y_{f;n} - y_{f;n-1}, \quad z_{f;n} - z_{f;n-1}, \quad x_{s;n} - x_{s;n-1}, \quad y_{s;n} - y_{s;n-1}, \quad z_{s;n} - z_{s;n-1},] \tag{3}$$

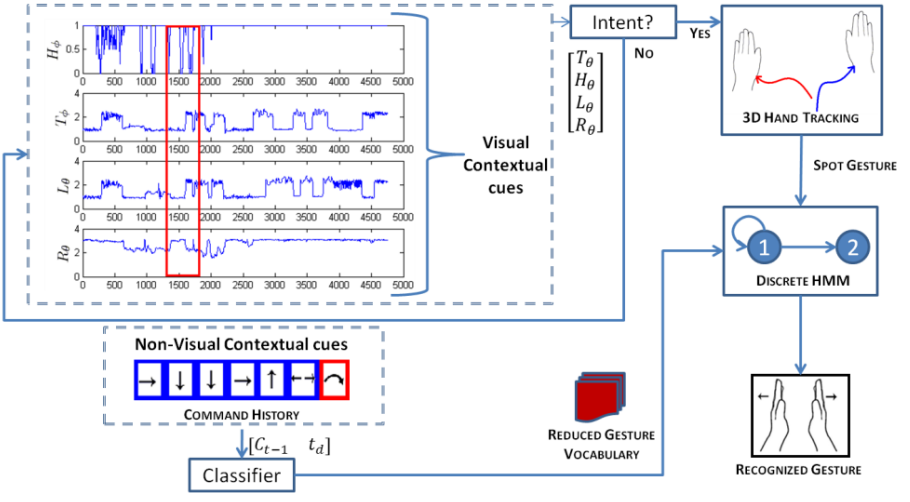

Fig. 3. Pruning the gesture lexicon with visual and non-visual contextual cues

A subset $\Lambda \subseteq \Gamma$ (determined in the pre-selection process) of discrete HMMs (one corresponding to each gesture), was used to recognize the trajectories. Each HMM is a left-right model [10] with 5 states. Each element in the vector u (velocities of the hands along each axis in $\mathbb{R}^3$) was quantized to three possible symbols; $\{+,0,-\}$. If $|u_i| \leq \tau$, then u was considered static (0 symbol). Else, the sign corresponding to the velocity along the corresponding axis was assigned.

Each vector u was quantized to 3^6 representations which form the set of observation symbols for each HMM. Ten HMMs (an HMM $\lambda_k = (A_k, B_k, \pi_k)$ per gesture k) was trained with labeled data with the Baum-Welch [10] algorithm. A gesture k was said to be recognized if λ_k resulted in the highest probability of the set of quantized observations $O = [O_1, \ldots, O_T]^T$. The probabilities were computed using the Viterbi algorithm [10] on the segmented trajectories, i.e.

$$k = \text{argmax}_k \, P(O|\lambda_k) \; \forall k = 1, \ldots, |\Lambda| \tag{4}$$

4 Experiments

The following section discusses the experiments conducted to validate the hypothesis that contextual information (intention) can be used to detect accurately the gestures evoked by the user.

4.1 Experiment 1: Intention Detection

The first experiment tested the prediction of intention based on contextual cues, as described in section 3.1. A dataset of 4750 observations was collected to train and test the decision tree, of which 44% represented "intentional" behavior, and the rest

"unintentional" behavior. An ROC curve (see Fig. 4) was generated through 2-fold cross-validation for ten discrete values of κ, the maximum depth of the decision tree (κ varied from 6-15). The ROC curve indicates that the peak operating point of the classifier has recognition accuracy of 97.9% with 1.36% false positive rate. A true positive is obtained when the user is correctly found to be facing the screen.

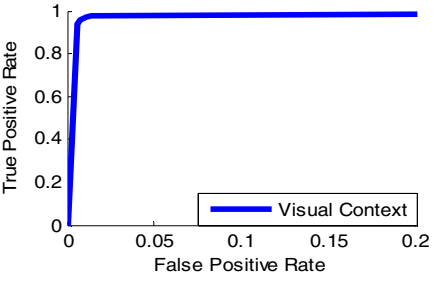

Fig. 4. ROC curve for intention recognition

4.2 Experiment 2: Gesture Recognition

The gesture recognition performance of the HMM was evaluated in this experiment. A dataset of 1000 gestures were performed by 10 users (10 gestures per user per class). Also, several configurations of the left-right HMM model with various discretization thresholds τ were tested over the dataset through 10-fold cross-validation. The value of $\tau = 28$mm was found to be optimal (see Fig. 5). The performance per class at this optimal operating point is described by the confusion matrix, presented in Fig. 6. Mean recognition accuracy of 97.23% was obtained. Fig.6 also shows that the *left* and *right* gestures have relatively lower mean accuracy since they are respectively sub-gestures of the *clockwise* and *counter-clockwise* gestures and are thus susceptible to be confused.

4.3 Experiment 3: In-task Recognition Performance

Twenty two students were asked to perform a specific browsing and manipulation task using the MRI image browser in a laboratory environment. The task consisted of searching for a landmark image and performing image manipulation tasks on the landmark image. All the data was recorded over 220 trials (10 per student) and each performed gesture was manually annotated. Data from 2 users were discarded as outliers due to the failure of the Kinect to reliably determine the skeleton of the user. A total of 4445 gestures were collected from users performing this task. At the end of each trial, each user was asked to assemble a surgical box. This activity served as a controlled way to force the user shift the focus of attention from the image browser. Without contextual information, such activity could potentially trigger accidental gestures. Fig. 7 displays the isolated gesture recognition accuracy of the 4445 annotated gestures captured when the user was interacting with the system. During the "non-intentional" phase of each trial, the gesture spotter was executed and the segmented gestures (false positives) were recognized. Intent was correctly determined 98.7% of the time and mean gesture recognition accuracy (ACC) of 92.58% and 93.6% was obtained for the system with and without context respectively.

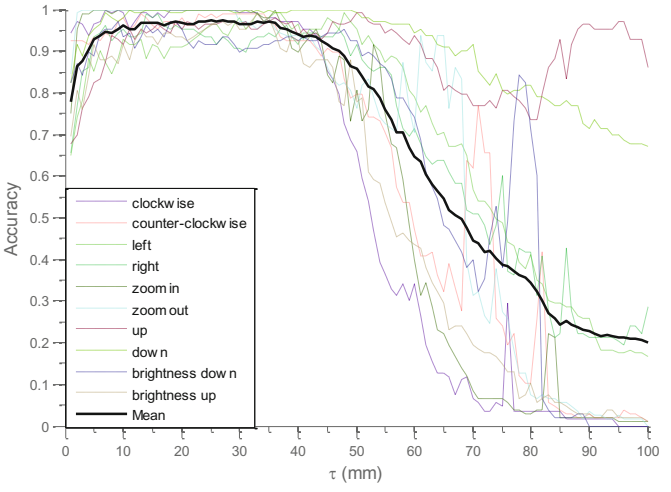

Fig. 5. Mean gesture recognition accuracy vs. discretization threshold. On the vertical axis is the accuracy of each gesture (in different colored line), and on the horizontal is the discretization threshold used to convert trajectories to sequences of discrete symbols.

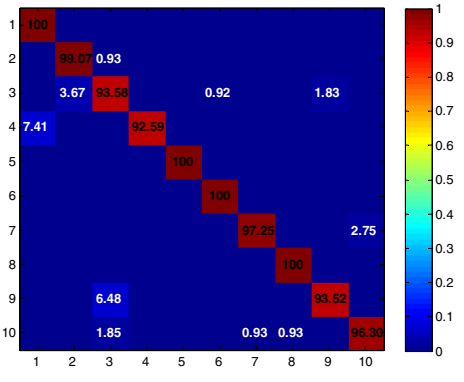

Fig. 6. Confusion matrix for $\tau = 28$mm. The rows represent the true class of the gestures labels and the columns represent the class assigned by the algorithm. High values on the diagonal elements indicate high gesture recognition accuracy.

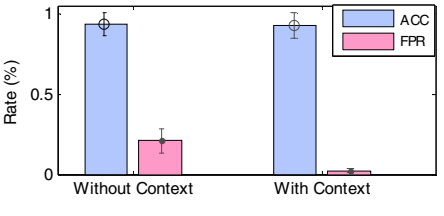

Fig. 7. Comparison of gesture recognition with and without context

The main advantage of incorporating context is visible in the reduction of the false positive rate (FPR) of 20.76% without context to 2.33% with context.

5 Discussion and Conclusion

The hypothesis that contextual information integrated with hand trajectory gesture information can significantly improve the overall system recognition performance was validated. It has been shown that the false positive rate is significantly reduced using context without affecting recognition performance. The intent recognition and gesture recognition systems have been shown to perform well (98.5% and 93.6% respectively) on data collected from user interactions. In the dataset of gestures, the average isolated gesture recognition rate was found to be 97.23%.

Future work includes building a more sophisticated gesture spotter which uses the gestural knowledge as well as local features of the trajectory to segment gestures. Additionally, tracking skeletal joints independently is required to handle the possible situation of failure in skeletal tracking by the OpenNI SDK.

Acknowledgments. This project was supported by grant number R03HS019837 from the Agency for Healthcare Research and Quality (AHRQ). The content is solely the responsibility of the authors and does not necessarily represent the official views of the AHRQ. We would also like to thank all the students who helped.

References

1. Albu, A.: Vision-based user interfaces for health applications: a survey. Advances in Visual Computing, 771–782 (2006)
2. Schultz, M., Gill, J., Zubairi, S., Huber, R., Gordin, F.: Bacterial contamination of computer keyboards in a teaching hospital. Infection Control and Hospital Epidemiology 24(4), 302–303 (2003)
3. Maintz, J., Viergever, M.A.: A survey of medical image registration. Medical Image Analysis 2(1), 1–36 (1998)
4. Ebert, L.C., Hatch, G., Ampanozi, G., Thali, M.J., Ross, S.: You Can't Touch This: Touch-free Navigation Through Radiological Images. Surg. Innov. (November 2011)
5. Wachs, J.P., Stern, H.I., Edan, Y., Gillam, M., Handler, J., Feied, C., et al.: A Gesture-Based Tool for Sterile Browsing of Radiology Images. J. Am. Med. Inf. Assoc. 15(3), 321–323 (2008)
6. Emery, N.: The eyes have it: the neuroethology, function and evolution of social gaze. Neuroscience & Biobehavioral Reviews 24(6), 581–604 (2000)
7. Langton, S.R.H.: The mutual influence of gaze and head orientation in the analysis of social attention direction. The Quarterly Journal of Experimental Psychology: Section A 53(3), 825–845 (2000)
8. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)
9. Kang, H., Woo Lee, C., Jung, K.: Recognition-based gesture spotting in video games. Pattern Recognition Letters 25(15), 1701–1714 (2004)
10. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)