A Visualization Tool for Displaying Hand Gestures

Beifang Yi Frederick C. Harris, Jr. Sergiu M. Dascalu
Department of Computer Science and Engineering
University of Nevada, Reno
Reno, NV 89557
{b_yi, fredh, dascalus}@cs.unr.edu

Abstract

This paper presents a visualization tool for displaying the output from a human hand gesture analysis system. This tool extends the environment of the hand gesture analysis system and produces hand gestures based on hand configurations identified by the system through a virtual hand in the constructed virtual environment. The virtual hand and virtual environment can be automatically aligned with the system. With the proposed visualization tool, hand gestures, rendered in the virtual environment, can be displayed on screen, and the rendered image sequences and their rendering parameters can be recorded for later processing.

Keywords: visualization tool, virtual reality, HCI, GUI

1 INTRODUCTION

The human hand has been considered one of the most natural and promising communication media in the human-computer interaction (HCI) [8, 12]. In order to take advantage of this new input modality, emphasis has been on vision-based studies, hand modeling, and hand gesture analysis and recognition [6, 10]. Hand images are processed and analyzed, and some hand gestures are recognized and represented as hand configurations (hand representations in some specific formats). These hand configurations can be used as input commands to computers in HCI applications.

The human hand is a highly articulated organic structure, and the Glove Box has been recently used in the computer vision research on the hand [3]. In such a box, four to eight cameras are mounted at (or near) the corners. Hand images are taken and sent to the hand gesture analysis and recognition system where meaningful hand gestures are identified and represented as specific hand configurations. The representations can be described as sets of hand parameters such as the hand joint coordinates, joint rotation axes and angles, and palm orientation.

A lifelike virtual hand created based on these parameters can greatly help the hand gesture analysis: with the virtual hand, we will be able to reconstruct the gestures of the real hand. In connection with this, an easy-to-use output environment for displaying the analysis results based on the virtual hand, with a user interface designed to adjust the environment, is also important. For example, a reconstructed hand gesture can be viewed from different viewing points and rendered with various styles such as wireframes and different colors.

We constructed a virtual hand and placed it into a virtual environment with a graphical user interface to form a hand gesture visualization tool. This paper concisely describes our work on developing a visualization tool for displaying hand gestures.

In the following sections, we first discuss the related work and the construction of a lifelike virtual hand, then describe issues related to designing and implementing the visualization tool and, finally, provide details of the tool's usage and pointers to future work.

2 RELATED WORK

Although the human hand is a highly complex biological organ, we can think sometimes of the hand as a mechanical machine and benefit from studies on the hand that apply mechanical principles [4]. In this view, we often interpret the hand motion with the movements of the hand bones. More precisely, we can use a linkage system of rigid segments of hand bones to describe and analyze hand motions, with hand joints between hand bones as rotation points. Each of the fingers (index, middle, ring, and pinky) and the thumb have three joints. The palm is often considered one whole part. The wrist bones, when in motion, have the most complicated movements, but their motions are limited to a very small 3D space. For convenience, we consider that the wrist bones share one common joint. Between any two joint points there is a part of hand and the whole hand is composed of 16 hand parts.

The movements at the joints can be simplified by introducing the concept of degree of freedom (DOF). There is only one DOF at the first two joints of the fingers (top and middle) and the first thumb joint because there is only one movement at each of these joints, a rotation around the flexion axis (flexion or extension). There are two DOFs at the finger base joints and the thumb's lower tow joints, corresponding to side-to-side movement (abduction or adduction) and to rotation around the flexion axis (flexion or extension). The wrist joint is represented with six DOFs: the displacement in 3D space (movements along 3 perpendicular axes), bending, side-to-side movement, and twist (as a result of the rotation of the forearm). Detailed description and measurement of hand joint motions are given in [5, 7].

Hand modeling is based on results from hand biomechanic studies and computer graphics techniques. Rotational axes and joints corresponding to the joint movements have been calculated in our hand model. The ranges of rotational angles have been carefully tested and determined. Thus, once a hand model is created, a hand configuration (gesture) is described as a set of rotational angles at all the hand joints with the hand position in 3D space. Some rendered hand gestures are illustrated in the following sections.

The hierarchical structure of the human hand can also be considered in hand modeling. The design of hand movements is greatly simplified by considering only the motions at one joint and relating these motions with those of the immediately connected joints, according to the hand biological structure. For example, a pointed hand made from a fist involves only the motions of the index finger: the top index joint will automatically follow the movement of the hand part connected with the middle index joint and the middle index joint will follow the movement of the hand part connected with the base index joint.

The virtual hand, modeled as a linkage system of rigid segments with large number of DOFs, can produce any arbitrary movements. But the human hand's movements are highly constrained such that it can only generate highly coordinated "natural" movements. We have also considered some hand constraints in our hand model. This guarantees that the virtual hand can only produce natural hand gestures. More information about hand modeling and hand gesture design can be found at [13].

3 DESIGN ISSUES

Our visualization tool serves as an output platform for a hand gesture analysis system. The hardware for such a system typically includes a glove box with four to eight cameras mounted close to the corners. The cameras capture pictures of real hands and these pictures are sent to the hand analysis system for estimation of the hand states in the form of the palm pose and hand position and shapes. This system then calculates the hand configuration parameters and these parameters can be finalized in the form of hand joint angles and hand part positions. These configuration parameters and camera parameters (such as camera positions and focuses) are finally be sent to the visualization tool, where a virtual hand is constructed in a virtual environment according to the parameters obtained from the analysis system.

The visualization tool serves as a displaying output platform where all the operations are based on a life-like virtual hand and its virtual environment. The design architecture for the tool is shown in Figure 1. The bold arrow lines in the figure give the directions of the hand gesture configurations: (1) The hand configurations (hand states) are calculated in the hand analysis system. A virtual hand is constructed based on these configuration parameters, matching the hand shape and orientation of the real hand in the analysis system. (2) A virtual environment is constructed based the glove box's size, background, and cameras' parameters. (3) The reconstructed virtual hand and virtual environment will be displayed on screen and the corresponding rendered hand images can be stored on disk.

The thin arrow control lines show operations on the virtual hand and virtual environment as well displaying and recording processes: (1) The coordinate systems used for the real hand and cameras in the analysis system and the position and orientation of the real hand are used to produce a virtual hand with the same orientation and position in the virtual environment. (2) Virtual cameras are created based on the real cameras' parameters in the system. A virtual hand is rendered for each of the virtual cameras. (3) The rendered images can be selected for display on screen and be recorded on disk. Also, the configuration parameters (joint angles and positions), camera parameters, and OpenGL rendering matrices can also be recorded in text files.

The design of the visualization tool includes two main processes: (1) Functional design, concerned with the operations on the virtual hand and environment as well as with the displaying and recording of hand images and parameters. (2) Interface design, focused on providing easy-to-use interface facilities that support these operations. During the first part of the design process, the functional design, we considered the following issues:

- The virtual environment should be constructed following the specifications of the glove box of the hand analysis system.

- Eight virtual cameras should be placed according to the real cameras' positions. To have more flexi-

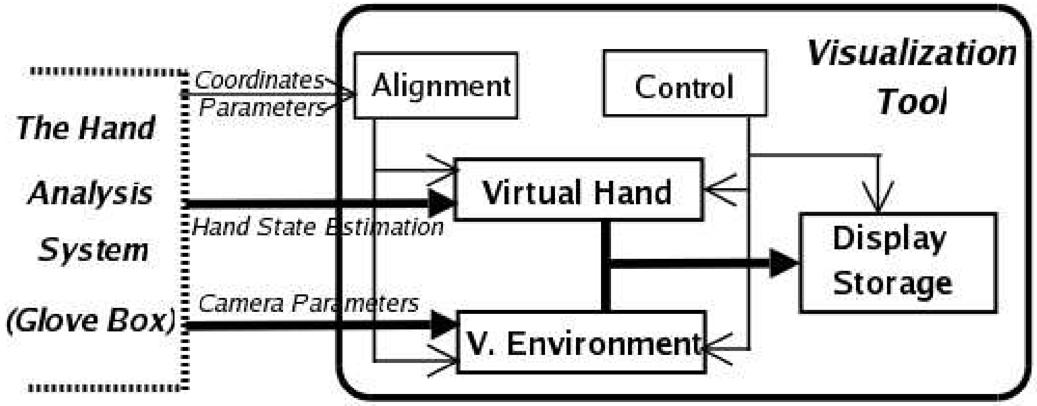


Figure 1: The visualization tool design architecture.

bility, a movable camera that can render a virtual hand from any viewing direction should be added. Also, the environment should be able to change its background, allowing for hand images of good quality.

- When the tool starts to work, it first aligns the coordinate system of the virtual environment with that of the glove box, and then sets the virtual hand position and posture according to the initial real hand configuration. The following this, subsequent outputs should be automatically processed.
- The relationship between a real hand gesture in the glove box and a virtual hand configuration in the virtual environment should be set up by matching the virtual hand parameters (such as joint angles and palm orientation) with the features of the first real hand gesture (such as a fist or a spread hand). Therefore, the subsequent hand gestures should be easily reproduced in the virtual environment if these hand gestures are identified in the hand gesture analysis system.
- As an extension to a merely passive output platform, flexible operations should be included in the visualization tool's interface. Such operations should be provided for the movable camera, cameras' focusings, rendering styles (such as lifelike hand, colored hand, and wireframe hand), and choice of recording images and rendering parameters (such as virtual hand's viewing and projection matrices).

During the second part of design process, the interface design, we followed the general principles for human-computer interaction and user interface design proposed in [9, 11]. More precisely, the program should perform operations the way users think the program should behave, that is, the users of the visualization tool should not have to "learn" how to use the tool and how to operate the virtual hand. These operations should be obvious to the users. Some of the specific user interface design considerations are as follows:

- The tool's interface should provide operations for controlling the rendering style for the virtual hand (e.g., a lifelike hand, colored hand, or wireframe hand).
- The use interface should use graphical widgets for operations on the virtual hand and virtual environment (e.g., slides for controlling camera focus or dials for adjusting hand joint angles).
- Because of the limited size of the computer screen, the hand gesture images rendered from all the virtual cameras need not be displayed on the screen at the same time. However, the tool should provide convenient operations to allow users to select a camera for displaying and recording the hand image sequences rendered from that camera.
- All the cameras' parameters should be adjustable and the background for the environment should be determined by the users.
- The rendered images should be recorded in different sizes and the recording process should be controlled the users.

4 IMPLEMENTATION & RESULTS

We implemented the hand model with the graphical library (Coin) Open Inventor [2] and the graphical user interface with Qt [1]. Both toolkits are cross-platform. Figure 2 shows several screen-shots from a hand animation process (a spread hand to a fist) rendered in different styles with various backgrounds. These hand images generated in different styles not only provide a variety of displaying features but also have potential applications in the hand gesture analysis system. For example, hand images rendered with base color model can be fed back into the system to test the accuracy of the hand gesture recognition; a set of colored hand images, generated with a set of hand joint angles corresponding to certain gestures can be used as input hand data (ground hand data) for a hand gesture analysis system.

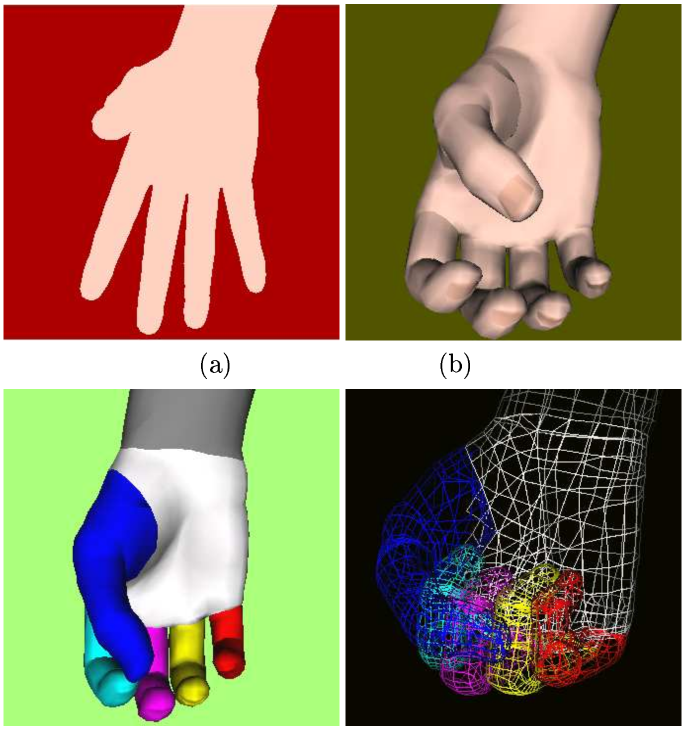
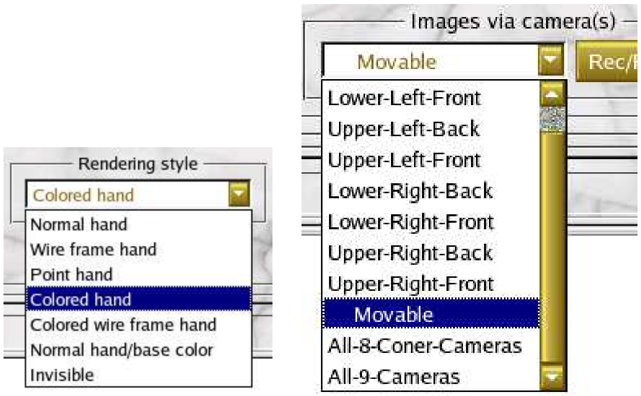

(a) (b)
(c) (d)

Figure 2: The virtual hand rendered in different styles: (a) a hand rendered with base color model; (b) a life-like hand; (c) a colored hand; (d) a hand in wireframe.

Figure 3 presents the screen-shot of our visualization tool's user interface. All the operations are available through a single panel. The icon widgets for various operations accessible via the interface are self-explanatory: buttons labeled with command names suggest different clicking operations; read-only combo boxes provide detailed parameter values within specific functions (such as camera selection and choice of rendering styles); spin boxes are used to specify image sizes; a slider is available for focusing the cameras; and a dial allows the rotation of the movable camera. All the operations are categorized into three groups: movement of the movable camera, camera and rendering control, and output control.

The right-hand side of the panel is designed for operations on the movable camera. The camera can rotate around three axes, which means that it can move to any position in a virtual sphere. Its look-up direction is automatically calculated such that it always looks at the hand. Also, it can be reseted to its initial position.

With the widgets in the middle part of the panel, the user can select any of the nine cameras (eight fixed cameras and a movable camera) to change its focus and determine the rendering style, as shown in Figure 4a. It is worth noting that each camera can have its own rendering style. For example, the user can select one camera for rendering a close image in wireframe and another camera for rendering faraway colored hand at the *same* time for the *same* hand configuration.

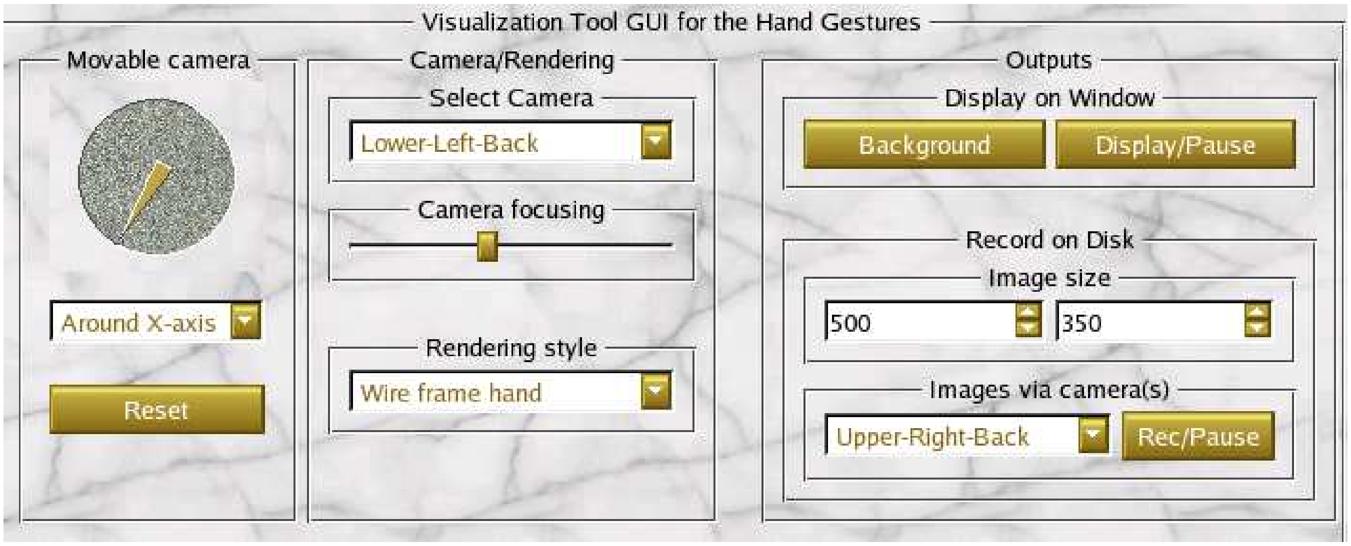


(a) (b)

Figure 4: The visualization tool interface: (a) choosing the rendering style; (b) selecting the camera for producing recording images.

Display and recording are controlled through the right-hand side of the panel. When the background button is clicked, a color palette shows up on the screen to allow the user to choose the most suitable background color for the hand image output. The output displaying can be paused and resumed. The rendered hand configurations can also be stored in image files. The image has a default size of 500x350 pixels, but this can be changed by the user through the two spin boxes available in the tool's interface. Up to nine hand images, rendered with the nine virtual cameras, can be saved for a single hand configuration. This is done with the selection of the "Images via camera(s)" read-only combo box (see Figure 4b). Figure 5 shows the same hand gesture rendered through six different cameras.

On-screen displaying of sequences of hand configu-


Visualization Tool GUI for the Hand Gestures

Movable camera

Around X-axis

Camera/Rendering

Select Camera

Lower-Left-Back

Camera focusing

Rendering style

Wire frame hand

Outputs

Display on Window

Record on Disk

Image size

500

350

Images via camera(s)

Upper-Right-Back

Figure 3: The visualization tool's main interface

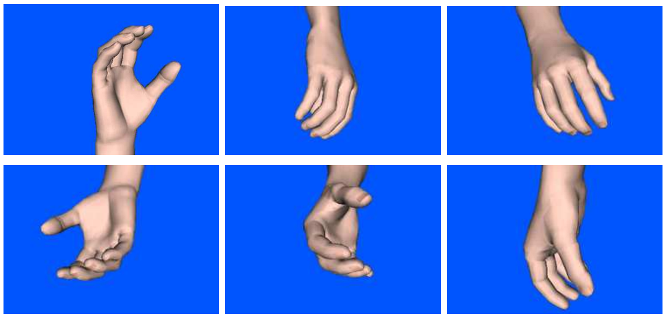
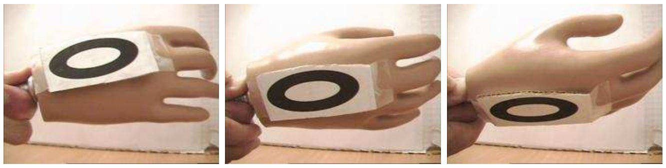


Figure 5: A hand gesture rendered through six cameras with the upper left one from the movable camera.

rations from the hand analysis system takes place in real time. Although saving images on the disk may slow down the recording process, none of images from the cameras will be lost. We solved this problem by forcing the rendering mechanism to stop its internal clock (thus preventing the rendition of next hand configuration) while the current renditions are in the process of disk saving. The rendering for the next hand configuration will starts only after the current rendition has been stored on the disk. All the hand configuration parameters (hand joint angles and orientation), virtual camera parameters and rendering matrices (all of which are used for the producing of the virtual hand gestures) are recorded in text files. Image file names and their corresponding parameter text file names are defined when the running of the visualization tool starts. Conveniently for the user, the recording process can be paused and resumed.

The alignment of the virtual hand with the real hand in glove box is done at the beginning of the program run. The parameters of the coordinate systems and cameras for the glove box are sent to the visualization tool when the tool starts to run, together with the hand parameters of the first images of the real hand in the glove box. The visualization tool uses these parameters and the initial conditions of the virtual hand and virtual environment to match the virtual hand with the real hand gesture and then to set a transformation table between the virtual environment and the glove box. The sequential hand configurations from the glove box are therefore automatically transformed into the corresponding hand configurations in the visualization tool. Figure 6 demonstrates this matching process (due to the initial stage of our hand gesture analysis system, we used only a dummy hand and its orientation).



(a) Images from the hand gesture analysis system

(b) The matched outputs in virtual environment.

Figure 6: The alignment of the virtual hand.

5 CONCLUSION AND FUTURE WORK

In this paper, we have presented a visualization tool for displaying hand gestures. Once estimated, the hand states in a hand gesture analysis system can be represented by a lifelike virtual hand in a virtual environment that is constructed with the same specifications as the system's glove box. Currently, we are extending this visualization tool to a human hand simulation system in which users can have full control of the virtual hand and virtual environment and can create any natural gestures and animation sequences. Due to the initial stage of our work on the gesture recognition system, the proposed visualization tool has not yet been connected to this system and we used text-files for conveying hand configurations (including coordinate system specifications). Nevertheless, this will not prevent the visualization tool from working correctly when connected to a gesture analysis system in the future. To ensure this, we have defined the system's output format as the format used in the visualization tool and have designed and implemented our visualization tool in a way that allows sufficient time for real time processing.

REFERENCES

[1] Qt Programming. Available from World Wide Web: `http://www.trolltech.com/` [cited November 18, 2004].

[2] The Coin Source [online, cited November 18, 2004]. Available from World Wide Web: `http://www.coin3d.org`.

[3] Geoge Bebis et al. Development of a nationally competitive program in computer vision technologies for effective human-computer interaction in virtual environments [online]. November 2002 [cited November 5, 2004]. Available from World Wide Web: `http://www.cs.unr.edu/~aerol/projecthome/publications.htm`.

[4] Paul W. Brand. *Clinical Mechanics of the Hand.* The C. V. Mosby Company, 1985.

[5] Edmund Y. S. Chao, Kai-Nan An, William P. Cooney III, and Ronald L Linscheid. *Biomechanics of the Hand: A Basic Research Study.* World Scientific Publishing Co. Pte. Ltd., Singapore, 1989.

[6] Nebojsa Jojic, Thomas Huang, Barry Brumitt, Brian Meyers, and Steve Harris. Detection and estimation of pointing gestures in dense disparity maps. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, DC, USA, 2000. IEEE Computer Society.

[7] American Academy of Orthopaedic Surgeons. *Joint Motion: Method of Measuring and Recording.* Churchill Livingstone, New York, 1988.

[8] Vladimir I. Pavlovic et al. Visual interpretation of hand gestures for human-computer interaction: A review. Available from World Wide Web: `http://citeseer.nj.nec.com/pavlovic97visual.html` [cited November 19, 2004].

[9] Jennifer Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design: Beyond Human-Computer Interaction.* John Wiley & Sons, Inc, 2002.

[10] F. K. H. Quek and M. Zhao. Inductive learning in hand pose recognition [online, cited November 19, 2004]. Available from World Wide Web: `http://csdl.computer.org/comp/proceedings/fg/1996/7713/00/77130078abs.htm`.

[11] Joel Spolsky. *User Interface Design for Programmers.* Apress, 2001.

[12] Ying Wu and Thomas S. Huang. Hand modeling, analysis, and recognition for vision-based human computer interaction. *IEEE Signal Processing Magazine*, 18(3):51–60, May 2001.

[13] Beifang Yi. Virtual Hand: a Human-Hand Simulation System for Human Computer Interactions in Virtual Environment, 2004. Available from World Wide Web: `http://www.cs.unr.edu/~b_yi/vHand/` [cited November, 2004].