Technology and Health Care 12 (2004)
IOS Press

A Non-Contact Mouse for Surgeon-Computer Interaction

C. Graetzel, T. Fong*, S. Grange, and C. Baur
Institut de production et robotique
Ecole Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland
http://vrai-group.epfl.ch

Abstract. We have developed a system that uses computer vision to replace standard computer mouse functions with hand gestures. The system is designed to enable non-contact human-computer interaction (HCI), so that surgeons will be able to make more effective use of computers during surgery. In this paper, we begin by discussing the need for non-contact computer interfaces in the operating room. We then describe the design of our non-contact mouse system, focusing on the techniques used for hand detection, tracking, and gesture recognition. Finally, we present preliminary results from testing and planned future work.

Keywords: vision-based interface, hand gesture recognition, surgeon-computer interaction

1 Introduction

1.1 Motivation

Information technology has dramatically changed medical practice in the past three decades, particularly in the areas of patient data management and preoperative planning. In the operating room (OR), however, computers tend to be used sparingly. Although there are numerous reasons for this (equipment bulk/clutter, software reliability, etc.), a primary factor is the manner in which surgeon-computer interaction currently occurs.

Computers and their peripherals are difficult to sterilize. As a result, when computer interaction is required, a common practice is for the supervising (i.e., sterile) surgeon to delegate some of portion of computer control to an assistant. For example, point-and-click interaction may be jointly performed: the assistant controls pointer position via a (non-sterile) mouse and the surgeon triggers button presses via floor pedals.

Such interaction, however, is awkward and slow. This is particularly true when the computer interface is complex and the assistant (or the surgeon) is unfamiliar with its operation. In such situations, time-consuming, spoken dialogue (e.g., “click the button on the left”) is required. Moreover, joint computer control can lead to error, especially when the surgeon and assistant have difficulty coordinating their actions.

To avoid the problems associated with delegated control, sterilizable interface hardware and speech recognition are sometimes used. These approaches allow the surgeon to interact directly with computer equipment. OR’s, however, are crowded environments, particularly near the surgical zone. Thus, it may be difficult to place touchscreens within the surgeon’s

*Corresponding author: Terrence Fong, IPR-LSRO, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland. Tel.: +41 (21) 693 78 14; Fax: +41 (21) 693 65 40; E-mail: terrence.fong@epfl.ch

reach. OR's also tend to be noisy, filled with the sounds of fans, pumps, and spoken dialogue. Hence, speech recognition is problematic, even if the surgeon is willing to wear a microphone.

1.2 Approach

Since 2001, a Swiss national research program has been investigating the potential that information technology offers for improving medical procedures and treatment. As part of this effort, we are developing user interface technologies to facilitate the use of computer equipment in the OR. Our long-term goal is to provide automated support services (equipment control, procedure monitoring, etc.) throughout the entire surgical process[7].

As a first step, we have developed a computer vision system that enables surgeons to perform standard mouse functions with hand gestures. The system uses color stereo cameras to detect 3D motion in a user-specified workspace and interprets hand gestures as mouse commands (pointer movement and button presses). The system is intended for use with minimally invasive surgery (MIS) because: (1) such procedures typically require computer support (imaging, navigation, etc.) and thus will benefit from improved HCI; (2) there are well-defined periods when the surgeon interacts only with the computer (e.g., software setup and configuration) and is not using his hands to operate; and (3) there is always at least one OR location (e.g., on top of computer displays) with a clear, unobstructed view of the surgeon.

We believe that visual gesture recognition is well-suited to the OR for several reasons. First, the OR presents a controlled, well-defined environment. Consequently, variation in illumination (color and intensity) is not a significant problem. Second, a vision-based interface does not require physical contact, which makes it usable even on top of a sterile surgical field. Third, modern CMOS cameras are small, lightweight, and easily movable. Thus, a vision system can be easily integrated into an OR. Finally, visual gesture recognition does not require the surgeon to wear additional hardware (e.g., electromagnetic trackers).

2 Related Work

Hand gesture recognition is an active area of research, particularly as a component of perceptual user interfaces. To date, a wide range of vision-based methods have been employed for detecting and classifying static hand postures and motions, including color segmentation, template matching, genetic algorithms, model-based tracking, elastic graph matching, and particle filtering[1, 3, 12, 16]. In [15], Pavlovic, Sharma, and Huang discuss different ways to model hand gestures and how such gestures are used in human-computer interaction, typically for command generation or pointing. A survey of more than 40 hand gesture recognition systems based on monocular and stereo vision is given in [10].

In recent years, numerous visual gesture mouse systems have been developed, including [8, 13, 14, 17]. Most systems, however, are designed primarily as demonstrations, or as proof-of-concept, with little regard for application constraints or performance evaluation. Two systems, that have undergone usability testing are the CameraMouse[2] and *Nouse*[5], both of which provide computer access (i.e., mouse control) to people with severe disabilities.

Our visual gesture mouse system is similar in some respects to those described in [2] and [17]. As with [2], our mouse supports the "wait to click" paradigm. Unlike [2], however, which continuously couples feature motion to pointer movement, our system requires the user to explicitly activate the mouse by engaging the tracker's attention. This interaction design is better suited for intermittent HCI and greatly reduces false-positive detection of control actions.

As in [17], we use stereo cameras for hand tracking and a finite-state machine for gesture classification. Unlike [17], our system does not require a constrained environment (a uniformly colored and illuminated background), nor does it rely on 2D contour extraction. Instead, we use normalized color segmentation, morphological filtering, and depth/shape matching.

To the best of our knowledge, our non-contact mouse is the first vision-based HCI system ever developed for the OR. Although other vision systems are used in surgery, they are restricted to functions such as image enhancement or optical tracking. As such, the design of these systems does not have to take interaction issues (usability, user variation, etc.) into consideration.

3 System Design

3.1 Methodology

We began developing our non-contact mouse system by conducting a survey of medical students and surgeons[6]. The questionnaire addressed a range of topics including: types of computer-based equipment that the surgeon would like to directly control himself in the OR, type of interaction required (on/off, pointing, etc.), locations for installing vision equipment in the OR, availability of fingers/hands for gesturing during surgery, and environmental factors (illumination, surgical clothing/gloves, etc).

The results of this survey led to the following design specifications: (1) the system should be compatible (i.e., easily integrated) with existing OR computers; (2) one-handed gestures should be recognized in a pre-defined 3D workspace located above the patient; (3) the system must function even if the surgeon is holding tools or equipment; and (4) the system must be able to ignore "parasite" gestures (hand motions not intended for interaction) and other hands/objects resembling hands.

To supplement the survey, we also observed the performance of an endoscopic nasal operation (removal of infectious tissues) in January 2003 at the Inselspital (Bern, Switzerland). This minimally invasive procedure is particularly relevant to our research because it includes use of a computer-aided navigation system[4], with which the surgeon must interact prior to, and during, surgery.

Our first observation was that the surgeon always maintained a non-occluded view of the surgical monitor that displays endoscope images and navigation data. For this procedure, therefore, installing and operating a vision system would not be a problem. We also observed that almost all human-computer interaction occurs before, or after, a surgical gesture. During the gesture itself, the surgeon's cognitive and motor workload impedes (or forbids) his ability to interact with the computer.

From a HCI standpoint, the most interesting phenomena we observed was the way in which the surgeon and his assistants made use of spoken dialogue. During the procedure, we witnessed many conversational exchanges such as:

surgeon. Move the mouse to the third button down.
assistant. This one?
surgeon. No, the next one down.
assistant. This one?
surgeon No, the other one... Yes, that's it.

This approach is sub-optimal for two key reasons: (1) it requires the surgeon to dedicate significant attention to giving orders and verifying their execution; and (2) the risk for error is high. For example, at one point, we observed a single mouse click (needed to configure the navigation system) that took 7 minutes to perform and that eventually involved four people

(including the surgeon who became de-sterilized in the process). Thus, it is clear that a non-contact mouse would greatly improve computer usability simply by allowing the surgeon, himself, to directly control the computer.

3.2 Architecture

Our current system setup is shown in Figure 1 for a MIS operating room. A color stereo camera (Videre Mega-D) is mounted on top of an OR display (e.g., the primary endoscope video display) and points towards the surgical zone, which is located 1-2 m away. Hand gesturing occurs in an 3D interaction zone (workspace), which can be adjusted by the surgeon and which typically measures 50x50x50 cm. The system is designed to work with bare hands or with colored surgical gloves.

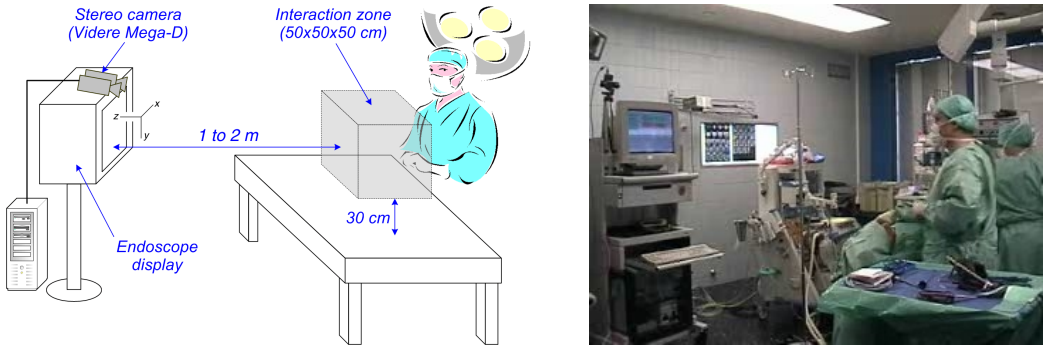

Figure 1: Non-contact mouse system setup. Left, Configuration in a MIS operating room. Right, Preliminary OR testing (Inselspital, Bern)

The system software architecture is shown in Figure 2. Color images (320x240, 24-bit) are acquired from the stereo camera and a disparity (range) image is computed using the SRI Small Vision System[11]. With proper lens selection and camera calibration, we have found that it is possible to obtain a useful (i.e., sufficiently dense) disparity image even with smooth, "untextured" gloves.

After image acquisition, hand-gesture recognition is performed in four steps: image pre-processing, hand detection, hand tracking, and gesture classification. We use a combination of color and depth processing to achieve reliable, high-speed hand detection and tracking. Our current system runs at 25 Hz on a typical office PC (2.4 GHz Pentium IV, 512 MB RAM, Windows 2000).

3.2.1 Image Pre-Processing

The initial processing step is to segment a color image into regions that correspond to the user's hands. We currently use a three-part segmentation method (Figure 2). First, we subtract a static background image (acquired at system initialization) to obtain an image that contains only foreground information (i.e., the user). From this image, we then identify pixels that are likely to be hand pixels through band-pass thresholding of normalized color values. Finally, we use morphological dilation and erosion to reduce noise and to connect closely separated pixels.

Although our segmentation approach works well in most situations, it does have two significant weaknesses, both of which we are working to address. First, because we subtract a static image, changes in the background, such as movement, can result in pixels being erroneously classified as foreground. Second, simple color filtering, even in the normalized color space, is sensitive to changes in lighting and surrounding pixel color.

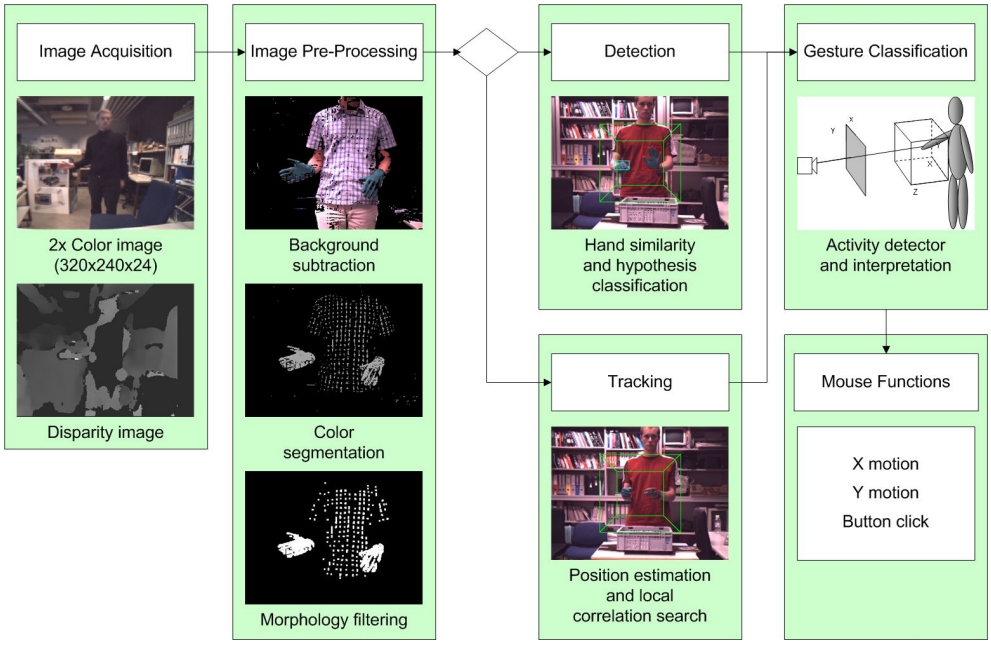

Figure 2: System software architecture

3.2.2 *Detection*

After segmentation has been performed, the resulting image contains connected regions of pixels ("blobs") that match pre-defined color ranges of the user's hands (normalized red-green of bare skin or gloves). For each blob, we then compute the 3D location and real-world size (bounding box) based on corresponding pixels in the disparity image. We consider each of these to be a hypothetical hand.

Hand detection occurs as follows. First, we discard all hypotheses that are located outside the 3D workspace. For each remaining hypothesis, we then compute a hand "similarity measure" s. Since our system only needs to track hand position (and not shape, nor finger pose), we use a simple metric that compares blob perimeter p_b and area a_b to the perimeter p and (2D projected) area a of an average adult hand.

$$s = \frac{1}{|p - p_b| + |a - a_b|}$$

If a similarity threshold is exceeded, we conclude that a hand has been detected. If multiple hypotheses exceed a similarity threshold, the hypothesis with the highest value is chosen. In either case, tracking begins. If, however, no measure is above the threshold, the system continues searching the workspace.

The primary weakness of our detection method is that we rely solely on the disparity image for depth and size estimates. In particular, if the disparity image is noisy, then the resulting estimates will be inaccurate. This can occur, for example, if there is insufficient background texture (e.g., a smooth wall) for stereo correlation.

3.2.3 *Tracking*

Once a hand has been detected, we apply local (small-window) correlation to track its movement. We use a Kalman filter to estimate hand velocity and to predict future hand position. Because hand motions are used only to control the relative mouse pointer position, absolute 3D localization accuracy is not critical. For our application, rapidly acquiring, maintaining, and releasing (when appropriate) tracking "lock" is more important.

We measure tracking quality in terms of the match (correlation fit) between the tracked shape and the image. If the quality is too low (poor match or loss of tracking "lock") or too high (multiple matches) for an extended period of time, tracking is stopped and the system switches back to detection. Tracking is also stopped if the hand is outside the workspace for too long.

3.2.4 Gesture Classification

We classify hand gestures using a simple finite state machine (Figure 3). When the surgeon wishes to engage ("pick-up") the non-contact mouse, he places his hand in the workspace and holds it stationary for a moment. We use this method to differentiate between "parasite" gestures and intentional gestures. The interaction is designed in this way because the workspace is situated just above the surgical zone. Hence, the surgeon will often have his hands in the workspace without intending to control the mouse.

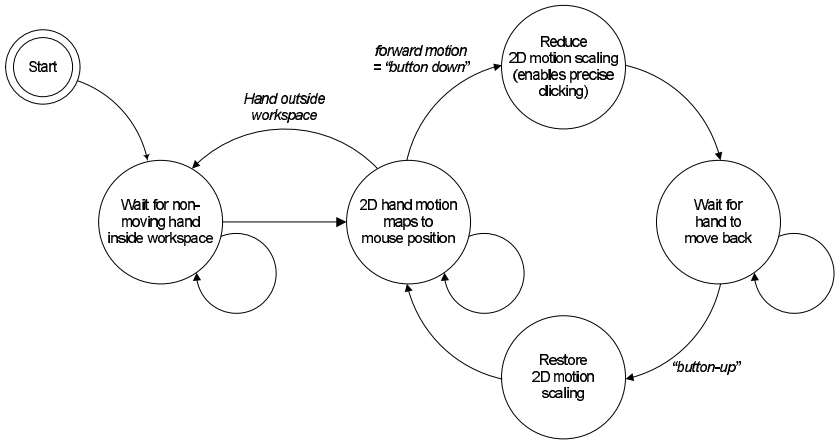

Figure 3: Gesture classification state machine

To facilitate positioning, we map hand motion to pointer movement using non-linear gains. Small, slow hand motion cause small pointer position changes. Large, fast movements cause large changes. In this way, the user can precisely control pointer alignment by moving his hand slowly, yet can also make the pointer move far when desired.

We currently provide two methods for generating mouse button clicks. The first method, "wait to click", consists in moving the cursor to the desired position and holding the hand stationary for a short time. The second method, "push to click", uses depth information to detect hand motion in the direction of the camera. A movement of 20 cm towards the camera triggers a click.

4 Results

4.1 Speed and Accuracy

To characterize vision processing, we installed the system on a typical office PC (2.4 GHz Pentium IV, 512 MB RAM, Windows 2000) and measured execution time of major processing blocks. Figure 4 shows the execution profile for detection mode (searching for a hand in the workspace) and tracking mode (following a detected hand). In both modes, the system runs at 25 Hz or greater, with the majority of time spent performing color and disparity image acquisition.

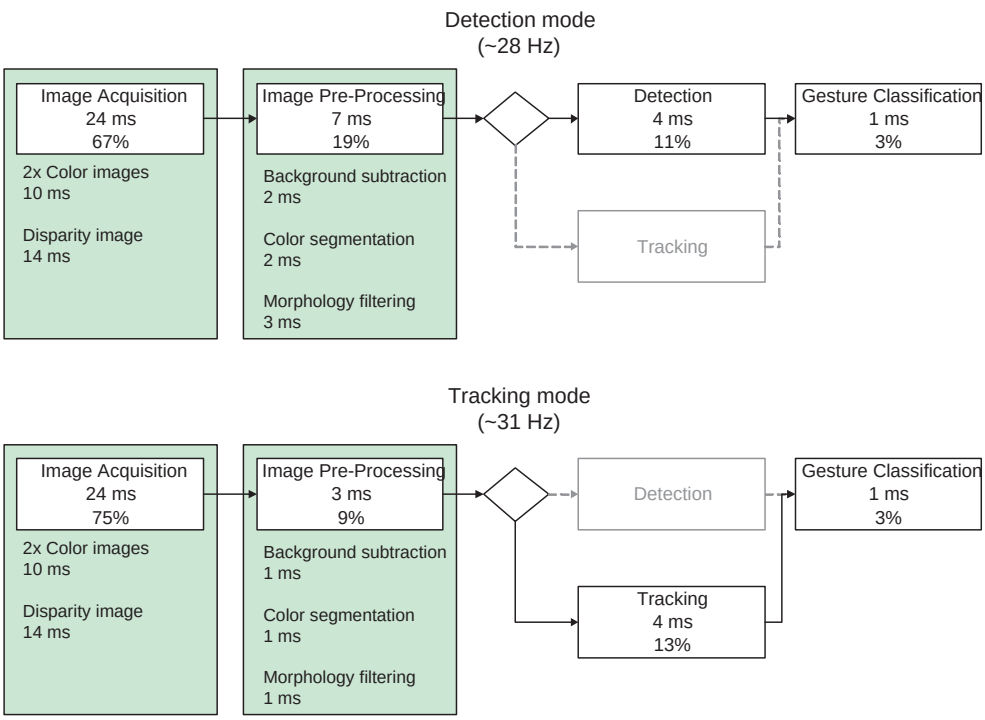

Figure 4: Execution profile. Top, detection mode; bottom, tracking mode

Tracking resolution depends on the focal length of the stereo camera lenses and the 3D workspace (location and extents), both of which can be changed by the user. With 12.5 mm lenses, a 600 cm^3 workspace located 1.7 m from the camera (see Figure 1) will be mapped to an image region measuring 146x140 pixels.

Taking into account image quantification errors, and using subpixel interpolation, the maximum theoretical spatial resolution is[6]:

$$\Delta x = 4mm, \Delta y = 4mm, \Delta z = 6mm$$

The actual spatial resolution was measured as:

$$\Delta x = 6 \pm 1mm, \Delta y = 6mm \pm 1mm, \Delta z = 10mm \pm 1mm$$

which agrees well with theory.

4.2 System Performance

4.2.1 Hand Detection

Because we use both color and depth processing, hand detection works quite well (Figure 5). In particular, having depth information provides two key benefits: (1) it enables us to restrict the search to a pre-defined 3D volume (workspace); and (2) it allows us to match hands using real-world size. As a result, the rate of false-positive and false-negative errors is low.

There are two primary situations in which false-positives (object incorrectly identified for tracking) may occur. First, an object located outside the workspace may be identified as trackable. However, because we use 3D information, this type of false positive is rared: it only occurs when the stereo camera system has been poorly calibrated (i.e., causing inaccurate position estimation).

The second false-positive situation is when there is no hand inside the workspace, but the system believes there is one. This type of error is difficult to quantify, as it depends

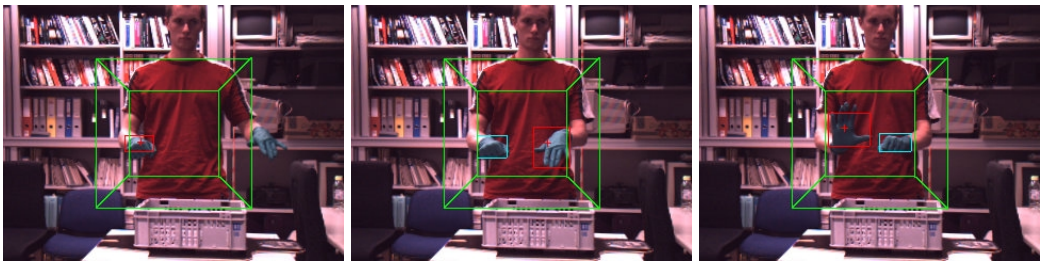

Figure 5: Hand detection. An object in the workspace is identified as a hand to track if: (1) it matches the pre-defined color range and (2) it has a perimeter and area similar to an average adult hand.

on the presence of moving "hand-like" objects inside the workspace. Appropriately setting the minimum similarity threshold may correct this (i.e., to prevent the objects from being detected). Static objects, which are permanently located in the environment, do not pose a problem because they are discarded during image-preprocessing.

False-negatives errors (hand is in the workspace but not tracked) generally occur when tracking is already locked on a "hand-like" object. Changing the similarity threshold may provide a solution. But if objects are too similar in both color and size to a hand, the only remedy may be to make the hand more easily discriminable (e.g., use a different color glove).

4.2.2 Hand Tracking

A hand may appear radically different from image to image, even if the posture seems identical from the human point of view. This is especially true when the hand is moving laterally (with respect to the camera), rotating out of the image plane, or changing form (e.g., switching from open palm to closed fist). Additionally, our current hand detection scheme sometimes identifies only portions of the hand, such as a finger or two. Thus, our hand tracker is designed to recognize changes in hand shape, size, and orientation and to adapt (re-initialize) tracking accordingly.

At the same time, however, the adaptation must be stable. If not, the system will have difficulty deciding what object to track. When this occurs, the system may fail to maintain tracking or may begin tracking a different object, which may be another hand or a non-hand. In the former case, mouse control will be intermittent (because the system has to frequently repeat the detection phase in order to reacquire a hand). In the latter, mouse control will be jumpy, unpredictable, and unreliable.

Figure 6 illustrates hand tracking performance in the presence of motion, size/shape changes, and rotation. Initially, the hand (held as a fist) is detected entering the workspace. The system then correctly follows the hand to the middle of the workspace, though the tracking center (shown as a small + symbol) is shifted. When the user opens his hand, tracking momentarily centers on the wrist. This is because the tracked shape, a fist, matches multiple places on the hand. The system detects this problem and re-adapts to track the "larger" hand. Similarly, the system re-adapts the tracked shape as the user flips and rotates his hand.

4.2.3 Impact of Multiple Hands

Since hand tracking is a local process, tracking is not disturbed by the presence of other hands (or objects similar to hands) inside the workspace. Once tracking is established, the system is fairly robust to the presence of other hands inside the workspace. Figure 7 shows the user's right hand (tracked) and left hand in various locations within the workspace. As the figure shows, even when the left hand has similar size and color, tracking remains fixed on the right hand.

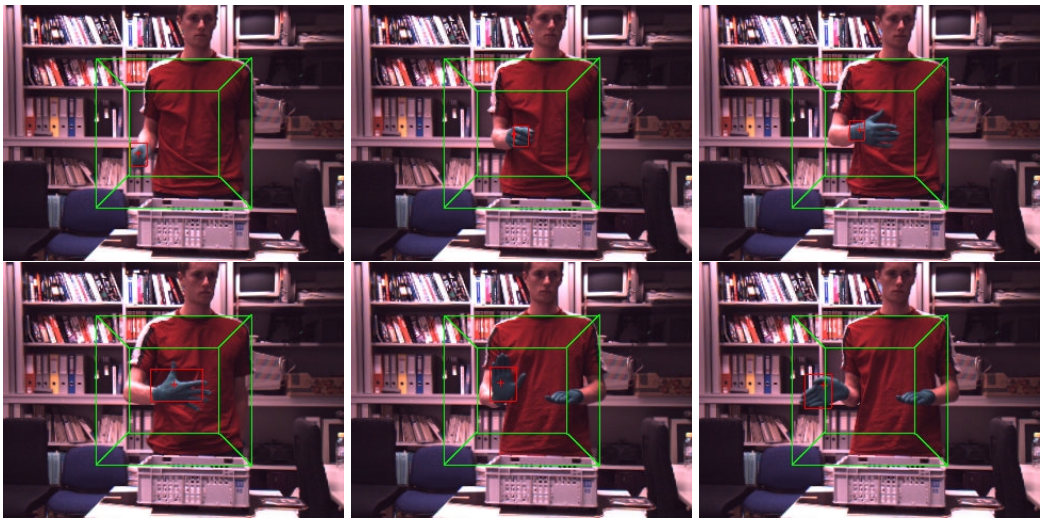

Figure 6: Hand tracking: the system continually adapts tracking in order to cope with changes in size, shape, and orientation.

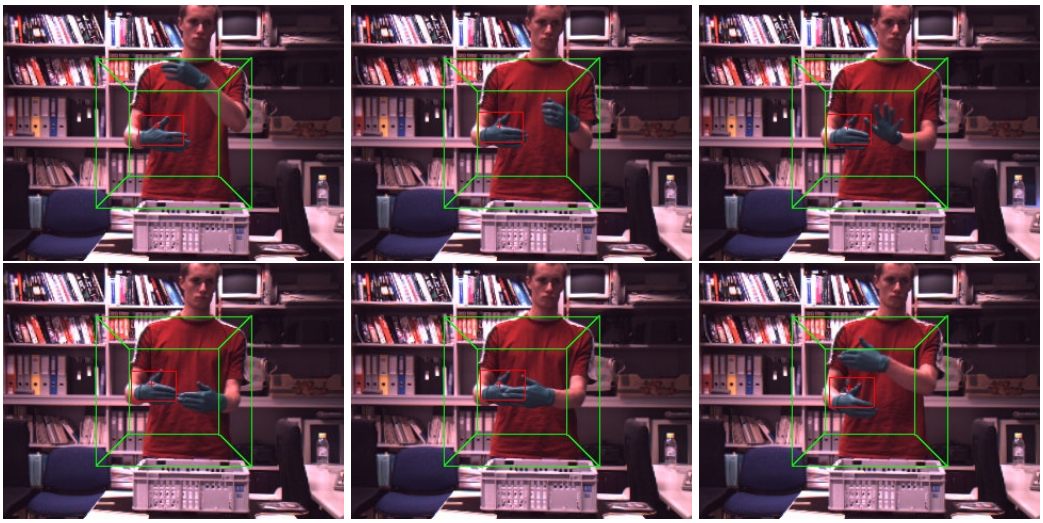

Figure 7: Multiple hands: Once a hand is "locked", the presence of other hands rarely perturbs tracking

It is possible, however, for hand swapping to occur. That is, if two (or more) hands are located close together, or are overlapped, the system can start swapping in between the hands, believing it is tracking a single hand. This results in sudden jumps of the pointer on the screen. With practice, though, this failure mode is rapidly identified and easy to avoid (i.e., the user learns to keep the hands well separated or easily differentiable).

4.2.4 Impact of Lighting Changes

As with all vision systems, lighting conditions can greatly influence performance. Although we use normalized color to reduce the impact of lighting, our current system has difficulty in situations dominated by saturation effects (e.g., full sunlight) and dynamic changes in intensity. However, since our system is designed for use in OR's, which have controlled lighting and generally do not have exterior windows, this is not a significant problem.

In a series of tests, we evaluated our system in a range of ambient lighting conditions (Figure 8). We found that between 200 lux (dim, fluorescent indoor) and 1200 lux (bright, indirect sunlight), both hand detection and tracking worked well.

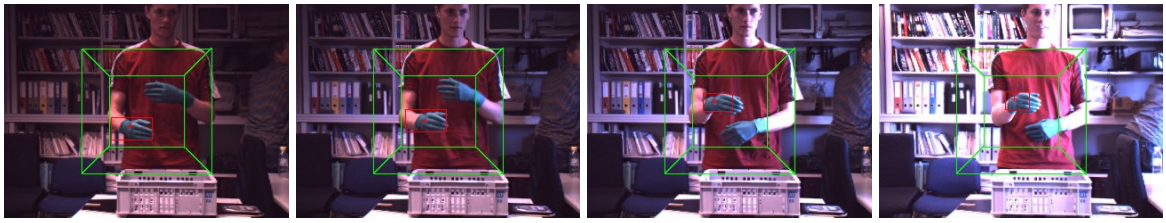

Figure 8: Lighting tests. Left to right: dim, fluorescent indoor (200 lux), mostly closed curtains (700 lux), partially open curtains (900 lux), bright, indirect sunlight (1200 lux)

5 Usability Tests

To evaluate the usability of the non-contact mouse, we developed a mock-up medical interface (Figure 9). This user interface tests a variety of interaction modalities: menu navigation, button presses, and analog scale setting (2D and 3D). To provide visual feedback, the cursor appearance changes to indicate when mouse control is acquired and when a click is about to be triggered.

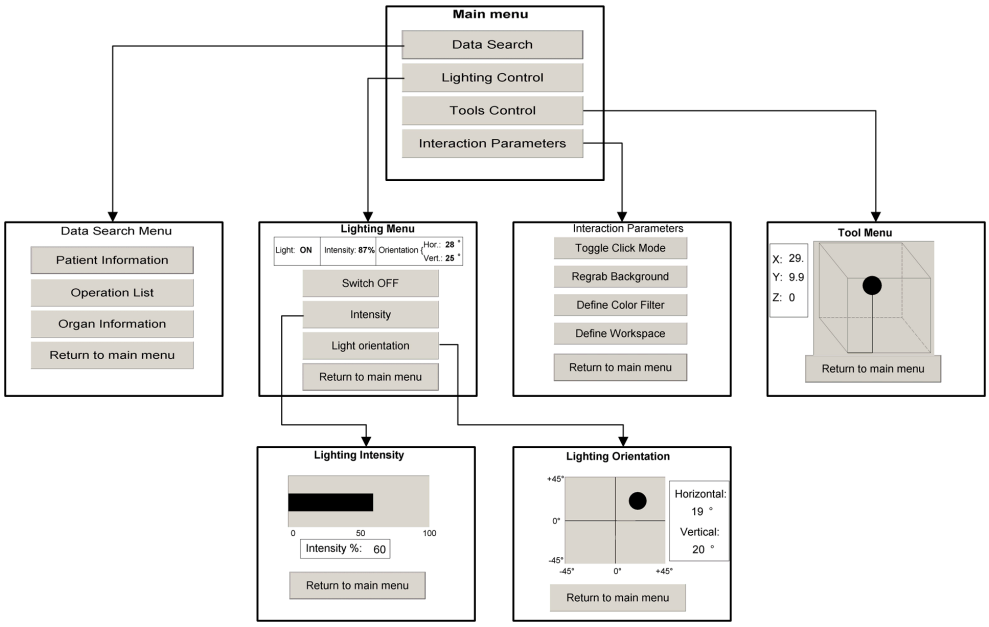

Figure 9: Mock-up “medical” user interface used for usability tests

In a first set of tests, 16 subjects (including 2 medical students and a perceptual user interface expert) with varied background and computer experience were asked to explore the interface and then to perform various tasks, some of which were timed. At the end of each test session, each subject completed a questionnaire and were asked questions about their experience.

Overall, we found the usability of the system to be good. All subjects were able to rapidly learn how to use the system. We found that navigation and button clicking were the fastest tasks: average time to click anywhere on the full-screen display was less than 5 sec. Setting an analog scale took more time, since cursor positioning needs to be precise. On average, setting a scale to within 1% of the target value required 12 sec.

We observed that all subjects initially had difficulty working inside the 3D workspace. At first, users would lose control of the mouse because their hand inadvertently passed out of the workspace. With experience, however, users learned to use rapid hand motion to access all points on the display while keeping their hand in the workspace.

A majority of subjects preferred "push to click" mode because it provides some (minimal) level of kinesthetic feedback. The main problem problem with this click paradigm is that users have difficulty moving their hand purely forward. As a result, undesired pointer motion sometimes occurred during clicking. This was particularly visible when users wanted to click in lower parts of the workspace: in this position, users have a strong tendency to extend their arm, thus moving vertically and horizontally.

To address this, we implemented an activity detector to classify the type of movement the user is trying to perform: pausing, positioning, and clicking. When a clicking movement is detected, the mouse gains (horizontal and vertical translation) are lowered, so that inadvertent lateral motion does not perturb the current pointer position.

We found that there are three primary weakness with the current system: (1) the mouse pointer jitters too much under dim lighting conditions; (2) the system has difficulty following rapid gestures; and (3) user confusion due to perceived differences between hand position and mouse pointer position.

6 Initial OR Testing

To assess strengths and weaknesses, we installed our system in an OR (Inselspital, September 2003) and collected image data during a computer assisted endoscopic operation (Figure 1, Right). We observed the following:

- There are numerous objects located in the workspace throughout the operation.
- The ambient lighting is generally very dim, in order to provide an acceptable endoscope camera image to the surgeon.
- The endoscope display provides an ideal location for the stereo camera: 1.5 to 2 m from the surgeon with a completely unobstructed view throughout the operation.

After the operation was complete, we conducted a cognitive walkthrough test with the surgeon. This testing revealed the following:

- The surgeon preferred the "wait to click" paradigm because he felt it was easier to use (i.e., requires less hand motion) while offering higher accuracy.
- Adding static hand posture recognition was not felt to be a necessary, nor beneficial, change. In fact, the surgeon argued that static hand gestures would require training and additional concentration, both of which are undesirable given the surgeon's already heavy workload.
- The possibility of a dynamic workspace, which would follow the surgeon's body, was also not seen as a necessary improvement. A fixed workspace, defined by surgeon, is more compatible with the plan-structured nature of surgery.
- Waiting to "pick-up" the mouse was not a problem. In fact, avoiding unintentional cursor control (by explicitly having to engage the system) is considered to be an important design feature.

Overall, the surgeon showed strong interest in the system and was confident that visual gesturing could be useful inside OR's. He emphasized, however, that it is important for the system not to impose additional cognitive load, nor interfere with the way surgical gestures are normally performed.

7 Future Work

We have recently begun packaging our system to facilitate setup and use. Our approach is to perform all vision and gesture processing on a laptop computer and to output mouse commands via a serial port (encoded with the Microsoft mouse communication protocol). We plan to use serial mouse adapters so that the system can easily be connected to a wide range of computers including Windows PC's, Macintosh, and Sun workstations.

We have also begun collecting stereo camera image sequences from a variety of OR's, both during and outside of surgery. This data will be used to test and refine the vision system. In particular, we wish to evaluate the efficacy of hand tracking and gesture recognition, as well as to characterize the impact of OR lighting and configuration variations.

For initial clinical trials, we intend to deploy the system at the Inselspital during 2004. Surgeons will use the non-contact mouse to configure and calibrate a computer-aided navigation system[4]. Because these tasks are performed prior to the start of surgery (but after the surgeon has sterilized his hands), no surgical risk will be incurred.

To improve tracking reliability and robustness, we plan to implement dynamic background subtraction and color histogramming. Dynamic background subtraction uses depth information to assist scene segmentation. This should lead to a significant reduction of noise and cleaner removal of background image regions. Color histogramming is well-known to improve color segmentation, particularly when significant variations in illumination or local pixel color are expected between image frames.

Acknowledgments

We would like to thank Dr. M. Caversaccio (Inselspital) and Dr. P.-Y. Zambelli (Orthopedic Hospital of Suisse Romande) for providing valuable comments and their medical insight. This work was supported by a grant from the Swiss National Science Foundation Computer Aided and Image Guided Medical Interventions (NCCR CO-ME) project.

References

[1] J. Aggarwal and Q. Cai, Human motion analysis: a review, Computer Vision and Image Understanding 73(3) (1999), 428-440.

[2] M. Betke, J. Gips, and P. Fleming, The Camera Mouse: visual tracking of body features to provide computer access for people with severe disabilities, IEEE Transactions Neural Systems and Rehabilitation Engineering 10(1) (2002), 1-10.

[3] M. Black and A. Jepson, A probabilistic framework for matching temporal trajectories: condensation-based recognition of gestures and expressions, in: Proceedings of the European Conference on Computer Vision, 1998, pp. 909-924.

[4] M. Caversaccio et al., The Bernese frameless optical computer aided surgery system, Computer Aided Surgery 4 (1999), 328-334.

[5] D. Gorodnichy, S. Malik, and G. Roth, Nouse 'Use Your Nose as a Mouse' - a new technology for hands-free games and interfaces, in: Proceedings of Vision Interface, 2002, pp. 383-390.

[6] C. Graetzel, Interface utilisateur baseé sur les gestes visuelles pour chirurgie, Technical Report, VRAI Group, Swiss Federal Institute of Technology, Lausanne, 2003.

[7] S. Grange, Vision-based human computer interaction for medical applications, Ph.D. thesis proposal, Technical Report, VRAI Group, Swiss Federal Institute of Technology, Lausanne, 2003.

[8] C. Hu et al., Virtual Mouse — inputting device by hand gesture tracking and recognition, in: Proceedings of the International Conference on Multi-modal Interface, 2000, pp. 88-95.

[9] T. Huang and V. Palvovic, Hand gesture modeling, analysis, and synthesis, in: Proceedings of the IEEE International Workshop on Automatic Face and Gesture Recognition, 1995, pp. 73-79.

[10] M. Kohler and S. Schröter, A Survey of video-based gesture recognition - stereo and mono systems, Technical Report 693, Informatik VII, University of Dortmund, 1998.

[11] K. Konolige, Small Vision System: hardware and implementation, in: Proceedings of the International Symposium on Robotics Research, 1997, pp. 111-116.

[12] J. LaViola, A survey of hand posture and gesture recognition techniques and technology, Technical Report CS-99-11, Department of Computer Science, Brown University, 1999.

[13] S. Li, W. Hsu, and H. Pung, A real-time monocular vision-based 3d mouse system, in: Proceedings of the International Conference on Computer Analysis of Images and Patterns, 1997, pp. 448-455.

[14] R. Lockton, and R. Fitzgibbon, Real-time gesture recognition using deterministic boosting, in: Proceedings of the British Machine Vision Conference, 2002, pp. 817-826,

[15] V. Pavlovic, R. Sharma, and T. Huang, Visual interpretation of hand gestures for human-computer interaction: a review, IEEE Transactions Pattern Analysis and Machine Intelligence 19(7) (1997), 677-695.

[16] A. Ruf, Bibliography on computer vision and graphics for motions of human hands, <http://www.inrialpes.fr/movi/people/Ruf/hand.htm> (April 2001)

[17] J. Segen and S. Kumar, GestureVR: vision-based 3d hand interface for spatial interaction, inn: Proceedings of the 6th ACM International Multimedia Conference, 1998.