Red-Handed: Collaborative Gesture Interaction with a Projection Table

Chris McDonald
School of Computer Science
Carleton University
Ottawa, ON, Canada
cmcdona3@scs.carleton.ca

Gerhard Roth
Computational Video Group
National Research Council
Ottawa, ON, Canada
Gerhard.Roth@nrc-cnrc.gc.ca

Steve Marsh
Computational Video Group
National Research Council
Ottawa, ON, Canada
Steve.Marsh@nrc-cnrc.gc.ca

Abstract

Collaboration is an essential mechanism for productivity. Projection tables such as the SociaDesk enable collaboration through the sharing of audio, video and data. To enhance this form of interaction, it is beneficial to enable local, multi-user interaction with this media. This paper introduces a computer vision-based gesture recognition system that detects and stores the gestures of multiple users at the SociaDesk. This system was designed to be a sub-module of high-level applications that implement multi-user interaction. With such a system, users can collaboratively interact with software-based tools at a projection table.

1. Introduction

The use of collaboration, both local and at a distance has always been essential for a productive workflow. The SociaDesk enables these forms of collaboration through its hardware design and software algorithms. This paper describes a hand gesture system that creates a natural mechanism for local interaction and collaboration. There are several projection tables being developed which incorporate hand gesture interaction. Our approach is to replace the hardware mouse with hand gesture input from the uniquely identified users at the table. Processing the images obtained from an overhead camera enables the system to detect and classify hand gestures. Since the projected background can vary unpredictably, we combine the information obtained from colour and infrared camera images. We also have small wrist-worn targets to distinguish between different users.

In order to perform gesture recognition reliably in real-time, a simple mouse replacement model is implemented. The finger and thumb tip position are tracked in real-time. This makes it possible to recognize the postures that are translated in to mouse-up and mouse-down events. Using this technique, the gesture input can be used with any software running within the Windows environment.

2. Related Work

Projection table technology has quickly become an attractive means for collaborative interaction. Several approaches to the problem of facilitating interaction with the table have been proposed [1,2,3,4]. Some approaches use complex key point motion to infer gesture [1], while others use posture [2,3].

3. The SociaDesk

SociaDesk was designed from the first as a technology that can facilitate action in distributed and co-located collaborative environments. We believe that collaboration is an underlying mode of behaviour that can be supported by technology regardless of the task the technology is being used to accomplish.

A new model of collaboration support, called Table Manners, is being developed. Table Manners is an autonomous agent based architecture that, it is hoped, will provide users of SociaDesk with cues, clues, and directions in the task at hand. This architecture has in fact benefited a great deal from the understandings of task and Quality of Experience (QoE) developed in [5]: that different task compositions require different environmental support structures in order to achieve the best possible QoE for the task participants. As well, Table Manners is closely related to the concept of the Artificial Chairman found in [6] as well as the Helper Agent [7].

Briefly, Table Manners presents each unique user in the system (hence the need for identification

Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'04)
0-7695-2122-3/04 $ 20.00 © 2004 IEEE

RedHanded answers) with a personal semi-autonomous agent (extended ACORN agents – see [8]) which will construct and maintain a personal and social model of the user based on their observable interactions with others, and their personal interactions with the agent (e.g., via handheld devices, or online, for individual input).

Additionally, the environment itself (the SociaDesk) will have its own agent. This agent will be cognizant of who is around the virtual table. It will also have some level of knowledge of the task at hand, or at least the process of collaboration that is required to be supported. Note that, this requirement is different in different tasks (design meeting, decision making discussions, command and control with some urgency, for example). It is also different based on the people actually in the environment, whether friends, informal, formal, and so on. Our current work is based on the definitions of these particularities.

The individual software agents will be able to negotiate with each other and the SociaDesk agent (and other environmental agents) in order to facilitate local and global control over aspects of the tabletop display (virtual objects, for example). In addition, agents will be able to manipulate group behaviours (by suggesting specific tasks, either individually or to the group) in order to facilitate group behaviours, and reduce group tension. These suggestions will be guided by the environmental agents' knowledge, as discussed above.

It is clear to us that the proposed agents must be inherently Socially Adept – that is, they must be able to model and to some extend understand user states of emotion, trust, and intention, for example, in order to be able to facilitate the interactions between users. They may also need to be capable of representing the user to other such agents in negotiations. Evidently, while there are some situations (command and control, for example) where some form of chain of command is clear, this is clearly not going to be the case in all situations.

To take the model further, we will be working towards an architecture where each manipulable virtual object in the space will be represented by an agent that understands something of what it represents. It may well be the case that this object represents a person, or something constrained physically in the space. The object's agent should be aware of this and inform user agents (and thus users) if these constraints are violated. An example may clarify our thoughts in this direction. In a command and control situation, for example in a large emergency where the SociaDesk is being used to manage and monitor the deployment of emergency services, the objects being monitored on the table surface are real people in physical space. A decision to commit these people by SociaDesk 'managers' will need to be safe and sensible for these real people before it can be committed. Agents within SociaDesk for each member of the personnel on the ground can help manage this situation.

It is clear that this architecture enables the environment in powerful ways, allowing the user to concentrate on interacting with other users, and with the environment itself.

Some of the projects associated with the Table Manners project include:

- Understanding group tensions in collaborative environments, and codifying them, in order to facilitate group interactions better.
- Individual collaborative environments – sharing data with tabletop and other displays using handheld devices.
- Networked robotics (remote control) and haptics.
- Advanced videoconferencing over broadband.
- Collaborative gaming.
- Virtual collaborative design (3D)

In order to achieve all of this, it was necessary to realise a control architecture for SociaDesk as a basic need. To that end, the RedHanded project has developed natural interaction techniques.

4. Natural Interaction

When collaborating at the SociaDesk, the standard mouse is far too cumbersome to share amongst different users. This fact demonstrates the lack of multi-user input within the current operating system technology. The Windows operating system was designed to accept input from a single mouse device, and therefore behaves unpredictably in the presence of input from multiple mouse-like sources. For this reason, an interface that supports multiple users interacting concurrently must have the ability to detect multi-user input and translate it into events that are understood by the operating system.

5. The "Red-Handed" Gesture System

To interface the projection table hardware with natural gestures, the Red-Handed computer vision library was developed. This library incorporates image processing techniques in combination with the Windows messaging queue to replace mouse clicks with simple hand gestures.

5.1. Computer Vision Input

To recognize gestures it is necessary to analyze the frames captured from an overhead video camera. In order to correlate the hand position in the camera frame with the hand position in desktop space, a homography transformation is pre-computed. This transformation essentially stabilizes the camera frames with respect to the desktop space. Within this stabilized reference, the geographic relationship between hand and virtual information is trivial as pixel locations in the stabilized frame are mapped directly to pixel locations on the Windows desktop. Figure 1(a) and (b) show an image captured by the overhead camera and the corresponding stabilized image respectively.

(a) (b)

Figure 1. Stabilized camera frames

In order to simplify the process of localizing hands and distinguishing between them, a target-tracking scheme was used.

5.2. Target Detection

Targets are worn on the wrists of the user's who wish to participate in the gesture interaction. These targets uniquely identify each user and also simplify the hand location process. The ARToolkit library was used to perform the initial detection of the targets. This library was modified to improve the detection by using the custom patterns suggested in [9], as shown in Figure 2.

Figure 2. Wrist Target Detection

5.3. Hand Detection

The goal of the hand detection phase is to collect a set of pixels for each user that represent the shape of the hand. The initial phase of the Red-Handed system uses a colour-based approach to hand detection. This requires a segmentation scheme that isolates skin-coloured pixels in the image. Much research has been done on the technique of mapping between colour spaces as a way of improving the clustering of skin pixels in colour-space. Several colour spaces have been surveyed in [10,11,12]. After testing a number of these spaces, it was concluded that the HSV (Hue-Saturation-Value) and YUV [13] spaces performed the best in our environment. It was also clear that the combination of these two spaces helped further reject unwanted colour. For this reason, the Red-Handed system uses the five colour channels Y,U,V,H and S (The Value channel from HSV is discarded as it is similar to the Y channel from YUV).

As a first step, a small rectangular region is identified for each user at a fixed location relative to that user's detected target. A skin colour model is built based on a histogram of pixel intensity values from this region in each of the five pre-defined color channels. All pixels in this region are assumed to be hand pixels, so they all contribute to the computation of the model. A peak in this 5D colour space histogram is computed from these pixels, and this peak represents the skin colour model for that user. This model is then used to classify pixels connected to this rectangular region; a pixel is classed as being skin colour if it is within a pre-determined maximum distance from the peak in colour-space. Starting from this local rectangular region a connected-region flood-fill algorithm is used to create a connected region of skin colour pixels. This connected region is defined as the hand pixel set for that user. Figure 3 shows a detected hand pixel set, shown in red along with the target and the predefined rectangular region.

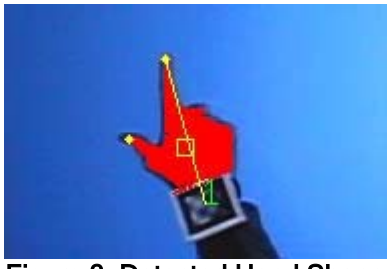


Figure 3. Detected Hand Shape

5.4. Key Point Extraction

In order to reduce the complex task of generalized gesture recognition, a key point detection scheme is used. This makes it possible to differentiate between the finite postures necessary for successful gesture inference. The key points used are a pointer location and point that identifies a posture change. First, we find the farthest point in the hand pixel set from the pre-defined target. Then we compute the principle axis of the entire hand pixel set. The pointer is located at the same distance as this farthest point but along the principle axis. This process helps stabilize the pointer, but there is still some jitter. To minimize this, the key point location is updated only when the detected motion is greater than a few pixels. The thumb is used to indicate a change in posture. The thumb is located by finding the point that lies farthest to the left (for right handed users) from the line joining the pointer and the target center. These key points are illustrated in Figure 3.

5.5. Gesture Inference

Since this system was designed to replace the behavior of a single button mouse, only two postures are necessary [14]. A point posture is defined as an extended index finger and a closed thumb. A select posture is defined as an extended index finger and an extended thumb. The thumb is considered extended if the thumb point is farther than a pre-determined distance from the line joining the pointer and target. The transition between these two postures represents the gestures.

(a) (b)

Figure 4. Gesture Inference

5.6. Mouse Event Generation

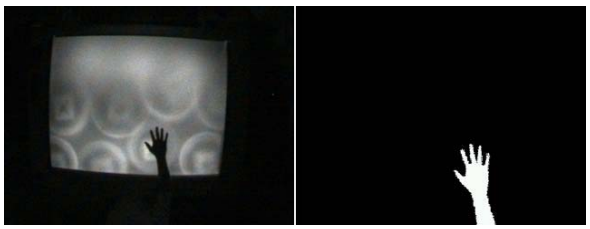
When a transition between the postures is detected, the appropriate mouse event is simulated. This is accomplished by adding a message to the Windows message queue. When the user transitions from a point posture to a select position, a mousedown event is generated. Similarly, when the user transitions from a select posture to a point posture, a mouseup event is generated.

6. Improving the System

6.1. Limitations of Colour-Based Detection

The robustness of the colour-based hand detection scheme is heavily dependent on the background projection. A background colour similar to the hand can influence the integrity of the detected hand pixel set. Since the Windows desktop varies over time, the unpredictability of the background colours forces the use of alternative methods for robust hand detection.

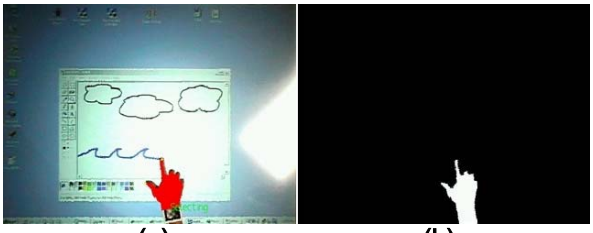
For this reason, the system was updated to incorporate the commonly used infrared shadow scheme [1,2] for hand detection. An overhead camera, equipped with an infrared filter, is mounted above the table. An infrared light source is positioned to project infrared light onto the table top from below. With this design, any tabletop occlusion will cast a shadow in the infrared-filtered camera image, as shown in Figure 5(a). Infrared technology produces a consistent tabletop image regardless of the projected desktop or artificial environmental lighting. This means that the occluded portions of the table surface will consistently appear significantly darker in the infrared camera image. Therefore, a simple threshold produces a binary image containing a clear representation of the shadows belonging to the table occlusions. The infrared images are stabilized using a pre-computed homography for the infrared camera. Figure 5(b) shows the stabilized infrared image after the threshold algorithm has been performed.



(a) (b)

Figure 5. Stabilized Infrared Frames

The target location in any given colour image overlaps a hand blob in the corresponding infrared image. This fact is used to match the pre-defined targets with infrared blobs, which simplifies the problem of determining the orientation of the hand relative to the image. Figure 6(b) shows the stabilized and thresholded hand occlusion over the table surface. It is clear that the hand pixel set integrity is unaffected by the actual desktop projection, shown in Figure 6(a).



(a) (b)

Figure 6. Gesture Interaction using Infrared Hand Pixel Set

Although the infrared solution improves the integrity of hand shape detection in the given environment, the colour-based method is not obsolete. In the presence of significant external infrared lighting, such as sunlight, the infrared-based hand detection is the inferior solution. In this situation the occluding hand may be lit from above by infrared light, creating a less significant shadow in the infrared-filtered camera image. This causes the hand shape integrity to be compromised when the image undergoes thresholding. The other advantage to the colour-based system is the hardware cost. Both systems require a high-resolution camera, but the infrared system also requires an infrared filter and light source. For this reason, both methods are available in the Red-Handed library. Any custom application built using this library can choose, at compile-time, the hand-detection method that best suits the custom environment.

6.2. Limitations of Key Point Gesture

The simplified gesture model used by the Red-Handed system is sufficient to translate simple postures into Windows events. Using key points to determine posture simplifies the gesture detection process. To improve the user experience a separate Kohonen-network has been implemented that enables this in a limited fashion, allowing the user to train the system a priori (that is, the system is not adapting on the fly as yet to different hand gestures). This implementation is not yet as robust as we would like, and is an ongoing development issue. A detection scheme based on shape was also added to the system. This scheme compares a stabilized representation of the current hand pixel set with a stored reference image. This method gives the user the freedom to choose the hand shape for each posture that is used for gesture interaction.

7. Collaborative Interaction

By design, the Windows operating system is not meant to handle simultaneous input from multiple mice. The behaviour of the mouse interface relies heavily on the order of events in the queue for a given application. Multiple mice adding messages to the event queue causes undefined behaviour.

To solve this problem, the Red-Handed system does not add mouse events to the queue directly, but stores the detected gesture information for higher-level applications to use. When a custom application is built to incorporate the Red-Handed gesture recognition, it implements the rules and outcomes of the detected behaviour.

The reason for this design is that the concepts of virtual collaboration are context-dependent, and are outside the scope of this system. To make a software-based interaction interface usable the social aspects of a shared space need to be examined. The rules of a software environment need to be well defined and should be similar to those experienced in the physical world if the interface is to be simple to use. It is in this domain that the Table Manners architecture is placed.

8. Results

The robustness of the system is heavily dependent on the lighting conditions of the environment. Poor lighting causes failure with respect to colour-based hand detection and target detection. The infrared-based method for hand detection is meant to improve the robustness under poor lighting conditions, but it too can fail in the presence of sunlight. Under normal conditions the gesture interface is very robust.

This system was designed for real-time interaction. It meets that goal, with only a slight delay noticed between a user's action and the response time of the system. Also, the system is accurate enough for fine-detail window manipulate (to within 1-2 pixels).

9. Conclusion

Overall, the system provides a robust gesture interface for mouse behaviour in real-time. With this system, natural collaboration at the projection table is made possible. With such a foundation for collaborative interaction, coupled with a developing understanding and facilitation of collaboration in such environments, we have the potential to achieve understanding of and support for heterogeneous users in collaborative, multi-user systems.

10. References

[1] K. Oka, Y. Sato, H. Koike, “Real-time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems”, In Proceedings of Automatic Face and Gesture Recognition 2002.

[2] T. Starner, B. Leibe, D. Minnen, T. Westyn, A. Hurst, J. Weeks, “The Perceptive Workbench: Computer-Vision based Gesture Tracking, Object Tracking, and 3D Reconstruction for Augmented Desks”, Machine Vision and Applications, Vol. 14(1), Springer, 2003, pp. 59-71.

[3] R.A. Metoyer, L. Xu, and M. Srinivasan, “A Tangible Interface for High-Level Direction of Multiple Animated Characters”, In Proceedings of Graphics Interface 2003, Halifax, Canada.

[4] J. Corso, D. Burschka, and G.D. Hager, “The 4D Touchpad: Unencumbered HCI With VICs”, In Proceedings of CVPRHCI, 2003.

[5] Corrie, B., Wong, H., Zimmerman, T., Marsh, S., Patrick, A.S., Singer, J., Emond, B., & Noel, S.. “Towards quality of experience in advanced collaborative environments”, in Third Annual Workshop on Advanced Collaborative Environments, June 22, 2003, Seattle, USA.

[6] M-S. Ekelin, D. Samuelsson and H. Verhagen, “Achieving a Rewarding Meeting with an Artificial Chairman “, in proceedings COLA 2003: Workshop on Collaboration Agents, at , Halifax, Canada.

[7] K. Isbister, H. Nakanishi, T. Ishida and C. Nass, “Helper agent: designing an assistant for human-human interaction in a virtual meeting space”, in Proceedings of the SIGCHI conference on Human factors in computing systems, CHI 2000, The Hague, the Netherlands. Pp. 57-64.

[8] S. Marsh, A. Ghorbani and V. Bhavsar, “The ACORN Multi Agent System”, Web Intelligence and Agent Systems: An International Journal, 1(1):65-86, 2003. IOS Press.

[9] C.B. Owen, F. Xiao, and P. Middlin, “What Is The Best Fudicial?”, First IEEE International Augmented Reality Toolkit Workshop, Darmstadt, Germany, Sept. 2002, pp 98-105.

[10] V. Vezhnevets, V. Sazonov, A. Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques". Graphicon-2003, Moscow, Russia, September 2003.

[11] J.-C. Terrillon, M.N. Shirazi, H. Fukamachi, S. Akamatsu, “Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images”. In Proceedings of the International Conference of Face and Gesture Recognition 2000, 54-61.

[12] J.Y. Lee, S.I. Yoo, “An elliptical boundary model for skin color detection”. In Proceedings of the 2002 International Conference on Imaging Science, Systems, and Technology.

[13] H. Wu, Q. Chen, M. Yachida, “Face Detection from Color Images Using a Fuzzy Pattern Matching Method”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 6, June 1999.

[14] C. McDonald, G. Roth. "Replacing a Mouse with Hand Gesture in a Plane-Based Augmented Reality System”, In *Proceedings of Vision Interface (VI) 2003*, Halifax, Canada.