Virtual Mouse Vision Based Interface

Paul Robertson
DOLL Inc.
9 Bartlet St. #334
Andover, MA 01810
probertson@doll.com

Robert Laddaga
MIT CSAIL
200 Technology Sq.
Cambridge, MA, 02139
rladdaga@ai.mit.edu

Max Van Kleek
MIT CSAIL
200 Technology Sq.
Cambridge, MA, 02139
emax@ai.mit.edu

ABSTRACT

A vision-based virtual mouse interface is described that utilizes a robotic head, visual tracking of the users head and hand positions and recognition of user hand signs to control an intelligent kiosk. The user interface supports, among other things, smooth control of the mouse pointer and buttons using hand signs and movements. The algorithms and architecture of real-time vision and robot controller are described.

Categories and Subject Descriptors

C.m [**Computer Systems Organization**]: Miscellaneous

General Terms

Human Factors

Keywords

Tangible Interfaces, Multimedia Interfaces, Vision, Intelligent UI

1. INTRODUCTION

In this paper we describe a virtual mouse input device, based on a vision system recognizing and tracking hand gestures. The system has been implemented to support a kiosk open space intelligent environment. Recent workplace studies have discovered that employees are spending increasing amounts of time holding unplanned meetings in public spaces, such as along corridors, or in lounges or kitchenettes[8]. In addition to serving as the crossroads for day-to-day activities, these spaces harbor a relaxed social atmosphere, where people feel naturally inclined to gather and talk casually about anything that may be on their minds. As a result, these spaces encourage social connections to be made, shared interests to be discovered, and, perhaps most importantly, collaborations to form among people who may otherwise never have realized the opportunity to work together. Despite the importance of such social encounters and informal collaborations in knowledge-driven organizations [4], these spaces still largely lack any information infrastructure. This inspired the Ki/o project [9] to design such an information infrastructure, which consists of an *intelligent kiosk* platform and software architecture to be integrated into these spaces.

We begin by explaining the context for our research, in Section 2. Section 3 describes the Kiosk application. Next we present a description of the virtual mouse, and its motivation. Then we describe the design, implementation, and operation of the virtual mouse in Section 4.

2. INTELLIGENT ENVIRONMENTS

The field of Intelligent Environments is concerned with studying how to use technological aids to improve the experience of humans in working, living, moving and other structured spaces. Improvement can be in productivity, comfort, or social interaction. It is a sub-genre of the field of ubiquitous computing.

Ubiquitous computing[1] is devoted to changing the relationship between humans and the computers with which we interact, towards allowing computers to become invisible and recede into the periphery of people's lives. In part, this task is proceeding quite naturally with respect to computers with which we don't normally directly interact: those computers in automobiles, washing machines, and watches, for example.

Ubiquitous computing is concerned with bringing the same degree of naturalness of interaction to the personal and business computers that are currently proliferating our work and play environments. As the world has become increasingly reliant on personal computers and the Internet, computers have begun to complicate and dominate, rather than simplify everyday tasks. Moreover, computers have come to occupy increasingly more physical space on desks, and in modern living environments, while, at the same time, they consume increasingly more amounts of time, require more attention, and demand more mental faculties to run simple tasks.

Computer systems today demand that the user be responsible for translating what users want to accomplish into a representation the systems can understand. Much of the exertion required to operate computers originates in having to

IUI'04, January 13–16, 2004, Madeira, Funchal, Portugal.
Copyright 2004 ACM 1-58113-815-6/04/0001 ...$5.00.

[1] The origin of the term and the concept is generally attributed to the late Mark Weiser from the Xerox Palo Alto Research Center in the early 1990's. The term "ubiquitous" is used interchangeably with the term "pervasive" in the research community. This paper adopts the former term.

continuously learn how to properly perform this translation, using whatever clues are provided by the system designers. Ubiquitous computing reverses this process, by making computers responsible for the translation from the physical world into the system's representation. Thus, ubiquitous computing systems act like intelligent personal assistants, capable of understanding what people are trying to accomplish in order to determine how best to intervene and assist them. Therefore, ubiquitous computing systems allow people to concentrate on what is truly important, i.e., their actual tasks, rather than focusing on the onerous steps of operating the computer systems to perform these tasks.

Ubiquitous computing also eliminates the artificial notion of a personal computer as an independent, isolated computational entity. It instead proposes that computation should be available everywhere as a shared natural resource, just like the air we breathe. This notion has been pursued aggressively by the massive and influential MIT Project Oxygen [5]. Our own virtual mouse project is part of MIT Project Oxygen.

Since ubiquitous computing systems need the ability to deduce users' intentions, preferences, and the state of the world, all automatically, they need to perceive the physical world, interpret these observations, make inferences, and then take appropriate action. When these systems, capable of perception, cognition, and action, are embodied in a physical space, they are collectively known as Intelligent Environments.

Within Project Oxygen is a wide array of subprojects that span various disciplines in computer science. The AIRE research group focuses on technologies related to building Intelligent Environments. Using its distributed agent architecture, Meta glue [3], AIRE has built software architectures for intelligent environments in the forms of offices and conference rooms (e21), handheld computers (h21), and now, kiosks (Ki/o).

3. THE KIOSK

The primary prototype test bed for the e21 architecture is embodied in a conference room known as The Intelligent Room [2]. A large number of projects have come out of work in the Intelligent Room, including Metaglue itself, the Metaglue resource manager, RASCAL [6], and a framework for specifying layered, reactive behaviors, ReBA [7]. To date, prototype IEs have been developed primarily for such spaces as conference rooms, classrooms, and offices. The Ki/o Kiosk Platform extends this notion by designing IEs specifically for informal public spaces, such as hallways, lounges, break rooms, and elevator lobbies. The ultimate vision of ubiquitous computing is that Intelligent Environments will pervade all physical spaces, thereby enabling access to digital information anywhere and at any time.

Three Kiosk prototypes were designed, for four separate environments. These were the lobby prototype [9], the hallway prototype, and the lounge prototype. They each have different requirements in terms of number of simultaneous viewers, collaboration requirements, proximity of interactors, etc. Consequently, the hardware to implement them is somewhat different in each case. The one we address in this paper is the lounge prototype.

This Ki/o installation will differ from the lobby prototype in several significant ways. First, users will most likely be interacting with it in a group, sitting around the glass table. Therefore, interaction sessions with this kiosk are likely to span a longer time, and potentially involve more people simultaneously than the lobby prototype. At the same time, since this lounge is not within a primary circulation route of the building, it will likely receive less visitor traffic than the lobby. Finally, since the users will be sitting, interaction must be done at a distance instead of directly through a touch screen tactile input.

Testing the first k:i/o prototype (the lobby version) which consisted of twin wall-mounted touch-screen 17" LCD displays quickly led us to the conclusion that a larger display area was desirable for the lounge prototype. For text to be readable from more than a few meters, a single textual item had to be maximized and made to occupy the entire 17" display surface. As a result, when users were not immediately in front of the display, any item with more text than an average sentence had to be broken up into multiple items, and each textual item had to be displayed serially in succession. This makes use of the screen resemble a ticker-window, and severely limits the amount of information a user can see at one time.

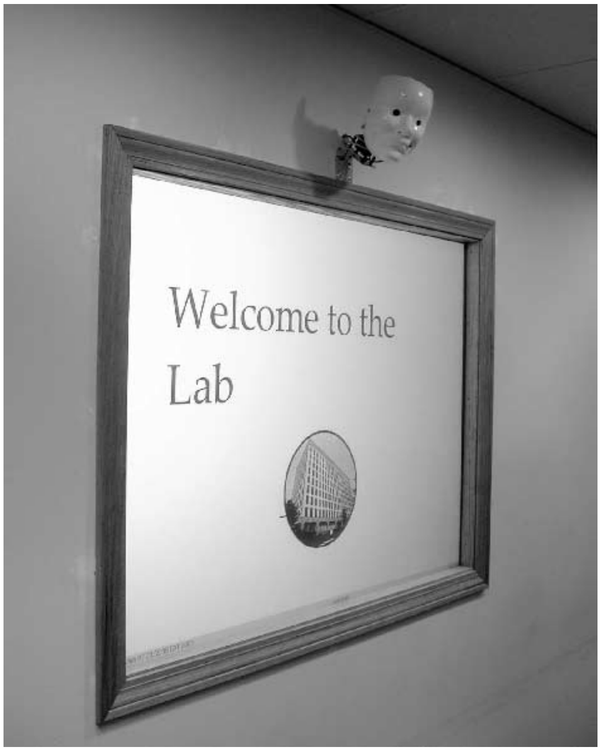


Figure 1: The Kiosk

To address these new demands, a projector and screen were chosen as the primary display medium and surface for the lounge. As visible in the figure, the large projection area should allow all users to be able to see and read text on the display effectively.

With a larger display, multiple items could be presented tiled in a billboard or collage fashion, while still maintaining readability at a distance. Therefore, for the kiosk prototype, ("K9"), a large, 4-by-3-foot rear-projection display was selected, and embedded in a wall within our laboratory. The display screen chosen was a plate of Plexiglas with a Polacoat diffusion coating manufactured by DaLite Inc. This surface was lit using a Sanyo PLCXU-38, 2000-lumens projector equipped with short-throw lens mounted to the ceiling in a room behind the wall. This special wide-angle lens allowed the projector to cast an image that is the full size of the display within 5 feet of the surface itself. The rear-projection kiosk is shown in Figure 1.

4. THE VIRTUAL MOUSE

The greatest challenge with this new display configuration was the means of user interaction. Touch-screens of the size are difficult to manufacture and obtain, and users could potentially have a difficult time reaching parts of such a display if it were made touch-screen. One common solution to this sort of problem is to use a laser pointer. This solution has a number of drawbacks, the most obvious of which is that it requires an additional piece of hardware that is unwieldy if attached (by say a cord) to the kiosk, and far to portable if it isn't attached. A subtler problem is that tracking a laser dot on a screen is itself a non-trivial task, and the red dot is a visible feedback signal to the user of what is being pointed at. If for example, an arrow cursor is used to indicate the pointer position, any discrepancy between the arrow and red dot will be disconcerting to the user.

Our solution was to develop a virtual mouse that enables users to control the kiosk with hand signs and movements. The kiosk has a standard visual user interface, with arrow cursor to indicate pointer movement. The user walks up to the kiosk. People approaching the kiosk are tracked by a robotic head called IGOR (Intelligent Gaze Oriented Robot) described below. When the user makes a recognized hand sign the kiosk allows movement of the hand to move the mouse pointer on the kiosk display. Separate hand signs allow for clicking of the mouse buttons for making selections on the kiosk display.

Note that the arrow pointer is the only feedback the user gets as to where the user is pointing. The user can use that feedback, adjusting to imperfections in tracking, without the distraction of a distinct and different other signal.

4.1 The Hand Signs and Operation of the Mouse

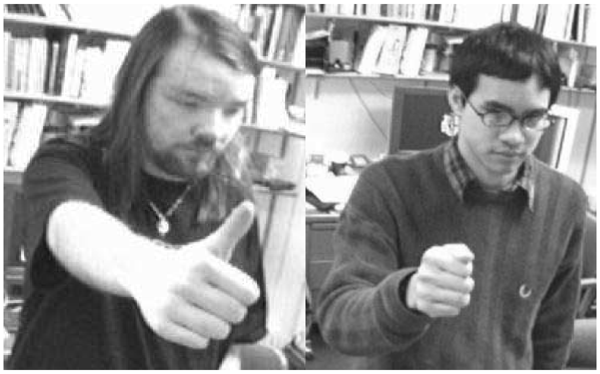


Figure 2: The Hand Signs

During training we trained the system for eight different hand signs. The current prototype uses two of those signs: "thumbs up" grabs the mouse and "fist" clicks the left mouse button. These signs are shown in Figure 2. Other gestures such as a palm may be used later to control scrolling of the kiosk display in future versions.

4.2 IGOR

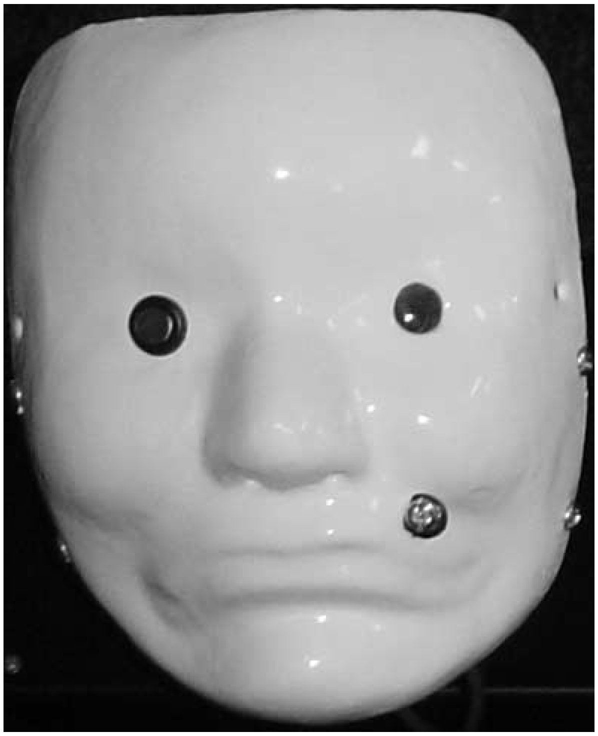


Figure 3: IGOR

The robotic head (Figure 3) that sits above the kiosk display is a robotic head with two degrees of freedom. IGOR has a microphone and speaker to communicating with the user although at present these are only used for debugging the virtual mouse. IGOR has a single camera that is used to track the hand signs and movements.

Motion of the robotic head allows the hand gestures to be centered within IGOR's field of view. Without an articulated head a wide angle camera would be necessary and detection of the hand signs would be difficult due to the poor resolution.

5. VIRTUAL MOUSE IMPLEMENTATION

IGOR is controlled by a state machine. The state machine supports light weight computations called "Actions" that occur:

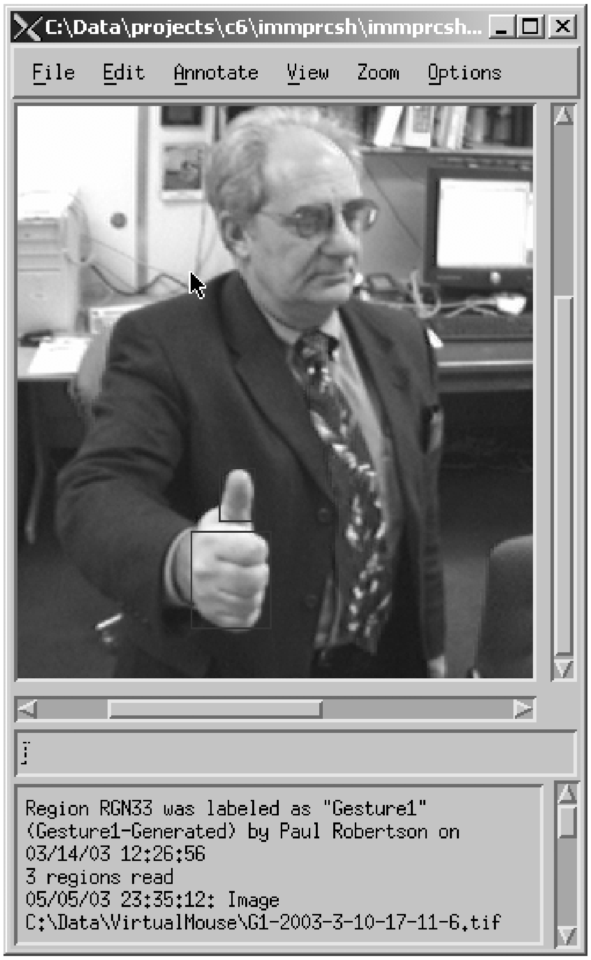
1. Upon entering a state.
2. When an event occurs.

While in a state heavy-weight computations called "components" are scheduled to be run when frames arrive from the camera. "components" raise events that are responded to by (1) actions or (2) a state change. The state diagram for the Virtual Mouse is described in Section 5.2.

Schedules are precompiled for each state so that at run-time switching state causes the current schedule to point at the new schedule and frame processing instantly switches to the new schedule. An outline of the scheduler is provided in Section 5.4 Are precompiled so that switching schedule only involves setting a pointer.

5.1 Detecting the signs

The hand signs are recognized by a modified boosting algorithm similar to that used by Viola and Jones [10] in their face recognition system. A face recognition system based on the same technology is used to recognize users as they approach the kiosk. In a future version identification of known users may be added so that the kiosk can customize the display to a particular users interests.



```
Region RGN33 was labeled as "Gesture1"
(Gesture1-Generated) by Paul Robertson on
03/14/03 12:26:56
3 regions read
05/05/03 23:35:12: Image
C:\Data\VirtualMouse\G1-2003-3-10-17-11-6.tif
```

Figure 4: Training

The system was trained by collecting thousands of examples of the hand signs from a dozen people in the lab. Each gesture was hand annotated to highlight the pieces of interest.

The annotations are stored as XML files such as the one below:

```
<GTAnnotations
  imageName="G1-2003-3-10-17-11-6.tif"
  author="emax"
  creationDate="03/11/01 23:23:46"
  modificationDate="03/14/03 12:26:56">
<GTRegion
  author="emax"
  regionType="Thumb1"
  regionUID="RGN9"
  regiondate="03/11/01 23:23:50"
  coordinates="209, 360, 229, 360, 229, 390,
               209, 390, 209, 360">
</GTRegion>
<GTRegion
  author="emax"
  regionType="Hand1"
  regionUID="RGN10"
  regiondate="03/11/01 23:23:56"
  coordinates="193, 396, 238, 396, 238, 451,
               193, 451, 193, 396">
</GTRegion>
</GTAnnotations>
```

The XML annotation files are read by the recognizer training system.

Figure 4 shows a thumbs up gesture in which the hand and the thumb are annotated. The annotated portions of the sign are used to arrange all examples of sign into a canonical size and position.

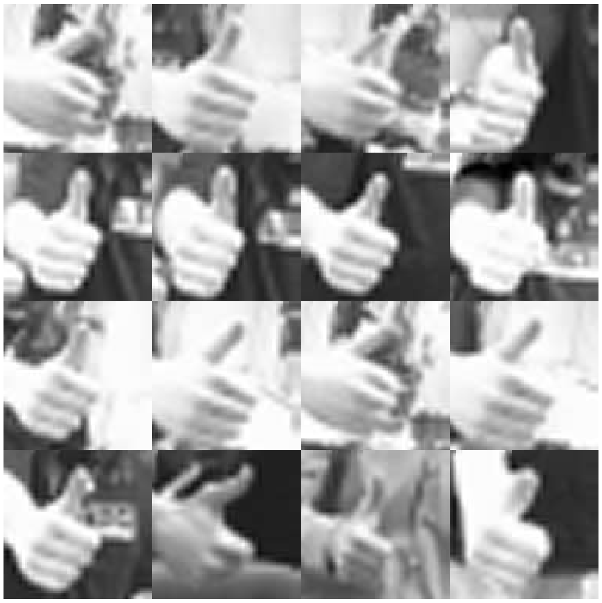


Figure 5: Sign Corpus

Figure 5 shows a subset of the corpus of thumbs up gestures used for training after the signs have been scaled and trimmed to the canonical shape.

The modified boosting algorithm produces a decision tree that can detect all of the trained gestures. The sign finder is applied to every position in the image frame at a variety of scales in order to find the presence of hand signs in an image.

5.2 State Machine for the Virtual Mouse

Figure 6 shows the state diagram for the Virtual Mouse.

The state machine starts in the “track faces” state. In the face tracking state IGOR finds faces in its field of view, selects one of the available faces as the “USER” and keeps that face centered in the image.

The largest face in the image is assumed to be the user on the grounds that it is the face that is closest to the kiosk. Only faces that are looking at the kiosk will be recognized

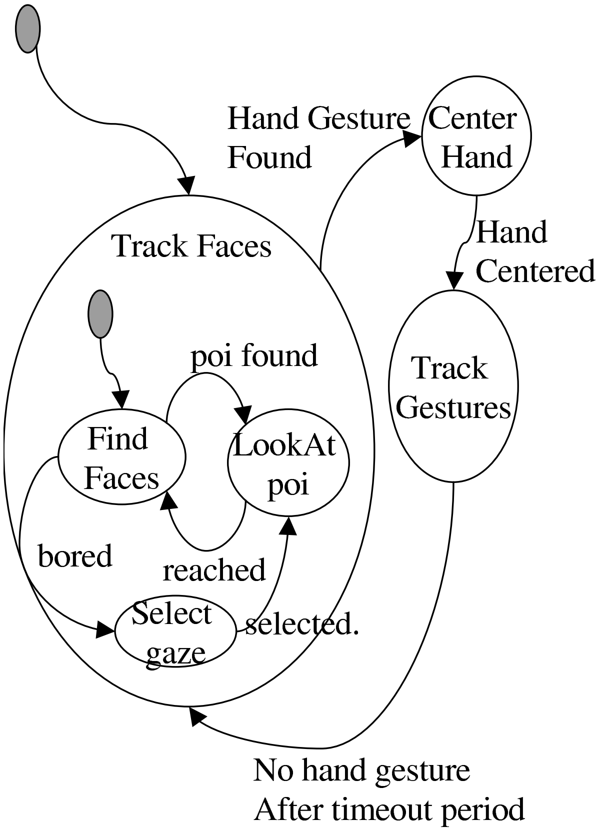


Figure 6: State Diagram

as faces so in practice this heuristic works well. By keeping the closest face centered in the image IGOR is ready to notice hand signs when the user takes control of the kiosk by making one of the recognized signs.

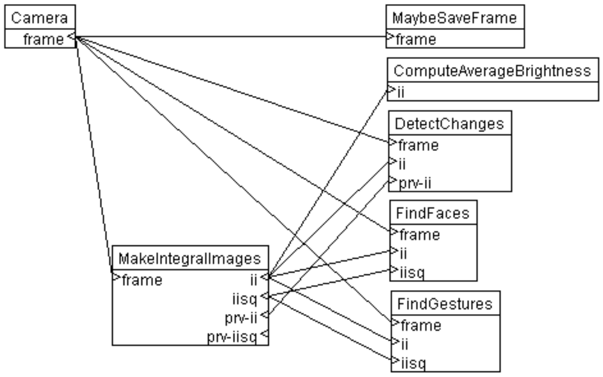


Figure 7: User Tracking Data Flow

Figure 7 is a dataflow diagram showing the processing that constitutes the heavy-weight processing that is performed in the "track faces" state. Briefly these are:

1. MakeIntegralImages: A preprocessing step in which the image is converted into a form that permits multi-scale operations to be computed efficiently ([10]).

2. ComputeAverageBrightness: Computes the average brightness of the image. If there is insufficient light to reliably process the image an event is generated to avoid false tracking and recognition events in darkness.

3. DetectChanges: Finds changes between frames so that a moving person can be identified before a face is recognizable.

4. FindFaces: Runs the face finding algorithm that can find upright full frontal face views in the image at a range of scales. this process produces a list of all faces in the image (referred to as points of interest) along with their size and position in the image.

5. FindGestures: Similar to FindFaces, FindGestures finds all instances of hand signs in the image at a range of scales and returns a list of points of interest along with their positions in the image.

In the face finding state the closest head is kept in the center of the image by moving IGOR's head.

Once a hand sign has been identified in the image the Virtual Mouse first of all switches into the "center hand" state in which the hand sign is centered in the image by moving the IGOR head and then switches into the "Gesture Tracking" state. No image processing is performed during the "center hand" state because IGOR's head is in motion and IGOR is blind during head motion. Once in the "Gesture Tracking" state IGOR begins tracking the hand motion within the image. during motion tracking IGOR makes no head movements because moving the IGOR head results in brief blindness which would interfere with smooth tracking of the motion. By centering the hand in the frame before entering the "Gesture Tracking" state IGOR ensures that the hand has the maximum amount of travel possible within the frame.

The Gesture Tracking state supports the following events:

1. Entry Event: Grab the mouse pointer, record the mouse pointer position, select the gesture box (explained below).

2. Click Gesture: When the "fist" gesture is recognized the kiosk mouse left button is clicked.

3. Motion in gesture box: Move the mouse.

4. Gesture found: Reset the recorded position of the mouse pointer, reselect the gesture box.

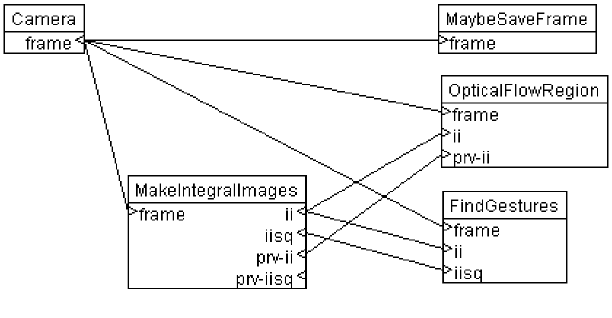


Figure 8: Hand Tracking

Figure 8 shows the heavy-weight processing that constitutes the image processing performed in the "Gesture Tracking" state.

In this state faces are not tracked but gestures continue to be tracked. In addition motion is tracked by an optical flow computation that is described below.

5.3 Gesture Tracking

Our initial attempt at providing mouse movement was to track the recognition of the hand signs and each time a new location for the hand sign was detected update the position of the mouse on the kiosk display. Unfortunately this resulted in unacceptably jerky movement of the mouse. This was caused by two factors:

1. During motion the sign was not always detected because of blurring in the image. This resulted in the sign being occasionally recognized and the mouse therefore being moved in jumps rather than smoothly.

2. The recognizer finds the bounding box for the gesture and takes the center point of the bounding box as the position of the sign. As the hand moves the shape of the bounding box changes and this gives rise to erratic position estimates that are unpleasing to the user.

We overcame the above problems by separating the motion of the kiosk mouse pointer from the recognition of the sign.

Current algorithm achieves smooth mouse movement by using two algorithms:

1. Optical Flow in a Region: The optical flow in the region that contains the hand sign is computed on each frame. This provides a very smooth estimate of movement of the sign.

2. Kalman Filter: The motion estimated by the optical flow algorithm is fed into a Kalman Filter [1] that additionally smoothes the trajectory of the mouse pointer.

The combination of the above two algorithms provides a very smooth and usable mouse movement on the kiosk.

Upon entry to the "Gesture Tracking" state a region is computed that is slightly larger than the bounding box of the hand sign (see Figure 9 which shows the bounding box and an outer box used for calculating the optical flow.

The optical flow is computed for the entire optical flow region and the average flow within the region is passed into the Kalman filter. whenever a hand sign is recognized the position if the gesture box is updated but recognition of a hind sign never moves the mouse pointer–only optical flow results in mouse movement.

5.4 Real-Time Considerations and the Vision Scheduler

The scheduler component is an essential part of the real-time management of the image processing routines in the application. Six priority levels are defined. Priority 1 processes are run on every frame. Priority 2 processes are run every other frame and so on.

```
Priority 1:  Every frame
  (0 1 2 3 4 5 6 7 8 9 10 11)
Priority 2:  Every other frame
  (0 2 4 6 8 10) (1 3 5 7 9 11)
Priority 3: Every third frame
  (0 3 6 9) (1 4 7 10) (2 5 8 11)
Priority 4: Every fourth frame
  (0 4 8) (1 5 9)
  (2 6 10) (3 7 11)
Priority 5: Every sixth frame
  (0 6) (1 7) (2 8)
  (3 9) (4 10) (5 11)
Priority 6: Every twelfth frame
  (0) (1) (2) (3) (4) (5) (6)
  (7) (8) (9) (10) (11)
```

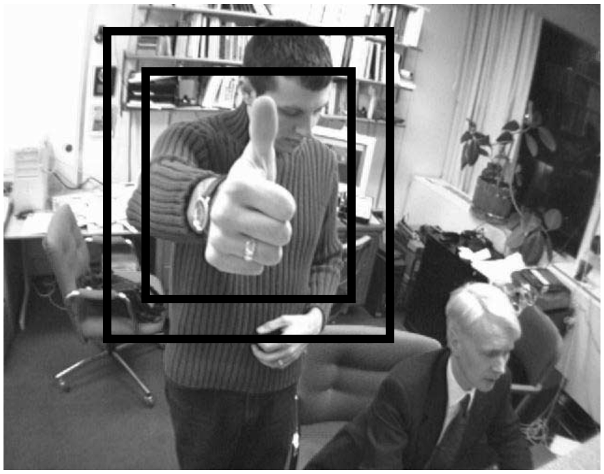


Figure 9: Tracking

When a frame comes in, the scheduler selects on a round robin bases which processes to run as shown in Figure 10. This allows us to engineer the priorities of the image processing modules so that all computation can be achieved in real-time.

In the face tracking state the face recognition is a higher priority than the sign recognition. In the sign tracking state the optical flow is given the higher priority.

In this way good responsiveness is achieved in both user tracking and gesture tracking.

6. CONCLUSIONS AND FURTHER WORK

The K9 Lounge Kiosk is still undergoing testing and user acceptance trials. However, we can already see that avoiding duplicate and disparate position feedback is helpful to users. Users so far have shown significant ability to quickly compensate for imperfections in tracking.

The visual tracking application has been an interesting challenge. Some things we learned are that:

1. State diagrams (represented as XML files) allow an application to be defined in terms of allocations of the real-time vision budget. The scheduler allows lower priority "events" to be detected while spending the bulk of the time budget on the primary task.

2. Optical flow allows smooth tracking of the hand gestures. It is robust because no recognition is required to achieve mouse motion, and it also provides smooth motion estimates.

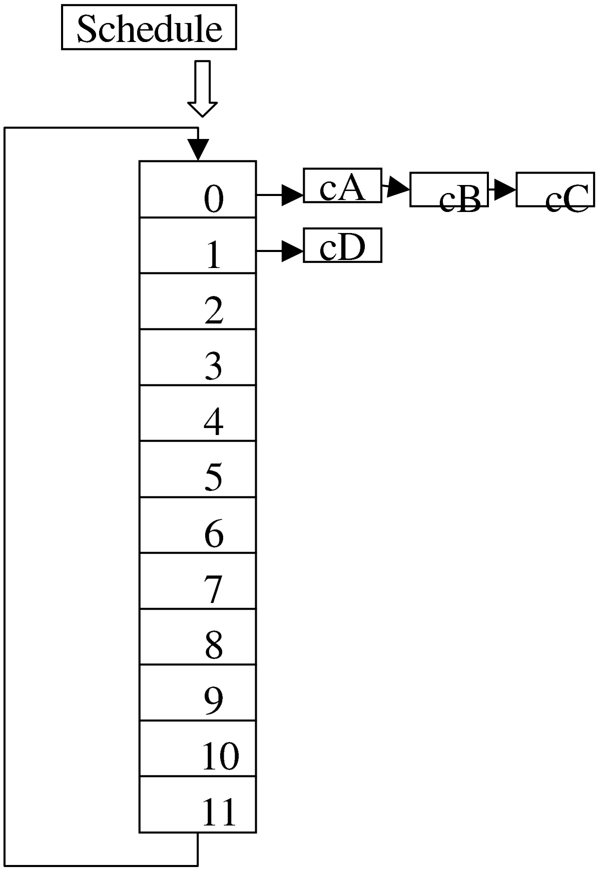


Figure 10: Scheduler

3. A useful development tool has been developed to specify software component networks visually, from their representation as XML files.

Future work includes more rigorous testing of the interface, improvement of the vision algorithms, and including a learning component, to dynamically learn to track gesture movements better for each recognized individual.

7. REFERENCES

[1] *Kalman Filtering: Theory and Application*. IEEE Press, 1985.

[2] Rodney Brooks. The intelligent room project. In *Proceedings of the 2nd International Cognitive Technology Conference (CT'97)*, Aizu, Japan, 1997.

[3] Michael Coen, Brenton Phillips, Nimrod Warshawsky, Luke Weisman, Stephen Peters, and Peter Finin. Meeting the computational needs of intelligent environments: The metaglue system. In *Proceedings of MANSE'99*, 1999.

[4] D. Cohen and L. Prusak. *In Good Company: How Social Capital Makes Organizations Work*. Harvard Business School Press, Cambridge, Massachusetts, 2001.

[5] Michael Dertouzos. The future of computing. *Scientific American*, 1999.

[6] Krzysztof Gajos. Rascal - a resource manager for multi agent systems in smart spaces. In *Proceedings of CEEMAS'01*, 2001.

[7] Ajay Kulkarni. A reactive behavioral system for the intelligent room. Technical report, MIT AI Lab, 2002.

[8] Employees on the move. *Steelcase Workplace Index Survey*, April 2002.

[9] Max Van Kleek. Intelligent environments for informal public spaces: the Ki/o Kiosk Platform. M.Eng. Thesis, Massachusetts Institute of Technology, Cambridge, MA, February 2003.

[10] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of Computer Vision and Pattern Recognition 2001*, 2001.