MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Television Control by Hand Gestures

William T. Freeman, Craig D. Weissman

TR94-24 December 1994

Abstract

We study how a viewer can control a television set remotely by hand gestures. We address two fundamental issues of gesture–based human–computer interaction: (1) How can one communicate a rich set of commands without extensive user training and memorization of gestures? (2) How can the computer recognize the commands in a complicated visual environment? We made a prototype of this system using a computer workstation and a television. The graphical overlays appear on the computer screen, although they could be mixed with the video to appear on the television. The computer controls the television set through serial port commands to an electronically controlled remote control. We describe knowledge we gained from building the prototype.

IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition, Zurich, June, 1995

Television control by hand gestures

William T. Freeman and Craig D. Weissman
Mitsubishi Electric Research Labs
201 Broadway
Cambridge, MA 02139 USA
e-mail: freeman@merl.com

From: IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition, Zurich, June, 1995.

Abstract

We study how a viewer can control a television set remotely by hand gestures. We address two fundamental issues of gesture–based human–computer interaction: (1) How can one communicate a rich set of commands without extensive user training and memorization of gestures? (2) How can the computer recognize the commands in a complicated visual environment?

Our solution to these problems exploits the visual feedback of the television display. The user uses only one gesture: the open hand, facing the camera. He controls the television by moving his hand. On the display, a hand icon appears which follows the user's hand. The user can then move his own hand to adjust various graphical controls with the hand icon.

The open hand presents a characteristic image which the computer can detect and track. We perform a normalized correlation of a template hand to the image to analyze the user's hand. A local orientation representation is used to achieve some robustness to lighting variations.

We made a prototype of this system using a computer workstation and a television. The graphical overlays appear on the computer screen, although they could be mixed with the video to appear on the television. The computer controls the television set through serial port commands to an electronically controlled remote control. We describe knowledge we gained from building the prototype.

1 Introduction

Our goal is to build computers and machines which are easy to use. Machines may be easier to use if we could operate them through natural language or gestural interactions.

Focussing on a concrete instance of the general problem, we study how to operate a television set remotely. This is a familiar, yet useful problem. People value the ability to control a television set from a distance. In a survey, Americans were asked what "high technology" gadget had improved their quality of life the most. The top responses were "microwave ovens" and "television remote controls" [1].

Contemporary hand-held television remote controls are very successful, yet not without flaws. They can be lost. There is a small industry for products related to losing remote controls—replacements remotes, devices which indicate where the remote control is, and televisions which locate the remote control. Even if the remote control is not lost, it can be an inconvenience to have to get it from another part of the room. It is reasonable to study additional ways to control the television remotely.

Voice and vision are two natural candidates. Voice has the advantage of a pre-established vocabulary (natural language). However, it may not be appropriate for the protracted issuing of commands, such as while "channel surfing", nor for changing parameters by increments, as with volume control. Gestural control may be more appropriate than voice for some tasks, yet lacks a natural vocabulary. For this work, we focussed on vision–based control. Future systems may ultimately use a combination of the two.

There has been much recent interest in the computer vision problem of hand gesture recognition [9, 5, 10]. However, most methods either do not operate in real–time without special hardware, or else are not appropriate for an unpredictable scene. We will tailor our recognition method to the user interface we will design.

We seek a user interface which new users can instantly master. Here we confront a fundamental problem in the control of machines by hand gestures: the lack of a vocabulary. We have many possible commands to give the television, such as "mute", "louder", "channel 37", yet no universal set of hand signals with which to specify them. We do not want to require the user to memorize complicated gestures (Fig. 1).

Figure 1: The fundamental problem of machine control by hand gestures. We may have many complicated commands we wish to issue, yet the manual vocabulary must be simple to form and easy to remember. Complicated hand signals are not appropriate.

There is a related problem from the computer's image processing perspective: How can a computer identify and classify the hand gestures quickly and reliably in a complex and unpredictable visual scene? Figure 2 shows a view from a camera near a living room television set. We have to find and decode a

hand signal which may be a small part of a large and cluttered image.

Figure 2: A typical visual scene which a camera looking out from a television set might encounter. It is complicated, unpredictable, and the hand is not a dominant part of the image.

2 Our Approach

Our solution to the above two problems exploits the capability of the television display for visual feedback. Based on graphics displayed on the television screen, the viewer sees how to move his hand to issue commands, Fig. 3.

Figure 4 shows a typical interaction session. When the user wants to control the television, he holds up his hand so it is facing the television We call this the *trigger gesture*.

When the television is off or playing a program, it continually looks for the trigger gesture. When the trigger gesture is detected, the television enters *control mode*. If the television display was off, it turns on. Graphics overlays appear over the bottom portion of the program displayed on the television screen. A hand icon appears on the screen, which follows the movements of the viewer's hand.

Figure 5 shows the graphical controls for the present implementation. There are sliders to adjust the television channel and volume, and buttons for mute and power off. The channel and volume may also be controlled incrementally through buttons for up and down increments.

The hand control is like using a buttonless mouse. The "hot–spot" of the hand icon is shown to the viewer by the solid circle below the hand. When the hot spot covers a control for 200 msec., the control changes color, and the command is executed. The slider responds by moving the slider bead. While a "mouse button press" could be signalled by some additional hand signal, simple control with the buttonless mouse is satisfactory for this interface.

After the viewer has adjusted the controls, he can leave control mode by closing his hand, Fig. 4. (Actually, any disappearance of the open hand image is sufficient). A closed–hand icon replaces the television's hand icon briefly, to echo the user's command, and then the television enters viewing mode. The graphical overlay disappears from the television program being displayed. If the user has pressed the "off" button, the television turns off, and delays for 2 seconds before resuming its search for the "on" trigger gesture.

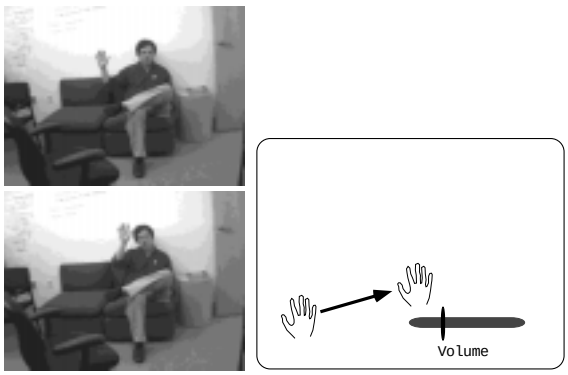


Figure 3: Our solution to the design constraints imposed by the user and the environment: exploit the visual feedback from the television. The user must only memorize one gesture: holding an open hand to the television. This hand position is relatively easy for the computer to find, even in a cluttered environment. The computer tracks the hand, echoing its position with a hand icon displayed on the television. The user moves his hand to operate the on–screen controls, much as one uses a mouse with a computer.

3 Image Processing

The open hand used for the trigger gesture and hand tracking is relatively straightforward to detect and track, even in a complicated scene. We use a normalized correlation method [2, 4].

Figure 6 shows the idea behind normalized correlation. One uses a template (a) of the image feature to be found. The normalized correlation between two vectors, $\vec{a}$ and $\vec{b}$, is the cosine of the angle between them, $\cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{|a||b|}$. The image pixels are the entries of the vector $\vec{a}$, and the corresponding pixels at some offset position the image form the second vector, $\vec{b}$. (c) shows the resulting normalized correlation for every offset position in the image (b). Note that the normalized correlation is highest at the position of the user's open hand. Note also that other regions of the image, where the local pattern of light and dark mimics hand digits, can have relatively high correlation values.

We could perform the normalized correlation using a variety of different image representations. We found that an orientation representation gave somewhat better performance than pixel intensities.

One can measure the local orientation in a variety of ways [8, 7, 6]. For reasons of speed, we used two–tap dx and dy filters to determine the orientation of the image gradient. One benefit to an orientation representation is robustness against lighting variations, illustrated in Fig. 7.

We find the hand position to sub–pixel accuracy by modeling the correlation surface as a quadratic polynomial and finding the position of maximum correlation.

We included in our system the ability to search for multiple hand templates. Searching the entire image for many templates would be costly. The system finds the position of best match with the current filter, then searches in the local area with the other

templates to find the position and value of the filter giving the best correlation match. This provides the flexibility of multiple hand templates with a neglible cost overhead.

For efficiency and to avoid potential false trigger gesture detections, we do not processes objects which are stationary for some time. We maintain a running average of the scene, and remove this stationary background from the incoming image, see Fig. 3.

4 The Prototype

To study ease of use and viewer behavior, we built a real–time prototype. The hardware is shown in Fig. 9. A Flex-Cam video camera acquired NTSC format television images. These were digitized at 640 x 480 resolution and downsampled by a factor of 2 by a Raster Ops VideoLive card in an HP 735 workstation. All image processing was performed in the workstation, on software written in C and C++. The prototype is a two–screen system. Graphics are displayed on the HP workstation's monitor, but could be overlaid on the television image with the proper video hardware. A television is controlled by a serial port connection to an All-In-One-12 television remote control. The accompanying software and interface for the remote control is from Home Control Concepts. Unfortunately for this application, commands can only be issued to the remote control at a rate of about 1 per second.

There is a tradeoff between the system response time and field–of–view. To obtain reasonable response, we limited the field of view to 25° during search for the trigger gesture, and 15° during tracking of the hand. This gave about a half second delay before recognition of the trigger gesture, and hand tracking at about 5 times a second. The precise timing depends on the size of the hand filter template. Typically, templates had 100 positions where the local contrast strength was above threshold to be a valid orientation value.

5 Lessons

Controlling a television set remotely through hand gestures seemed to be exciting for the people who tried the prototype. This may or may not be due to the novelty of such control.

The open hand gesture was found to be somewhat tiring for extended viewing. An improvement may be to maintain the open hand as a trigger gesture, but allow a more restful command gesture, once the trigger gesture has been detected and the hand located. The contour tracking algorithms of Blake and Isard [3] may be useful for such commands.

Multiple templates are useful for robust operation of the prototype, although too many templates makes false detection of the trigger more possible. Further development has to be undertaken to determine whether this simple correlation–based image processing could be made robust enough for home use.

References

[1] reported in Telecommunications Policy Review, July 31 1994. Porter/Novelli survey.

[2] D. H. Ballard and C. M. Brown, editors. *Computer Vision*. Prentice Hall, 1982.

[3] A. Blake and M. Isard. 3D position, attitude and shape input using video tracking of hands and lips. In *Proceedings of SIGGRAPH 94*, pages 185–192, 1994. In *Computer Graphics*, Annual Conference Series.

[4] T. J. Darrell and A. P. Pentland. Space-time gestures. In *Proc. IEEE CVPR*, pages 335–340, 1993.

[5] J. Davis and M. Shah. Gesture recognition. Technical Report CS-TR-93-11, University of Central Florida, Orlando, FL 32816, 1993.

[6] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Pat. Anal. Mach. Intell.*, 13(9):891–906, September 1991.

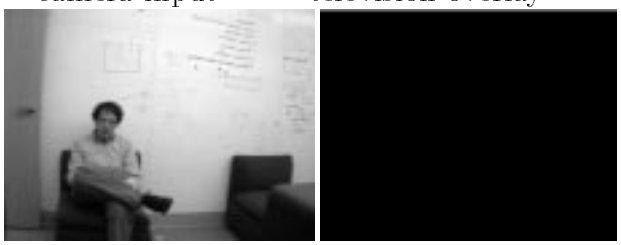
[7] M. Kass and A. P. Witkin. Analyzing oriented patterns. In *Proc. Ninth IJCAI*, pages 944–952, Los Angeles, CA, August 1985.

[8] H. Knutsson and G. H. Granlund. Texture analysis using two-dimensional quadrature filters. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pages 206–213, 1983.

[9] J. M. Rehg and T. Kanade. Digiteyes: vision-based human hand tracking. Technical Report CMU-CS-93-220, Carnegie Mellon School of Computer Science, Pittsburgh, PA 15213, 1993.

[10] J. Segen. Gest: a learning computer vision system that recognizes gestures. In *Machine Learning IV*. Morgan Kauffman, 1992. edited by Michalski et. al.

camera input television overlay



(a) television off

(b) turn on television

(c) channel control

(d) change channel

(e) end adjustment

Figure 4: Sample session of television viewing. (a) Television is off, but searching for the trigger gesture. (b) Viewer shows trigger gesture (open hand). Television set turns on and hand icon and graphics overlays appear. (c) The hand icon tracks the user's hand movement. User changes controls as with a mouse. (d) User has moved hand icon to change channel. (e) User closes hand (or takes it out of view) to leave control mode. Computer echoes this with the hand icon closing. After one second, the hand icon and controls then disappear.

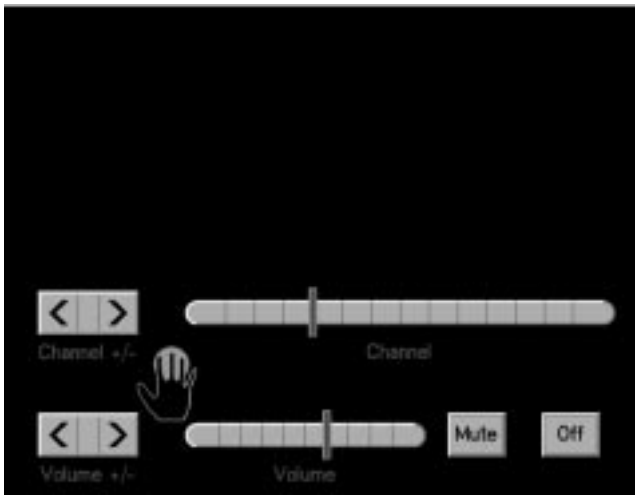


Figure 5: Close-up of graphical controls used for this implementation. Two sliders control channel and volume. These may also be controlled by increment up/down buttons at the left of each slider. A "mute" button is to the right of the volume slider, and to the right of that is the "off" button. The hot spot of the hand icon is shown to the user by the solid area underneath the hand drawing.

(a) template

(b) image

(c) normalized correlation

Figure 6: The hand recognition method used is normalized correlation. The normalized correlation of the hand template, (a) with the incoming image, (b) is shown in (c). Note that the position of maximum correlation is at the user's hand. In our implementation, we used an orientation representation for the template and image, not the pixel intensities shown.

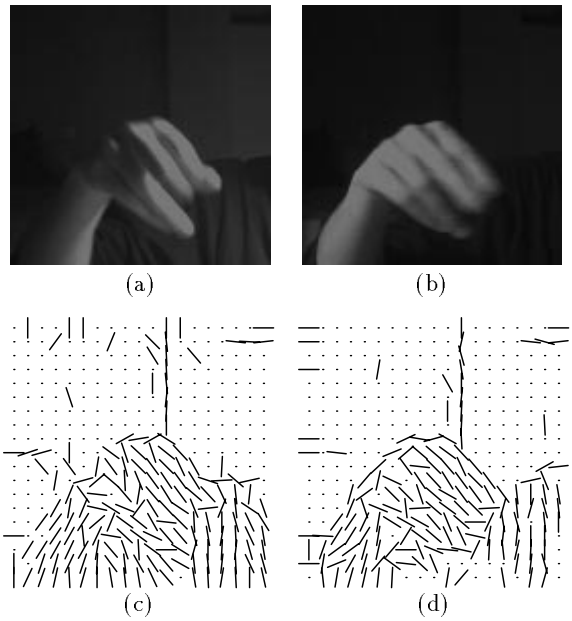


Figure 7: Showing the robustness of local orientation representation to lighting changes. (a) and (b) show the same hand gesture illuminated under two different lighting conditions. The pixel intensities change significantly as the lighting changes. The maps of local orientation, shown in (c) and (d), are more stable. (The orientation maps were computed using steerable filters [6] for this figure. For the real–time implementation, we used two–tap gradient filters). Orientation bars below a contrast threshold are suppressed.)

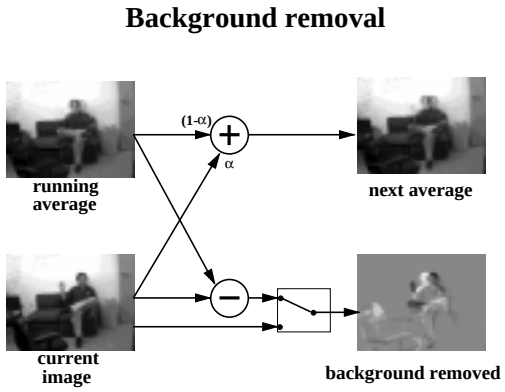


Figure 8: To avoid analyzing furniture and other stationary objects, we peformed a simple background removal. We linearly combined the current image with a running average image to update the running average. We subtracted the two images to detect image positions where the change was above a pre–set threshold. We only processed those positions above the change threshold.

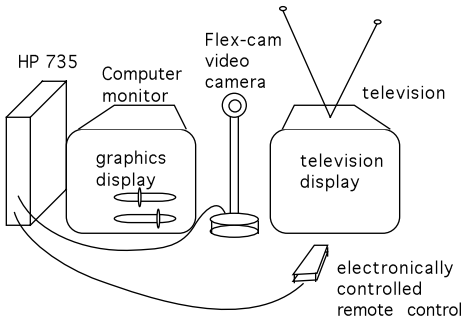


Figure 9: Hardware components for prototype. A Flex-cam video camera produces a video image, which is digitized by a RasterOps video digitizer card in the HP-735 workstation. The computer analyzes the image and displays the appropriate graphics on the computer display screen. The user moves his hand to adjust the on–screen controls. The computer then issues the appropriate commands over a serial port to an electronically controllable remote control. While this prototype uses two display screens, future versions could display the graphics overlay directly on the television screen.