Armchair Interface: Computer Vision for General HCI

Daniel R. Schlegel, Jeffrey A. Delmerico
University at Buffalo
Buffalo, NY, USA
{drschleg,jad12}@buffalo.edu

Abstract

We present a system for general human-computer interaction which utilizes a webcam-enabled computer and allows the user to override the standard pointing device, and instead interact with the computer via hand gestures. We employ the Continuously Adaptive Mean Shift algorithm to perform finger tracking on the colored fingertips of a glove that the user wears. A homography is used to map the tracked finger locations from a camera-relative coordinate system to a screen-relative coordinate system. The tracked index finger controls the location of the cursor on screen, and several gestures allow the user to "left-click" and scroll to interact with the operating system. We use a virtual "plane of interaction" in between the user and the camera to enable the user to gesture in three dimensions; its function is analogous to that of a touch screen. The gestures produce different results depending on whether they are in front of or behind this plane. This interface represents the primary novelty of our system. The longterm goal of this work is to produce a human-computer interaction system which permits the user to replace a standard pointing device with a robust and lightweight tracking and gesture recognition system. Thus the user will be able to physically disconnect from the computer and interact in three-dimensional space.

1. Introduction

Visions of hands-free computer interfaces have permeated science fiction (i.e. Minority Report) for many years, but we are now on the cusp of being able to implement systems which capture at least the same spirit as the fictional ones using off-the-shelf webcam technology and open source libraries. Attempts have been made in the past [1], and projects are currently in development [10], to create systems which replace the mouse with some method of tracking or gesture recognition system, but these systems do not naturally map the users interactions in the world with those on the screen and without modifications to the software the user is accustomed to. The system we present here attempts to create an interface for the user to interact with their computer using finger tracking, natural gestures, and only with off-the-shelf hardware.

Mean shift has been shown to be effective in finding the modes of a multimodal density, in this case the color space of an image [3]. By extending that concept to sequences of images, an iterative mean shift algorithm has been implemented for real-time color-based tracking in video [4]. The same group augmented these methods to improve target localization by masking the target with an isotropic kernel before performing mean shift on the smoothed feature space [5]. An algorithm called Continuously Adaptive Mean Shift (CAMSHIFT) further extends this concept to handle the dynamic color space of a video sequence [2].

Some work has been done in gestures for general computing but there has been no clearly successful method. Work has been done in using eye tracking [7] and even nose-tracking to move a mouse on the screen [11]. Nose tracking tends to use gestures that may be unnatural to click the mouse (such as ”dwell time” [6] which could be a problem if the user is performing some action which doesn't involve head movement). These two interfaces are more useful for persons with disabilities who lack hand control. In addition there has been work in tracking a hands movement on a surface or within a 2D plane which would simulate a mouse [8]. Kleek, Robertson, and Laddaga produced a system in 2004 which uses gestures for general computer use. Their system uses a ”thumbs-up” gesture to enter the state where the mouse is being moved, and a fist gesture is used to click the mouse. It is unclear how the system deals with the transition between the two states though (i.e. does the mouse still move while the hand is moving towards being a fist) [9].

Our system combines the CAMSHIFT algorithm with a small set of natural gestures to provide a computer vision interface for general human-computer interaction. The primary goals of this system are to use entirely off-the-shelf vision components (i.e. webcams) and to not require modification to any software on the client side.

2. Tracking

In order to reliably track the hand we are using OpenCVs native CAMSHIFT method, which is based on [2], and having users wear gloves with colored fingertips. The CAMSHIFT algorithm has proven to be fairly robust, and leaves a lot of computational room for further components of our system. One aspect of this method which required significant investigation is the problem of reinitialization. Moving too quickly, moving out of the frame, occlusion, or a similarly colored background all proved to be challenges for the CAMSHIFT algorithm. In order to cope with these challenges, we first smooth the color densities with a Gaussian kernel, and then we use a Kalman filter on the tracked finger location observations. We use the predicted locations from the Kalman filter to reinitialize the tracker if it loses the maximum of a color density representing a finger. These features have proven to be adequate in providing robust enough tracking to make our system usable, however, we are continuing to experiment in this area in search of improvement in efficiency and robustness.

3. Gestures

The implemented finger tracking system is used to control the mouse pointer on the computer screen. The user then moves their hands around in the air to manipulate the mouse pointer. The tracker submits *observations* to the gesture controller. These observations contain the coordinates of each of the fingers being tracked along with the radius of the observation area for each. Calculations are then performed on the observation to determine relative distance between fingers on the same hand, velocity of movement, and coordinates for the fingers are converted from camera to screen coordinates. Observations are then submitted to all registered gesture recognizers.

Our system is built to be highly extensible - allowing the addition of gestures easily. An eventual goal for the system is to allow applications which may wish to use additional gestures to merely register them with the system, though this is currently not completely implemented.

The gestures we use center around what is the primary novelty of our system, the *plane of interaction*. We use the plane of interaction to extend our gestures into three dimensions. For example, to move the mouse pointer around the screen the user need only move their pointer finger in the air until the mouse pointer is in the desired position. To click the user then moves their finger towards the screen until it passes into the plane of interaction. Another gesture we support is using two fingers in the plane as a method of scrolling up or down. This allows users to use gestures which they are familiar with for touch screens which many users have already become accustomed to due to the increased use of multi-touch smartphones and tablets.

Since one of the emphasis points for this system is that it should be usable with off the shelf parts and built-in laptop webcams, it is important to create a proper mapping between the camera and the screen. Other systems do not form a natural mapping between the gestures and screen, but since we are emulating a touch screen type interface, this is a must. To do this we have developed a quick calibration system which defines the parameters for a homography to virtually place the camera behind the users screen.

4. Conclusion

We have presented a computer vision system allowing the user to interact naturally with their computer, using gestures instead of a traditional pointing device, with only off-the-shelf components. We have conducted initial tests using the system to navigate websites and use basic software packages. While what we have created has proved adequate for simple tasks, there is still much to be done in improving the robustness of our tracker and developing new gestures which provide a more natural user experience.

References

[1] J. Alon. *Spatiotemporal gesture segmentation*. PhD thesis, Boston University, 2006.

[2] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. Technical report, Intel Technology Journal, Q2 1998.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 603–619, 2002.

[4] D. Comaniciu, V. Ramesh, , and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 142–149, 2000.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 564–577, 2003.

[6] J. Gips, M. Betke, and P. Fleming. The camera mouse: Preliminary investigation of automated visual tracking for computer access. In *Proceedings of RESNA 2000*, pages 98–100, 2000.

[7] M. Kumar. *Gaze-Enhanced User Interface Design*. PhD thesis, Stanford University, May 2007.

[8] B. Mitchell, D. Crosta, and Z. Pezzementi. Camera based mouse. http://www.cs.jhu.edu/ ben/vision/lab5/index.html.

[9] P. Robertson, R. Laddaga, and M. V. Kleek. Virtual mouse vision based interface. In *Proceedings of IUI04*, pages 177–183, January 2004.

[10] Toshiba Research Europe Ltd. Cambridge Research Laboratory. Projects: Gesture user interfaces, May 2010.

[11] L. Zhang, F. Zhou, W. Li, and X. Yang. Human-computer interaction system based on nose tracking. *HCI*, pages 769–778, 2007.