

# COMPUTER VISION BASED MOUSE

*Aykut ERDEM<sup>1</sup>, Erkut ERDEM<sup>1</sup>,  
Yasemin Yardimci<sup>1</sup>, Volkan ATALAY<sup>1</sup>, A. Enis ÇETİN<sup>2</sup>*

<sup>1</sup>Middle East Technical University, Ankara, Turkey

<sup>2</sup>Faculty of Engineering, Sabanci University, Istanbul, Turkey

## ABSTRACT

We describe a computer vision based mouse, which can control and command the cursor of a computer or a computerized system using a camera. In order to move the cursor on the computer screen the user simply moves the mouse shaped passive device placed on a surface within the viewing area of the camera. The video generated by the camera is analyzed using computer vision techniques and the computer moves the cursor according to mouse movements. The computer vision based mouse has regions corresponding to buttons for clicking. To click a button the user simply covers one of these regions with his/her finger.

## 1. INTRODUCTION

Recently, there is interest for alternative flexible and versatile ways for humans to communicate with computers. Examples to the alternative communication systems include touch screens, speech recognition systems, key systems, and vision based hand gesture and face expression recognition systems [1]-[7]. In addition, human-like capabilities such as perception would be a good feature of systems targeted for man-machine interaction, a specific gesture or a sign of a hand can be used as a key to a database system. Vision based methods provide both conventional and unconventional means for entering data into computers. Application areas of vision based text and data entry systems include regular computers as well as wearable computing in which flexible and versatile man-machine communication systems other than the ordinary tools of keyboard and mouse may be needed. Computer vision based man-machine communication systems can be developed by taking advantage of the character recognition systems developed in document analysis [8]-[10], and image analysis methods. In [6] we developed a text entry method for recognizing the characters drawn by hand gestures or by a pointer captured by a digital camera which is a new form of data entry into a computer. This system recognizes unistroke isolated characters which are successfully used in personal digital assistants in which

people feel easier to write rather than type on a small size keyboard [11],[12]. In [6] it is assumed that each character is drawn by a single stroke as an isolated character. One of the alphabets that has this property is the Graffiti™.

In this paper, we describe a computer vision based mouse which can control and command the cursor of a computer or a computerized system using a camera. In this system there is no need to have a cable connection between the computer and mouse nor a wireless transmitter-receiver pair as the mouse movements are transferred to the computer by the camera. The concept of vision based finger mouse is first proposed by Quek *et al.* [4] in which the user controls the cursor by moving his fingers in the three-dimensional space. In our mouse system the approach is simpler than [4] in the sense that we place specific reference points on the mouse and the mouse moves on a two-dimensional surface. We track the reference points and the location of the cursor is updated accordingly. This approach is computationally more efficient than finding the finger tip in a cluttered background and tracking it. In addition to the above features our mouse has well defined regions corresponding to buttons to implement clicking. To click a button the user simply covers a button region for some time by his or her finger or a pointer.

The resulting image processing system can be also used in mobile communication and computing devices such as mobile phones, laptop computers, handheld computers, and PDAs. The advantages of our computer vision based mouse system compared to earlier systems are the following:

- The background is controlled by the user. A mousepad or a white sheet of paper which is clearly distinguishable from the background can be used. This simplifies the image analysis process for extracting the mouse boundaries or to track the reference point on the mouse.

- Reference marks can be placed on the mouse. These marks are easier to find and track by the vision system of the computer.
- Well defined regions corresponding to mouse buttons can be placed on the mouse. To click a button the user covers one of these regions momentarily by his or her finger or pointer.
- There is no need to place an electronic circuit inside the mouse which is essentially a passive device. The mouse can be made of any hard material.

In Section 2 the outline of the recognition system is described.

## 2. VIDEO ANALYSIS AND RECOGNITION SYSTEM

The video transmitted by the camera is analyzed image by image in real time. Let  $I_t$  be the image at time instant  $t$  extracted from the video generated by the camera. The main image processing problem that we encounter is to find the location of the mouse in the image  $I_t$ . This can be done in various ways. In this section we describe three different methods.

The first method is based on edge detection. In this method, the edges of the mouse are detected by an edge detection method. As described in the previous section the mouse must be clearly distinguishable from the background or from the mousepad. It is preferable to have a mousepad which is distinguishable from both the background and the mouse. In this way it is easier to detect the edges of the mouse and the mouse pad (if there is any). Let us assume that  $I_t(n_t, m_t)$  be the pixel corresponding to one of the corners of the mouse or the center of an edge of the mouse or the location of a specific reference mark on the mouse etc. The pixel  $I_t(n_t, m_t)$  is obtained by performing edge detection and after some simple image processing operations such as adaptive thresholding etc. Whenever the next image  $I_{t+1}$  is available the same edge detection operation is repeated over the new image  $I_{t+1}$  and the reference mark is extracted. Let  $I_{t+1}(n_{t+1}, m_{t+1})$  be the pixel corresponding to the reference mark on the mouse. If  $(n_t, m_t)$  and  $(n_{t+1}, m_{t+1})$  are the same and the pixel values are close to each other then it is assumed that the mouse has not moved and the cursor remains wherever it is on the screen. If  $(n_t, m_t)$  and  $(n_{t+1}, m_{t+1})$  are different from each other and the pixel values are close to each other then the cursor is moved in the direction of the vector  $(n_t, m_t) - (n_{t+1}, m_{t+1})$ . The length of the cursor movement is also proportional to the difference vector. This can be adjusted by the user and according to the viewing area of the camera.

If there is no mousepad then the viewing area of the camera which is the image  $I_t$  is mapped onto the screen of the computer. If the reference mark exceeds the viewing area of the camera then the cursor is frozen at the corresponding edge of the screen. Whenever the user moves the mouse back to the viewing area the cursor becomes active again.

If there is a mousepad then the edges of the mousepad are mapped onto the screen of the computer. In the normal mode of operation the north edge of the mousepad is assumed to correspond to the upper edge of the screen. If the reference point of the mouse leaves the mousepad the cursor is frozen at the corresponding edge of the screen. Whenever the reference point of the mouse enters the mousepad the cursor becomes active again. The use of a texture-free unicolor mousepad distinguishable from the background increases the robustness of the system. Once the edges of the mousepad is detected then the image processing operations can be carried out inside the region corresponding to the mousepad.

In ordinary mouse devices there are buttons for clicking to mark certain regions on the screen. In the computer vision based mouse there are specific regions corresponding to mouse buttons. To click a button the user covers a region by his or her finger or a pointer for some time corresponding to a button. The edges of these button regions are also detected during image analysis. If a change is detected inside one of these regions it is assumed that it is assumed that it is pressed.

The second method for mouse detection and tracking method that we describe is based on color analysis. In this approach a small reference mark which has a different color from the mouse, the mousepad and the background is placed on the mouse. This mark is known by the computer a priori. This mark is used as the reference point of the mouse. Whenever the mouse is moved by the user the location of the reference point changes. The detection of the reference point can be carried out by adaptive thresholding. By tracking the reference point the cursor is moved by the computer as described above. This color coding approach is easier and more robust than the edge detection method as the viewing area of the camera is more or less known by the image analysis system.

Similarly the mouse buttons can be color coded, too. For example the left mouse button may be a green region and the right mouse button may be a blue region etc. Whenever the green region is covered by a finger or a pointer momentarily it is assumed that the left mouse button is clicked. The user should not cover the reference point and the regions corresponding to mouse buttons during the normal mode of operation.

The third detection and tracking approach is based on motion analysis. In this case two consecutive images  $I_t$  and  $I_{t+1}$  are processed together. The difference image  $I_t - I_{t+1}$  contains only the moving regions in the viewing area of the camera (assuming that the camera is fixed) [13]. If the reference point is fixed it does not appear on the difference image. If it moves it appears on the difference image and the cursor is updated accordingly.

### 3. IMPLEMENTATION DETAILS

We implemented the color coding method to realize the computer vision based mouse. Our setup is composed of only an ordinary mouse not connected to computer with a cable, and an USB web camera which produces 176 pixel by 144 pixel color images at 13.3 frames per second. To convert the ordinary mouse into a vision based mouse, we made the following modifications and used it on a controlled background consisting of a white sheet of paper:

- *Reference Mark*  
In our implementation, the reference mark is a blue region (different than the color of the mouse and background) placed on a safe position on the mouse so that the user cannot cover it while using the mouse. We placed the blue mark in the middle of top-end of the mouse as shown in Figure 1(a).
- *Mouse Buttons*  
The mouse buttons are also color-coded. They are colored in green. The region G1 (G2) in Figure 1 corresponds to the left (right) mouse button. The left and right mouse button assignments are made based on their computed locations with respect to the reference mark.
- *Background*  
To distinguish the mouse from the background, we placed the mouse on a white sheet of paper.

We assume that the color-coded regions produce local maxima in normalized green and blue channels of the video, respectively. The normalization of the channels as

$$\overline{G}_{ij} = \frac{G_{ij}}{R_{ij} + G_{ij} + B_{ij}}$$

and

$$\overline{B}_{ij} = \frac{B_{ij}}{R_{ij} + G_{ij} + B_{ij}}$$

where  $i, j$  addresses a pixel in the image;

are necessary to avoid the high values of the green and blue channels of the controlled background (white). After determining the maximum color values for the normalized blue and green channels, the threshold value for the green channel is determined as 90 per cent of the maximum normalized green level.

The vision-based mouse operates according to the steps below:

#### *Step 1: Initialization phase*

- Analyze all pixels in the window and find the maximum of the normalized green and blue channels.
- Determine the position of the reference mark as the pixel with maximum normalized blue level.
- Compute the areas of the color-coded mouse buttons as the number of pixels of the button above the calculated threshold value for the green channel.

#### *Step 2: Using the vision-based mouse*

- Find the new position of the reference mark as the pixel with the maximum normalized blue level in the next frame and update the mouse cursor position accordingly.
- Compute the areas of the color coded mouse buttons and decide whether a button is clicked by checking if more than 75 % of the original area of the button is covered.

#### *Step 3: Keeping track of the illumination changes in the environment*

- Recalculate the threshold value of the green channel by computing the 90 percent of the present value of the maximum of the normalized green channel.
- Go to Step 2.

The mouse cursor position is set according to the new position of reference mark by using the ratio between camera window size and actual screen resolution. A sample image of a case in which the user covers 75% of a region corresponding to click action is shown in Figure 1(b).

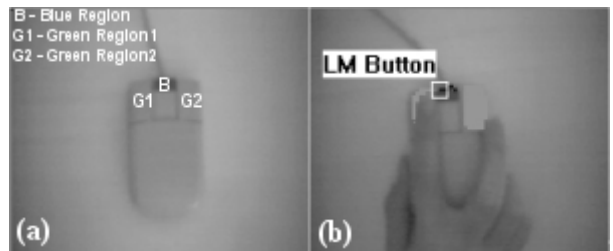


Figure 1. (a) A sample image of the mouse showing the color-coded regions, (b) A sample image captured when the user covers the left mouse button.

#### 4. CONCLUSION

In this paper a computer vision based mouse system is developed. To move the cursor on the computer screen the user simply moves the mouse shaped passive device placed on a surface within the viewing area of the camera. The mouse has well defined regions for buttons. To click a button the user simply covers the corresponding region with his or her finger.

The CV based mouse is implemented in real-time in an ordinary PC having a camera. In order to achieve a real-time working prototype the image analysis problem is simplified by defining a colored reference mark for tracking. Similarly, the mouse buttons are also color coded to simplify the clicking problem.

#### 5. REFERENCES

[1] D. Hall, J. Martin, and J.L. Crowley, "Statistical Recognition of Parameter Trajectories for Hand Gestures and Face Expressions", *Computer Vision and Mobile Robotics Workshop*, Santorini, Greece, September 17-18, 1998.

[2] I. Laptev and T. Lindeberg, "Tracking of multi-state hand models using particle filtering and a hierarchy of multi-scale image features", *Technical report CVAP245, ISRN KTH NA/P--00/12--SE*, Department of Numerical Analysis and Computer Science, KTH, Sweden, March 2000.

[3] F. Quek, D.J. McNeill, R. Ansari, X. Ma, R. Bryll, S. Duncan, K.E. McCullough, C. Kirbas, "Gesture cues for conversational interaction in monocular video", *Proceedings of Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'99)*, Corfu, Greece, September 1999.

[4] <http://wearcam.org/septambic> 'as accessed on September 1, 2001'

[5] [www.handykey.com](http://www.handykey.com) 'as accessed on September 1, 2001'

[6] O. Faruk Ozer, O. Ozun, V. Atalay, A. Enis Cetin, "[VISGRAPH: Vision Based Single Stroke Character Recognition for Wearable Computing](#)," *IEEE Intelligent Systems and Applications*, May-June 2001.

[7] Wu, Y., Shan, Y., Zhang, Z., and Shafer, S. "[Visual Panel: Towards a Mobile Visual Input Interface for Anywhere](#)". Submitted to CHI, April 2001.

[8] O.N. Gerek, A.E. Cetin, A. Tewfik, and V. Atalay, "Subband Domain Coding of Binary Textual Images for Document Archiving", *IEEE Transactions on Image Processing*, Vol.8, No.10, pp.1438-1446, October 1999.

[9] E. Oztop, A.Y. Mulayim, V. Atalay, and F. Yarman-Vural, "Repulsive Attractive Network for Baseline Extraction on Document Images", *Signal Processing*, Vol.75, No.1, pp.1-10, 1999.

[10] M.E. Munich and P. Perona, "Visual input for pen-based computers", *13<sup>th</sup> Int. Conf. Pattern Recognition*, pp.33-37, Vienna, 1996.

[11] D. Goldberg and C. Richardson, "Touch-typing with a stylus", *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, pp.80-87, New York, 1993.

[12] I.S. MacKenzie and S. Zhang, "The immediate usability of Graffiti", *Proc. of Graphics Interface '97*, pp.129-137, 1997.

[13] R. Zeibi, A.E. Cetin, Y. Yardimci, 'Small Moving Object Detection in Video Sequences', *IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, June 2000.

[14] A. Vardy, J. A. Robinson, L-T Cheng, "The Wristcam as Input Device", *Proceedings of the Third International Symposium on Wearable Computers*, San Francisco, California, Oct 1999, pp 199-202.

[15] T. Starner, J. Weaver, and A. Pentland, "A Wearable Computing Based American Sign Language Recognizer", *Proc. of the First International Symposium on Wearable Computers*, Cambridge, MA, IEEE Computer Society Press, Oct. 13-14, 1997.

[16] S. Mann and R.W. Picard, "Video orbits of the projective group: a simple approach to featureless estimation of parameters", *IEEE Trans. Image Processing*, Vol.6, pp. 1281-1295, 1997.

[17] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision", Cambridge University Press, 2000.

[18] J. C. Russ, *The Image Processing Handbook*, by IEEE Press and CRC Press, 1998.