ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Recognition of Hand movements to move cursor

Mr. Swapnil M. Gulhane*
Department of Information Technology
P.R.M. Institute of Technology & Research, Badnera
Amaravati, India
swapnilgulhane@gmail.com

Mr. A. S. Alvi
Department of Information Technology
P.R.M. Institute of Technology & Research, Badnera
Amaravati, India
abrar_alvi@rediffmail.com

***Abstract*—** This paper presents a new approach for recognition of Hand Movements. Humans and computers have interacted primarily through devices that are constrained by wires. Typically, the wires limit the range of movement and inhibit freedom of orientation. In addition, most interactions are indirect. The user moves a device as an analogue for the action in the display space. We envision an untethered interface that accepts gestures directly and can accept any objects the user chooses for interaction. This approach is used for controlling cursor movement using a real-time camera it will remove hardware dependencies like mouse. Our method is to use a camera and computer vision technology, such as image segmentation and gesture recognition, to control mouse tasks (left and right clicking, double-clicking, and scrolling) and we show how it can perform everything current mouse devices can. This paper shows how to build this mouse control system.

Keywords- Convex Hull algorithm. Image processing, Segmentation algorithm, Morphology alogorithm

I. INTRODUCTION

As computer technology continues to develop, people have smaller and smaller electronic devices and want to use them ubiquitously. There is a need for new interfaces designed specifically for use with these smaller devices. Increasingly we are recognizing the importance of human computing interaction (HCI), and in particular vision-based gesture and object recognition. Simple interfaces already exist, such as embedded keyboard, folder-keyboard and mini-keyboard. However, these interfaces need some amount of space to use and cannot be used while moving. Touch screens are also a good control interface and now days it is used globally in many applications. However, touch screens cannot be applied to desktop systems because of cost and other hardware limitations. By applying vision technology and controlling the mouse by natural hand gestures, we can reduce the work space required. In this paper, we propose a novel approach that uses a video device to control the mouse system. This proposed system can remove the hardware dependencies to move the cursor. We employ several image processing algorithms to implement this.

The recent trend in the field of computer vision includes recognition of human face and gestures as well as environment recognition with human movement. Humans might perform various kinds of activities according to their purpose and also in a certain relationship to the circumstance. Instead, even in the area of environment recognition, most of researchers have pursued recognition of objects, humans and human movements separately without taking their interaction into account.

In this paper we will try to present new approach to recognize the human finger movement and move the cursor with respect to finger. Person will show his hand to camera which is attached to computer. Software will identify the fingers or related object in his hand. When user moves finger or object, software will start tracing movement and will send message to window to change cursor position. In this paper we tried to enhance approach proposed by Mr. Hojoon Park's in paper "A Method of controlling mouse movements using real time camera" for Hand recognition system. This approach is diversified into multiple areas like Human computer interaction as it is new way to communicate with computer, Image processing, Artificial intelligence for identifying the movement and gesture and Computer vision.

II. LITERATURE REVIEW

Many researchers in the human computer interaction and robotics fields have tried to control mouse movement using video devices. However, all of them used different methods for motion tracking. One approach, by Erdem et al, used fingertip tracking to control the motion of the cursor. A click of the mouse button was implemented by defining a screen such that a click occurred when a user's hand passed over the region [1, 11]. Another approach was developed by Chu-Feng Lien [3]. He used only the finger-tips to control the mouse cursor and click. His clicking method was based on image density, and required the user to hold the mouse cursor on the desired spot for a short period of time. Paul et al, used still another method to click. They used the motion of the thumb (from a 'thumbs-up' position to a fist) to mark a clicking event thumb. Movement of the hand while making a special hand sign moved the mouse pointer. Our project was inspired by a paper of Asanterabi Malima et al. [2]. They developed a finger counting system to control behaviours of a robot. We used their algorithm to estimate the radius of hand region and other algorithms in our image segmentation part to improve our results. The segmentation is the most important part in this project. Our system used a colour calibration method to segment the hand region and convex hull algorithm to find fingertip positions [4].

III. COMPUTER VISION TECHNIQUES FOR HAND MOVEMENTS RECOGNITION

Most of the complete hand interactive systems can be considered to be comprised of three layers: detection, tracking and recognition. The detection layer is responsible for defining and extracting visual features that can be attributed to the presence of hands in the field of view of the camera(s). The tracking layer is responsible for performing temporal data association between successive image frames, so that, at each moment in time, the system may be aware of what is where". Moreover, in model-based methods, tracking also provides a way to maintain estimates of model parameters, variables and features that are not directly observable at a certain moment in time. Last, the recognition layer is responsible for grouping the spatiotemporal data extracted in the previous layers and assigning the resulting groups with labels associated to particular classes of gestures. In this section, research on these three identified sub problems of vision-based gesture recognition is reviewed.

A. Detection

The primary step in gesture recognition systems is the detection of hands and the segmentation of the corresponding image regions. This segmentation is crucial because it isolates the task-relevant data from the image background before passing them to the subsequent tracking and recognition stages. A large number of methods have been proposed in the literature that utilize a several types of visual features and, in many cases, their combination. Such features are skin colour, shape, motion and anatomical models of hands. In a comparative study on the performance of some hand segmentation techniques can be found. Different features by which we can detect hands are given below[12].

a. Colour
b. Shape
c. Learning detectors from pixel values
d. 3D model-based detection
e. Motion

B. Tracking

Tracking or the frame-to-frame correspondence of the segmented hand regions or features is the second step in the process towards understanding the observed hand movements. The importance of robust tracking is twofold. First, it provides the inter-frame linking of hand/finger appearances, giving rise to trajectories of features in time. These trajectories convey essential information regarding the gesture and might be used either in a raw form (e.g. in certain control applications like virtual drawing the tracked hand trajectory directly guides the drawing operation) or after further analysis (e.g. recognition of a certain type of hand gesture). Second, in model-based methods, tracking also provides a way to maintain estimates of model parameters variables and features that are not directly observable at a certain moment in time. Different methods for tracking the objects are given below[12].

a. Template based tracking
b. Optimal estimation techniques
c. Tracking based on the Mean Shift algorithm
d. Particle filtering

C. Recognition

The overall goal of hand gesture recognition is the interpretation of the semantics that the hand(s) location, posture, or gesture conveys. Basically, there have been two types of interaction in which hands are employed in the user's communication with a computer. The first is control applications such as drawing, where the user sketches a curve while the computer renders this curve on a 2D canvas. Methods that relate to hand-driven control focus on the detection and tracking of some feature (e.g. the fingertip, the centroid of the hand in the image etc) and can be handled with the information extracted through the tracking of these features. The second type of interaction involves the recognition of hand postures, or signs, and gestures.

Naturally, the vocabulary of signs or gestures is largely application dependent. Typically, the larger the vocabulary is, the hardest the recognition task becomes. Two early systems indicate the difference between recognition and control. The first recognizes 25 postures from the International Hand Alphabet, while the second was used to support interaction in a virtual workspace. The recognition of postures is of topic of great interest on its own, because of sign language communication. Moreover, it also forms the basis of numerous gesture-recognition methods that treat gestures as a series of hand postures. Besides the recognition of hand postures from images, recognition of gestures includes an additional level of complexity, which involves the parsing, or segmentation, of the continuous signal into constituent elements. In a wide variety of methods, the temporal instances at which hand velocity (or optical flow) is minimized are considered as observed postures, while video frames that portray a hand in motion are sometimes disregarded. However, the problem of simultaneous segmentation and recognition of gestures without being confused with inter-gesture hand motions remains a rather challenging one. Another requirement for this segmentation process is to cope with the shape and time variability that the same gesture may exhibit, e.g. when performed by different persons or by the same person at different speeds. The fact that even hand posture recognition exhibits considerable levels of uncertainty casts the above processing computationally complex or error prone. Several of the reviewed works indicate that lack of robustness in gesture recognition can be compensated by addressing the temporal context of detected gestures. This can be established by letting the gesture detector know of the grammatical or physical rules that the observed gestures are supposed to express. Based on these rules, certain candidate gestures may be improbable. In turn, this information may disambiguate candidate gestures, by selecting to recognize the most likely candidate. Different approaches for recognition of are given below[12].

a. Template matching
b. Methods based on Principal Component Analysis
c. Boosting
d. Contour and silhouette matching
e. Model-based recognition methods

IV. HAND RECOGNITION

In this section we present basic prototype for recognizing hand. Hand recognition system will start with capturing image using web camera. It removes the background image frame from by using background subtraction mechanism. For skin detection, convert the colour space RGB to YCbCr. Then, we define a range of colours as ‘skin colour’ and convert these pixels to white; all other pixels are converted to black. Then skin colour object will be hand implement convex hull algorithm to identify fingers. Next step is to identify the gesture after identifying the gesture move cursor with respect to finger position.

In Recognition human movements by motion tracking, we will try to present approach to recognize the human finger movement and move the cursor with respect to finger. Person will show his hand to camera which is attached to computer. Software will identify the fingers. When user moves finger, software will start tracing movement and will send message to window to change cursor position. The functional flow is shown in Figure 1.

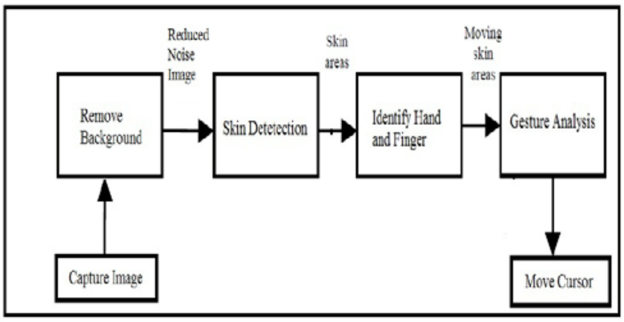


Figure 1. Functional flow of Recognition of human movements

Proposed approach for Recognition of Human Movement by motion tracking contains 7 processing layer every image pass through this seven layer. Figure 2 shows the detailed functional flow of recognition of human movements by motion tracking. Video is made up of frames and we need to analyse each frame for determining the hand and then moving cursor. Each frame passes through seven layers. Hand is detected by two approaches first one is using motion and second using colour. For detecting hand using motion we initially learn the background. This background is subtracted from frame to identify the changes in foreground. We can detect hand using motion in only controlled environment and object that move will considered as hand. While after that we again detect hand using colour where we provide skin range for detect hand but it is vulnerable to background noise and illumination.

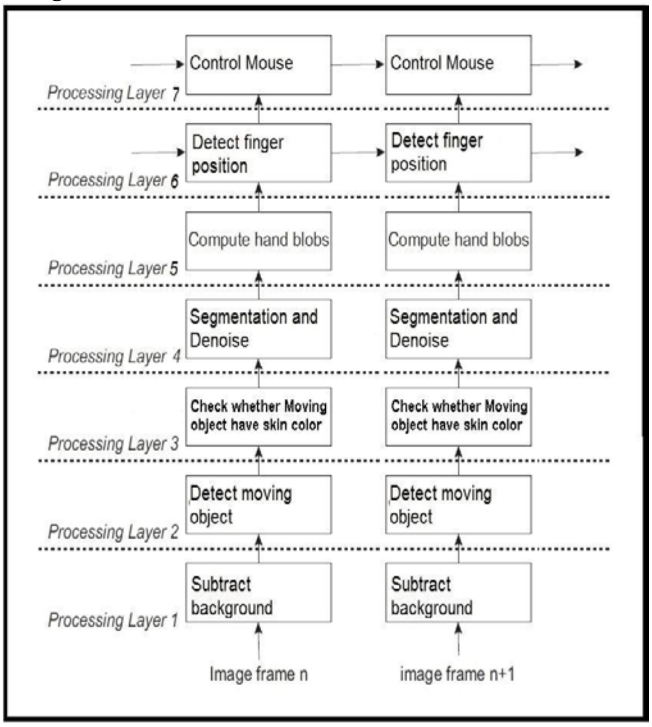


Figure 2 Detailed functional flow of recognition hand movements by motion tracking.

A. Subtract Background and Identify Moving Object

First step, we capture image from web camera and remove the background by background subtraction mechanism. So when we compare current frame with previous frame we get newly added object in frame. From this we will able to identify moving object in frame. In fig 3a) is required static background while in fig 3b) we have inserted our hand. Hand is clearly visible in fig 3b) after subtracting the background from images. So anything change occurs in foreground will get detected. It will also detect the moving object.

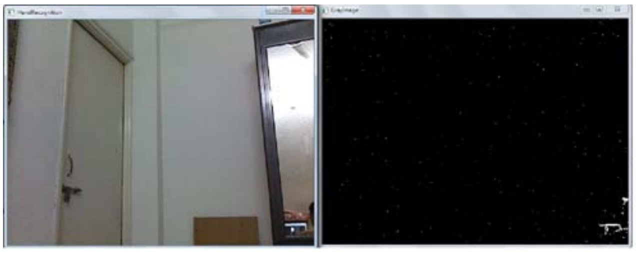


Figure 3a) Original image with static background. In second image we have just subtracted background from original image. As there are no change in foreground second frame is blank.

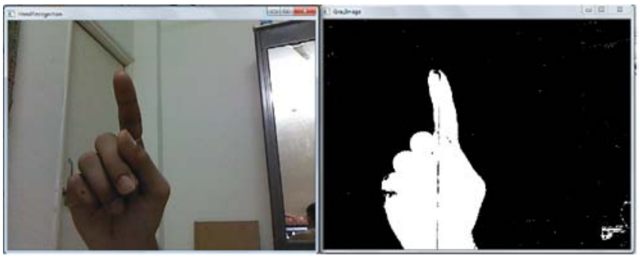


Figure 3b) Hand is inserted in next frame. Result of background subtraction is shown in second image.

Checking moving object have skin colour

Next, we need to separate the hand area from a complex background and also we want to detect skin colour. It is difficult to detect skin colour in natural environments because of the variety of illuminations and skin colors. So, we need to carefully pick a colour range. To get better results, we converted from RGB colour space to YCbCr colour space, since YCbCr is insensitive to colour variation. Specify the skin colour range and convert all the pixel come into skin colour range to white and all others to black. The conversion equation is as follows.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.29900 & 0.587000 & 0.114000 \\ -0.168736 & -0.331264 & 0.500000 \\ 0.500000 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where Y is luminance, Cb is the blue difference chroma component and Cr is the red difference chroma component. From this information, we can select a particular color range for Cb and Cr for detecting skin color. Value range selected for skin color regions are: Y is 0 to 255, for Cr is 77 to 127 and for Cb is 133 to 173. Then we loop over all the image pixels, changing pixels within the skin color range to 0, and all others to 255. Hence, we obtain a binary image of the hand. Figure 4 shows the results.

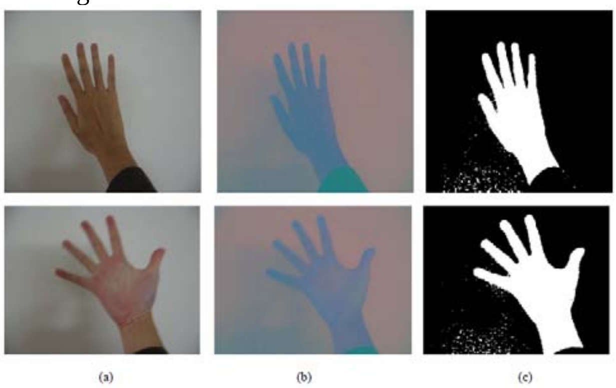


Figure 4. Convert RGB to YCbCr color space and segment hand region from original input image. (a) RGB image. (b) YCbCr image. (c) Binary image.

Next processing layer is check whether moving object have skin color, in this layer we compare the contour of skin colour to the contour of moving object. If both contours are same then we have detected proper hand. i.e. we check whether moving object have skin colour or not. Here we are initially detecting moving object and then again compare to skin colour because if we have complex background then skin colour frame contents too much noise. It is hard to detect hand from that noise. By subtracting background we eventually reduced the noise.

B. Segmentation and Delete Noise

In processing layer 4 we do segmentation and remove noise from image frame, In Segmentation, we separate the hand area from a image frame and specify hand as a region of interest. After setting region of interest we remove the noise. This noise may occur due to variation in lightning condition. For getting a better estimate of the hand, we need to delete noisy pixels from the image. We use an image morphology algorithm that performs image erosion and image dilation to eliminate noise. Erosion trims down the image area where the hand is not present and Dilation expands the area of the Image pixels which are not eroded.

Mathematically, Erosion is given by,

$$A \Theta B = \{x \mid (B)\ x \cap A\ c = \varphi\}$$

where A denotes input image and B denotes Structure elements. The Structure element is operated on the Image using a Sliding window and exact matches are marked. Figure 5 shows a graphical representation of the algorithm. Dilation is defined by,

$$A \oplus B = \{x \mid (\hat{B}) \cap A \neq \phi\}$$
$$= \{x \mid [(\hat{B})_x \cap A] \subseteq A\}$$

where A denotes the input image and B denotes the structure element. The same structure element is operated on the image and if the center pixel is matched, the whole area around that pixel is marked. Figure 6 shows the algorithm. Erosion and Dilation are changed by the shape of B. Thus, B should be selected in advance. We performed erode function with structure of 10x10 square pixels three times and dilate function with 6x6 square pixels structure three times to get a clearer hand. Figure 7 is the result of this algorithm.

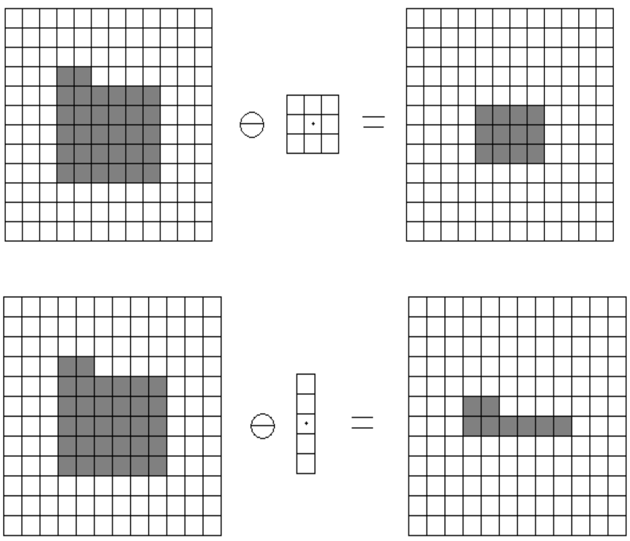


Figure 5. The result of image erosion. The structure element scans the input image. When the structure element matches, the central pixel is kept; when it does not match, all pixels are discarded.

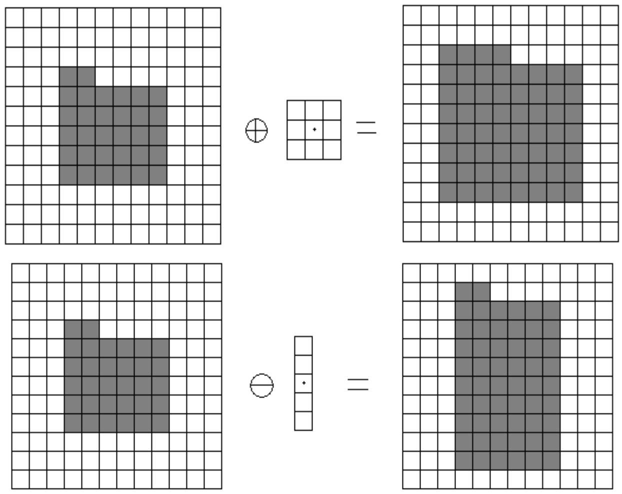


Figure 6. The result of image dilation. The structure element scans the input image. When the center of the structure matches any pixels, the bin of the structure element is filled.

(a) (b)

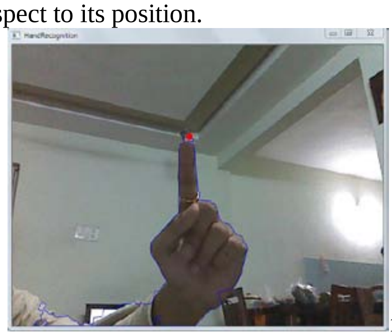
Figure 7. The result of image morphology. Perform erosion and dilation on binary image. (a) Original image. (b)The result of image morphology.

C. Compute Hand Blob

In processing layer 5, we compute hands Blob, and blobs are assigned to hand hypotheses which are tracked over time. This blob will track over the frame. Blob is indistinct shapeless form where we put mark. Hand is blob is tracked for moving cursor in next layer. In Compute blob we compute the size of hand and assign blob to hand hypotheses which are tracked over time. Draw contour in frame it will help identify the object and shape. First locate the center of the hand; we compute the radius of the palm region to get hand size. To obtain the size of the hand, we draw a circle increasing the radius of the circle from the center coordinate until the circle meets the first black pixel. When the algorithm finds the first black pixel then it returns to the current radius value. This algorithm assumes that when the circle meets the first black pixel, after drawing a larger and larger circle, then the length from the center is the radius of the back of the hand. Thus, the image segmentation is the most significant part because if some of the black pixels are made by shadows and illuminations near the center, then the tracking algorithm will meet earlier than the real background and the size of the hand region becomes smaller than the real hand.

D. Detect Finger Tips

In processing layer 6 we detect fingers, we try to detect finger. For detecting fingers we used a convex hull algorithm. The convex hull algorithm is used to solve the problem of finding the biggest polygon including all vertices. Using this feature of this algorithm, we can detect finger tips on the hand. We used this algorithm to recognize numbers of fingers are open. Identify the index finger. And start moving object with respect to its position.



16 a)

Figure 8 a) Index finger has been detected.

In final processing layer we will move the cursor; to move cursor initially we calculate Screen resolution and camera frame resolution. Now calculate ratio of screen resolution to camera image resolution. This ratio is used to calculate cursor coordinate with respect to finger position. Now Calculate the index finger coordinate in frame. Multiply ratio of screen resolution to camera frame resolution by finger coordinate, this will return cursor coordinate with respect to our finger. Send message with calculated cursor coordinate to operating system to move mouse. It is necessary that cursor coordinate should properly map with finger coordinates in camera frame. If it is not properly mapped then cursor will not move properly.

All the image frames will pass through seven processing layer for recognising hand movements to move cursor. We are using colour and motion approach to detect hand. This system required static background as it is using motion detection approach to recognise hand. Motion detection approach required controlled environment with static background.

V. CONCLUSION

We present a novel approach to control the mouse cursor by recognition of human movements using motion tracking. In this paper we have presented a novel gesture recognition system intended for natural interaction between Human and computer. We are trying to present new interface system between human and computer to control cursor which removes hardware dependencies to move cursor.

This system is based on computer vision algorithms and can extend to do all mouse tasks. However, it is difficult to get stable results because of the variety of lighting and skin colors of human races. Most vision algorithms have illumination issues. After implementation, we can expect that cursor will move with respect to hand movements. if the vision algorithms can work in all environments then our system will work more efficiently. This system could be useful in presentations and to reduce work space. In the future, we plan to add more features such as clicking mouse button for performing different activities like enlarging and shrinking windows, closing window, etc. by using the palm and multiple fingers.

VI. ACKNOWLEDGMENT

First & foremost, I would like to express my sincere gratitude towards my project guide, Prof. A. S. Alvi for his valuable guidance, encouragement, and optimism. I feel proud of his eminence and vast knowledge, which will guide me throughout my life.

I will also express my sincere thanks to Prof. A. P. Bodkhe (Head, Department of Information Technology) & other staff members for helping & providing me all the facilities for my project work. Last but not the least, I would like to express my sincere thanks to my institute Prof. Ram Meghe Institute of Technology & Research, Badnera. For providing me all the needful.

VII. REFERENCES

[1] A. Erdem, E. Yardimci, Y. Atalay, V. Cetin, A. E. Acoustics, “Computer vision based mouse”, Speech, and Signal Processing, 2002. Proceedings. (ICASS). IEEE International Conference.

[2] Asanterabi Malima, Erol Ozgur, and Mujdat Cetin, “A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control”

[3] Chu-Feng Lien, “Portable Vision-Based HCI - A Real-time Hand Mouse System on Handheld Devices”.

[4] Gary Bradski, Adrian Kaehler, “Learning OpenCV”, O’Reilly Media, 2008.

[5] Hailing Zhou, Lijun Xie, Xuliang Fang, “Visual Mouse: SIFT Detection and PCA Recognition”, cisw, pp.263-266, 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007),2007

[6] Hojoon Park, “A Method for Controlling Mouse Movement using a Real-Time Camera”.

[7] James M. Rehg, Takeo Kanade, “DigitEyes: Vision-Based Hand Tracking for Human-Computer Interaction”,Proc. of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects, Austin, Texas, November 1994, pages 16-22.

[8] J. Serra, “Image Analysis and Mathematical Morphology”, New York: Academic Press, 1983.

[9] K. Homma and E.-I. Takenaka, “An image processing method for feature extraction of space-occupying lesions,” journal of Nuclear Medicine 26 (1985): 1472-1477

[10] Robertson P., Laddaga R., Van Kleek M., “Virtual mouse vision based interface”, 2004.

[11] Vision based Men-Machine Interaction, http://www.ceng.metu.edu.tr/

[12] X. Zabulisy, H. Baltzakisy, A. Argyroszy, “Vision-based Hand Gesture Recognition for Human-Computer Interaction”