



DIGITAL ART FOR A SMARTER ENVIRONMENT

G.Valarmathi, Priyanga.S, Yazhini.S.A and Sowmya Devi.R

Sri Sairam Institute of Technology, Chennai, Tamil Nadu. Email: priyangasoundararajan@gmail.com

ABSTRACT

This paper presents an advanced and user friendly Human Computer Interaction (HCI) technology using Raspberry Pi to effortlessly translate the users' intentions into corresponding commands and output. The proposed system overcomes the disadvantages present in the existing systems such as hardware complexity, dependence on external object and limitation in recognition of the movements. It is a gestural interface that allows the physical world around us with digital information and uses gesture and color recognition algorithm based on Open CV. This recognizes the hand, tracks the hand movements and provides information about it on the screen, monitor or any component used as display.

Keyword: color tracking, gesture recognition, background subtraction, erosion, dilation

I.INTRODUCTION

Gesture Recognition is a technology which is used to identify human gestures with the help of mathematical algorithms. Gesture recognition recognizes the hand, tracks the hand movements & also provides information about hand position orientation and flux of the fingers.

Gesture recognition is interpreting human motions via mathematical algorithms. Gesture recognition can be seen as a way for a computer to understand human body language, thus building a bridge between machines and humans than text interfaces or graphical user interfaces, which still limit the input to keyboard and mouse. Gesture recognition can be applied into various applications.

Hence, gesture recognition is a technique by which the system will understand what is going to be done by the gesturer. Gesture recognition can be done with techniques from computer vision and image processing. Gesture recognition enables humans to interface with the machine and interact without any mechanical devices. Using gesture recognition, it is possible to point a finger at the computer screen so that the cursor will move according to the movement.

Hand gesture recognition system can be used for interfacing between computer and human using hand gesture. Hand extraction is the essential requirement of the image frame for the design of a hand gesture based application. These image frames can be captured from a low cost webcam for use in a vision based gesture recognition technique. Removal and variable lighting conditions are found to be efficiently handled by the system.

Color markers are placed at the tip of the user fingers. This helps the camera to identify the movement of hand and the gesture recognition. The signals generated from hand movements are fed into ARM 7 for processing using python c coding .The processed signals are fed to monitor for display. Techniques used for processing are Background Removal, Segmentation, Erosion, and Dilation. The drawing application allows the user to draw on any surface by tracking the fingertip movements of the user's finger. The pictures that are drawn by the user can be displayed on any other surface.

The color tracking also depends on the intensity of the color markers used. For different intensities different outputs are obtained. Before fixing the variant of the color to be used its hue value should be checked so that it matches with the hue value which is programmed. An interesting fact about this technology is that any surface can be used for display by simply interfacing a projector Noise present in the segmented image due to dynamic background can be removed with the help of this adaptive technique which is found to be effective for the application required.

II.PROCESSING TECHNIQUES

A. Motion estimation

Motion estimation describes the process of determining the motion between two or more frames in an image sequence. Motion compensation refers to the technique of predicting and reconstructing a frame using a given reference frame and a set of motion parameters. Motion compensation can be performed once an estimate of motion is available.

Motion estimation is not only used in the field of video compression but also in the field of spatial – temporal segmentation, scene cut detection, frame rate conversion, de-interlacing, object tracking etc.

Motion estimation and compensation have traditionally been performed using block Input, video Sub-sampling, Transform Quantization, Inverse Quantization, Inverse Transform, Motion Compensator, Motion Estimator. They offer the advantage of being fast, easy to implement and fairly effective over a wide range of video content. Block-based motion estimation is the most practical approach to obtain motion compensated prediction frames. It divides frames into equal sized rectangular blocks and finds out the displacement of the best-matched block from previous frame as the motion vector to the block in the current frame within a search window. Based on block distortion measure or other matching criteria, the displacement of the best matched block will be described as the motion vector to the block in the current frame. The best match is evaluated by a cost function such as Mean Square Error (MSE), Mean Absolute Error (MAE), or Sum of Absolute Differences (SAD).

B. Color tracking

Tracking objects based on color is one of the quickest and easiest methods for tracking an object from one image frame to the next. The speed of this technique makes it very attractive for near-real time applications. Blob detection is a fast and simple method that can be used for many machine vision tasks, such as tracking a red ball, finding a blue marker or detecting a person's skin (Skin Detection can be very useful with Face Detection and Face Recognition using a skin mask, as well as for Hand Gesture Recognition).To find color blobs, you should convert the color image from RGB to HSV format so that the colors are easier to separate. Graphics software can't be used to see which HSV

thresholds to use because the HSV values are different in Open CV than in other software. For example, if you want to detect a blue ball, Hue value for blue is roughly between 85 and 130 in Open CV.



Fig1: Ball used as color object

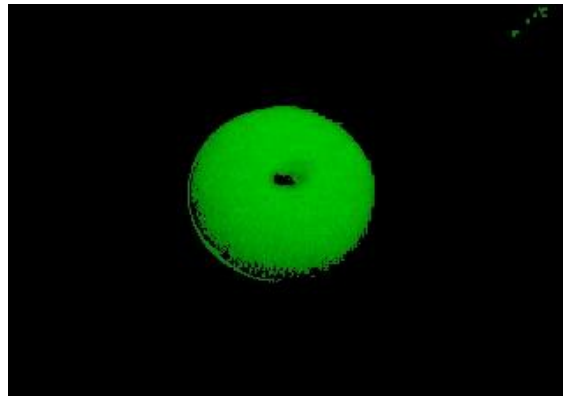


Fig2: Tracked color

C. Background subtraction

Background subtraction also known as Foreground Detection, is a technique in the fields of image processing and computer vision wherein an image's foreground is extracted for further processing (object recognition etc.). Generally an image's regions of interest are objects (humans, cars, text etc.) in its foreground. After the stage of image preprocessing object localization is required. Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called background image. Background subtraction is mostly done if the image in question is a part of a video stream. Background subtraction provides important cues for numerous applications in computer vision, for example surveillance tracking or human poses estimation. However, background subtraction is generally based on a static background hypothesis which is often not applicable in real environments. With indoor scenes, reflections or animated images

on screens lead to background changes. In a same way, due to wind, rain or illumination changes brought by weather, static backgrounds methods have difficulties with outdoor scenes.

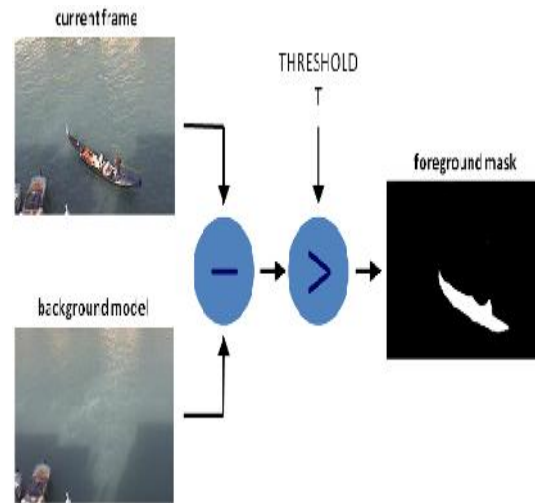


Fig3: Tresholding in background subtraction

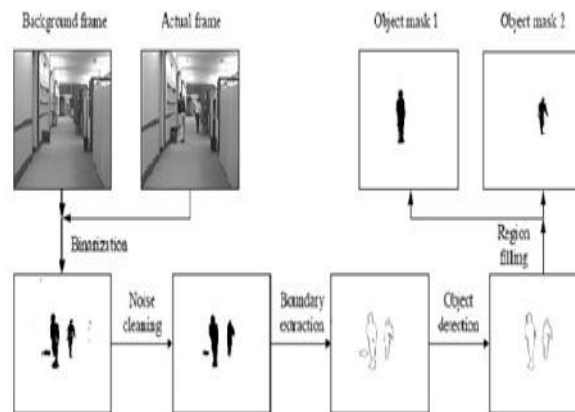


Fig4: Background subtraction process

1. Conventional Approaches

A robust background subtraction algorithm should be able to handle lighting changes, repetitive motions from clutter and long-term scene changes. The following analyses make use of the function of $V(x,y,t)$ as a video sequence where t is the time dimension, x and y are the pixel location variables.

E.g. $V(1,2,3)$ is the pixel intensity at (1,2) pixel location of the image at $t = 3$ in the video sequence.

1.1.Using frame differencing

A motion detection algorithm begins with the segmentation part where foreground or moving objects are segmented from the background. The simplest way to implement this is to take an image as background and take the frames obtained at the time t , denoted by $I(t)$ to compare with the background image denoted by B . Here using simple arithmetic calculations, we can segment out the objects simply by using image subtraction technique of computer vision meaning for each pixels in $I(t)$, take the pixel value denoted by $P[I(t)]$ and subtract it with the corresponding pixels at the same position on the background image denoted as $P[B]$.

In mathematical equation, it is written as:

$$P[F(t)] = P[I(t)] - P[B]$$

The background is assumed to be the frame at time t . This difference image would only show some intensity for the pixel locations which have changed in the two frames. Though we have seemingly removed the background, this approach will only work for cases where all foreground pixels are moving and all background pixels are static. A threshold “Threshold” is put on this difference image to improve the subtraction (see Image thresholding).

$$|P[F(t)] - P[F(t - 1)]| > \text{Threshold}$$

This means that the difference image’s pixels’ intensities are ‘thresholded’ or filtered on the basis of value of threshold. The accuracy of this approach is dependent on speed of movement in the scene. Faster movements may require higher thresholds.

D. Erosion.

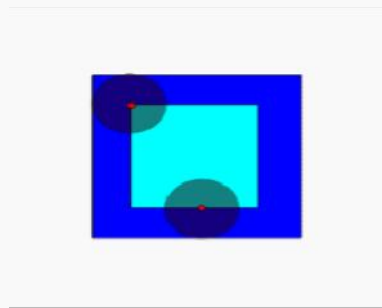


Fig 5: The erosion of the dark-blue square by a disk, resulting in the light-blue square.

Erosion is one of two fundamental operations (the other being dilation) in morphological image processing from which all other morphological operations are based. It was originally defined for binary images, later being extended to gray-scale images, and subsequently to complete lattices.

Suppose that X is the set of Euclidean coordinates corresponding to the input binary image, and that K is the set of coordinates for the structuring element.

Let Kx denote the translation of K so that its origin is at x . Then the erosion of X by K is simply the set of all points x such that Kx is a subset of X . The mathematical definition for gray-scale erosion is identical except in the way in which the set of coordinates associated with the input image is derived. In addition, these coordinates are 3-D rather than 2-D.

As an example of binary erosion, suppose that the structuring element is a 3×3 square, with the origin at its center as shown in figure. Note that in this and subsequent diagrams, foreground pixels are represented by 1's and background pixels by 0's.

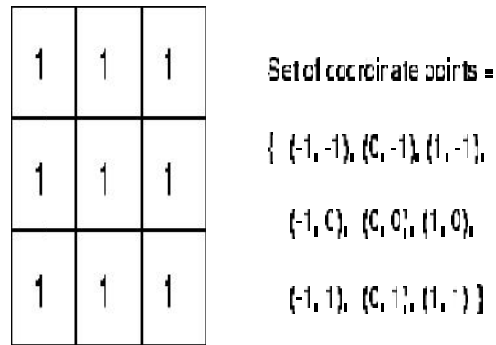


Fig 6: 3×3 square structuring element

To compute the erosion of a binary input image by this structuring element, we consider each of the *foreground* pixels in the input image in turn. For each foreground pixel (which we will call the *input pixel*) we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel coordinates. If for *every* pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is. If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value.

For our example 3×3 structuring element, the effect of this operation is to remove any foreground pixel that is not completely surrounded by other white pixels. Such pixels must lie at the edges of white regions, and so the practical upshot is that foreground regions shrink (and holes inside a region grow).

Erosion is the dual of dilation, *i.e.* eroding foreground pixels is equivalent to dilating the background pixels.

E. Dilation

Dilation is a basic operation in mathematical morphology. Originally developed for binary images, it has been expanded first to gray-scale images, and then to complete lattices. The dilation operation usually uses a structuring element for probing and expanding the shapes contained in the input image.

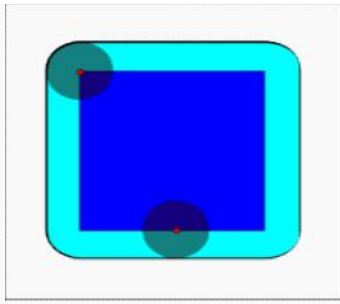


Fig 7: The dilation of dark-blue square by a disk, resulting in the light-blue square with rounded corners.

F. HSV COLOR MODEL

The hue (H) of a color refers to which pure color it resembles. All tints, tones and shades of red have the same hue. Hues are described by a number that specifies the position of the corresponding pure color on the color wheel, as a fraction between 0 and 1. Value 0 refers to red; $1/6$ is yellow; $1/3$ is green; and so forth around the color wheel.

The saturation (S) of a color describes how white the color is. A pure red is fully saturated, with a saturation of 1; tints of red have saturations less than 1; and white has a saturation of 0.

The value (V) of a color, also called its lightness, describes how dark the color is. A value of 0 is black, with increasing lightness moving away from black. The outer edge of the top of the cone is the color wheel, with all the pure colors. The H parameter describes the angle around the wheel.

The S (saturation) is zero for any color on the axis of the cone; the center of the top circle is white. An increase in the value of S corresponds to a movement away from the axis.

The V (value or lightness) is zero for black. An increase in the value of V corresponds to a movement away from black and toward the top of the cone.

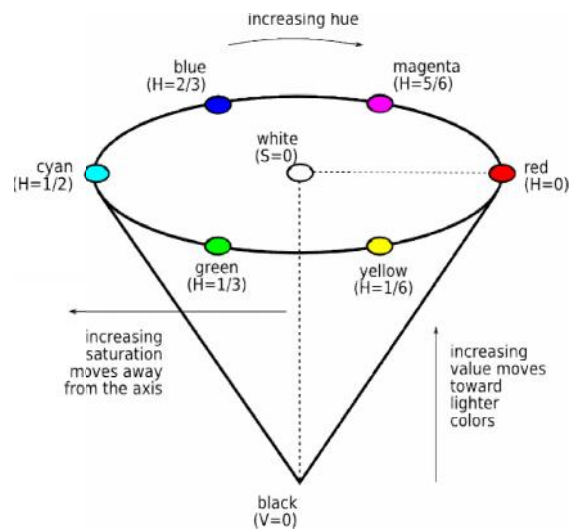


Fig 8: HSV color model

Here we convert the RGB to HSV for accurate output

Conversion Of RGB TO HSV:

The R,G,B values are divided by 255 to change the range from 0..255 to 0..1:

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

Hue calculation:

$$H = \begin{cases} 60^\circ \times \left(\frac{G' - B'}{\Delta} \text{mod} 6 \right), & C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right), & C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right), & C_{max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0, & C_{max} = 0 \\ \frac{\Delta}{C_{max}}, & C_{max} \neq 0 \end{cases}$$

Value calculation:

$$V = C_{max}$$

RESULTS

The following outputs were obtained from the system.

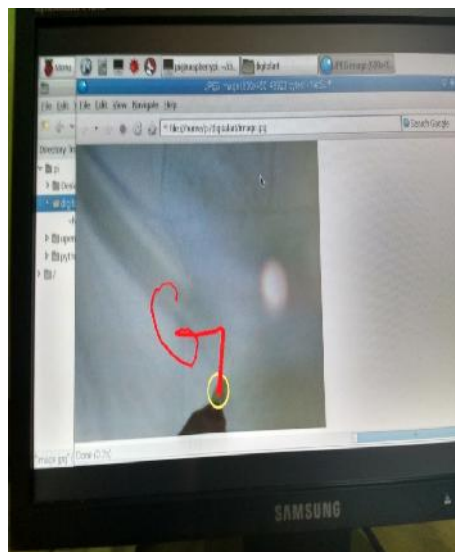


Fig 7.1 Sample Output : Alphabet “G”

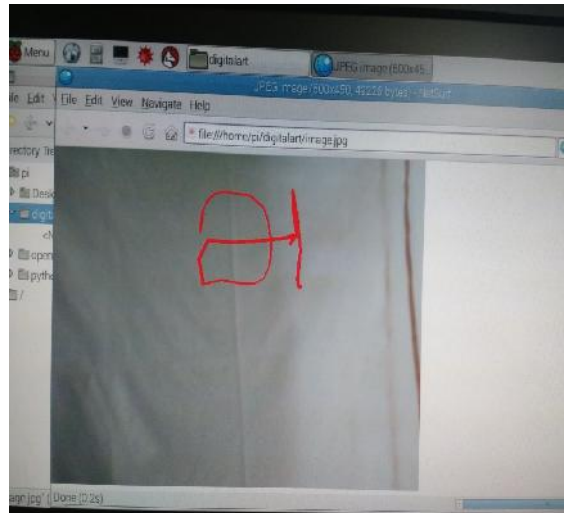


Fig 7.2 Sample Output: Tamil Letter

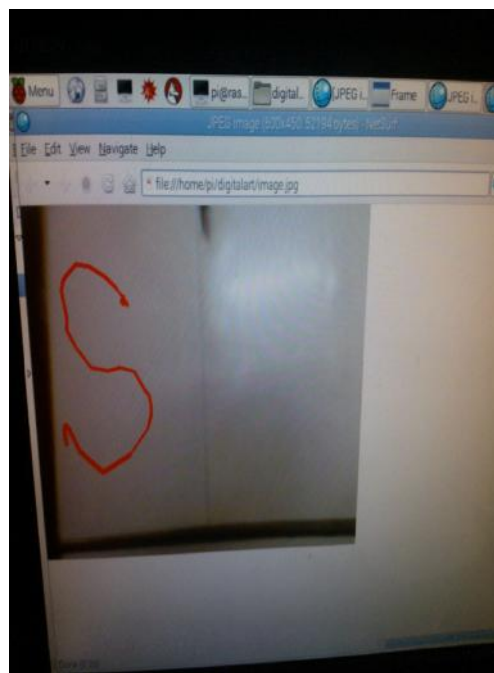


Fig 7.3 Sample Output : Alphabet “S”

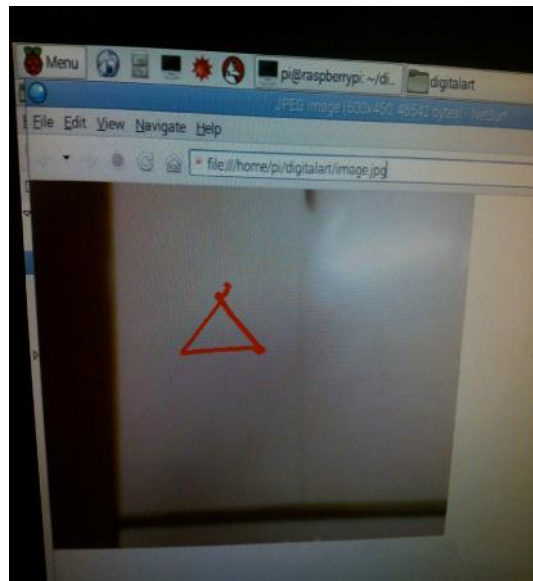


Fig 7.4 Sample Output: Triangle

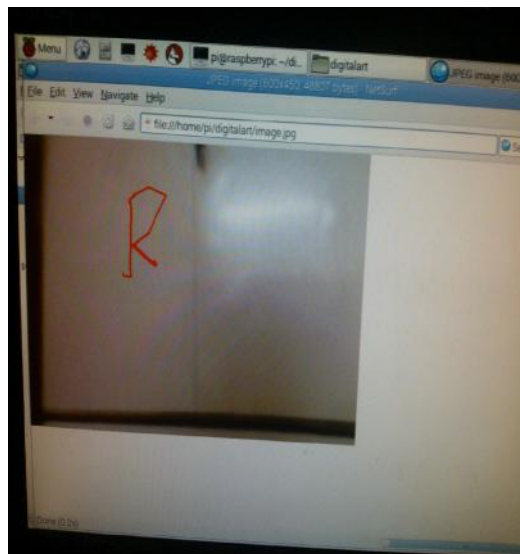


Fig 7.5 Sample Output: Alphabet "R"

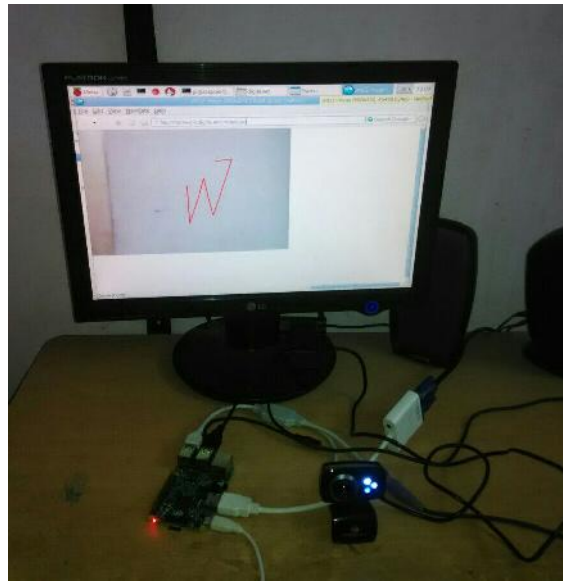


Fig 7.6 Sample Output: The Alphabet “W”

CONCLUSION

This system has presented a new Human-computer Interface method to communicate with the world. The proposed digital art uses camera, Raspberry Pi and projector. The projector projects visual information enabling surfaces, walls and physical objects around us to be used as interfaces; while the camera recognizes and tracks users' hand gestures and physical objects using computer-vision based techniques. The software program processes the video stream data captured by the camera and tracks the locations of the colour markers at the tips of the user's fingers. Compared to the previous techniques, the technique presented in this paper does not involve any mechanical tool for writing. Any object around with the programmed colour can be used for obtaining the output. The device is also very compact and less expensive. A more interesting factor is that a monitor or any kind of screen is not required to display the output. Just by interfacing a projector with the device any surface (such as palm, wall, and paper) can be used as screen.

VI.FUTURE SCOPE

The proposed system can be applied in digital classrooms, seminar halls, conferences, etc. It can also be used as a writing aid for paralyzed people. This can be made into a compact device which can be carried in the form of touch free anywhere and used for any purpose. This system can be further improved by increasing the processing speed and be implemented.

VII. REFERENCES

- 1.“ Similarity Matching-Based Extensible Hand Gesture Recognition”, **R.Xie X.Sun; X. Xia ; J. Cao**,IEEE Sensors Journal, Year: 2015,Volume: 15, Issue: 6
2. “A robust background subtraction approach with a moving camera ”Xiaochun Zou; Xinbo Zhao; Zheru Chi Computing and Convergence Technology (ICCCT), 2012 7th International Conference on,Year: 2012

- 3“.Real-Time and Robust Compressive Background Subtraction for Embedded Camera Networks”, Y.Shen; W. Hu; M. Yang; J. Liu; B. Wei; S. Lucey; C. T. ChouIEEE Transactions on Mobile Computing Year: 2016, Volume: 15, Issue: 2
4. “A review on real time object tracking in video sequences”, S. V. Kothiya; K. B. Mistree
Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 International Conference on ,Year: 2015
- 7.“ Real-time 3D game using sixth sense and haptic technology”
A. A. Kadu; A. S. Nagdive
Computation of Power, Energy, Information and Communication (ICCPEIC), 2014 International Conference on Year: 2014
8. “Sixth-Sense: Context Reasoning for Potential Objects Detection in Smart Sensor Rich Environment”
B. Guo; S. Satake; M. Imai
Intelligent Agent Technology, 2006. IAT '06. IEEE/WIC/ACM International Conference on Year: 2006
- 10.“Handicapped people virtual keyboard controlled by head motion detection ”
O. Krejcar
Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on Year: 2011
11. W. C. Bang, W. Chang, K. H. Kang, E. S. Choi, A. Potanin, and D. Y. Kim, “Self-contained spatial input device for wearable computers,” in Proc. IEEE Int. Conf. Wearable Comput., Oct. 2003, pp. 26–34.
12. G. Bailador, C. Sanchez-Avila, J. Guerra-Casanova, and A. de Santos Sierra, “Analysis of pattern recognition techniques for in-air signature biometrics,” Pattern Recognit., vol. 44, nos. 10–11, pp. 2468–2478, 2011