MULTI MODEL HUMAN COMPUTER INTERACTION SYSTEM BASED ON GAZE TRACKING AND SPEECH RECOGNITION

Howida Abdelfattaah Shedeed[1], Abdlel-Badeeh M. Salem[2]
Faculty of Computer and Information Sciences, Ain shams University,
Cairo, 11566, Egypt.
[1]dr_howida@yahoo.com , [2]abmsalem@yahoo.com

Keywords: Human Computer interaction, iris detection and tracking, gaze estimation and mapping, disabled people support systems, speech recognition.

Abstract. *This paper presents an architecture and design for a multi-model human computer interface system used to control the mouse movements and execute mouse and windows commands using gaze tracking and speech recognition. The system is to help handicapped people, whom have severe disability in using their hands. The paper proposed optimum iris detection and tracking methodology integrated with an investigated method for gaze estimation and mapping. Our proposed method for gaze estimation and mapping is a hybrid method of the "2D Direct mapping" method and the "2D Reference mapping" method. Experimental results show that our system achieved 92% accuracy for iris detection and tracking. Our investigated 2D Gaze mapping method also achieved a high mapping accuracy and smoothing cursor pointer transition with the ability to recover when the user moves away from the camera and returns again. The method reduced the computational and calibration effort more than any previous used 2D mapping methods. We extend our system ability by adding a speech command recognition feature that enables the disabled user to execute other mouse and windows commands that are complex to recognize using the eyes.*

1. INTRODUCTION

Many physically disabled individuals are deterred from using computers due to their inability to utilize a hand-controlled mouse and keyboard because they cannot control their hands in the right way.

This paper presents a complete architecture and design for a system that can help physically disabled individuals to use the eye's iris position and movement to control the cursor movement on the computer's screen. Speech recognition feature is also used to order mouse and windows commands that are complex to recognize using the eyes.

The traditional phases for the methodology that has been used to control the cursor movement using gaze mapping are: 1-face and eye detection, 2-iris detection, 3-iris Tracking, and then 4- gaze estimation and mapping on screen coordinate (to map the iris movements to cursor movements on screen coordinates). Many researches have been done in every phase.

In this paper we proposed optimum iris detection and tracking methodology integrated with an investigated method for gaze estimation and mapping. Our investigated 2D Gaze mapping method achieves a high mapping accuracy and smoothing cursor transition with less computational and calibration effort than any previous used 2D mapping methods. We extend the system capabilities by using the speech recognition feature to facilitate some mouse and windows commands that are complex to recognize using the eyes. These commands include, the mouse commands: left click, right click, drag and drop and other windows commands like: minimize, maximize, back, and forward.

The remainder of this paper is organized as follows: Section 2 presents the previous work in each phase of the system development methodology. Sec 3 presents the proposed system architecture explaining the proposed techniques for each phase. Sec 3 illustrates the experimental results and finally conclusion and future work are drawn in Sec 4.

2. PREVIOUS WORK

In this section we illustrate the previous research techniques for "iris detection and tracking" and "gaze estimation and mapping", which are the two main phases for the methodology that has been used to control the cursor movement using gaze estimation and mapping.

2.1. Iris detection and tracking techniques

Many research have been done for iris detection and tracking include: [1], [2], [3], [4],[5] , [6], [7], [8], [9] and [10]. Most of them used one of the three main algorithms for iris detection and tracking: segmentation, Hough transform and template matching, or a combination with some enhancements added to improve the results.

Many researches such as [4], [5] and [6] used the segmentation algorithm with some added enhancements in each to detect the iris. Image segmentation is the process of assigning a label to every pixel in an image such that, pixels with the same label share certain visual characteristics. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images.

First step in segmentation method is to convert the image to grayscale image, in order to remove non-uniform illumination, to be segmented properly and then detect the iris properly. Background modeling and subtraction techniques are then used to detect the foreground object (the iris) and also to track the iris in the next video frames. Our preliminary work for detecting the iris using segmentation techniques showed low time processing, high error rate and law accuracy so we proposed a methodology for iris detection and tracking based on the other two algorithms: Hough transform and template matching, which will be discussed briefly in the next subsections.

2.2. Hough transform algorithm

The Hough transform [8] can be described as a transformation of a point in the x-y plane to the parameter space.

The parameter space is defined according to the shape of the object of interest.

The first step to detect the iris in an image using Hough transform is to convert the shape of the human's iris to parametric equations. In the case of human's iris, the shape will be a circle and the parametric equations are:

$$x = a + R \cos(\theta) \qquad (1)$$
$$y = b + R \sin(\theta) \qquad (2)$$

Where “a” and “b” can be the coordinates of the pupil center’s position and “R” is the radius of the iris. The algorithm is as follows: First we find all edges in the image using any edge detection technique; Canny, Sobel or Morphological operations. At each edge point we draw a circle centered at this point with the desired radius. This circle is drawn in the parameter space, such that our x coordinate is the "a" value and the y coordinate is the "b" value while the z coordinate is the radii. At the coordinates which belong to the perimeter of the drawn circle we increment the value in our accumulator matrix which essentially has the same size as the parameter space. In this way we sweep over every edge point in the input image drawing circles with the desired radii and incrementing the values in our accumulator. When every edge point and every desired radius is used, we can turn our attention to the accumulator. The accumulator will now contain numbers corresponding to the number of circles passing through the individual coordinates. Thus the highest numbers (selected in an intelligent way, in relation to the radius) are corresponding to the centers of the circles in the image [9].

Using Hough transform for iris detection has some advantages such as: it doesn’t need any database as template matching method and also no need for special light. It provides a high accuracy results. It enables us to find iris with different sizes in a given range. The only disadvantage for Hough transform method is that, it takes long time according to the size of the image and the trained range for iris radius. For these reasons we prefer to use the Hough transform method for detecting the iris in the first frame only in the video image due to its high accuracy. This detected iris is then used as a template for tracking the iris in the subsequent frames using template matching method which provides law computational effort with reasonable accuracy and then less computational time in real time application.

2.3. Template matching using affine template algorithm

To locate the iris by its color and shape information, template matching using an affine iris template is used in [10]. Assuming that the iris template size is 20 X 20, it is not assured that the incoming frame has an iris area dimensions equal to these dimensions. Choosing an appropriate size for the given iris template is variable due to many variations like, the distance between the user and the input camera. This problem can be solved by selecting a range of sizes e.g. (from 15 X 15 to 25 X 25) and then resizing the iris template to match the given current size. For each iris template size, after resizing operation, we apply the template matching algorithm using the iris template over the incoming camera frame. If the maximum correlation coefficient of the template matching algorithm exceeds a predefined threshold, the resizing operation diverges and the current iris template size will be used during the tracking phase.

2.4. Gaze Mapping Techniques

In general, most of the non-intrusive vision-based gaze tracking techniques can be classified into two groups: 2D mapping-based gaze estimation method [11],[12], [13], [14],[15],[16],[17],[18],[19] and direct 3D gaze estimation method [20], [21], [22], [23], [24], [25], [26]. In the following section, each group will be discussed briefly.

2.5. 2D Mapping-Based Gaze Estimation Technique

For the 2D mapping-based gaze estimation method, the eye gaze is estimated from a calibrated gaze mapping function by inputting a set of 2D eye movement features extracted from eye images, without knowing the 3D direction of the gaze. Usually, the extracted 2D eye movement features vary with the eye gaze so that the relationship between them can be encoded by a gaze mapping function. In order to obtain the gaze mapping function, an online calibration needs to be performed for each person. Unfortunately, the extracted 2D eye movement features also vary significantly with head position; thus, the calibrated gaze mapping function is very sensitive to head motion [19]. Hence, the user has to keep his head unnaturally still in order to achieve good performance.

In order to correctly estimate visual gaze it would be reasonable to consider the head position and orientation to give a rough initialization of the visual gaze, and then use the information about the eye centers and corners to fine tune the information. Unfortunately, head pose estimators often involve many assumptions in order to achieve a realistic modeling (i.e. the shape and size of the head, the possible rotation angles of the eye, etc.). Furthermore, the high computational requirements of head pose estimators are not in line with the lightweight requirements of our system. Finally small mistakes in pose estimation might introduce additional errors in the final visual gaze estimation.

Other methods tend to simplify the problem by assuming that the eye doesn't rotate but it just shifts. This assumption is reflected in commercial eye trackers, which deal with high resolution images of the eyes. This simplification comes from the assumption that the face is always frontal to the screen so the head pose information can be discarded. Therefore **[**13] and [14] used the linear mapping method suggested by [11]. The user needs to perform a calibration procedure by looking at several known points on the screen as shown in Figure 1. A 2D linear mapping is then constructed from the vector between the eye corner and the iris center and recorded at the known position on the screen as shown in Figure 1. This vector is then used to interpolate between the known screen locations. For example, if we have two calibration points P1 and P2 with screen coordinates (α_1, β_1) and (α_2, β_2) respectively and eye-center vectors (taken with origin from the anchor point) (x1, y1) and (x2, y2), we can interpolate a new reading of the eye center vector (x, y) to obtain the screen coordinates (α, β) by using the following interpolant equations [13]:

$$\alpha = \alpha_1 + \frac{x - x_1}{x_2 - x_1}(\alpha_2 - \alpha_1) \qquad (3)$$

$$\beta = \beta_1 + \frac{y - y_1}{y_2 - y_1}(\beta_2 - \beta_1) \qquad (4)$$

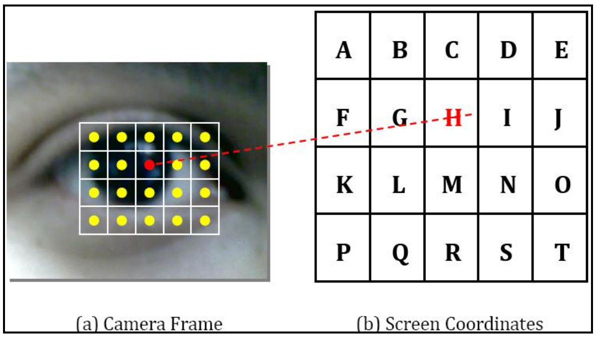


Fig. 1. Calibration Procedure

The advantage of this approach is its low computational cost and a decent accuracy with respect to more complex systems. In facts the reported error introduced by this approximation is just 1.2^o. Unfortunately, this method doesn't allow head movements, so the user will need to recalibrate in case of big horizontal or vertical shifts. [15] Proposed a Dynamic Head Compensation Model to compensate for the head movements so that whenever the head moves, the gaze mapping function at a new 3D head position can be updated automatically. But this compensation process added an extra computational load to the original system which may take extra time and decreases the real time system performance [15]. Used the specific gaze mapping function $Ss = f(v)$, based on Pupil Center Cornea Reflection (PCCR) technique, which modeled by the following nonlinear equations [16]:

$$x_{gaze} = a0 + a1 * v_x + a2 * v_y + a3 * v_x * v_y \quad (5)$$
$$y_{gaze} = b0 + b1 * v_x + b2 * v_y + b3 * v_y^2 \quad (6)$$

Where the extracted pupil-glint vector v is represented as (v_x, v_y) and the screen gaze point Ss is represented by (x_{gaze}, y_{gaze}). The coefficients $a0$, $a1$, $a2$, $a3$ and $b0$, $b1$, $b2$, $b3$ are estimated from a set of pairs of pupil-glint vectors and the corresponding screen gaze points. These pairs are collected in a calibration procedure. During the calibration, the user is required to visually follow a shining dot as it displays at several predefined locations on the computer screen. In addition, the subject must keep his head as still as possible.

The main advantage of the direct 2D mapping method (linear or nonlinear) is its low computational cost. But unfortunately, we have a disadvantage of idle long calibration process, which required at each startup of the system.

[28] Intended to fix this problem using a new 2D mapping approach called "reference mapping" method. In reference mapping, there is no need to calibrate the system neither at every startup nor with different user, the fact that makes this method more reusable than the 2D direct mapping method. Reference mapping method is based on the assumption that the eye does not rotate but it just shifts, so we treat the iris movements as translations in 2D space. As a result, movement of the iris across two subsequent frames is used to move the cursor pointer by a relative distance computed by the difference in x-direction and y-direction, i.e. $dx=(X_{f2} – X_{f1})$ & $dy=(Y_{f2} – Y_{f1})$, multiplied by a reference mapping constant in the same direction as the iris as shown in Figure 2. The reference mapping constants Sx & Sy are computed by scaling the search window, to the screen dimensions in pixels in x and y directions respectively. The search window is defined to contain the centers of all sub-images tested for the used feature (the iris) [28].

2.6. Direct 3D gaze estimation technique

For the direct 3D gaze estimation techniques, the 3D direction of the gaze is estimated so that the gaze point can be obtained by simply intersecting it with the scene. Therefore, how to estimate the 3D gaze direction of the eye precisely is the key issue for most of these techniques. Several attempts [24], [23], [20], [21], [26] have been proposed to estimate the 3D direction of gaze from the eye images.

Most of the existing 3D gaze tracking techniques either require knowledge of several user-dependent parameters about the eye [24], [23], [20], or cannot work under certain circumstances [21] or with large head movement [26]. But in reality, these user-dependent parameters of the eyeball, such as the cornea radius and the distance between the pupil and the cornea center, are very small (normally less than 10mm). Therefore, accurate indirect estimation techniques like the one proposed in [26] to estimate these eye parameters is a prerequisite, which added additional effort and time to the mapping process. Furthermore, [15] concluded that, in terms of accuracy, the experiments indicate that the 2D mapping-based gaze estimation technique is more accurate than the 3D gaze mapping technique.

3. PROPOSED SYSTEM ARCHITECTURE

Our system consists of two modules. The first module ("Control using gaze" module) is used to control the mouse movement using "gaze tracking". The second module used the speech command recognition feature to facilitate the execution of mouse and windows commands that are complex to recognize using the eyes. Our proposed system methodology for the first module is as shown in figure 3. The proposed methodology reduced the effort needs for the first phase of similar systems (face and eye detection) by using a small webcam stands in front of user's eye at short distance (about a foot). Then the first phase in our methodology is to detect the iris from the first camera frame using the accurate Hough transform algorithm. The detected iris image will then be used as an iris template for the second phase, iris tracking, in the next frames. The third and last phase in our methodology is the gaze estimation and mapping. Gaze mapping techniques has been investigated as 2D or 3D mapping based. [15] Concluded that, in terms of accuracy, the experiments indicate that the 2D mapping-based gaze estimation technique is more accurate than the 3D gaze mapping technique. This is why we prefer to investigate a new 2D mapping based technique.

In this paper we introduced a new investigated 2D mapping technique, called "hybrid mapping", that we used to map the gaze to a screen coordinate.

Our investigated gaze mapping technique achieved a high mapping accuracy, smoothing cursor transition and the ability to recover when the user moves away from the camera and returns again.

Also our investigated technique required less computational and calibration effort than the previous used 2D mapping techniques.

The next subsections illustrate the design for each module of the system.

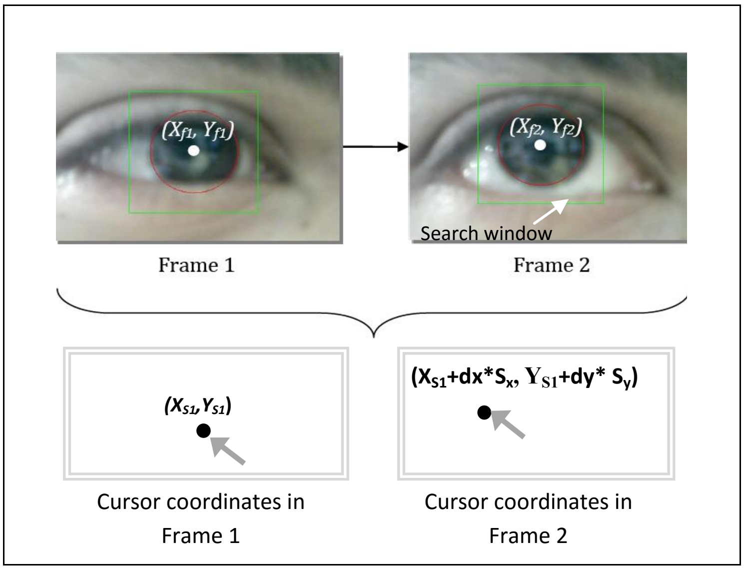


Fig. 2. Reference Mapping Method Demonstration

3.1. "Control Using Gaze" Module

Figure 3 shows the main phases for the methodology used for this module which are: iris detection, iris tracking, and "gaze estimation and mapping". Our proposed technique for every phase is as follows.

3.2. Iris Detection

The first phase in our system methodology, for "Control using gaze" module, is to detect the iris of the eye. We used a camera that, was fixed at average distance of about one foot from the user face to enable the user to use the system under natural head movement but still able to detect the iris accurately using our proposed iris detection methodology.

Accurate detection of the iris is an important goal in achieving higher accuracy in tracking the iris.

Hough transform is an accurate technique that has been used to detect the iris but unfortunately, it requires large computational effort and then large execution time which reduces the system performance.

So our proposed methodology for iris detection and tracking is to use the "Hough transform" technique for accurate iris detection from the first frame only and then use the template matching algorithm, which gives a reasonable accuracy but with less computational effort, to track the iris in the next frames.

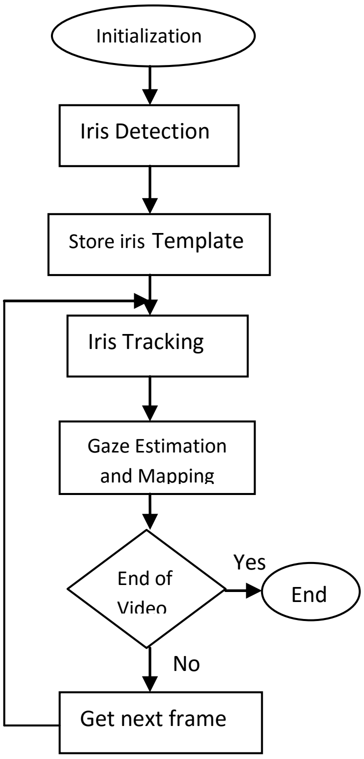


Fig. 3. Proposed Methodology for the "Control using Gaze" Module

To achieve high accurate detection using Hough transform, we proposed to do some preprocessing operations on the image of the first camera frame, to enhance the accuracy of the Hough transform, as follows and as shown in Figure4:

1. High contrast operation to delete the small details in the image. As a result, the iris will be the dark region and the others features will be bright. The used value for the contrast depends on the iris color and the light intensity in the surrounding environment. Thus for "light color" iris this step can be avoided.
2. Convert the image to gray scale image. By this step all color values of the image pixels will be between 0 and 255, i.e. one channel.
3. Use Gaussian filters to remove the noise from the image, such as salt and pepper noise, to be clearer. After this step the iris will be the largest dark region in the image.

Figure 5 shows the output of our implemented "Hough transform" method.

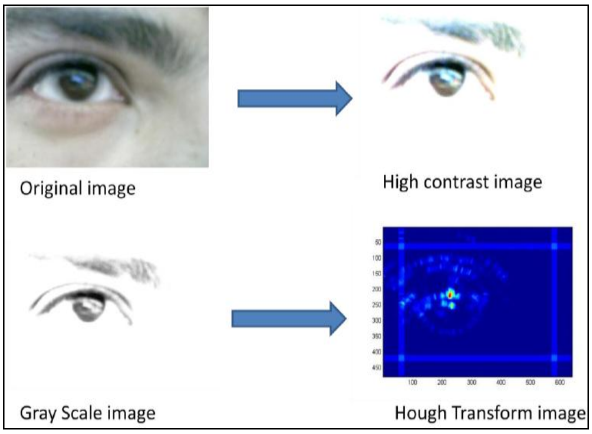


Fig. 4. Preprocessing and Hough transform implementation

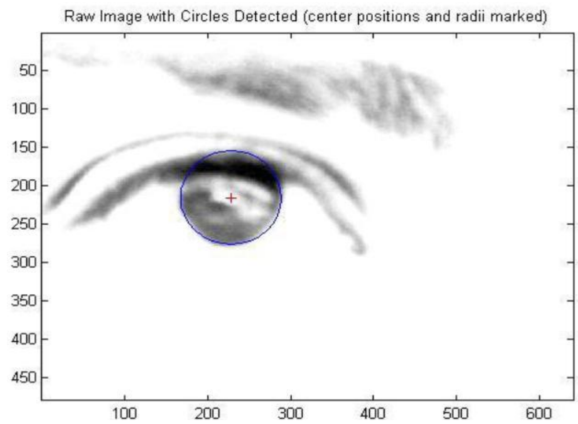


Fig. 5. Hough Transform result

3.3. Iris Tracking

Template matching Technique is used for iris tracking in our proposed methodology. After detecting the iris in the first camera frame using the accurate "Hough transform" technique, the detected iris's image is then used as an iris template for tracking the iris in the next frames. Assuming a fixed distance for the subject from the computer screen, then the template will resemble the iris in size and we didn't need to do any resizing operation. This added an enhancement to our system performance. Each sub-image s(x, y) in the search window is matched with the template sub-image t(x, y). As a measure of match, the normalized correlation coefficient is used according to the following equation:

$$r(s,t) = \frac{A\sum s(x,y)t(x,y) - \sum s(x,y)\sum t(x,y)}{\sigma_s \sigma_t} \quad (7)$$

Where:

$$\sigma_s = \sqrt{\sum s(x,y)^2 - (s(x,y))^2}$$
$$\sigma_t = \sqrt{A\sum t(x,y)^2 - (t(x,y))^2}$$

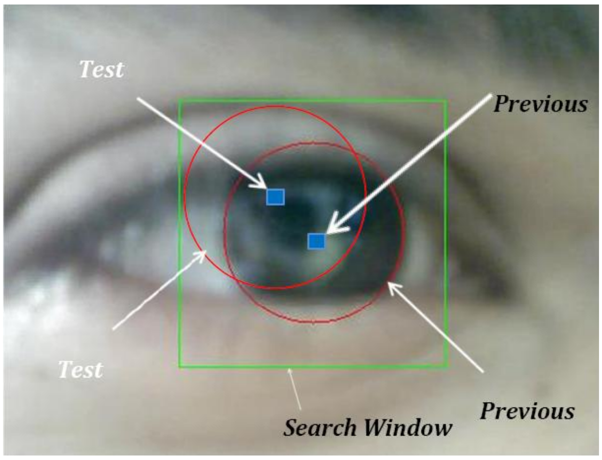


Fig. 6. Template matching Technique Demonstration

The sub-image with the highest correlation coefficient among all sub-images in the search window is determined as shown in figure 6. It is used to find the feature center and then mapped to a screen_coordinates. [28] Cropped the sub-image of the feature from the current frame and used it as a template that is used to search for the best matching sub-image in the next frame. But we find that updating the iris template is not a necessary step and take a time that may reduce the system performance, so we proposed to use the same iris template detected using the accurate "Hough transform" method from the first camera frame. If at any time a low correlation is obtained and the template no longer resembles the iris, the detection process can be processed once again to begin refreshed iris tracking. Pseudo code for the proposed algorithm is as shown in Figure 7.

```
maxCorrelationCoeffecient
🡐 minValue;
FOR x=0 TO
x<=searchWindowHeight-
templateHeight
{
FOR y=0 TO
y<=searchWindowWidth-
templateWidth
{
correlationCoeffecient 🡐 0;
FOR i = 0 TO i <
templateHeight
 {
FOR j = 0 TO j < templateWidth
{
pixel searchIMG
🡐search[x+i][y+j];
pixel templateIMG
🡐template[i][j];
correlationCoeffecient+=abs(se
archIMG.Gray-templateIMG.Gray
); }}
if(maxCorrelationCoeffecient<c
orrelationCoeffecient )
{
maxCorrelationCoeffecient🡐cor
relationCoeffecien; posi-
tion.bestRow 🡐x;
position.bestCol 🡐y;
position. corrCoeffecient
🡐correlationCoeffecient ; }
  }    }
```

Fig. 7. Pesudo code for The Proposed Template matching algorithm

3.4. Gaze Estimation and Mapping

We proposed an investigated technique for gaze mapping called "2D hybrid mapping technique". The investigated technique is a hybrid technique from the two known 2D techniques that we explained in Sec 2; 2D direct linear mapping technique and 2D reference mapping technique.

Our new hybrid technique combines the advantages of the previous two techniques. Moreover, it minimized the calibration effort done by the user.

(x_0, y_0) Top-Left point					
					(x_f, y_f) Bottom-Right point

Fig. 8. Hybrid mapping demonstration

In our investigated "hybrid mapping" technique, the user needs to perform a calibration procedure by looking at two points only (top-left & bottom-right) on the screen that treated as an n-order square matrix and is equally divided into 2N areas as shown in Figure 8. The higher will be the N value; the thinner will be the screen division. At each gaze fixation on the given two screen points, the iris center location is stored. These locations are then be used as reference points during the mapping process. Figure 9 list the pseudo code for the calibration procedure for the proposed "Hybrid mapping" technique.

```
GET TOP-LEFT POINT AT FRAME 1:
 { topLeftX ← irisCenter.x;
 topLeftY ← irisCenter.y;
referencePoint.x← topLeftX;
 referencePoint.y ← topLeftY; }
GET BOTTOM-RIGHT POINT AT FRAME
2:
{ bottomRightX ← irisCenter.x;
 bottomRightY ← irisCenter.y;
XDifference ← bottomRightX -
topLeftX;
 YDifference ← bottomRightY -
topLeftY; }
XmappingConstant= SC_X/
XDifference;
YmappingConstant= SC_Y/
YDifference;
```

Fig. 9. Calibration Procedure for the investigated "Hybrid mapping" technique.

The reference window shown in Figure 10 in white color is the window where the iris centers locations exist and are mapped to screen coordinates.

Here we consider (X_0, Y_0) as the reference point (anchor) to which an iris center location given a camera frame is mapped with reference to. Thus the hybrid mapping method combines the 2D direct mapping method and the reference mapping method. So in any incoming camera frame, the cursor pointer position (Xs, Ys) can be calculated using the following formula:

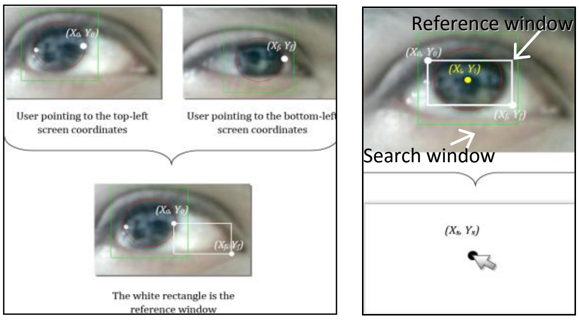
$$Xs=(X_i-X_0)\times((SC_x/(X_f-X_0)) + 1) \quad (8)$$

$$Ys=(Y_i-Y_0)\times((SC_y/(Y_f-Y_0)) + 1) \quad (9)$$

Where: SC_x and SC_y are the width and the height of the screen respectively.

Figure 11 list the pseudo code which illustrates the investigated "Hybrid mapping" technique. Notice that, XmappingConstant & YmappingConstant are the reference mapping constants computed (as in Fig 9) using the reference window (shown in Fig 10) dimensions instead of the search window (shown in Fig 10) dimensions as used in the "reference mapping" technique [28].

That makes our investigated technique to be more accurate than the "reference mapping" technique.



(a) (b)

Fig. 10. (a) Calibration for the Hybrid mapping (b) The hybrid mapping reference.

```
POINT previousPointerPosition;
GetCursorPos(previousPointerPo
sition);
IF(IrisCenter IS OUTSIDE THE
REFERENCE WINDOW)
 { return; }
 POINT pointerPosition;
PointerPosition.x
←(IrisCenter.x -
referencePoint.x)
*  (XmappingConstant + 1);
 PointerPosition.y
←(IrisCenter.y -
referencePoint.y)
*(YmappingConstant + 1);
 IF(PointerPosition.x >
screenX)
{ PointerPosition.x
```

```
 🡐  screenX; }
  IF(PointerPosition.y >
 screenY)
  { PointerPosition.y =
 screenY; }
  IF(pointerPosition.x < 0)
  { PointerPosition.x 🡐  0; }
 IF(PointerPosition.y < 0)
  { PointerPosition.y 🡐  0; }
 SetCursorPos(pointerPosition);
```

Fig. 11. Pesudo code for the investigated "Hybrid mapping" Technique.

Experiments showed that, our investigated 2D "Hybrid mapping" technique achieved high mapping accuracy than the other two previous used techniques. Also our technique showed the ability to recover when the user moves away from the camera and returns again than the "reference mapping" technique and it required less computational and calibration effort than the "direct mapping" technique.

3.5. Speech Recognition Module

To facilitate the mouse commands, some Human-Computer Interaction (HCI) systems like [27] detected the user's eye blinks and analyzed the pattern and duration of the blinks, to use them to provide input to the computer in the form of a mouse click. [28] Tracked some of the body-features such as the tip of the user's nose, the lower lib and the thumb to use them to spell out messages or play games. [29] Presented a novel (HCI) system with calibrated mono-camera which integrated active computer vision technology and embedded speech command recognition technology. Mainly by tracking the nose tip motion robustly as the mouse trace, this system completed mouse mission with recognition rate more than 85% at the speed 15 frame per second.

So we extend our system capability by adding a speech command recognition feature that enables the disabled user to execute other mouse and windows commands that are complex to recognize using the eyes or anybody-features. The supported speech words will explain in the following Sec. We used Microsoft Speech API (Speech Application Programming Interface) to implement the speech recognition in our system because it is free to use and compatible with windows SAIP used in Microsoft Office, Microsoft Agent and Microsoft Speech Server.

4. EXPERIMENTAL RESULTS

We provide a human computer interaction system that integrates both gaze mapping and speech recognition to give a system with higher capabilities than the systems that used the gaze only to interact because most of the mouse and windows commands are complex to recognize using the eyes or any of the body-features. "Control using Gaze" module is used to track and map the gaze to a mouse cursor on screen coordinates. This is used to select or concentrate on icon or object on the screen. Then "speech recognition" module is used to order commands considering this object such as drag, drop, open …etc.

For gaze tracking and mapping, we get 50 eye images for different subjects with different iris color; the proposed iris detection algorithm using Hough transform detected the iris correctly in 42 images, but for 3 images it detected another circle in the image because some features in the eye's image may classified as circle.

In 5 images "Hough transform" algorithm wasn't able to detect the iris, because it had a bright color but when we canceled the high contrast step for these images our implemented Hough transform detected 4 of them accurately. Thus we reached an accuracy of about 92% using our proposed iris detection methodology.

The implemented Hough transform function takes the following five parameters:

CircularHough_Grd(Image, radius range, gradient_thr, filter, multirad); Where:

Image is the eye image after applying the preprocessing on it.

Radius range: is the possible minimum and maximum radius of the iris to be searched.

Gradient_thr: is the threshold value on the gradient magnitude, applied before the voting process of the circular Hough transform to remove the uniform intensity. In other words, pixels with gradient magnitudes smaller than Gradient_thr are not considered in the computation.

Filter: is the radius of the filter used in the search of the local maxima in the accumulation array, to detect circles whose shapes are less perfect. **Multirad**: In case of concentric circles, multiple radii may be detected corresponding to a single center position. This argument sets the tolerance of picking up the likely radii values. It ranges from 0.1 to 1, where 0.1 corresponds to the largest tolerance, meaning more radii values will be detected, and 1 corresponds to the smallest tolerance, in which case only the "principal" radius will be picked up.

The following values for the function's parameters are used in our experiment:
CircularHough_Grd(Gaussian_image, [50 70], 8, 30, 0.7).

Figure 12 shows a screen shot for the output of the iris tracking procedure, where the "brightness" and "contrast" track bars are used to adjust the search window area. The circle bounding the iris shows the detected iris at a given frame.

Figure 13 shows the interface screen during the experiment of the "Control using Gaze" module.

Using our proposed gaze mapping technique we found that, the average gaze error for 10 subjects is about 1.1^o visual angle. This angular error is corresponding to about 15 pixels on screen coordinate according to our used screen size and resolution when the subject is sitting about 550 mm from the computer screen.

This error is acceptable since our system application, deals with screen icon that at least 25*25 pixels in size.

Also this average error has been calculated using the "reference mapping" technique and was found to be about 1.3^o which explains the added enhancement due to our proposed technique.

Fig. 12. Iris Tracking Output

For speech recognition module of the system, we tried the speech recognition on 10 persons on the words used in our system, such that each word trained 20 times.

The following average results were detected.

The Word	Correct times	Incorrect times
Click	17	3
Right	18	2
Open	18	2
Close	18	2
Maximize	16	4
Minimize	15	5
Drag	18	2
Drop	17	3
Back	16	4
Forward	18	2

TABLE 1. Speech Recognition Results

So the average accuracy of the implemented speech recognition module is about 86% but this accuracy dependent on the number of words and the surrounding environment.

If the recognized words are big different in sound the accuracy will increase.

5. CONCLUSION AND FUTURE WORK

This paper presents an architecture and design for human computer interface system used to control the mouse movements and events using gaze tracking and speech recognition.

The paper proposed optimum iris detection and tracking methodology integrated with an investigated technique for gaze estimation and mapping.

Our investigated technique for gaze estimation and mapping is a hybrid of the two 2D mapping techniques; "Direct mapping" technique and "Reference mapping" technique.

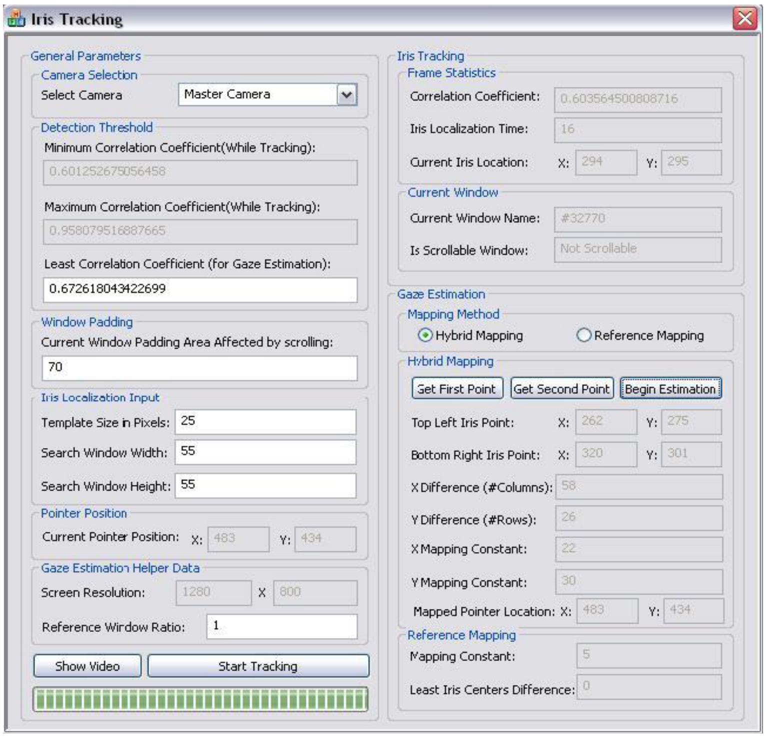


Fig. 13. Interface screen for the "Control using Gaze" experiment.

Our investigated technique achieved a high mapping accuracy and smoothing cursor transition with the ability to recover when the user moves away from the camera and returns again.

Also our technique required less computational and calibration effort than the other used 2D mapping techniques.

We extend our system ability by adding a speech commands recognition feature that enables the disabled user to execute other mouse and windows commands that are complex to recognize using the eyes.

Future work will include enhancing our investigated technique for gaze estimation and mapping by adding a head compensation model to overcome the problem of head movement. Recognizing additional mouse and windows commands using speech recognition will be done also in future work.

6. REFERENCES

[1]. H. EL-BAKRY, *"Human iris detection using fast cooperative modular neural nets and image decomposition"*, Machine Graphics & vision International Journal, Volume 11, issue 4 (2002).

[2]. J. DAUGMAN, *"New methods in iris recognition,"* IEEE Transactions on Systems, Man, and Cybernetics, Part B, 37(5), pp. 1167–1175, 2007.

[3]. N. BARZEGAR AND M. SHAHRAM MOIN, *"A new approach for iris localization in iris recognition systems,"* in Proceedings of the 6th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '08), pp. 516–523, Doha, Qatar, March-April, 2008.

[4]. N. BARZEGAR AND M. SHAHRAM MOIN, *"A New User Dependent Iris Recognition System Based on an Area Preserving Pointwise Level Set Segmentation Approach"*, EURASIP Journal on Advances in Signal Processing, 2009.

[5]. Y. DU, E. ARSLANTURK, , Z. ZHOU AND C. BELCHER, *"Video-Based Noncooperative Iris Image Segmentation"*, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, 41(1), PP. 64 - 74, April 2010.

[6]. J. CUI, L. MA, Y. WANG, T. TAN AND Z. SUN, *"A Fast and Robust Iris Localization Method Based on Texture Segmentation"*, Proc. of SPIE, Vol. 5404, pp. 401-408, 2004, USA.

[7]. J. TANG AND J. ZHANG, *"Eye Tracking Based on Grey Prediction"*, First International Workshop on Education Technology and Computer Science, 2009. ETCS '09.

[8]. S. JUST AND K. PEDERSEN, "*Circular Hough Transform"*, Aalborg University, Vision,

Graphics, and Interactive Systems, November 2007.

[9]. J. GAO, S. ZHANG, W. LU, *"Application of hough transform in eye tracking and targeting"*, 9th International Conference on Electronic Measurement & Instruments, 2009. ICEMI '09.

[10]. G. SARAVANA KUMAR AND S. RAVI, *"Eye detection and tracking using genetic algorithm"*, MASAUM journal of computing, 1(1), 2009.

[11]. K. TAN, D. KRIEGMAN, AND H. AHUJA, *"Appearance based eye gaz estimation"* in Proceedings of the IEEE Workshop on Applications of Computer Vision, 2002, pp. 191–195.

[12]. J. KANG, E. GUESTRIN, W. MACLEAN, M. EIZENMAN, *"Simplifying the Cross-Ratios Method of Point-of-Gaze Estimation",* 30th Canadian Medical Biology and Engineering Conference, June 16-19, Toronto, Canada, 2007.

[13]. R. VALENTI, J. STAIANO, N. SEBE AND T. GEVERS, *"Webcam-Based Visual Gaze Estimation"*, International Conference on Image Analysis and Processing, 2009

[14]. G. BOULAY, *"Eye Pose Tracking & Gaze Estimation",* Master of Engineering, done at the National Institute of Applied Sciences (INSA Lyon, France) in the Telecommunications, Services & Uses department, Aug 2008.

[15]. Z. ZHU AND Q. JI, *"Novel Eye Gaze Tracking Techniques Under Natural Head Movement"*, IEEE Trans. on biomedical engineering, 54(12), 2007.

[16]. "LC Technologies, Inc," *http://www.eyegaze.com*.

[17]. Z. ZHU AND Q. JI, "Eye and gaze tracking for interactive graphic display," Machine Vision and Applications, vol. 15, no. 3, pp. 139–148, 2004.

[18]. J. ZHU AND J. YANG, *"Subpixel eye gaze tracking"* in Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, Washington D.C., 2002, pp. 131–136.

[19]. C. MORIMOTO AND M. MIMICA, "*Eye gaze tracking techniques for interactive applications"* Computer Vision and Image Understanding, Special Issue on Eye Detection and Tracking, vol. 98, no. 1, pp. 4–24, 2005.

[20]. D. BEYMER AND M. FLICKNER, *"Eye gaze tracking using an active stereo head"* in Proceedings of the International Conference on Computer vision and Pattern Recognition, 2003.

[21]. S. SHIH AND J. LIU, *"A novel approach to 3-D gaze tracking using stereo cameras"* in IEEE Transactions on Syst. Man and Cybern., part B, 2004, vol. 34, pp. 234–245.

[22]. J. WANG, E. SUNG, AND R. VENKATESWARLU, *"Eye gaze estimation from a single image of one eye"* in Proceedings of Int. Conference on Computer Vision, 2003.

[23]. T. OHNO, N. MUKAWA, AND A. YOSHIKAWA , "*Freegaze: A gaze tracking system for everyday gaze interaction"* in Proceedings of the symposium on ETRA 2002, 2002.

[24]. C. MORIMOTO, A. AMIR, AND M. FLICKNER, *"Detecting eye position and gaze from a single camera and 2 light sources"* in Proceedings of the Int. Conference on Pattern Recognition, 2002.

[25]. Y. MATSUMOTO, T. OGASAWARA, AND A. ZELINSKY, *"Behavior recognition based on head pose and gaze direction measurement"* in Proceedings of 2000 IEEE/RSJ Int. Conference on Intelligent Robots and Systems, 2000.

[26]. E. GUESTRIN AND M. EIZENMAN, *"General theory of remote gaze estimation using the pupil center and corneal reflections"* IEEE Transactions on biomedical engineering, vol. 53, no. 6, pp. 1124–1133, 2006.

[27]. M. CHAU AND M. BETKE, *"Real Time Eye Tracking and Blink Detection with USB Cameras",* Computer Science Dept. Boston University, 2005.

[28]. M. BETKE, J. GIPS AND P. FLEMING, "The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access For People with Severe Disabilities", IEEE Transactions on Rehabilitation Engineering, June 19, 2008.

[29]. L. ZHANG, F. ZHOU, W. LI AND X. YANG, *"Human Computer Interaction Based on Nose Tracking",* J. Jako(Ed.), Human Computer interaction, PartIII, LNCS 4552, pp.769-778, © Springer-Verlag Berlin Heidelberg 2007.