Smart Scrolling based on Eye Tracking

Trevor Lewis
Computer Engineering Dept.
St. Francis Institute of Technology
Mumbai - 400103, India

Tobin Pereira
Computer Engineering Dept.
St. Francis Institute of Technology
Mumbai - 400103, India

Donita Almeida
Computer Engineering Dept.
St. Francis Institute of Technology
Mumbai - 400103, India

ABSTRACT
In this paper, we propose an application that is able to scroll the screen with the human face. The eyes are detected and tracked in real-time to use their actions for scrolling events. The basic strategy for detection is fast extraction of the face from the image in front of the screen with a Six-Segmented Rectangular (SSR) filter. In the SSR filter, Between-the-Eyes is selected as a face representative because of its characteristic is common to most people and is easily seen for a wide range of face orientation. After detecting the eyes, the eye movement is used to trigger the scrolling event based on our proposed algorithm. This has practical applications in document readers used on computers, laptops, tablets and smartphones.

Keywords
Six-Segmented Rectangular filter, SSR filter, face detection, eye detecting, eye tracking, scrolling, human-computer interaction

1. INTRODUCTION
Over the years computer technology has evolved drastically. Today computers and hand-held devices with high-speed processing power are available which are inexpensive and affordable to more and more people. This has enhanced various innovative applications. As technologies progressed, there has been an improvement in the field of image processing. What was once considered complex has now become possible because of the hardware availability to support it. This in turn led to the exploration in the field of human-computer interaction (HCI). In recent years, HCI has received attention and many innovative, effective and tireless ways to interact with the computer has been achieved

In this paper, we propose an application that is able to scroll the screen with the human face. The eyes are detected and tracked in real-time to use their actions for scrolling events. This application needs a capturing device with a moderate resolution and frame rate to feed the program with the video stream. The basic strategy for detection is fast extraction of the face from the video stream, in front of the screen, with a Six-Segmented Rectangular (SSR) filter. The SSR filter is a rectangle divided into 6 segments, operated by using the concept of bright-dark relation around Between-the-Eyes area. In the SSR filter, Between-the-Eyes is selected as a face representative because its characteristic is common to most people and is easily seen for a wide range of face orientation. After detecting the eyes, the eye movement is tracked and used to trigger the scrolling event.

This has practical applications in document readers used on computers, laptops, tablets and smartphones in order to scroll the text on the document reading devices very easily and effectively.

2. RELATED WORK
A number of research works has been published in the field of face detection and eye tracking in the past. In the early approaches, skin tone based segmentation (Yang & Waibel, 1996; Wu et al., 1999; Hsu et al., 2002; Terrillon & Akamatsu, 1999; Chai & Ngan, 1999) was utilized to detect facial features. However, it was not feasible to adapt the skin tone model because skin color information is sensitive to lighting conditions.

Viola and Jones proposed [1,3] machine learning approaches for visual object detection which was capable of processing images extremely rapidly and achieving high detection rates. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection. But, the machine learning approaches required training.

Chiang, Tai, Yang, Huang and Huang proposed [2] an eye detector based on the darkest part of the face. In their method, the color image is converted to equalized grey-level histogram. A threshold was applied to identify the eyes. Although this method is relatively fast, it fails to work with images of people with dark skin because their eyes are not always the darkest parts of their faces.

An eye tracking technique [4-6] based on Six-Segmented Rectangular (SSR) filter which operates on integral images was proposed. The Between-the-Eyes is selected as face representative because the characteristic of intensity and symmetry in eye region is common to most people and is used as robust cues to find possible eye pairs. In this technique, a rectangle, divided into six segments, is scanned throughout the face image. Then their bright-dark relations are tested if its center can be a candidate of Between-the-Eyes. The proposed method is robust to deal with illumination changes and glasses wearing. This system can run at real-time speed of 30 frames per second. Because of the high speed tracking, this technique can be utilized in different real-time human-computer interaction applications [8-10].

Hence, we used the SSR filter in our work as it was very fast and reliable and could be used in real time without any prior training.

3. SSR FILTER FOR FACE DETECTION

SSR filter is an acronym for Six-Segmented Rectangular filter. The SSR filter is used to detect the candidate of Between-the-Eyes. As the name suggests it is a filter consisting of six rectangles as show in Fig 1. Each block represents a sector and the sum of the pixels of that sector is represented by S_r.

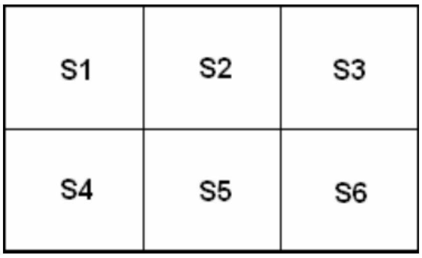

Fig 1: Six-Segmented Rectangular (SSR) Filter

SSR filter makes use of intermediate representation for image called "Integral Image" to detect face candidates. To find face candidates the SSR filter will be used in the following way.

At first, the SSR filter is scanned throughout the input image to calculate the integral image by using the equations:

$$s(x,y) = s(x,y-1) + i(x,y) \quad (1)$$

$$ii(x,y) = ii(x-1,y) + s(x,y) \quad (2)$$

Where s(x, y) is the cumulative row sum;
s(x,-1) = 0 and ii(-1, y) = 0

Using the integral image, the sum of pixels within rectangle D (S_r) can be computed at high speed with four array references as shown in Fig 2.

$$Sr = \big(ii\,(x,y) + ii\,(x-W, y-L)\big) - \big(ii\,(x-W, y) + ii\,(x, y-L)\big) \quad (3)$$

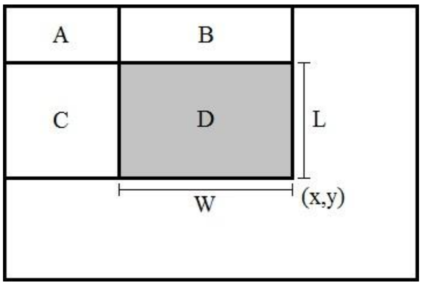

Fig 2: Integral Image

The SSR filter is used to detect the Between-the-Eyes based on two characteristics of face geometry.

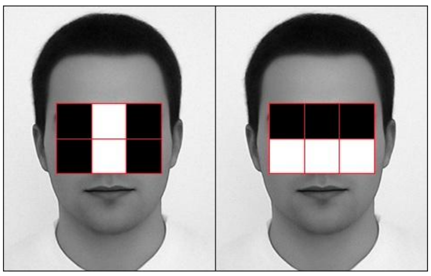

Fig 3 (a) **Fig 3 (b)**

1) The nose area (S_n) is brighter than the right and left eye area (S_{er} and S_{el}, respectively) as shown in Fig.3 (a), where

$$Sn = S2 + S5$$

$$Ser = S1 + S4$$

$$Sel = S3 + S6$$

Then,

$$Sn > Ser \quad (4)$$

$$Sn > Sel \quad (5)$$

2) The eye area (both eyes and eyebrows) (Se) is relatively darker than the cheekbone area (including nose) (Sc) as shown in Fig. 3 (b), where

$$Se = S1 + S2 + S3$$

$$Sc = S4 + S5 + S6$$

Then,

$$Se < Sc \quad (6)$$

The center of the rectangle can be a candidate for Between-the-Eyes if equations (4), (5) and (6) are satisfied.

The ideal location of the SSR filter is shown in Fig 4, where its center is considered as a face candidate.

Fig 4: Ideal Location of SSR Filter for Face Candidate

(x, y) is the location of the filter (upper left corner) and the plus sign is the center of the filter which is the face candidate. In this ideal position the eyes fall in sectors S1 and S3, while the nose falls in sector S5.

To determine the face candidates, the upper left corner of the SSR filter is placed on each pixel of the image where the filter falls entirely inside the bounds of the image. For each location (x, y) check equations (4), (5) and (6); if the conditions are fulfilled then the center of the filter will be considered as a face candidate. Eventually the candidates will be grouped in clusters as shown in Fig 5.

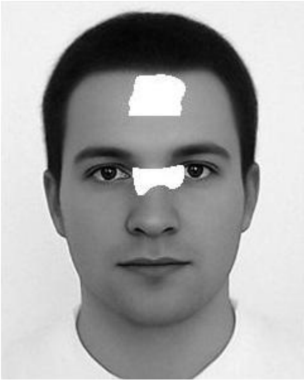

Fig 5: Clusters of Face Candidates

In order to find the most suitable filter size, for detection and tracking; the inherent fact that the human face is governed by proportions that define the different sizes and distances between facial features is used.

4. SELECTION OF FACE CANDIDATE

After determining the face candidates by using the SSR filter; groups of clusters as face candidates are formed. It is not feasible to check all the face candidates. The final candidate is chosen by taking the center of each cluster using the clustering algorithm

Clustering Algorithm

In the first pass, the image is scanned from the upper left corner to the lower right one.

For each face candidate fc:

- If all neighbors are not face candidates assign a new label to fc.
- If one of the neighbors is a face candidate assign its label to fc.
- If several neighbors are face candidates assign the label of one of them to fc and make a note that the labels are equal.

In the second pass, each group of same labels is assigned a unique label, so the final labels will become the clusters' labels. A threshold value is used that is relevant to the size of the currently used SSR filter, to eliminate clusters that are small.

The center of each cluster that is big enough is set with the following equations:

$$x = ([\sum x(i)])/n \qquad (7)$$

$$y = ([\sum y(i)])/n \qquad (8)$$

Where, i is the pixel from the cluster, n is the cluster's area.

This is shown in Fig 6 where the centers of the clusters that passed the threshold value, are marked as white dots.

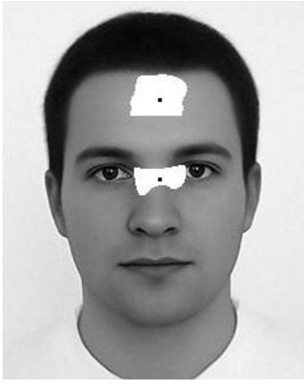

Fig 6: Clusters of Face Candidates and their Centers

5. EYE TRACKING

In order to track the eyes, the Between-the-Eyes is used as the reference as it is well tracked in each frame. While selecting the eyes, to avoid the effect of eyebrows, ignore a few pixels at the border. Also both eyes areas are assumed to be at 14×12 pixels on each side of face; so neglect a few pixels in the middle of Between-the-Eyes template as the nose area. To find the eye's template search for the darkest 5×5 region because the eye pupil is black. Then check the angle (A_e) between the located eyes and center of the filter using the expression:

$$115° < Ae < 180° \qquad (9)$$

In the next frame after locating the Between-the-Eyes, place the eyes' template at the same relative positions to the new Between-the-Eyes, assuming that the eyes have kept their relative locations to the Between-the-Eyes template.

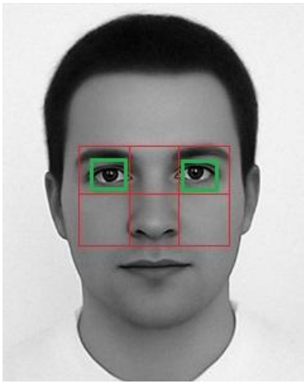

Fig 7: Eye Tracking

In our application we need to track only one eye (the right eye) instead of both the eyes. This will be used in the scrolling algorithm to trigger the scrolling event.

6. SCROLLING ALGORITHM

After tracking the eye, the image in the eyes' template is converted into binary in such a way that the eyeball is black and the other region is white as shown in Fig 8. For this an edge detection algorithm is used to find the edge of the eyeball.

Fig 8: Converting the Eye Template from Gray-scale to Binary

The system is calibrated the first time the user starts the application to get the eyeball looking at the center of the screen. This is done to draw a horizontal line such that eyeball is divided into two equal parts. This horizontal line is used in reference to the eyes' template in all the next frames.

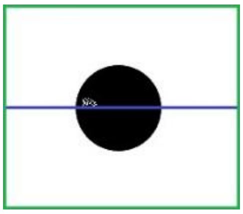

Fig 9: Eyeball looking at the Center of the Screen

The scrolling event takes place on the following two conditions:

1. If the user looks at the bottom of the screen, the number of black pixels in lower region (n_2) to the reference line is greater than number of pixels in upper region (n_1) by threshold T_1 then the text scrolls up.

   If $(n2 - n1) > T1$ then scroll up

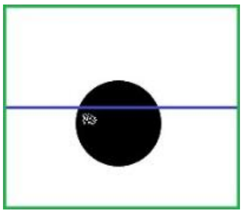

Fig 10: Eyeball looking at the Bottom of the Screen

2. If the user looks at the top of the screen, the number of black pixels in upper region (n_1) to the reference line is greater than number of pixels in lower region (n_2) by threshold T_2 then text scrolls down.

   If $(n1 - n2) > T2$ then scroll down

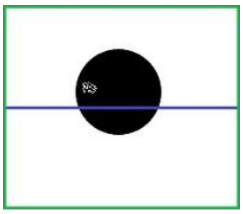

Fig 11: Eyeball looking at the Top of the Screen

7. FUTURE SCOPE

This application has scope for further enhancement in terms of highlighting the paragraph which is being read. It can also include the functionality to read the paragraph using text to speech converters; this would help the people with reading disabilities. Further work could include detection of head movement, to dim the screen when the user is looking away from the screen.

8. CONCLUSION

We have presented an approach for hands free scrolling on document readers using eye tracking. A Six-Segmented Rectangular (SSR) filter is used, which is scanned over the entire input image, to detect the Between-the-Eyes region. The Between-the-Eyes region is used for eye detection. After detecting the eyes, the eye movement is used to trigger the scrolling event based on our proposed algorithm. This has practical applications in document readers used on computers, laptops, tablets and smartphones.

9. REFERENCES

[1] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features", Computer Vision and Pattern Recognition, 2001.

[2] C. C. Chiang, W. K. Tai, M. T. Yang, Y. T. Huang and C. J. Huang, "A Novel Method for Detecting Lips, Eyes and Faces in Real Time", Real-Time Imaging, vol. 9, no.4, pp. 277- 287, Aug. 2003.

[3] P. Viola and M. Jones, "Robust Real-Time Face Detection Robust Real-Time Face Detection", International Journal of Computer Vision 57(2), 137–154, 2004.

[4] O. Sawettanusorn, Y. Senda, S. Kawato, N. Tetsutani and H. Yamauchi, "Detection of Face Representative Using Newly Proposed Filter", Journal of Signal Processing, vol. 8, no. 2, pp. 137-145, 2004.

[5] S. Kawato, N. Tetsutani and K. Hosaka, "Scale Adaptive Face Detection and Tracking in Real Time with SSR filter and Support Vector Machine". IEICE Transactions on Information and Systems, vol. 88, no. 12, pp. 2857-2863, Dec. 2005.

[6] Q. Wang, J. Yang and W. Yang, "Face Detection using Rectangle Features and SVM", International Journal of Intelligent Technology, vol. 1, no. 3, 2006.

[7] N. Bhoi and M. N. Mohanty, "Template Matching based Eye Detection in Facial Image", International Journal of Computer Applications (0975 – 8887), vol. 12, no. 5, Dec. 2010.

[8] A. Gupta, A. Rathi and Dr. Y. Radhika, "Hands-free PC Control - Controlling of Mouse Cursor using Eye movement", International Journal of Scientific and Research Publications, vol. 2, no. 4, Apr. 2012.

[9] S. Gupta, D. Dholakiya and S. Barve, "Real-Time Feature based Face Detection and Tracking I-Cursor", International Journal of Computer Applications (0975 – 8887), vol. 48, no. 24, Jun. 2012.

[10] M. Nehete, M. Lokhande and K. Ahire, "Design an Eye Tracking Mouse", International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 2, Feb. 2013.