Eye Tracking

Tamara Miller
University of Wisconsin, Madison

Vision REU Summer 2000

Abstract

This paper details an eye tracking system for use in determining a person's direction of gaze or focus of interest. The eyes are tracked over time with a large range of motion. Several well developed utilities of computer vision are used: color predicates for locating skin colored pixels and edge detection using a canny edge detector. Our image is compared to a template image in order to establish relative sizes between the image and the template. Then, a maximal function is used to determine placement of the image pupil. This system presents a fast and simple way of tracking eyes that is quite accurate.

1 Introduction

A system such as this could be used for a variety of purposes, a few examples being: human computer interaction through visual input to a mouse or determining regions of interest and focus of attention through detecting where the gaze lingers. This has often been touted as a possible use to advertisers who wish to determine which parts of their advertisements draw the consumers gaze and interest. An eye guided mouse could be used by disabled people for whom a hand guided mouse is not a possibility or for freeing up the hands of a person for other purposes.

Our eye tracker presents a system which is completely external, using a digital camcorder which is not intrusive to the user and which can be easily obtained, unlike infrared illumination or head mounted displays which are often used in other eye tracking systems. The person's head is free to move and does not need to be restrained, but can shift from side to side and turn as long as the person's eyes remain within the viewing region. The camera is placed 2 feet away from the person and zoomed in so that the person's eyes are the main focus of the image. The camera is assumed to be in a fixed location so that color predicates, sensitive to lighting conditions, need only be trained once for the lighting in the room. The eye region is not restricted, but is usually from the middle forehead to the bottom of the nose. This could be expanded for uses where the whole face is needed.

1.1 Previous work

Many methods of eye tracking have been tried. However many of these are intrusive to the user or require head mounted displays or restriction of the head in order to track the eyes accurately. We decided to use a less accurate method, but which is not intrusive and requires only a video camera, instead of specially designed equipment.

Ummm this obviously isn't finished. I need to find actual references.

1.2 Input Data

The movies were acquired using a digital camera placed about 2 feet from the user and zoomed in around the eye region from the middle forehead to the bottom of the nose. The system uses 640x480 size images taken at 30 fps. A data set of 25 videos was tested using a variety of skin and eye colors.

1.3 Methods Used from Other Research

A color predicate developed by Kjeldsen et al [1] was used for skin detection in the images. This is trained to mark subsets of RGB color space as skin color in order to recognize skin pixels in future images. Also used is an edge detector developed by Canny [2] which uses a

2 Algorithm

1. Start with a zoomed in video of a face.
2. Apply a skin color predicate to the image.
3. Find the 2 largest non-skin regions contained within the face.
4. Remove Eyebrows.
5. Apply edge detector to eye regions and find approximate size of eyes.
6. Compare this to the size of the eyes in a template image.
7. Find an approximate size of the pupil in the image.
8. Detect white and dark pixels in the eye image.
9. Slide our calculated pupil over white and dark map to place the pupil in the correct location.

3 Determining Eye Region

The first step in determining the direction of a person's gaze is determining the region which contains the two eyes. The image is zoomed in around the person's eyes, but needs to be segmented further. In order to separate this region from the rest of the face a color predicate is used that has been trained to detect pixels which are of skin color. Once this has been run over the image we are left with a black and white image of the pixels thought to be skin and those which are non-skin as seen in figure 1, skin shown in black and non-skin in white. Extracting the two largest connected non-skin regions which are surrounded by skin pixels leaves us with the two eye regions and gives us an approximate size for the eyes.

3.1 Separating Eyebrows

Often, dependent upon lighting conditions and facial features this will also leave us with a piece of the eyebrow included in the eye region (figure 2). When taking the two largest connected components we then retrieve the eyebrow, or part of the eyebrow along with the two eyes. Besides giving us a false region this also gives us incorrect dimensions for the eye. In order to segment the eyebrow from the eye, the number of non-skin pixels in a row part of the way down the region is calculated. Moving upwards, we continue calculating the number of non-skin pixels

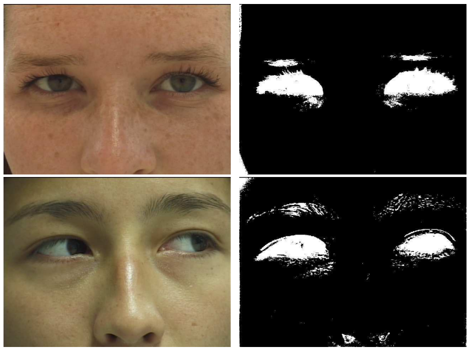

Figure 1: Original and skin detected images.

Figure 2: Skin detected image with eyebrows attached.

per row until either the number of non-skin pixels begins to increase or goes to zero. This determines where if at all the eyebrow is joined to the eye and allows us to separate the region into eye and eyebrow regions. If the eyebrow is not connected to the eye region at all as in figure 1 the width will go to zero and therefore give us back the same eye region. However, if as in figure 2 the eyebrow region is partly connected it will give us a line where the probable separation should occur and we can split the region into two parts, that of the eye and that of the brow. Now we have a tentative eye region which allows us to reduce the search space of possible eye positions and also gives an idea of where to apply a more accurate search.

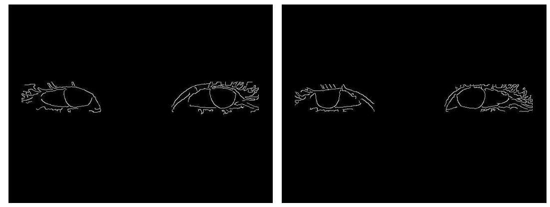

Figure 3: Canny edge detected image

4 Edge Detection

Using a Canny edge detector, we run this over the region of the image found by the skin detection to be the eye region. The Canny edge detector was used since it allows a specified sigma to be used and seems to give the best edge detected results. Since the skin detection has narrowed down the area which the edge detector is used, we cut down on the number of extraneous edges produced and get an edge map which clearly outlines the eyes and the pupils (figure 3). It also cuts down on the time which the canny detector takes, since it would take much longer to run across the whole image. From this edge map we can determine the approximate size of the image eyes by looking at the coordinates of the eyes, their left, and right corners and the highest and lowest points. Comparing this to a template image for which the size of the pupils is known, an approximate radius is then found for the image pupil. $\frac{Image\,Radius}{Template\,Radius} = \frac{Image\,Edge\,Height}{Template\,Edge\,Height}$.

5 Templates

One template is used consisting of an edge map of the template image and an image for which the pupils have been hand selected (figure 4). This allows us to have a sample image against which we can compare dimensions of the test image. Since we know the

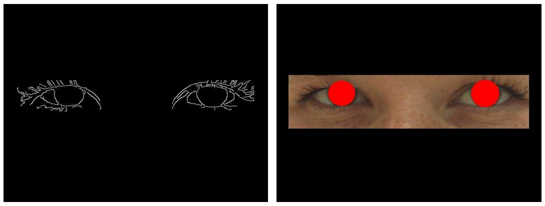

Figure 4: Template edge map and highlighted pupil.

size of the template pupil by comparing the template dimensions to those of the image we can calculate the approximate size of the pupil in our test image.

6 Maximizing function

6.1 Whites and Darks

To determine the white and dark regions of the eyes a relative color threshold is used. The average intensity of the eye region is first calculated. Then, pixels which have a magnitude which is small compared to the average intensity of pixels in the eye region are determined to be dark while pixels with larger magnitudes are labeled as whites. An example of this is shown in figure 4, where the whites of the eyes and outlying skin regions are marked as whites and dark regions are marked in a darker color as dark pixels.

6.2 Maximal Pupil Placement Function

Next, we take our circle whose radius was calculated above and slide it across the eye region, trying to maximize the Pupil Placement Function = (Number of Dark pixels covered by the circle - Number of White pixels covered). This is done in two steps. First for a quick approximation every 10 pixels across and down are tried, and then once the maximal region is found each pixel in the resulting 10x10 region

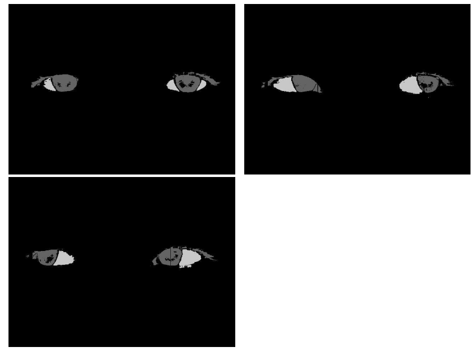

Figure 5: Whites and darks of eyes.

is tested for maximality. During this second step, the radius is also allowed to grow and shrink in order to get a better fit for the pupil region. If a larger or smaller radius than previously calculated gives us a larger value for the Pupil Placement Function then that radius is used.

7 Accuracy Improvements

Sometimes the maximal function value is not necessarily the optimal placement of the pupil, due to shadowing or other areas being falsely labelled as dark pixels. Two methods are used to improve the accuracy of the pupil position.

7.1 Dependency of eyes

A person's pupils are not independent, which means that they will always be the same distance apart in moving as for example the distance between the left corners of their eyes (assuming the eyes are not in a crossed position). Therefore, we can use this distance as determined from the edge map in order to find a better placement. Therefore, we calculate the distance between the left corners of each eye in our edge map and assume that the centers of the pupils

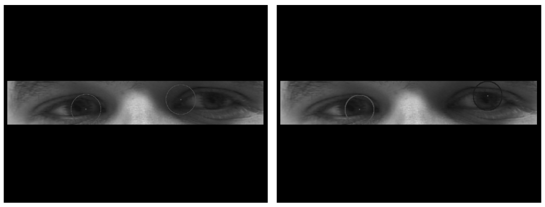

Figure 6: old and new image with dependency corrected.

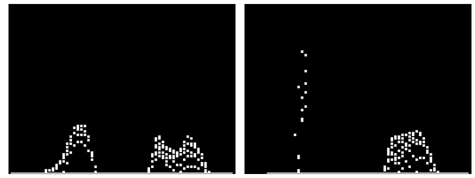

Figure 7: Graph of function values vs. Column. Note two peaks in right eye of left figure.

will have a distance apart close to this value. When choosing the placement, a position with a large function value, but also with a distance between the two pupils which is close to the distance between the eye corners is looked for (figure 5).

7.2 Maximal graph

Graphing the function values at each column in the eye region, we can detect peaks of larger values as we run across the image. These peaks will give us possible placements of the pupil centers. Those images with multiple peaks such as figure 6, often lead to poor placement of the pupil. Multiple peaks usually means there was a region of shadow in the image.

Shadow makes pixels in our image darker than the average intensity and therefore leads to some white pixels being incorrectly labeled as darks. So, when choosing a placement of the pupil shadow can lead to a poor placement of the pupil if we were to simply use the largest value obtained by our Pupil Placement Function. Therefore, we can instead use this curve of function values in order to decide whether the pupil should be placed at a position other than the maximal one. An eye with more than one peak is considered differently than one with only one peak. When multiple peaks appear each peak placement point is tried and that with the highest confidence level is chosen to be the correct placement.

8 Confidence levels

In order to determine the confidence level of a placement, the pupil's circumference is matched with the edge map and the pixels around the eye are also looked at. Two things must appear in order to assign a pupil a high confidence: there must be a good match with the edges and there must a large number of white pixels around the pupil. Otherwise, the placement is given a low confidence level.

When determining whether a placement other than maximal should be used all peak placements are considered and that which has the best edge match and largest number of surrounding white pixels is chosen.

9 Conclusion And Future Work

This method could be expanded to a more robust tracker. One possible future step would be to utilize the shape of the eyes as in Kanade's upper face tracking method [3]. It could also be further enlarged to tracking a person's gaze using direction and angle of the head placement and to use more clues or facial features to determine position of the head.

Ummm.....

10 References

[1] Rick Kjedlsen and John Kendor "Finding Skin in Color Images." Face and Gesture Recognition, pp312-317, 1996.

[2] J. Canny "A Computational Approach to Edge Detection." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov. 1986

[3] Kanade's upper facial feature tracking. I forget the name at the moment.