Proceedings of The National Conference
On Undergraduate Research (NCUR) 2006
The University of North Carolina at Asheville
Asheville, North Carolina
April 6 – 8, 2006

Eye Feature Detection Towards Automatic Strabismus Screening

Kenneth Allen, Khanh Nguyen
Department of Computer Science
Gettysburg College
300 North Washington Street
Gettysburg, Pennsylvania 17325 USA

Faculty Advisor: Dr. Sunghee Kim

Abstract

Strabismus is an eye defect which, if left untreated, can result in inability to attain binocular vision or blindness in the affected eye. Early detection is critical for successful treatment. Current methods for detection involve going to an eye specialist or going to a lab with specialized equipment and lighting setup. These options can be expensive and thus aren't always available to all people. The purpose of this project was to determine the feasibility of a software-only solution that could be deployed to photo studios as a low-cost early screening mechanism. The eventual goal is to be able to detect signs of strabismus from a single high resolution portrait photograph of an infant. The strabismus screening algorithm consists of two phases. In the first phase the locations of certain eye features such as pupil center, eye corners, specular highlights, etc are determined. In the second phase the locations of the highlights in both eyes are examined in relation to the pupil centers. The success of the project depends heavily on the accuracy of eye feature detection and is the focus of our work. Several existing computer vision algorithms for detecting various eye features were investigated and the ones that would handle the requirements best for implementation were chosen. So far a good degree of detection accuracy (roughly 80%) has been achieved and currently the eye detection and corner detection routines are being fine tuned to improve detection accuracy even more.

1. Introduction

Strabismus is an eye defect characterized by a lack of coordination between the eyes, causing the eyes to look in two different directions and thus be unable to fixate on a single image. If children with strabismus are left untreated, the brain may learn to ignore the input from the weaker eye. Consequently the children may suffer permanent loss of vision in the ignored eye. Newborn babies often appear cross eyed but this does not mean that they have strabismus. Unfortunately this common condition means that by the time a child with strabismus is taken to an eye specialist muscles in the weaker eye may have weakened to such a degree that the possibilities for treatment may be limited. Early diagnosis makes successful treatments possible and hence is crucial for normal development of affected children. The goal of the work presented in this paper was to develop and deploy a software-only method for strabismus testing to photo studios as a free pre-screening system, so that cases can be caught while the condition can still be treated.

The corneal light reflex test (Hirschberg test) assesses ocular alignment to detect strabismus. This method uses the positions of the corneal light reflection from a light in each eye when the child fixates on the light and follows it. Displacement of the corneal light reflex in one eye could signal strabismus. This procedure is currently performed manually by a specialist but it could be applied in a software solution if the locations of different eye features in a frontal photograph of the head are found. This paper focuses on the process of detecting these eye features from a single portrait photo of infants and young children.

Though our program is intended to be used in photo studios and thus will be operating on professional quality studio photographs some assumptions (described later) were made about the target photograph's characteristics. However, any photograph that possesses these characteristics is suitable as input.

Before the eye feature detection can begin the search region in the photo is narrowed down by locating the bounds of the face and then locating the bounds of the eye region within that face. It is within the eye region that the necessary eye features can be located. The features to be detected are the pupil centers, the centers of a matched pair of highlights and the corners of the eyes. In the next sections, the methods for detecting these five features are discussed in detail

2. Face Detection

The goal of the face detection in this project was only to provide a focus-of-attention mechanism for the eye-detection algorithm used later. Four different methods were evaluated for determining the face region. The first was a neural network based face detection package described by Jones and Rehg[2]. This method did not provide acceptable results on infant faces, and the time and resources to train our own neural network were not available.

The next two methods, mixture model and histogram-based color detection, both described in Rowley et al.[2], were also found to be inadequate due to unavailability of the dataset large enough to build a quality model. Finally the Jones-Viola face detector provided in the OpenCV[4] library was considered. This detector uses a haar-like cascade classifier.

After evaluating the performance of all four face classifiers included with OpenCV, the “frontalface_alt” classifier was found to be the best. Detection is done in a two-stage process. The detector is run once, then it is run again with lower accuracy if no faces are detected. Because the input photos are studio pictures, with the face as the dominant feature, a large rectangle size was used in the detection. This greatly increases the speed of the operation without affecting accuracy (in these types of pictures). The face can be safely assumed to be the main feature so false positives could be easily culled by discarding matches smaller than 25% of the image. This number could potentially be made larger, but it was decided to err on the side of caution.

One unusual characteristic exhibited by this classifier is that it generally fails to detect the face in pictures that had been cropped fairly closely around the face as shown in Figure 1. Because most studio pictures were not expected to be cut this close, this behavior was deemed acceptable.

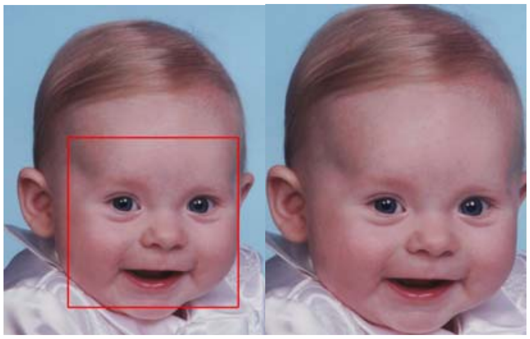

Figure 1. Cropped faces are generally not detected.

This method provided good results, accurately detecting the face in all but one of the non-cropped faces in our dataset of 30 photos. This face detection step can also be omitted by having the photographer specify the face region manually.

3. Eye Regions Detection

After the facial region is isolated the straightforward work of finding the eye regions can begin. Given the face image, the candidates for eye locations can be extracted by an adaptive binary thresholding algorithm. After experimenting with several common threshold methods, it was decided to threshold the image at the value calculated by the method of minimizing fuzziness[4]. The returned threshold value then was enhanced, based on the knowledge about the facial features, by comparing the area of black pixels with that of white pixels. After this point, there are generally two to five candidate areas left. The other thresholding techniques that were examined use the mean of pixel values, the black percentage (the percentage of black pixels in the candidate region), or iterative selection.

Most of the experimented methods produced similar results, so the less complex method, algorithmically and implementation-wise, was chosen.

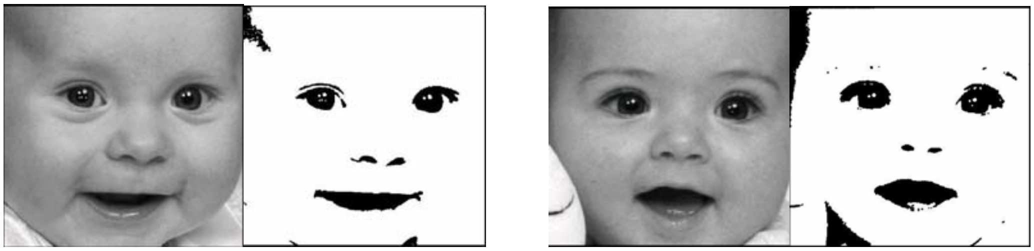

Figure 2. Typical results from adaptive binary thresholding.

Next, the best pairs are determined using the geometrical properties of the typical human eyes. Under the assumption that the eyes should be in the top half of the image, the elimination process can filter out the mouth and the nose. All the candidate areas near the boundary of the image also are eliminated. Notice that the line passing through the center of the eyes has a strong horizontal bias, normally parallel with the x-axis. Also the areas of the two eye regions are approximately equal. Based on these two observations, the algorithm locates the two eye regions in all our test images[7].

Once the positions of the eyes are approximately located, a region growing algorithm was used to find the bounding boxes for the eyes. Next the image is thresholded again with a lower value and the black areas resulting from both thresholds are grouped together.

The program approximates the upper and lower bounds of the bounding boxes by looking for the top and bottom black pixels in each eye. To make sure that the eyes are completely contained in the regions, a small adjustment is added to top and bottom y-axis.

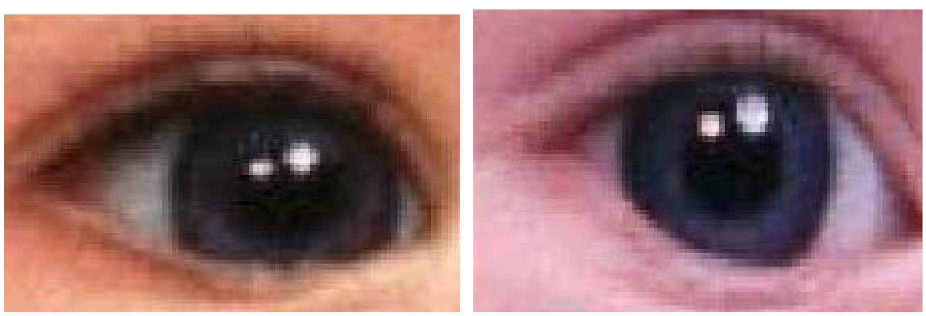

Figure 3. Typical eye regions.

To find the left and right boundaries of the eyes, the Sobel filter first is applied for edge detection. Assuming that the left eye region will most likely to be in the left half of the image, while the right eye is in the right half. For the left eye, we first scan from the middle of the picture to the left until we encounter the first white pixel, that is considered the right boundary line, then continue scanning until we find three continuous black pixels, that is the left boundary line. The same process is repeated for the right eye. The success of this task depends very much on the results of the Sobel Filter. If the image has a lot of noise, the detection of the eye corners, and consequently the iris and pupil, becomes very difficult.

4. Iris Detection

In this phase, the Hough Transformation algorithm is applied to the eye region twice to locate irises. In the process, the Hough Transformation is used two times. When there is a highlight present in the iris, its location usually has the highest pixel value in the hot spot image. This leads to an incorrect iris center detection. Therefore, the first Hough Transformation is executed to locate the highlight and dim it so that it won’t affect the detection process.

Once the second Hough Transformation produces its hot spots image, the candidates for the iris center are sought in a circular window search whose center is the highlight. This is because in most cases the highlights are quite close to the true iris centers. For each iris center candidate, we try to fit the best circle. The best of those detected circles will be the iris. Detecting the circle for each iris center candidate is the most difficult part of the implementation since we have to deal with noise and many other uncertainties. The circle fitting function looks for boundary points based on pixel red channels because, as observed by Vezhnevets and Degtiareva,[6] the iris and eyes tend to have lower pixel values in the red channel than the skin. The boundary points detected on the sides of a candidate circle score more than those on the upper and lower parts. The reason is that the upper and lower parts of an iris are often hidden by the eyelashes. To make a better approximation, a point on a candidate circle is selected. If this point and the point opposite it are both on the boundary it is likely that the candidate circle actually corresponds to the iris. The largest of the candidate circles is then chosen because we have observed that the largest one is usually the best match.

Detecting the circle for each iris center candidate is the most difficult part since we have to deal with noise and many other uncertainties. For instance, one of the challenges is that there are pictures whose irises are partially occluded by the eyelids or the eyelashes, making it very difficult for the Hough Transformation to work well. Another challenge is that sometimes the boundary between the irises and the skin is difficult to recognize because the differences are not significant enough. This happens more often with babies with lighter colored eyes.

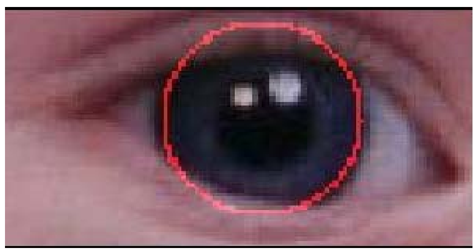

Figure 4. A successfully detected iris.

5. Specular Highlight (Corneal Light Reflection) Detection

The process of detecting the highlights in the eyes is a straightforward one. Pixels belonging to highlights, when examined in HSB/HSV color space, are usually high in brightness and low in saturation. The eye region (already determined in Section 3) is scanned for the brightest pixel with saturation below a certain threshold. This pixel is region-grown to determine its extents, and a bounding rectangle is found for this region. The pixels of the region are marked on a binary mask. Pixels in subsequent scans are ignored if their corresponding mask pixel is so marked. On occasion the whites of the eyes are detected, so if a detected region's bounding rectangle is larger than a quarter of the eye region, it is ignored. This process repeats until all pixels are either below the minimum accepted brightness or have their corresponding mask pixels marked. This results in a set of possible highlights for one eye. The whole sequence is repeated for the opposing eye and the process of selecting a matched set can begin.

A match cannot be determined merely by comparing the positions (relative to the eye region) of the highlights in question because in children afflicted with strabismus the highlights can at times be in somewhat different locations. It was noted however, that a matched set of highlights were very similar in size, even when the head is at an angle. Therefore, each highlight in one eye is compared to each highlight in the other, and the difference in size is noted. The pair with the smallest size difference is assumed to be the best match. At times however two highlights may have a very small difference in dimension only. This would seem to be a better match than two highlights which differed slightly in both dimensions, but a 1D difference indicates that the highlights are of a different shape or that one of them is covered somewhat by the eyelid. Neither of those conditions is acceptable in a matched set so the algorithm was designed to prioritize proportionally differing highlights, i.e. those of the same shape.

Figure 5. Partially occluded highlights (as in the left eye) are unacceptable for matching.

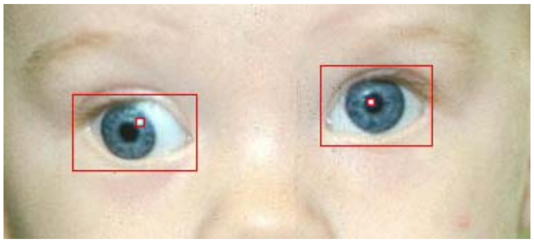

Figure 6. An instance in which the highlights are not in proportional positions relative to the eye

As the highlights tend to be circular and the bounding rectangles are usually very tightly bound to them, it was decided the finding the center of the rectangle would be sufficient for finding the center of the highlight. This method is also far less computationally expensive than fitting a circle and finding its center, thus freeing processor cycles for use in the more computationally intensive work done in the other half of the project.

This detection method proved to be highly accurate in all images tested and quite robust in the face of image noise, even though noise should be minimal in the studio photos this it is meant to be used on.

6. Eye Corner Detection

The positions of the corners of the eyes are required to determine the angle of the eye. The angles of the eyes are required to make the correlation between the iris and highlights meaningful. The corners of the eyes, while clearly visible to humans, are quite difficult to distinguish programmatically. While accuracy is important in this operation, it is more important for the corners to be identified consistently between both eyes.

The image is first smoothed with a Gaussian blur to remove noise. The resulting image is then run through a canny edge detector. This finds the primary contours of the eyes. In addition, a set of gradient points is built where each point represents a pixel where there is an appreciable increase or decrease in luminance in the picture. Finally, a filter based on luminance, saturation, and the red channel is run on the eye region and the resulting area is grown outwards to the nearest edge (as discovered in the first step).

It was found that the corner of the eye was nearly always found at a point on one of the detected edges that bordered on the grown region (corresponding roughly to the whites of the eyes) and surrounded by a high concentration of gradient points. Each of these three features is assigned a weight and the point on each side of the eye with the highest weight is chosen as the corner.

As the distance between the point in question and the outer edge of the eye increases a counter-weight does as well, to discourage the selection of points in the strong edges and gradients near the highlights. The counter-weight is given by formula (1).

$$\text{distance}^3 / 2 \qquad (1)$$

The corner detection is quite consistent across the photos in our test set, with only a few false positives. Usually the feature detected in the false positives is the corner of the fold above the eyelid. The tendency to detect this fold is unfortunately not consistent between eyes, or it would not be such a problem. Fortunately, however, the distance between the detected point and the actual corner is quite small.

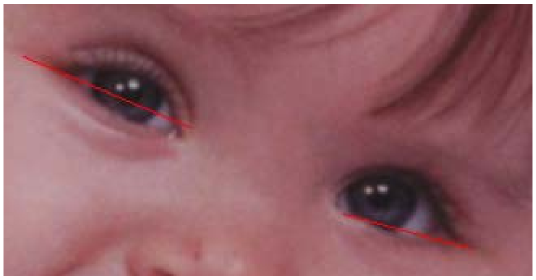

Figure 7. The right corner of the left eyelid is detected, rather than the actual corner

Figure 8. Typical results of the corner detector

The biggest problem with this method for corner detection is that it is not very robust across differences in lighting, and occasionally across differences in skin color. Because the pictures will be taken in a studio under fairly consistent lighting, this will not be as big of a problem as it could be but it is still an issue that needs to be addressed.

Other possible methods for corner detection[3] rely on contour fitting using the iris as a point of reference and would be a more robust method. However the algorithm for detecting the iris was being developed concurrently with this one and, because of time constraints, it was decided to pursue a solution that did not depend on the iris.

7. Future Work

So far the various steps in the process have been running separately in test programs. Our next task is to fully incorporate all the separate code into one integrated system. This will be the prototype for the application that will eventually be deployed, so simplicity and ease of use will be a major focus in creating the user interface.

More importantly, even though we have developed methods for getting the information required by the strabismus screening algorithm, we have yet to implement the algorithm itself. Fortunately it is mostly a matter of simple geometry and will be fairly easy to implement, once the foundation code has been integrated into the prototype. The screening algorithm examines the relative distance and location of the specular highlight in relation to the location of the pupil/iris center. If the data (placement, angle, and distance) from the left eye deviates significantly from the data from the right eye, the child in the photograph may have strabismus.

In addition, a new method contour fitting method[6] is being examined as a replacement for the current eye corner and iris detection algorithms. It is believed that contour fitting will produce better, more consistent results than the current algorithms.

8. Conclusion

The results of the work are very favorable in that they are reasonably accurate across a wide range of test cases and run acceptably fast. This gives reason for optimism about the success of the overall project. It seems that a pure software solution for the detection of strabismus is quite possible, and will merely require additional effort to produce the accuracy and consistency needed to make it a reality.

9. Acknowledgment

The work was originally brought to our attention by Drs. Al Yonas, Simon Christiansen, and Victoria Interrante at the University of Minnesota. We thank them also for providing us with the test dataset. We would also like to gratefully acknowledge Gettysburg College for providing a faculty development grant for this project.

References

1. Michael J. Jones and James M. Rehg. Statistical Color Models with Application to Skin Detection. *International Journal for Computer Vision*

2. Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20, number 1, pages 23-38, January 1998.

3. http://www.intel.com/technology/computing/opencv/index.htm

4. Parker, Jim R. Algorithms for image processing and computer vision. 1st ed. : Wiley computer publishing, 1996.

5. Parker, Jim R. Practical computer vision using C. 1st ed. : Wiley computer publishing, 1993.

6. Vezhnevets V., Degtiareva A., *"Robust and Accurate Eye Contour Extraction"*. Proc. Graphicon-2003, pp. 81-84, Moscow, Russia, September 2003.

7. Zhang and Lenders, “Knowledge-based eye detection for human face recognition” in *Conf. on Knowledge-based Intelligent Systems and Allied Technologies*,, 2000