Subpixel Eye Gaze Tracking

Jie Zhu, Jie Yang
Interactive Systems Laboratory
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{jzhu, yang+}@cs.cmu.edu

Abstract

This paper addresses the accuracy problem of an eye gaze tracking system. We first analyze the technical barrier for a gaze tracking system to achieve desired accuracy, and then propose a subpixel tracking method to break the barrier. We present new algorithms for detecting the inner eye corner and the center of an iris in subpixel accuracy, and we apply these new methods in developing a real-time gaze tracking system. The experimental results indicate that the new methods achieve an average accuracy to within 1.4° using normal eye image resolutions.

1. Introduction

Human gaze provides several functions in communication, such as: giving cues of people's interest and attention, facilitating turn-taking during conversations, giving reference cues by looking at an object or person, and indicating interpersonal cues such as friendliness or defensiveness [22]. It also has been an important modality in many human computer interaction applications. Most commercial gaze tracking products, however, require users to wear cumbersome head-mounted equipment.

Much research has been directed to non-intrusive gaze tracking in the past few years [4, 9, 11, 13, 15, 16, 19]. These systems fall into two categories: analytical approaches that compute eye gaze by analyzing locations of certain facial features; and neural network approaches that use eye images as inputs. The goal is to relieve user from any head-mounted equipments; and to allow user unrestricted freedom of movement. Major challenges in developing a real-time gaze tracking system include: tracking speed, accuracy, and robustness.

In an analytical approach, the accuracy of gaze tracking greatly depends upon the resolution of the eye images. However, imposing a zoom-in constraint would limit a user's movement. If a user's freedom of movement is to be preserved, we currently must settle for low resolution eye images. Low resolution eye images are a major obstacle in achieving a high accuracy of gaze tracking in an analytical approach. That is why all reported high precision non-intrusive gaze tracking results are achieved only through neural network methods, which can utilize the intensity information of the low-resolution eye images. However, neural network based methods suffer from a lack of robustness against environmental changes.

After carefully investigating neural network approaches, we have discovered that not all the intensity information is being used by neural network based methods. What a neural network based method really benefits from is the pixel intensity information along edges within the eye images. In this paper, we present an analytical method that uses edges and local patterns to obtain detection of eye features with subpixel precision. We developed two novel detection algorithms that can robustly detect the inner eye corner and iris edges, both with subpixel accuracy. Unlike previous approaches, which use Hough transform to fit the iris edge with a circle, we employ an efficient and robust ellipse fitting algorithm to find the center of the iris from the detected iris edge points. A simple, but accurate and robust estimation scheme is used to calculate the gaze direction from the tracked features. All of these new approaches contribute to a robust and accurate real-time eye gaze tracking system. Experiments demonstrate that the new method robustly keeps on average tracking error under 1.4° using normal resolution eye images, which is superior to any reported results obtained through any of the other methods. In addition, the proposed algorithm is computationally efficient. It can provide real-time performance with a PIII 900MHz PC.

2. Accuracy of Gaze Tracking

2.1 The desired accuracy

Gaze is defined as the direction the eyes are pointing in space. Some parameters about eye anatomy and movement are listed in Table 1 [21]. There are two kinds of photoreceptors in the retina of human eyes: rods that are responsible for vision in dim light, and cones that are responsible for high-acuity vision and color vision in moderate or bright light. Fovea is a small cone-concentrated area in the retina responsible for high-acuity vision with a diameter of about 5.2°. An even smaller rod-free area (with a diameter of 1.7°) in the fovea has even higher spatial resolution. When an eye is "gazing" a direction, what actually happens is that the eyeball rotates to make light rays from that direction fall on the fovea, or even on the rod-free fovea. Therefore, 1.7° should be a pretty good accuracy benchmark for a gaze tracking system.

Table 1. Some parameters of human eyes

Size of one eye's visual field	~135°×~160°
Range of eyeball rotation	~70°×~70°
Diameter of the fovea	~5.2°
Diameter of the rod-free fovea	1.7°~2°
Radius of the eyeball	1.3cm

2.2 The technical barrier

A non-intrusive video based gaze tracking system, however, has difficulty achieving the desired accuracy. This difficulty mainly comes from the insufficient resolution of the eye images. Suppose that the size of an eye image is 40×25, which is a typical situation, the range of the iris or pupil movement is then about 20×20 pixels. At pixel level, the best possible resolution in gaze tracking is about 70°/20 ≈ 3.5° in both azimuth and elevation angles. More precisely, let's consider a method of calculating the gaze direction from the relative positions of the iris and the center of the eyeball, or equivalently any facial features that serve as a stationary reference point in calculations. Figure 1 shows the structure of an eye. In Figure 1, $|l| \in [0,10]$ pixels, $r \approx 20$ pixels. The smallest pixel level unit of l is $|\Delta l| = 1$ pixel. Then,

$$\theta = \arcsin {l}/{r} \tag{1}$$

$$\Delta\theta \approx \frac{\partial\theta}{\partial l}\Delta l + \frac{\partial\theta}{\partial r}\Delta r = \frac{1}{r\sqrt{1-\left({l}/{r}\right)^2}}\Delta l - \frac{l}{r^2\sqrt{1-\left({l}/{r}\right)^2}}\Delta r \tag{2}$$

Suppose $\Delta r = 0$, then for $|l| \in [0,10]$ pixel, $|\Delta\theta| = [2.9^\circ, 3.3^\circ]$. Thus, the smallest unit of θ, i.e. the resolution of gaze direction is about 3°. This is consistent with the previous intuitive reasoning. If the iris detection has a small error of 1 pixel, it will generate a tracking error of about 3° of gaze direction.

In addition, other factors also decrease the accuracy of a gaze tracking system, such as an intrinsic error in the construction of an eye model. Some researchers have modeled the eyeball as a sphere [9, 11]. This can bring in errors in Δr in equation (2). For example, this model can cause up to a 6° error in gaze estimation if there is a 0.2cm error in the estimation of the eyeball radius.

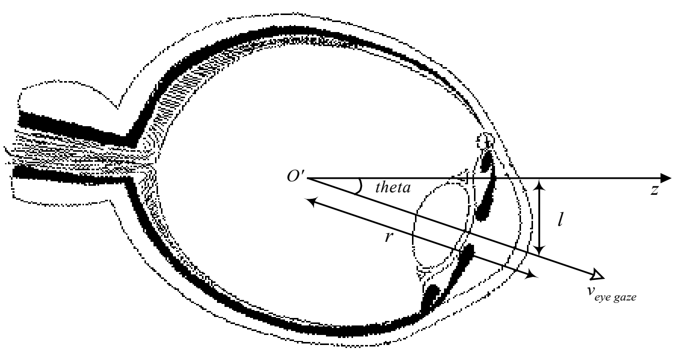


Figure 1. Eye structure for error analysis.

Many researchers tackled this resolution problem by using a zoomed-in camera or putting the camera very close to the user. These approaches require the user to fix his or her head position to keep the eyes in the camera's narrowed field of view.

2.3 Approach

While analytical gaze tracking systems suffer problems of inaccuracy, neural network based approaches have demonstrated good accuracy. Excellent accuracy in offline evaluations, such as 1.7°, has been reported [4]. A neural network based approach takes eye images as input and learns patterns from examples. The reason they can achieve a higher accuracy is that they take advantage of the pixel intensity information of the whole eye image. Therefore, the possible variations are not 20×20 different iris center positions, but $256^{40\times25}$ different images, assuming an 8-bit gray-scale eye image (of course, the number of actual possible variations are far less than this, which we discuss later). Although neural network systems have achieved higher accuracy in offline evaluations, few systems have been applied to real applications, because the trained neural network is too sensitive to changes in users, lighting conditions, and even changes within the user. What caused these problems is the very property that enables them to overcome the resolution barrier – the utilization of the pixel intensity information. Changes other than gaze direction can often have far greater effects on the pixel intensities. Furthermore, it also cannot distinguish pixels in "useful" areas like the iris from those "distracting" areas, such as eyelid.

Although a neural network based approach has inherent problems for real applications, its property of coding more information can be used to improve the accuracy of an analytical approach. We would like to investigate in more detail how a neural network encodes the information. Figure 2 shows two eye images (a), (b), and their difference (c). The gaze directions of these two images are slightly different. When we send these two images into a neural network, what makes the output different are mainly those pixels that have large differences. That is, although the neural network takes in the whole images as inputs, what really contribute to gaze estimation are only a few pixels. These pixels, as shown in Figure 2, are mainly at the edges of the iris and the sclera. The other pixels, such as the pixels in the eyelid areas, can only bring trouble and make the system unstable. Inspired by this observation, we propose to estimate gaze based on local features and patterns instead of single pixels. In this way, an analytical gaze tracking system can obtain subpixel accuracy by encoding the intensity of edge pixels.

(a) (b) (c)

Figure 2. Two eye images and their difference.

An analytical gaze estimation algorithm employs at least one moving point on the eyeball and one stationary facial reference point to compute a gaze direction. Many researchers use the center of an iris as the moving point, and eye corner or some special marks on the face as the reference point. These systems can achieve, at the best, gaze estimation accuracy at a pixel level. Instead of using points, we use local patterns and local features to achieve gaze estimation accuracy at a subpixel level. We choose the iris as the moving part because it has a high contrast with the sclera, and the inner eye corner as the reference point because it is the most stable feature in a face and relatively insensitive to facial expressions [18]. As discussed earlier, to achieve subpixel accuracy in gaze estimation, both the inner eye corner and the iris center must have subpixel accuracy. Because the shape and intensity distribution of an iris within the eye image changes dramatically while the gaze changes, we track its edge points and then fit them with a dynamic ellipse to find the center of the iris. We use a local feature filter to detect the eye corner with subpixel accuracy. We discuss these algorithms in detail in next section.

3. Gaze Tracking at Subpixel Accuracy

Subpixel techniques have been developed in various applications in image processing and computer vision community: image registration [2, 17], stereo vision [20], edge and corner detection [1, 3, 6, 7, 10, 12], and feature detection [5].

Among previous work, the closest related includes: subpixel edge and corner detection and feature detection. Unfortunately, the previous work cannot be applied directly to a real-time gaze tracking system because of the following reasons:

- **Computationally expensive**: Almost all algorithms require heavy computation to process one image, which is not possible in a real-time application. For example, the corner detection algorithm in [1] requires computation of several levels of auto-correlation type operations and Laplacians.
- **Inappropriate models**: The popular paraboloid fitting of correlation technique in feature tracking [5] works poorly for asymmetric features, especially when the template is small. Almost all subpixel edge detection algorithms assume a straight edge or other edge models, which are obviously inappropriate for a gaze tracking task [6, 7, 10, 12].
- **Insufficient data**: Some previous algorithms developed for high resolution images, such as curvature edge models [10], perform poorly in low resolution eye images because insufficient iris edge points are available.

In order to solve these problems, we have developed a novel eye corner tracking algorithm and a local intensity information based iris edge detection algorithm. In the following subsections, we first introduce these tracking algorithms in detail and then describe gaze estimation based on these tracked features.

3.1 Subpixel eye corner tracking

Although we are facing many difficulties common in subpixel feature tracking and real-time development, the eye corner tracking problem also has a characteristic that we can leverage: the shape and orientation of the eye corner is very stable across different people and facial expressions. We do not have to use a general corner detector that looks for corners in every orientation. Having taken into account all these considerations, we propose the following eye corner tracking algorithm:

1. *Define the eye corner search area as a small section (SA) in the image with the prior eye location as a starting point.*
2. *Perform 2-dimentional convolution on SA with a preset corner filter (CF) shown in Figure 3:*
   $FO = SA \otimes CF$ *(3)*
3. *Define a smaller search area (SA') as the area around the pixel with the highest FO value. Define the convolution result in this area as FO'.*
4. *Apply bi-cubic interpolation [8] to FO' to get the convolution results at subpixel positions in a finer grid.*
5. *The subpixel position with the highest interpolated FO' is considered the eye corner point at subpixel precision.*

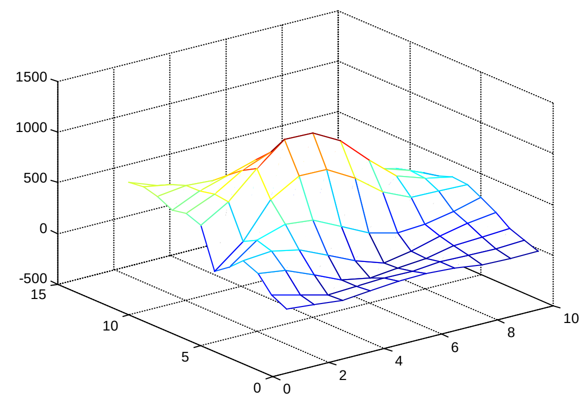
The above algorithm has the following characteristics:

- It is computationally very efficient.
- By using a simple preset eye corner filter, not a template extracted from an eye image, it can accommodate changes in user, lighting, and other situations.
- The eye corner filter is designed to be balanced in positive and negative values, which helps to eliminate the possibility of spurious high convolution values at extremely bright areas.

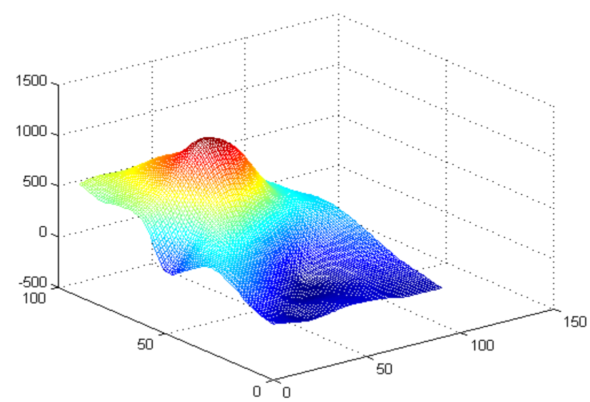
One example of FO', and its interpolation are shown in Figure 4.

$$\begin{bmatrix} -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Figure 3. A pair of corner filters.



(a)



(b)

Figure 4. The result of the corner filtering and its subpixel interpolation.

3.2 Subpixel iris tracking

In this subsection, we first present a new subpixel edge detection algorithm used in searching for the edge of the iris. Then we discuss the approach of locating the center of the iris with an ellipse fitting algorithm. Finally we introduce the procedure of distinguishing edge points at the boundaries between the iris and the sclera from those at the boundaries between iris and eyelids.

3.2.1 Subpixel Edge Detection for Iris Boundary Tracking

Since there is a high contrast between the sclera and iris areas, the boundary pixels of the iris can be easily detected and tracked. To obtain the precise location of the iris center, however, we need edge points at a subpixel level. We approach the problem using a similar algorithm to the subpixel corner tracker:

1. *For every iris-sclera boundary pixel e, we apply horizontal and vertical Sobel edge operators to it, and get two values u and v.*
2. *Calculate the gradient direction of the Sobel edge magnitude at e using:*
   d(e) = arctan(u / v) *(4)*
3. *Calculate the Sobel edge magnitudes at several adjacent pixels along the gradient direction.*
4. *Interpolate the Sobel edge magnitudes at subpixel positions with 1-dimentional cubic interpolation [8].*
5. *Find the subpixel level edge point located at the subpixel position where the interpolated Sobel edge magnitude gets its maximum (minimum).*

In a real-time application, since the iris-sclera boundary points that are not occluded by the eyelids extend near vertically, the above procedure is simplified by assuming a horizontal gradient direction of the Sobel edge magnitude. The y coordinates of the subpixel edge points are set at the mid-point of a boundary pixel, and the x coordinates are calculated by cubic interpolation. This is an approximation to the procedure described above, but it gets satisfactory result in practice.

Besides increasing the precision of iris detection, the subpixel iris edge detection algorithm also solves a practical problem in the subsequent ellipse fitting step, which we discuss next.

3.2.2 Iris Center Estimation with Ellipse Fitting

When we have the iris edge points, we will use them to find the iris center. A common approach is to use the Hough transform to fit a circle to these points [9, 11]. This approach has several drawbacks:

- For a large range of gaze directions, the iris in an image cannot be well approximated by a circle.
- Eye images available for gaze tracking are typically small in size. Thus the number of available iris edge points are usually very limited. This makes the Hough transform, which employs a "voting" procedure, questionable.
- The diameter of an iris is often assumed to be known, while it obviously varies from person to person, and varies when the distance between the user and the camera changes.

In our system, we use an efficient and accurate ellipse fitting algorithm [14] to replace the popular Hough transform to solve this problem. The ellipse fitting algorithm assumes that the input points rest on an ellipse, and use these points to estimate the parameters of the ellipse by solving a least-square fitting problem. Our approach has the following advantages:

- In practice, ellipses are far better approximations of the iris than circles. From our experiments, we have discovered that it is good enough for a range of gaze direction up to $\pm 20^\circ$ in both azimuth and elevation.
- Although the algorithm works best when there are a number of input points, it can still give good result with as few as six points. This is a highly desirable property in practice. Actually, in our experiments, we did have extreme situations when there were only 7 points available.
- No assumption about the size of the iris is required. The ellipse can even be skewed relative to the coordinate axes.

The previously discussed subpixel edge detection is also essential for situations when there are very few edge points. If only pixel level edge points are used, we may encounter a singular matrix problem in solving the ellipse because edge points that should have different x coordinate values can be in the same column at pixel level. By feeding the ellipse fitting algorithm with subpixel edge positions, we successfully circumvent this problem.

3.2.3 Distinguishing Boundary Points

With the occlusion of the iris by the eyelids, there are two groups of edge points around the iris area: points between the iris and the sclera, and points between the iris and the eyelids. We only want to include the former points in fitting the iris. To distinguish between the two types of boundary points, we use the following method: track the uppermost point of the upper eyelid edge and the lowermost point of the lower eyelid edge. These points and the eye corners are connected with straight lines to define a rough contour of the eye socket. Only edge points falling inside this contour are considered as iris edge points.

3.3 Estimation of gaze direction

As discussed in other gaze tracking literatures [9, 11], the calculation of gaze direction from eye features is very difficult. It involves many geometrical models and assumptions, such as the size and movement of the eyeball. These complex models and calculations not only reduce the robustness of the gaze direction estimations, but also often introduce errors into the estimations. Inspired by commercial gaze tracking products, we designed an extremely simple scheme in calculating the gaze direction. It is not only robust, but also accurate, despite its simplicity. The scheme is as follows:

1. Have the user look at several known points on a plane, such as corners and other anchor points. Record the corresponding eye corner and iris center positions. These points serve as the calibration points.
2. We construct a 2-D linear mapping from the vector between the eye corner and the iris center to the gaze angle. Gaze directions in successive frames are calculated by interpolation. For example, suppose the gaze angle and the vector from the eye corner to the iris center for calibration point P_1, P_2 are respectively $\{(\alpha_1, \beta_1), (x_1, y_1)\}$ and $\{(\alpha_2, \beta_2), (x_2, y_2)\}$, then if we observe a corner-iris vector (x, y), the corresponding gaze angle is calculated as follows:

$$\alpha = \alpha_1 + \frac{x - x_1}{x_2 - x_1}(\alpha_2 - \alpha_1) \tag{5}$$

$$\beta = \beta_1 + \frac{y - y_1}{y_2 - y_1}(\beta_2 - \beta_1) \tag{6}$$

The above estimation scheme has the following advantages:

- It is computationally simple and efficient.
- It is extremely accurate. This may be surprising, but if we examine this scheme (please refer to Figure 5) we will find that the only approximation it employs is a simplification of the rotation angle of the eyeball as the projection of the movement of the iris center. As is well known, this $\theta \approx \sin(\theta)$ type approximation is very close for small θ's. Even for extreme gaze angles such as $\pm 35^\circ$, the error introduced by this approximation is just 1.2°. For the most frequent and useful gaze directions, which are in the range of $\pm 15^\circ$, the error introduced by this approximation is less than 0.17°. Actually, a very small deviation from the assumptions that other more complicated methods take bring even larger errors than this.

- It is fast, easy, and adaptive. In many applications, for example tracking gaze point on a desktop screen, we do not need to know the gaze angles. The α's and β's can be just the coordinates on the screen. This simplifies the calibration process. In our experiments, the actual calibration process takes just a few seconds.

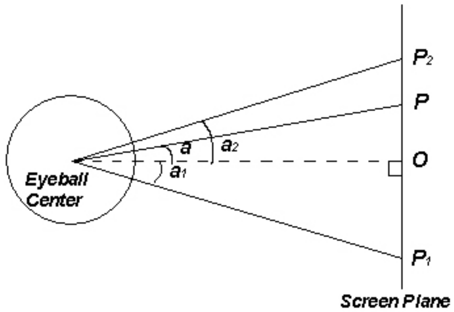


Figure 5. Calculation of gaze direction.

4. Experimental Results

We have applied the proposed algorithms to a real-time gaze tracking system. In order to evaluate the accuracy of the proposed algorithms, we have tested the system using several image sequences. In the data collection, subjects were asked to follow a cursor on a desktop screen using his/her gaze and maintain head pose. The cursor was designed to move in a horizontal or vertical zigzag manner. A video camera was placed at the bottom of the screen and input image size was 320x240 pixels. In order to save storage space and keep real-time performance, the system only saved eye images in each frame to the hard disk. The length of each sequence is between 1700 and 2000 frames. Figure 6 shows an example of an input image. The figure illustrates that the user has enough space for free movement. A larger size of an image, e.g., 640x480 pixels, can be used to provide bigger freedom of movement in a more general application such as an intelligent working space application.

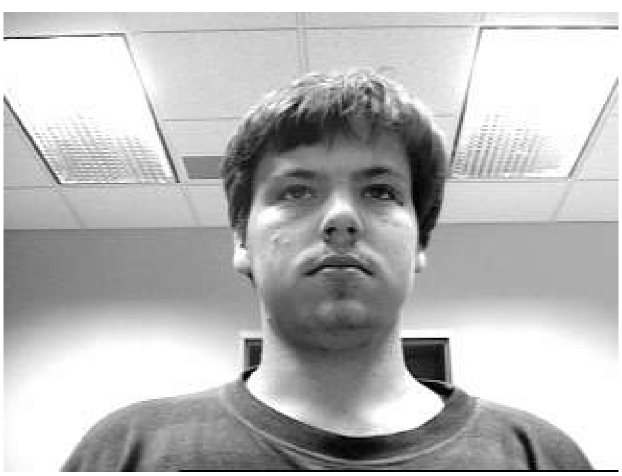


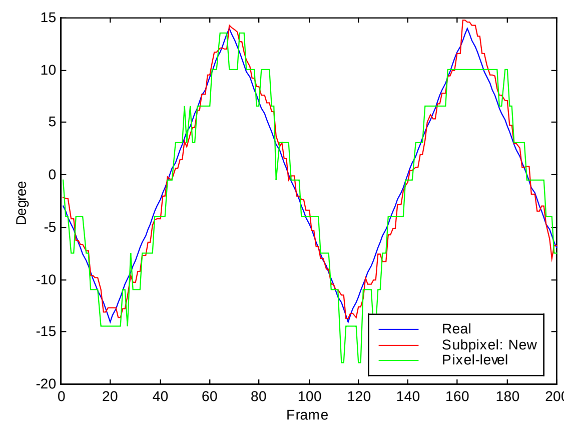
Figure 6. An example of the input image.

We have evaluated the accuracy of the proposed subpixel gaze tracking method, and compared it to a pixel level gaze tracking method and another subpixel method [5]. The average error of these algorithms' performance on three different sequences are listed in Table 2. The average error from the new subpixel gaze tracking algorithm is already less than the diameter of the rod-free fovea. As mentioned in the previous section, a pixel level gaze tracking system, even if its tracking algorithm works perfectly, would generate larger error due to quantization effects. This gives a strong proof to the benefit of using the proposed subpixel feature tracking algorithms.

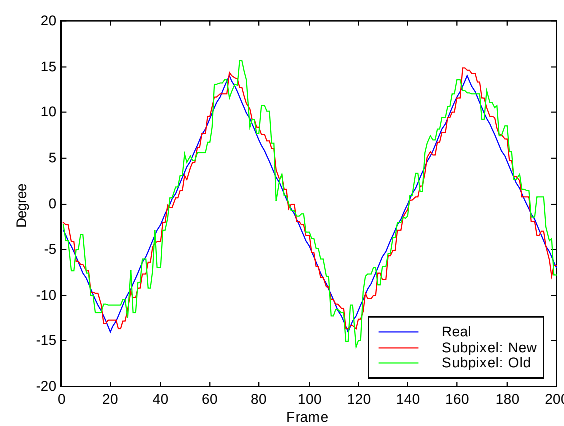
Table 2. Average errors of gaze tracking for three sequences.

Algorithm \ Sequence	1	2	3
New Subpixel	1.2°	1.4°	1.1°
Old Subpixel	2.8°	3.3°	2.7°
Pixel Level	3.1°	3.7°	3.3°

The azimuth angle tracking result of a sequence segment (200 frames) is shown in Figure 7. For a comparison, we also show the result of a pixel level analytical algorithm in Figure 7(a). The discontinuity of the pixel level result is obvious. In Figure 7(b), the result of the proposed method is compared to the result of using a traditional subpixel template matching algorithm [5]. The proposed method is superior to the traditional subpixel method.



(a)



(b)

Figure 7. Azimuth angle tracking result of a sequence segment.

Figure 8 illustrates gaze tracking results randomly selected from an evaluation sequence. The detected edge points, inner eye corner, and the fitted iris center are marked by white points and lines.

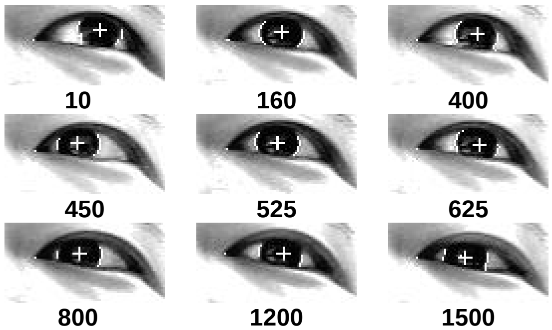


Figure 8. Detected iris edge, inner eye corner, and the fitted iris center.

5. Conclusion

Gaze is an important communication cue in both human-human communication and human-computer interaction. Gaze accuracy has been a major obstacle in developing a real-time gaze tracking system. In this paper, we have investigated the desired accuracy of a gaze tracking system, the technical barrier to achieve such accuracy, and a subpixel approach to break this barrier. We have developed new algorithms for detecting the inner eye corner and the center of an iris with subpixel accuracy. We have applied these algorithms to a real-time gaze tracking system, and the feasibility of the proposed method has been demonstrated by experiments. The proposed method achieved an accuracy of 1.4° on average with normal resolution eye images, which is superior to any reported results obtained through any of the other methods. We are currently applying the proposed eye gaze tracking algorithm together with a head pose estimator to ubiquitous computing applications. We believe that the proposed subpixel feature tracking algorithm can also enhance the accuracy of head pose estimation.

6. Acknowledgements

This Research is supported by the National Science Foundation under Grant No. IIS-9980013.

7. References

[1] C. Achard, E. Bigorgne, and J. Devars. A sub-pixel and multispectral corner detector. *International Conference on Pattern Recognition*, pages 959-962, 2000.

[2] R.J. Althof, M.G.J. Wind, and J.T. Dobbins III. A rapid and automatic image registration algorithm with subpixel accuracy. *IEEE Trans. Medical Imaging*, 16(3): 308-316, June 1997.

[3] K. Astrom and A. Heyden. Stochastic modeling and analysis of sub-pixel edge detection. *International Conference on Pattern Recognition*, pages 86-90, 1996.

[4] S. Baluja and D. Pomerleau. Non-intrusive gaze-tracking using artificial neural networks. *Neural Information Processing Systems*, Morgan Kaufman Publishers, New York, 1994.

[5] S.S. Gleason, M.A. Hunt, and W.B. Jatko. Subpixel measurement of image features based on paraboloid surface fit. *SPIE*, Vol. 1386.

[6] D.C.D. Hung and O.R. Mitchell. Subpixel edge estimation using geometrical edge models with noise miniaturization. *IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 112-117, 1994.

[7] K. Jensen and D. Anastassiou. Subpixel edge localization and the interpolation of still images. *IEEE Trans. Image Processing*, 4(3): 285-295, March 1995.

[8] R.G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 29(6):1153-1160, December 1981.

[9] K.N. Kim and R.S. Ramakrishna. Vision-based eye-gaze tracking for human computer interface. *International Conference on Systems, Man, and Cybernetics*, pages 324-329, 1999.

[10] M. Kisworo, S. Venkatesh, and G.A.W. West. Detection of curved edges at subpixel accuracy using deformable models. *IEE Proceedings on Vision, Image, and Signal Processing*, 142(5): 304-312, 1995.

[11] Y. Matsumoto and A. Zelinsky. An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. *International Conference on Automatic Face and Gesture Recognition*, pages 499-504, 2000.

[12] L. O'Gorman. Subpixel precision of straight-edged shapes for registration and measurement. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(7): 746-751, July 1996.

[13] S. Pastoor, J. Liu, and S. Renault. An experimental multimedia system allowing 3-D visualization and eye-controlled interaction without user-worn devices. *IEEE Trans. Multimedia*, 1(1): 41-52, March 1999.

[14] M. Pilu, A.W. Fitzgibbon, and R.B. Fisher. Ellipse-specific direct least-square fitting. *International Conference on Image Processing*, pages 599-602, 1996.

[15] M. Pomplun, B.M. Velichkovsky, and H. Ritter. An artificial neural network for high precision eye movement tracking. In B. Nebel and L. Drescher-Fischer (Eds.), *Lectures Notes in Artificial Interlligence*, Springer Verlag, Berlin, 1994.

[16] R. Stiefelhagen and J. Yang. Gaze tracking for multimodal human-computer interaction. *International Conference on Acoustics, Speech, and Signal Processing*, pages 2617-2620, 1997.

[17] P. Thevenaz, U.E. Ruttimann, and M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Trans. Image Processing*, 7(1): 27-41, January 1998.

[18] Y.L. Tian, T. Kanade., and J.F. Cohn. Dual-state parametric eye tracking. *International Conference on Automatic Face and Gesture Recognition*, pages 110-115, 2000.

[19] K. Toyama. 'Look, Ma – no hands!' hands-free cursor control with real-time 3D face tracking. *Workshop on Perceptual User Interfaces*, 1998.

[20] K. Umeda and T. Takahashi. Subpixel stereo method: a new methodology of stereo vision. *International Conference on Robotics and Automation*, pages 3215-3220, 2000.

[21] B.A. Wandell. *Foundation of Vision*. Sinauer Press, Sunderland, MA, 1995.

[22] S. Whittaker and B. O'Connail. The role of vision in face-to-face and mediated communication. In K.E. Finn, A.J. Sellen, and S.B. Wilbur (Eds.), *Video-mediated Communication*, Lawrence Erlbaum Associates, Mahwah, NJ, 1997.