Eye Tracking using Monocular Camera for Gaze Estimation Applications

Guojun Yang and Jafar Saniie

Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, Illinois, United States

***Abstract*—Gaze estimation reveals valuable information in terms of human behavior. For example, gaze location can be used as a mean of human computer interaction. In practice, locating the eyes position is a challenging task. Moreover, it is more challenging to estimate the location of the gaze. It requires a system to extract the information of the human eye location then estimate the direction of the gaze. There are existing systems that achieve similar tasks. Many of these systems rely on complex systems, such as wearable sensors or an infrared camera to find the exact focus of eyes gaze. Consequently, these systems are not only complex but also suitable for limited applications. In this paper, we present efficient eye focus methods using a monocular camera, which brings more flexibility and convenience to the system. Three different algorithms for locating the eye's focus have been examined and compared. These algorithms are highly efficient and able to discern the location of the eye's focus in real-time using a regular laptop or desktop computers.**

I. Introduction

Movements of eyes reveal valuable information of human behavior. This information can be used in many applications. For example, the cognition process of human is reflected on the movement of the eyes. This suggests that an object a person is looking at is the observer's cognitive goal [1], [2]. Such relationship between an observer's eye movements and intention can generate a series of applications. For example, when using such information in human computer interfaces, computers may be able to understand human intentions more accurately [3]. The movements of the eyes can also reflect person's mental or medical condition [4]. It is well known that under the influence of alcohol, it is more difficult to keep track of a slow moving object smoothly [5]. As a result, law enforcement personnel will conduct a horizontal gaze nystagmus test to determine the physical condition of a driver. Whenever participants show nystagmus behaviors, they will be considered as driving under the influence of alcohol. However, this field test can sometimes be confusing and inaccurate [6]. While with the help of a portable device that is capable of quantifying eye movements, law enforcement personnel can make better informed decisions.

The demands and applications for eye tracking systems already exist. For example, there are business owners who use eye trackers to identify what displays a customer's gaze is attracted to this leads to a better understanding of a customer's interest. Many of these systems are built using video based sensors that uses an infrared light to illuminate the retina which helps to reduce interference from the surrounding environment. Locating the eyes focus using either 2D images or videos is a challenging task in general. Since eyes take only a small proportion of the overall facial area, the eye movement appears to be relatively frequent. Furthermore, pupils are partially covered by the eye lid which makes template matching or circle detecting methods ineffective in many cases. All these conditions make the eye detection a challenging task. Therefore, achieving robust eye locating, a system become either very complex in terms of the hardware or demands robust algorithms. In this paper, we focus on algorithms that allow a monocular camera to capture the pattern of the eyes' focus.

II. Background

There are different approaches to identifying the eyes' focus problems, most of these can be categorized into three main categories [7]. The first is a sensor based approach, such as sensors attached to the skin around eyes, or contact lens sensors. The second category is the corneal reflection method, which locates the eyes' focus by projecting infrared lights to create recognizable reflection on the retina. The last category is using pure image processing techniques with a monocular camera. These approaches toward the eye locating problem were developed with different backgrounds. As a result, each of these techniques fit different applications. For example, in 1901, Dodge and Cline studied the eye fixation problem using mechanical equipment that attaches to the eye [8]. Today, it is not considered appropriate to apply an invasive device to research participants. Modern devices can detect the eye movements without even been noticed. With improving technologies, eye tracking is becoming easier. However, each method still has its advantages and disadvantages.

A. Sensor Based Approaches

Using a designated sensor to find the location of eyes gaze is one of the most straightforward solutions to this problem. A widely used method in eye tracking is the 2D scleral search coil [9]. Horizontal and vertical positions of the eye can be determined by measuring the voltage induced in the copper windings, which corresponds to the position of the coil with

978-1-4673-9985-2/16/$31.00 ©2016 IEEE

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on October 02,2020 at 15:13:42 UTC from IEEE Xplore. Restrictions apply.

respect to the external AC magnetic field generated by Helmholtz coils [10]. As its name suggests, the 2D scleral search coil can only measure the direction the user is looking in. There is no distance information that can be extracted from such sensors. However, by measuring the torsion of the eye ball, the distance of gazing location can be determined. This task can be performed using the 3D scleral search coil. Since the coil is directly attached to user's eyes, the accuracy of the measurement is persistent regardless of the light conditions or the head movement. However, compared with other video based sensors, coil sensors are more invasive and less desirable. Therefore, 2D and 3D scleral search coils are more suitable for clinical research purposes.

B. Corneal Reflection

Corneal reflection is an efficient method to find where the gaze of the eye is being directed. The theory is based on a common fact that human corneal reflects infrared lights. Considering the *red-eye effect* when taking photographs using the flash light, such phenomenon is the result of the eye structure. When the light projects on the pupil, a beam of light can get through the retina and reaches the rear surface inside the eyeball, then reflects and travels back to camera through the retina.

Corneal reflection can be used to locate the eye center using the glint alone [11]. The reflection on the rear surface inside the eyeball will appear at the exact same location as the retina. Such a bright glint makes it possible to find the location of eye's gaze by simply applying threshold on the image intensity. Some other researchers employed the use of reflections on the eyeball to calculate the gaze location directly [12]. To be more specific, the direction of the gaze can be calculated with a fixed camera and infrared light position by finding the reflection from the rear surface inside the eyeball as well as the reflection on the corneal surface. Also, there are many different configurations of corneal reflection systems, such as multiple cameras and multiple light sources. Moreover, the light sources can either be mounted on goggles or on a table. Such a method can perform accurate gaze estimation as well. However, using the infrared light and camera is not intuitive. Moreover, the relative position of the camera and light source must remain fixed, and the orientation of the camera and light source needs to be accurate. Hence, existing products using the corneal reflection method are designed for specific applications which results in limited flexibilities.

C. Monocular Camera

Unlike the scleral search coil or corneal reflection method, using a monocular camera has limited requirements in terms of the hardware. When using a monocular camera, the robustness and accuracy of the system relies more on eye focus locating algorithms. In this paper, three different eye focus locating algorithms are studied and compared. When building an eye focus locating system using a monocular camera, there are three challenging steps. The first is to locate the user. The second is to find the approximate location of eyes' focus. The third is to find the eye centers. The system is highly re-configurable due to the simplicity of the hardware, which can be easily tailored for different applications. Therefore, potential applications of the monocular camera eye tracking system can be enormous.

III. MONOCULAR CAMERA EYE LOCATING ALGORITHMS

In this study, different eye focus locating algorithms were explored which utilize the same monocular camera. All systems use a template base face detection method to locate the face. With a known face location, the eye regions can be decided accordingly. Then, the location of eyes is computed using three different methods: Hough circle method [13], Isophote curvature method [14], [15] and mean of gradient method [16].

A. Hough Circle

The Hough Circle method is an effective algorithm to find circles in images [13]. The fundamental idea of Hough circle transform is similar to Hough transform. In Hough transform, each point in regular 2D space is transformed to a sine-like periodic curve in Hough space, as shown in Equation (1). A point in (x, y) is represented by a vector, starting from the origin with length R and angle θ.

$$R = x \sin\theta + y \sin\theta \quad (1)$$

However, the circle Hough transform uses a set of circles in the transformed space to represent a point in the original space [13]. As shown in Equation (2), to represent any points (x, y) on a circle with the center (a, b) and radius R in $x - y$ space, we can draw a circle for each point on the circle at (a_i, b_i) with the radius R. All these circles intersect at the center (a, b).

$$\begin{aligned} x &= a + R\sin\theta \\ y &= b + R\sin\theta \end{aligned} \quad (2)$$

As shown in Figure 1, given three points (shown as black dots) on a circle whose radius is R. After conducting the circle Hough transform, each dot on the circles creates a circle in Hough space by finding the intersection of these circles. Thus, the center of the circle in original image can be located (shown as the orange dot).

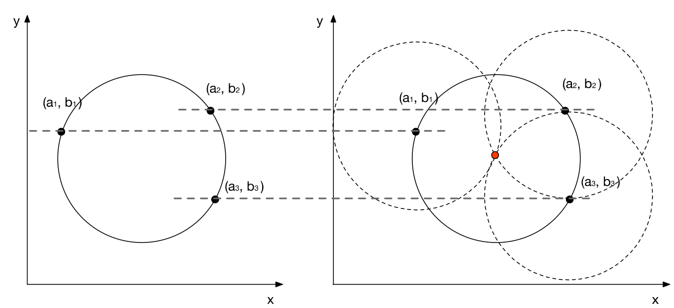


Figure 1. Illustration of Hough transform

In practice, the radius R of the original circle is unknown. Also, it is impossible to find out if an arbitrary pixel is on the circle before processing. Therefore, when conducting the circle Hough transform, each point generates a series of circles with different radius. As a result, the point with the most intersections is decided as the center of the circle.

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on October 02,2020 at 15:13:42 UTC from IEEE Xplore. Restrictions apply.

B. Isophote

Using Isophote curvature is a more direct method to find the center of the eye. This algorithm is based on the gradient information of images [14], [15].

The first step of the Isophote algorithm is to derive the gradient in horizontal L_x and vertical L_y directions. The gradient information is only calculated on the perimeter of circles. The center can be estimated using the gradient information. To be more specific, circles with smaller radius generate curves with more significant curvature, which results in larger gradient value. Also, the center of the circle lies on the direction, along which the gradient changes most rapidly. The relationship between the vector to the center of circle and the gradient are described in Equation (3), where $D(x, y)$ is the vector points to the center of the circle from the point at (x, y), L_x is the first order derivative of the image in x direction (gradient change in the horizontal direction). L_{xx} and L_{yy} are the second order derivative of image in x and y directions respectively, L_{xy} can be calculated by finding the first order derivative in x direction and then in y direction.

$$D(x, y) = -\frac{\{L_x, L_y\}(L_x^2 + L_y^2)}{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}} \quad (3)$$

Using the Isophote algorithm, each point on a segment of the circle or the curve generates a vector point to a potential center of the eye. The point that most vectors pointing to is more likely to be the center. Using voting as an analogy, the more votes a point has, the higher chances it is the center of the eye. Therefore, the candidate point with the most votes is determined as the center of the circle, which is the eye location in our system. As shown in Figure 2, the given original image (on the left) includes multiple curves, the algorithm is able to generate the vector pointing to the center and vote for center accordingly (on the right). Result shown on the right in Figure 2, each point on the curve votes for its center according to the curvature and the direction of the curve, each vector (shown as green ray) starts from the point on curve and points to the center that the point is voting.

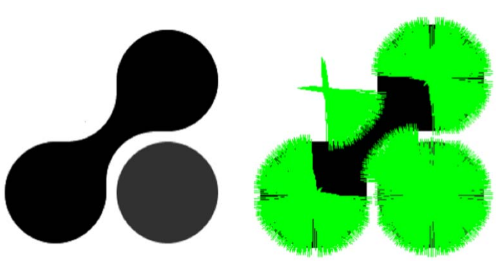


Figure 2. Illustration of center voting [15]

C. Mean of Gradient

Mean of gradient (MoG) [16] algorithm is another efficient algorithm to locate the eye center. Similar to the Isophote curvature method, the MoG algorithm locates the center of the eye using the curvature information knowing the fact that gradient vectors on the perimeter of the iris passes its center. Therefore, points that are away from the center will generate smaller dot product. As a result, the center of the iris can be found by computing a dot product of each potential point and all gradient vectors on perimeter.

In MoG, the algorithm, to locate the eye center, gradient vectors g_i need to be found for each pixel p_i within the eye region. Each gradient vector contains gradient information in horizontal and vertical directions, represented by L_x and L_y respectively. Then, the algorithm searches the center region of the eye through a voting process. For each point p_i, it connects to all points q_i on the perimeter of the iris and creates vectors d_i. The voting or weight value of each point p_i is calculated with the all dot product, as illustrated in Figure 3. In Figure 3, candidate point p_1 (on the left) generates a larger dot product than p_2 (on the right).

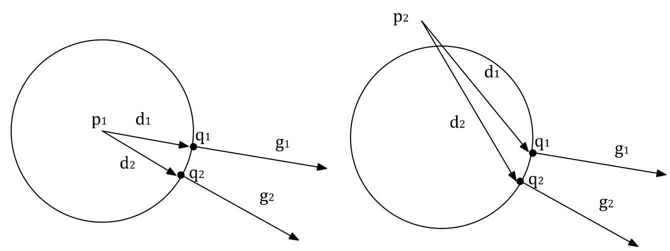


Figure 3. Illustration of dot products of each point

Then the among all points, the point p_i with largest the weight is considered as the center the iris. The mathematical description of the weighting process is described in Equation (4), where the weight W_p of an arbitrary p is calculated as following.

$$W_p = \frac{1}{N}\sum_{i=1}^{N}(d_i^T g_i)^2 \quad (4)$$

Among the regions of interest (ROI), the point that generates the largest weight value should be considered as the center of the eye. The parameter N is the number of points on the perimeter. There are situations that a segment of curve which has a large diameter been detected and mistakenly considered as the center of the eye. Therefore, $\frac{1}{N}$ was applied to the equation to fix this problem.

IV. RESULTS

To verify the validity and performance of these three algorithms, a series of simple experiments were conducted. In these experiments, same pre-processing procedure and post-processing procedure were conducted on the images taken by camera. Pre-processing process includes face detection, eye locating and applying thresholds on other common image processing processes. Then, different post-processing procedures will visualize the results. A flowchart of the evaluation process is illustrated in Figure 4.

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on October 02,2020 at 15:13:42 UTC from IEEE Xplore. Restrictions apply.

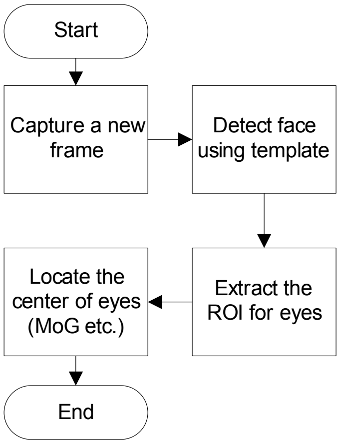
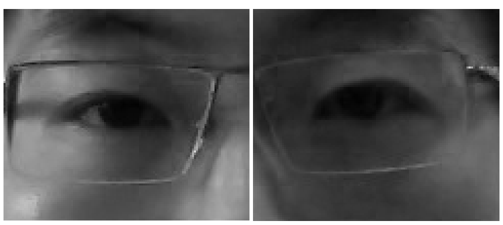


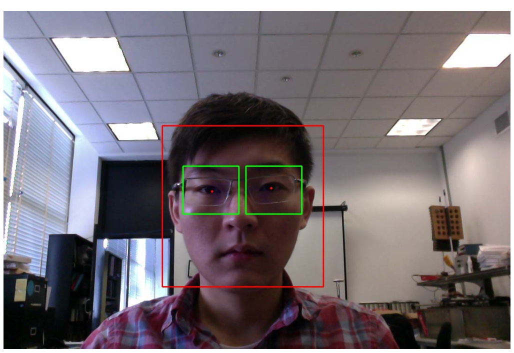
Figure 4. Flowchart of the evaluation process.

Each single step in these processes is identical during the evaluation phase except when locating the eye centers. The outcome of each single step is shown in Figure 5.

(a) Original image (b) ROI of face



(c) ROI for both eyes



(d) The result of eye locating

Figure 5. Outcome of each single step

In Figure 5, subfigures show the outcome of each step from the implemented algorithm. Figure 5(a) shows the original image (frame) being sent to the algorithm. Figure 5(b) shows the face detected using multi-scale template based method. To accommodate processing speed and accuracy as well as consistency, each detected face is scaled to the same fixed size regardless of its original size. In Figure 5(c), once the location of the face has been detected, the region of eyes can be generated accordingly, since the relative location of eyes and face are fixed. Finally, using MoG or other algorithms, the eye center can be located, as shown in Figure 5(d).

A. *Accuracy*

In this experiment, accuracy is measured by the distance between the eye location found by algorithm and the manually determined eye location.

To verify the robustness of this algorithm, a 3 by 3 pad (shown in Figure 6) was displayed on the computer screen. The user was sitting 590 millimeters in front of the computer screen (where the webcam is embedded) and the edge length of each square is approximately 44 millimeters. By looking at different squares, the robustness of the algorithm can be estimated. The result is shown in Figure 7, subfigures (a) to (i) are images captured when staring at square A-I respectively.

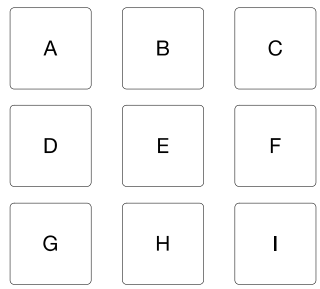
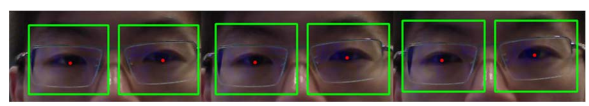
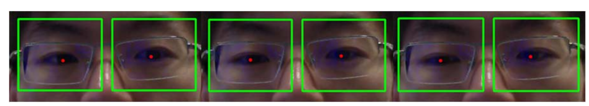


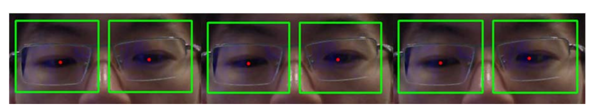
Figure 6. The square pattern used during experiments



(a) (b) (c)



(d) (e) (f)



(g) (h) (i)

Figure 7. The image captured starring at square A to I

According to the results shown in Figure 7, location changes of the eye balls are insignificant, even when the distance between two squares on the screen is significant. To quantitatively estimate the error in each of the algorithms discussed in this paper, the error E is calculated by the Euclidean distance D between true eye center p and the estimated eye center p', then divided by the width of eye S (distance between eye corners) as described in Equation (5).

$$E = \frac{D(p, p')}{S} \tag{5}$$

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on October 02,2020 at 15:13:42 UTC from IEEE Xplore. Restrictions apply.

TABLE I. ERROR RATE

Block	A	B	C	D	E	F	G	H	I
IsoPhote	0.1374	0.1241	0.1253	0.1320	0.1075	0.1162	0.0867	0.1363	0.0429
Hough Circle	0.1413	0.1245	0.1248	0.1370	0.1094	0.1052	0.0961	0.1418	0.0528
Mean of Gradient	0.1399	0.1259	0.1243	0.1402	0.1010	0.1043	0.0979	0.1373	0.0491

After constantly computing the error rate while staring at a fixed location (block A to I), the average error rates of algorithms can be obtained. The error E results are shown in Table . These results demonstrate the accuracy of IsoPhote, Hough Circle and Mean of Gradient algorithms.

This result also confirmed the challenges in the monocular camera based eye tracking systems.

B. *Efficiency*

Due to the nature of these algorithms, the complexity of the program is always linear. To elaborate, the steps of operation is linear to the size of eyeballs (size of the perimeter). The efficiency is acceptable but with slight differences. When using the MoG, the performance of the eye detection process is degraded due to the fact that all pixels in the eye ROI are calculated with pixels on the perimeter of the eyeballs (detected by applying threshold to gradient value). Therefore, this method appears to be relatively slower while the Isophote and circle Hough transform appear to be faster.

As mentioned in previous sections, the detected face is scaled to a fixed size in order to accommodate both speed and accuracy. A larger size increases accuracy, but slows down the process. During the experiment, the system can perform eye focus locating in real-time.

V. CONCLUSION

Eye focus locating using a monocular camera is a challenging task. In this paper, the study on eye tracking methods is presented with possible solutions. Each solution fits different applications, since gaze location reflects cognition process. When people turn their eyes to look at some objects, almost immediately, they will then turn their heads toward the object, eventually, the eyes will return to the center. Therefore, with merely by locating the eye centers, it is inadequate to estimate the gaze location. However, algorithms implemented in this study have demonstrated their capabilities of locating eye enters regardless of the direction of gazing. Moreover, the system can operate in real-time, this advantage that benefits applications require responsive interactions.

From the experimental results, we discovered that when staring at a certain object, people will turn their eyes towards it, then turn their heads towards it, and eyes return to the center eventually. This fact makes it possible to estimate the gaze of human using head direction only [17]. Another fact we discovered during the experiment is that, turning the head towards the eye gaze direction is done in a subconscious state of mind, which makes it difficult to gaze at certain locations without moving the head. This fact implies that eye movements can be used to predict head movements. Eye tracking algorithms presented in this paper can be used in human computer interfaces when assisted with other techniques, such as finding the direction of the head by 3D face modeling. Moreover, results from the conducted experiments may benefit researches requiring eye tracking tools, such as research on human behavior or medical conditions [6].

REFERENCES

[1] M. Hayhoe and D. Ballard, "Eye movements in natural behavior," *Trends in cognitive sciences,* vol. 9, no. 4, pp. 188-194, 2005.

[2] S. P. Liversedge and J. M. Findlay, "Saccadic eye movements and cognition," *Trends in Cognitive Sciences,* vol. 4, no. 1, pp. 6-14, 2000.

[3] K.-N. Kim and R. S. Ramakrishna, "Vision-based eye-gaze tracking for human computer interface," in *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, Tokyo, 1999.

[4] B. Cohen, V. Matsuo and T. Raphan, "Quantitative analysis of the velocity characteristics of optokinetic nystagmus and optokinetic after-nystagmus," *The Journal of Physiology,* vol. 270, no. 2, pp. 321-344, 1977.

[5] K. E. Money and W. S. Myles, "Heavy water nystagmus and effects of alcohol," *Nature,* vol. 247, no. 5440, 1974.

[6] S. E. Busloff, "Can your eyes be used against you? The use of the horizontal gaze nystagmus test in the courtroom," *Journal of Criminal Law and Criminology,* vol. 84, no. 1, pp. 203-238, 1993.

[7] H. Drewes, Eye gaze tracking for human computer interaction, Munich: Ph.D Thesis, Ludwig Maximilian University, 2010.

[8] R. Radach, J. Hyona and H. Deubel, The mind's eye: Cognitive and applied aspects of eye movement research, Holland: Elsevier, 2003.

[9] T. Imai, K. Sekine, K. Hattori, N. Takeda, I. Koizuka, K. Nakamae, K. Miura, H. Fujioka and T. Kubo, "Comparing the accuracy of video-oculography and the scleral search coil system in human eye movement analysis," *Auris Nasus Larynx,* vol. 32, no. 1, pp. 3-9, 2005.

[10] D. A. Robinson, "A Method of Measuring Eye Movemnent Using a Scieral Search Coil in a Magnetic Field," *IEEE Transactions on Bio-medical Electronics,* vol. 10, no. 4, pp. 137-145, 1963.

[11] C. H. Morimoto, D. Koons, A. Amir and M. Flickner, "Pupil detection and tracking using multiple light sources," *Image and Vision Computing,* vol. 18, no. 4, pp. 331-335, 2000.

[12] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *Biomedical Engineering, IEEE Transactions on,* vol. 53, no. 6, pp. 1124-1133, 2006.

[13] H. Yuen, J. Princen, J. Illingworth and J. Kittler, "Comparative study of Hough transform methods for circle finding," *Image and vision computing,* vol. 8, no. 1, pp. 71-77, 1990.

[14] R. Valenti and T. Gevers, "Accurate eye center location through invariant isocentric patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 34, no. 9, pp. 1785-1798, 2012.

[15] R. Valenti and T. Gevers, "Accurate Eye Center Location and Tracking Using Isophote Curvature," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, Anchorage, 2008.

[16] F. Timm and E. Barth, "Accurate Eye Center Localization by Means of Gradients," in *Proceedings of the int. conference on computer theory and applications*, Algarve, Portugal, 2011.

[17] R. Newman, Y. Matsumoto, S. Rougeaux and A. Zelinsky, "Real-time stereo tracking for head pose and gaze estimation," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, Grenoble, 2000.

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on October 02,2020 at 15:13:42 UTC from IEEE Xplore. Restrictions apply.