

Isophote-Based Low-Computing-Power Eye-Detection Embedded-System

C. Ordoñez, E. Blotta, and J. Pastore

Abstract—The eye tracking field is an area highly studied by different researchers. In addition, there are companies that offer commercial eye trackers, but they are usually economically restrictive. However, current technology allows to design embedded eye tracking systems, capable of running on low-cost hardware. This paper presents the implementation of a real-time eye-detection method that uses the properties of “isophotes”, to achieve robustness against changes in illumination, eye rotation and pupil size. The method is implemented in a portable platform, reduced both in computing power and in RAM memory. The performance is evaluated using different eye tracking methods implemented by the authors on the same platform. For this purpose, a database generated on this hardware is used. This database is composed of a set of low-resolution near-infrared eyes-images created with the dark pupil technique, using a lightweight glasses head unit. Results show the robustness of the method and a significant accuracy improvement over the methods analyzed.

Index Terms—Eye tracking, Isophotes, Low cost, Portable, Real-time.

I. INTRODUCCIÓN

LOS avances tecnológicos han permitido grandes logros en el campo de seguimiento de objetos y, en particular, en el de seguimiento de ojos. Este último campo resulta de interés, ya que puede ser aplicado a diversas áreas como, por ejemplo: videojuegos, publicidad, asistencia de personas con discapacidad, biometría, detección de somnolencia al volante, y cualquier otro tipo de interacción persona-computadora [1]. Existen una gran variedad de sistemas de este tipo, los cuales se pueden agrupar, según el tipo de plataforma de cálculo que emplean, en tres categorías: basados en computadoras personales, en Dispositivos Lógicos Programables en Campo (FPGA) o en microcontroladores.

La plataforma más utilizada por la mayoría de los investigadores que trabajan en el campo de seguimiento de ojos, son las computadoras personales. En este sentido, al disponer de una buena capacidad de procesamiento y memoria, existen diversos enfoques en los cuales el costo computacional no es un variable tenida en cuenta. Algunos autores han obtenido recientemente buenos resultados empleando técnicas de detección y selección de bordes [2]; técnicas de búsqueda de contornos [3]; clasificadores Ada-

boost combinados con Máquinas de Soporte Vectorial (SVM) y filtros de Kalman para ahorrar tiempo de procesamiento [4]; técnicas de cálculos de gradientes [5]; detectores de esquinas, como el detector Shi-Tomasi combinado con el conocido método de Viola-Jones [6] y el algoritmo de clasificación K-means para determinar regiones candidatas a ojos [7]; técnicas de seguimiento empleando el algoritmo Unscented Kalman filter (ST-UKF) para lidiar con la no-linealidad en el seguimiento de ojos [8]; algoritmos de filtros de partículas [9]; técnicas de patrones isocéntricos invariantes combinadas con Mean-Shift y Machine Learning [10]; entre otros. Mayoritariamente, los métodos mencionados, emplean fuentes de iluminación infrarroja para generar el efecto de pupila oscura o, en su defecto, el de pupila brillante para mejorar el contraste pupila-iris y facilitar la detección.

Dada la capacidad de procesamiento en paralelo de los sistemas embebidos basados en FPGA se ha incrementado su aplicación en distintas investigaciones. Este tipo de tecnología se utiliza para implementar sistemas de seguimiento de ojos mediante técnicas de substracción de fondo [11], técnicas que emplean luz diferencial a través de dos fuentes de iluminación infrarroja sincronizadas y posicionadas convenientemente [12], y técnicas de detección de bordes combinadas con aproximación elíptica [13], entre otros.

La tecnología actual permite desarrollar sistemas embebidos de seguimiento de ojos en tiempo real que se ejecuten en plataformas basadas en microcontroladores, logrando sistemas portables, de bajo costo y de bajo consumo de energía. A pesar de esto, en la bibliografía no resulta fácil encontrar trabajos que hayan obtenido resultados destacables en esta área. Generalmente, las técnicas robustas que se presentan en la literatura resultan demasiado complejas como para poder ser implementadas en plataformas con estas características.

En este trabajo se presenta la implementación de un método robusto de detección de pupila, basado en el trabajo de patrones isocéntricos invariantes desarrollado por Theo Gevers et. al. [10], capaz de ser utilizado en sistemas críticos, donde se requiere un alto nivel de precisión y robustez (por ejemplo, para el control de una silla de ruedas motorizada, a través del seguimiento de pupila). El método aprovecha las ventajas del empleo de isocurvas para lograr robustez frente a cambios de iluminación, rotación, tamaño y color de ojos. Además, se compara dicho método con distintos algoritmos desarrollados por los autores, diseñados bajo una perspectiva de minimización del costo computacional. Todos estos algoritmos fueron implementados en una plataforma de bajo costo, basada en un microcontrolador Cortex M4 [16].

Para evaluar el desempeño del método propuesto, se realizará un análisis comparativo con los siguientes métodos de seguimiento de pupila basados en: la Transformada de

C. E. Ordoñez, Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICyTE), Universidad Nacional de Mar del Plata - CONICET, Argentina, (e-mail: ordonezc@fi.mdp.edu.ar).

E. L. Blotta, ICyTE, Universidad Nacional de Mar del Plata - CONICET, Argentina.

J. I. Pastore, ICyTE, Universidad Nacional de Mar del Plata - CONICET, Argentina.

Hough para Circunferencias (THC) [17], la THC combinada con factores de forma [18] y redes neuronales Feedforward [19]. Dichos métodos serán descritos brevemente en la sección “Resultados” (III).

El trabajo se organiza de la siguiente manera: en la sección II se presenta el método basado en patrones isocéntricos y se detalla su implementación; en la sección III se presentan los resultados, donde se realiza un análisis exhaustivo del método, comparándolo con los métodos mencionados, y analizando el impacto que generan distintas variables que permiten evaluar la robustez del método.

II. MÉTODO BASADO EN PATRONES ISOCÉNTRICOS INVARIANTES

El método propuesto para la detección de pupila en tiempo real se basa en el empleo de curvas de intensidad constante (*isocurvas*, conocidas en inglés como *isophotes*). Una propiedad importante que tienen este tipo de curvas es que son invariantes a cambios de iluminación y rotación [25].

Partiendo de la premisa de pupila circular, las isocurvas se procesan convenientemente y se genera un espacio de acumulación de votos que se utiliza para encontrar el centro de la circunferencia buscada.

La Fig. 1 muestra las cuatro etapas principales que componen el método: filtrado de entrada, curvatura y centro de isocurvas, generación de mapa de centros y filtrado de salida.

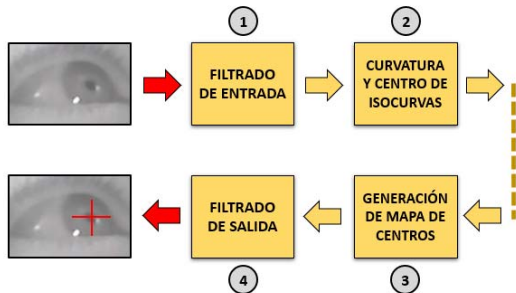


Fig. 1. Etapas principales del método.

A. Filtrado de Entrada

La etapa de filtrado de entrada es una etapa clásica de “pre-procesamiento”. Permite suavizar los datos de la imagen de entrada a través de un filtrado gaussiano, con el objetivo de mejorar los resultados del procesamiento. Si bien, en la búsqueda de acotar costos de procesamiento, resulta tentador no implementar esta etapa, esto puede generar una pérdida significativa en la precisión del método. Este efecto se detalla en la sección “Resultados” (III).

B. Curvatura y Centro de Isocurvas

Las isocurvas son una de las curvas características de cualquier superficie. Estas se definen como curvas con la propiedad de que sus elementos poseen la misma intensidad de luz. Pueden ser consideradas como una consecuencia de la ley del coseno de Lambert que establece que la intensidad de iluminación sobre una superficie difusa es proporcional al coseno del ángulo generado entre el vector normal a la superficie y el vector de luz [26]. Estas curvas son muy útiles para caracterizar la geometría de una superficie, y utilizadas exitosamente en el campo de visión computacional [27][28].

En una imagen digital, las isocurvas son representadas por curvas que conectan píxeles de igual intensidad. En otras palabras, si visualizáramos una imagen en tres dimensiones, las isocurvas serían cortes horizontales de la función intensidad, tal como se muestra en la Fig. 2.

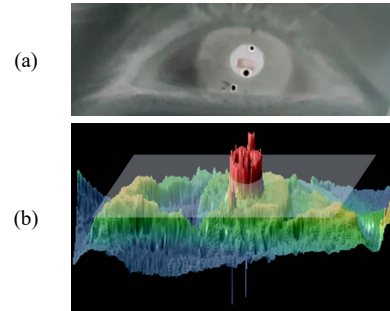


Fig. 2. Ejemplo de isocurva. La figura (a) muestra una imagen de pupila, cuyos niveles de intensidad fueron invertidos. La figura (b) muestra la función intensidad en \mathbb{R}^3 , y un plano horizontal, que permite identificar píxeles de igual intensidad.

Tradicionalmente, la mayoría de los métodos mencionados en la literatura, emplean información de bordes para poder discernir acerca de la geometría del objeto. Si bien es cierto que este enfoque puede ofrecer información valiosa al respecto, restringe el procesamiento a una región del plano de la función intensidad del objeto, desperdiciando la información del resto de los píxeles que componen la imagen.

Dada la característica semi-circular de la pupila, un atributo que interesa analizar es la curvatura de las isocurvas, la cual representa la tasa de cambio del vector tangente a la misma.

P.W. Verbeek [29] demuestra que la curvatura asociada a las isocurvas pertenecientes a una imagen $A \in \mathbb{R}^{M \times N}$ (siendo M y N , la altura y ancho de la imagen, respectivamente), con valores en escala de grises $A(x,y)$, puede estimarse como:

$$\kappa = - \frac{A_y^2 A_{xx} - 2A_{xy} A_x A_y + A_x^2 A_{yy}}{(A_x^2 + A_y^2)^{\frac{3}{2}}} \quad (1)$$

Donde $\kappa \in \mathbb{R}^{M \times N}$, y verifica que la dirección de las isocurvas es perpendicular al gradiente (A_x, A_y) , siendo:

$$\begin{aligned} - A_x &= \frac{\partial A(x,y)}{\partial x} \\ - A_y &= \frac{\partial A(x,y)}{\partial y} \\ - A_{yy} &= \frac{\partial^2 A(x,y)}{\partial y^2} \\ - A_{xy} &= \frac{\partial^2 A(x,y)}{\partial x \partial y} \\ - A_{xx} &= \frac{\partial^2 A(x,y)}{\partial x^2} \end{aligned}$$

Tal como muestra la Fig. 3, el centro de curvatura, o centro de circunferencia, asociado a una posición (x,y) , se obtiene empleando un vector de desplazamiento \vec{D} . La magnitud de dicho vector es igual al radio de curvatura, el cual se define como la inversa del valor de curvatura κ . La dirección del desplazamiento, se obtiene a través del vector normal (de módulo unitario) a la isocurva, es decir, a través del gradiente unitario en dicho punto. Formalmente, los

vectores de desplazamiento D_x y D_y , se obtienen mediante la siguiente expresión:

$$D_x = -\frac{A_x(A_x^2 + A_y^2)}{A_y^2 A_{xx} - 2A_{xy}A_x A_y + A_x^2 A_{yy}} \quad (2)$$

$$D_y = -\frac{A_y(A_x^2 + A_y^2)}{A_y^2 A_{xx} - 2A_{xy}A_x A_y + A_x^2 A_{yy}} \quad (3)$$

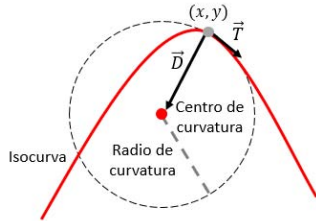


Fig. 3. Centro de curvatura de isocurva tangente al punto (x, y) .

C. Generación de Mapa de Centros

El mapa de centros es una matriz de acumulación de votos que se genera a partir de los centros de circunferencia de las isocurvas, teniendo en cuenta lo siguiente:

- Se consideran solamente los votos generados por curvaturas con signo positivo. La curvatura es positiva, si la intensidad de la región exterior a la curva, es más clara que el interior [28]. Debido al efecto de pupila oscura, la pupila suele ser más oscura que el iris, esto genera que la curvatura tome valores positivos.
- Se utiliza una matriz de pesos para ponderar los votos generados por los centros de isocurvas. Con esto se busca que las isocurvas con forma circular, tengan mayor peso que el resto, ya que, debido a reflejos y reflexiones, la forma de las isocurvas no siempre coincide con la forma del objeto. Para esto se utiliza un factor que mide la característica de curvatura de la región [30]:

$$\gamma = \sqrt{A_{xx}^2 + 2A_{xy}^2 + A_{yy}^2} \quad (4)$$

De esta manera, los votos que provienen de isocurvas con un valor alto de curvatura, es decir, con valores altos de gradiente (bordes de la imagen), tendrán mayor peso que el resto.

D. Filtrado de Salida

El mapa de centros generado en la etapa anterior posee un nivel alto de discretización, tal como se muestra en la Fig. 4b y c.

Como el centro de pupila se obtiene a través del máximo global de dicho espacio, es necesario realizar un filtrado previo para suavizar la discretización y mejorar la estimación de la detección. Tal como se analiza en la sección "Resultados" (III), la etapa de filtrado de salida resulta fundamental para obtener niveles altos de precisión en la detección.

E. Implementación

La implementación que se propone busca minimizar el costo de procesamiento y el uso de memoria para lograr que el método funcione satisfactoriamente en plataformas de

bajos recursos. La Fig. 5 muestra las etapas que componen la implementación del método: filtrado de entrada, cálculo de derivadas parciales, generación de mapa de centros y filtrado de salida.

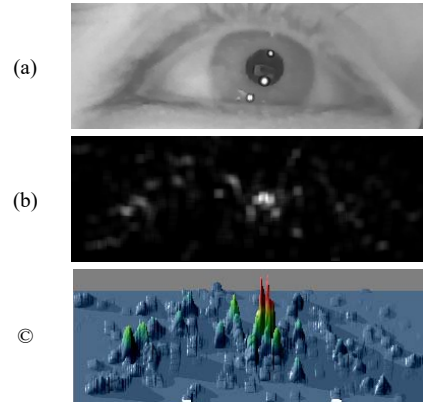


Fig. 4. Ejemplo de mapa de centros. Se grafica el mapa de centros correspondiente a la imagen de entrada (a), en \mathbb{R}^2 (b), y en \mathbb{R}^3 (c).

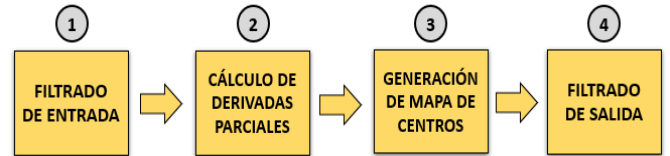


Fig. 5. Etapas de la implementación del método.

- Filtrado de entrada:

El filtrado se realiza convolucionando, por columnas y por filas, la imagen de entrada, con un kernel gaussiano. Se utiliza un kernel, que llamaremos H_i , de una dimensión, de cinco elementos, donde:

$$H_i = [0.160945 \ 0.218261 \ 0.241587 \ 0.218261 \ 0.160945] \quad (5)$$

- Cálculo de derivadas parciales:

Las derivadas parciales también se calculan por convolución. Se utilizan los coeficientes propuestos por Farid y Simoncelli en su trabajo "Differentiation of Discrete Multidimensional Signals" [31], expresados en la Tabla I. La Tabla II muestra cómo se utilizan dichos coeficientes para obtener cada una de las derivadas parciales.

TABLA I
COEFICIENTES PARA DIFERENCIADORES ÓPTIMOS

	Número de muestra						
	0	1	2	3	4	5	6
p	0.004711	0.069321	0.245410	0.361117	0.245410	0.069321	0.004711
d1	0.018708	0.125376	0.193091	0	-0.193091	-0.125376	-0.018708
d2	0.055336	0.137778	-0.056554	-0.273118	-0.056554	0.137778	0.055336

TABLA II
CÁLCULO DE DERIVADAS PARCIALES

Salida	Entrada	Kernel de convolución por columnas	Kernel de convolución por filas
FX	Imagen	-d1	p
FY	Imagen	-p	d1
FXX	Imagen	d2	p
FYY	Imagen	p	d2
FXY	FX	-p	d1

- Generación de mapa de centros:

El mapa de centros se genera a partir de la información de curvatura de las isocurvas, los vectores de desplazamiento y la característica de curvatura. Para reducir el consumo de

memoria, estos cálculos se realizan para cada pixel de la imagen, en lugar de hacerse en forma matricial.

El procedimiento para generar el mapa de centros es el siguiente:

- Primero, se analiza el signo de la curvatura de la isocurva, ya que solamente interesan las que sean positivas. A partir de (1), podemos inferir que para que la curvatura sea positiva, el numerador tiene que ser positivo, ya que el denominador siempre es positivo. De esta manera, lo condición de curvatura positiva que surge es la siguiente:

$$A_y^2 A_{xx} - 2A_{xy} A_x A_y + A_x^2 A_{yy} > 0 \quad (6)$$

- Si se verifica la condición anterior, a continuación, se calculan las componentes (x, y) y el módulo del vector de desplazamiento \vec{D} mediante (2) y (3).

- Luego, para que este elemento genere un voto en la matriz de acumulación se debe verificar:

- 1) Que el centro de isocurva, el cual se obtiene desplazando la posición que se está analizando empleando los vectores D_x, D_y , se encuentre dentro de la dimensión de la imagen.
- 2) Que la magnitud del vector de desplazamiento esté comprendida dentro del rango de radio de búsqueda.

Si se cumplen estas condiciones, entonces, mediante (4), se calcula el valor de característica de curvatura, y se utiliza dicho valor para incrementar la matriz de acumulación en la posición de centro de isocurva.

- *Filtrado de salida:*

El alto nivel de discretización que posee el mapa de centros generado en la etapa anterior, provoca la necesidad de utilizar un filtro de suavizado de mayor dimensión que el utilizado en la etapa de filtrado de entrada. Se utiliza un kernel gaussiano, que llamaremos H_o , de dos dimensiones, de 9×9 elementos, donde:

$$H_o = \begin{bmatrix} 0.0034 & 0.0039 & 0.0043 & 0.0045 & 0.0046 & 0.0045 & 0.0043 & 0.0039 & 0.0034 \\ 0.0039 & 0.0044 & 0.0049 & 0.0052 & 0.0053 & 0.0052 & 0.0049 & 0.0044 & 0.0039 \\ 0.0043 & 0.0049 & 0.0054 & 0.0058 & 0.0059 & 0.0058 & 0.0054 & 0.0049 & 0.0043 \\ 0.0045 & 0.0052 & 0.0058 & 0.0061 & 0.0062 & 0.0061 & 0.0058 & 0.0052 & 0.0045 \\ 0.0046 & 0.0053 & 0.0059 & 0.0062 & 0.0064 & 0.0062 & 0.0059 & 0.0053 & 0.0046 \\ 0.0045 & 0.0052 & 0.0058 & 0.0061 & 0.0062 & 0.0061 & 0.0058 & 0.0052 & 0.0045 \\ 0.0043 & 0.0049 & 0.0054 & 0.0058 & 0.0059 & 0.0058 & 0.0054 & 0.0049 & 0.0043 \\ 0.0039 & 0.0044 & 0.0049 & 0.0052 & 0.0053 & 0.0052 & 0.0049 & 0.0044 & 0.0039 \\ 0.0034 & 0.0039 & 0.0043 & 0.0045 & 0.0046 & 0.0045 & 0.0043 & 0.0039 & 0.0034 \end{bmatrix} \quad (7)$$

Finalmente, la posición de centro de pupila se obtiene buscando el máximo global dentro de la matriz de acumulación filtrada.

III. RESULTADOS

Para analizar el desempeño del método se analiza la precisión en la detección en función de los siguientes parámetros:

- Radio de búsqueda, indica qué tan sensible es el método al tamaño de la pupila buscada.
- Radio máximo de error, se refiere a la relación precisión-exactitud del método.
- Cambios de iluminación, indica que tan sensible es el método a los cambios de iluminación.
- Filtrado de entrada y salida, se analiza el impacto en la precisión que genera el empleo de estos filtros.
- Costo de procesamiento.

Para enriquecer los resultados con fines comparativos, se evalúan otros tres métodos desarrollados por los autores en

[17] [18] [19] basado en: la Transformada de Hough para Circunferencias (THC), la THC basada en factores de forma (THC+F), y uso de proyecciones con redes neuronales Feedforward (PFF). A continuación, se describen brevemente cada uno de estos métodos.

A. Métodos de Detección de Ojos de Bajo Costo Computacional

1) Transformada de Hough para Circunferencias (THC)

La THC es una técnica robusta frente al ruido presente en las imágenes, ampliamente utilizada en distintos trabajos de detección de ojos [20][21][22]. Utiliza la información de bordes de la imagen para generar un matriz de acumulación a partir de los pixeles que satisfacen la ecuación de la circunferencia buscada [23]. Si bien es una técnica simple, computacionalmente es costosa. Sin embargo, esta puede ser optimizada para reducir sustancialmente dicho costo [17].

2) Transformada de Hough y Factor de Forma (THC+F)

Una de las grandes limitaciones de la THC es que no establece ninguna extracción de características de los candidatos a centros de circunferencias. Estos se obtienen mediante una votación en el espacio de parámetros, seleccionando los máximos globales en la matriz de acumulación. Algunas imperfecciones en la imagen, en el detector de bordes, o simplemente la forma del contorno bajo análisis, generan regiones con máximos locales o globales en el acumulador a pesar de no asemejarse a una circunferencia. Aplicando un factor que mida la forma de las regiones espaciales que generan los candidatos a centros de circunferencia, es decir, aquellos que generan máximos en la matriz de acumulación, es posible mejorar considerablemente la precisión del método [18].

La Fig. 6 muestra un ejemplo donde la THC genera dos máximos globales en la matriz de acumulación. A partir de la información de bordes, visualmente, se puede concluir que la región de centro de pupila es la que tiene mayor forma circular. Esto significa que, caracterizando la forma de dicha región mediante un factor de redondez, es posible seleccionar el candidato correcto.

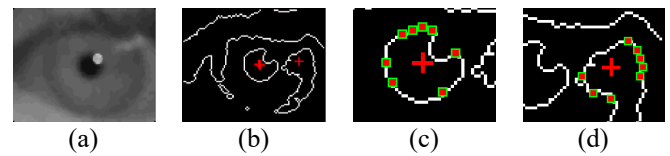


Fig. 6. THC y factor de forma. Se muestra la imagen de pupila (a), la información de bordes junto con la posición de los máximos globales en la matriz de acumulación (b) y los pixeles que generan cada uno de estos máximos (c) (d).

3) Proyecciones con Redes Neuronales Feed-Forward (PFF)

El aprendizaje de una red neuronal se realiza ajustando los pesos entre conexiones en respuesta a patrones de entrenamiento. En redes neuronales de tipo Feed-Forward, esto implica hacer un mapeo desde un espacio de entrada a un espacio de salida. Cuando se utiliza aprendizaje supervisado el proceso se resume a encontrar una buena función de mapeo, es decir, encontrar una función que mapee correctamente los patrones de entrenamiento y generalice el mapeo a patrones de testeo no utilizados en el entrenamiento [24]. Por este motivo, cuando se trabaja con dispositivos de bajo poder de cálculo, el entrenamiento se

suele hacer fuera de línea y se utiliza una tabla de búsqueda almacenada en memoria para consultar la red.

La topología de la red es otro factor fundamental que determina el costo de procesamiento requerido para realizar cada consulta. En redes Feed-Forward, la salida de cada capa se obtiene mediante la siguiente ecuación:

$$a = F(\omega * p + b) \quad (8)$$

Donde:

- ω es una matriz de dimensión $S \times R$, que contiene los pesos entre cada una de las capas, siendo S el número de neuronas de la capa, y R el número de entradas de la red.
- p es el vector de entrada, de dimensión $R \times 1$
- b el vector de polarización, de dimensión $S \times 1$
- F es la función de activación de la red neuronal.

De esta manera, se puede concluir que la complejidad de los cálculos va a estar atada a la cantidad de entradas de la red, la cantidad de capas ocultas y la cantidad de neuronas por capa.

Para implementar este tipo de sistemas en plataformas de bajos recursos, es necesario emplear redes neuronales de baja complejidad y seleccionar características de la pupila que puedan obtenerse mediante cálculos simples, ya que esta información es la que se utiliza para alimentar la red neuronal.

El efecto de pupila oscura que se produce cuando se ilumina el ojo mediante una fuente de iluminación infrarroja, fuera del eje óptico, provoca que la pupila se vea más oscura de lo normal y mejore notablemente el contraste pupila-iris. Este efecto genera una acumulación de píxeles oscuros en la región de pupila. Una forma simple de encontrar la posición de pupila consiste en analizar esta proyección de píxeles oscuros sobre la imagen, ver Fig. 7. Esta información se utiliza para alimentar una red neuronal que funciona como clasificador de no pupila / pupila para determinar la posición horizontal de la pupila. A continuación, se utiliza otra red de las mismas características, para obtener la posición vertical [19].

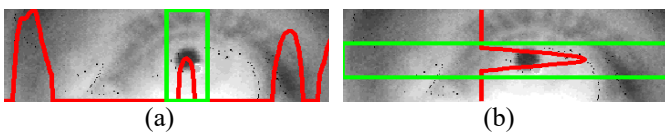


Fig. 7. Proyección de píxeles oscuros. Proyección horizontal (a), proyección vertical (b).

B. Base de Datos de Testeo

Para analizar el desempeño de cada método se utiliza una base de datos propia, libre de descarga [32], obtenida a través de un par de anteojos de seguimiento, que poseen una cámara con iluminación infrarroja, conectada a una placa Discovery STM32F4. Dicha base está compuesta por 800 imágenes de ojos, de infrarrojo cercano, de baja resolución (cuyos píxeles toman valores entre 0 y 31), correspondientes a personas de diferente edad, color de ojos y tamaño de pupila, adquiridas bajo condiciones de baja, media y alta iluminación ambiental. Un inconveniente que surge al querer clasificar las imágenes a partir de la condición de iluminación ambiental bajo la cual fueron adquiridas, es la dificultad que existe para cuantificar el nivel de dicha iluminación. Para sortear este inconveniente, las imágenes

de testeo se dividen en tres categorías, empleando la intensidad promedio (I_p) de los píxeles que las componen:

- Categoría 1: $I_p < 15$
- Categoría 2: $15 \leq I_p \leq 18$
- Categoría 3: $I_p > 18$

C. Precisión

La precisión representa la proporción de detecciones correctas para un conjunto de datos dado. La detección es considerada correcta si la estimación de centro de pupila del algoritmo se encuentra dentro de un radio de n píxeles con respecto al *ground-truth* del centro de pupila.

Teniendo en cuenta que en el banco de imágenes de prueba no existen imágenes de no-pupila, la precisión se obtiene mediante la siguiente ecuación:

$$\text{Precisión} = \frac{TP}{TP + FN} \quad (9)$$

Donde TP es el número de verdaderos positivos, que representa la cantidad de veces en las cuales la distancia entre el centro de pupila detectado y el *ground-truth*, se encuentra dentro de un radio r_0 de píxeles. Por su parte, FN es el número de negativos falsos que representa la cantidad de veces en las cuales esta distancia reside fuera del radio r_0 .

D. Precisión en Función del Radio de Búsqueda

La THC, la THC+F, y el método basado en isocurvas, para poder realizar la detección, necesitan conocer de antemano el tamaño de pupila buscado. Este tamaño se establece mediante un rango de búsqueda comprendido entre un valor mínimo y máximo preestablecidos. La Fig. 8 muestra la precisión de cada método en función del radio de búsqueda (r_b), para imágenes de una misma categoría, donde, $r_{\min} = r_b - 1$ y $r_{\max} = r_b + 1$.

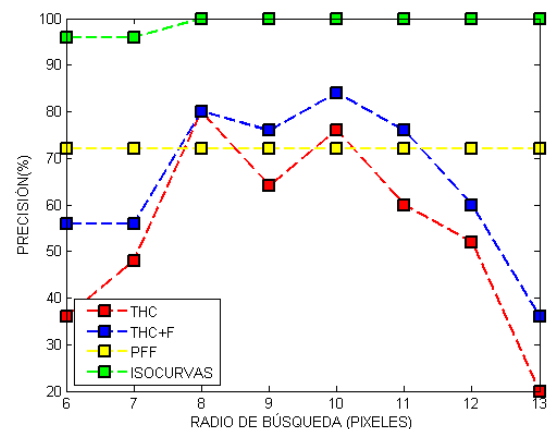


Fig. 8. Precisión de cada método en función del radio de pupila buscada.

Como puede verse, tanto en la THC como en la THC+F, la precisión toma una forma de campana centrada en $r_b = 10$; este valor coincide aproximadamente con el radio real de las pupilas analizadas. En el método PFF, la precisión es constante debido a que el mismo no utiliza un radio de búsqueda, sino que emplea una ventana deslizante que clasifica regiones en pupila o no-pupila. Por su parte, el método basado en isocurvas se destaca por presentar un nivel alto de precisión, muy superior al resto de los métodos, y prácticamente constante, para todo el rango de búsqueda.

E. Precisión en Función del Radio de Error

La Fig. 9 muestra la precisión de cada método en función del radio máximo de error permitido (r_0) para imágenes de una misma categoría. Dicho radio es la máxima distancia radial permitida entre el *ground-truth* y el centro de pupila detectado. Nuevamente, el método basado en isocurvas se destaca notablemente de los otros métodos, presentando un nivel de precisión cercano al 100% para valores de r_0 mayores a 2 píxeles.

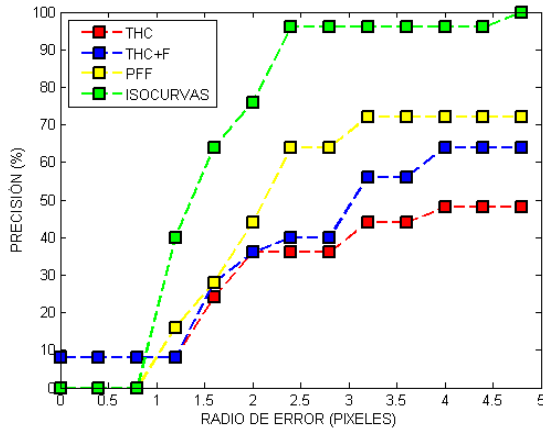


Fig. 9. Precisión de cada método en función del máximo radio de error permitido.

F. Precisión en Función de los Cambios de Iluminación

Los sistemas de detección de ojos pueden ser aplicados a una gran cantidad de áreas donde las condiciones de iluminación ambiental pueden ser variables. Por este motivo, es esencial analizar el efecto que producen estas variaciones en la precisión de cada método. Dicho efecto se analiza en la Fig. 10, donde se grafica la precisión de cada método en función de la categoría de intensidad definida anteriormente.

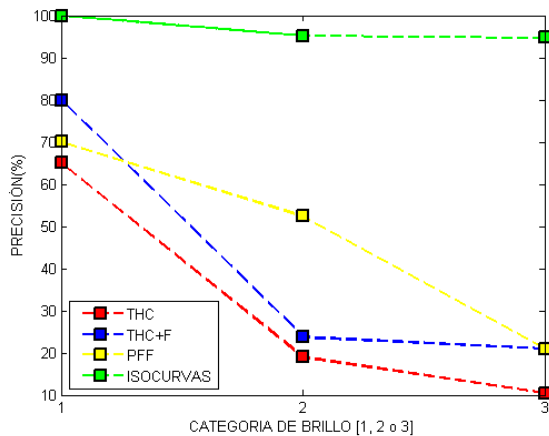


Fig. 10. Precisión de cada método en función de la categoría de intensidad de la imagen.

Los parámetros de configuración de cada método se ajustan para obtener los mejores resultados para imágenes de la categoría 1. De acuerdo a la respuesta de cada método, las conclusiones que se pueden sacar al respecto son las siguientes:

- La THC, la THC+F y el método PFF, muestran una alta dependencia a las condiciones de iluminación. La THC y la THC+F, realizan la búsqueda de centros de pupila a partir de la información de bordes de la imagen. Este tipo de

información, al ser sensible a los cambios de iluminación, genera que la precisión de la detección también lo sea. El método PFF, por su parte, al emplear la información de proyección de píxeles oscuros, restringe el procesamiento a un conjunto acotado de píxeles, cuya extensión depende, nuevamente, de la condición de iluminación.

- El método basado en isocurvas, al considerar el aporte de cada uno de los píxeles que componen la imagen, muestra un alto grado de robustez frente a los cambios de iluminación, presentando un nivel de precisión que prácticamente se mantiene constante para todas las categorías de iluminación, con un valor superior al 95% para todos los casos.

G. Filtrado de Entrada/Salida

Las etapas de pre-procesamiento y post-procesamiento cumplen un rol fundamental en cualquier algoritmo de detección de objetos. El objetivo principal de estas etapas suele ser simplificar cálculos e incrementar la precisión de la detección. En aplicaciones de tiempo real, que sea ejecutan en plataformas de bajos recursos computacionales, estas etapas no pueden ser demasiadas complejas.

Una característica presente en las imágenes con las que se trabaja, es el ruido de adquisición. Este se genera debido a perturbaciones eléctricas en el transductor de la cámara. El ruido de adquisición obstaculiza la detección y en algunos casos los efectos pueden ser significativos. Una forma simple de reducir el impacto de este ruido es a través de un filtro de suavizado.

La Tabla III muestra la precisión de cada método con y sin filtro de entrada, los datos se obtienen procesando imágenes de tipo 1, empleando el kernel que se presenta en (5). Las conclusiones que se pueden al respecto son:

- La precisión del método de isocurvas, aumenta un 20%. cuando se emplea el filtro de entrada. Esto se debe a los motivos mencionados anteriormente.

- Por su parte, la THC, la THC+F, y el método PFF, sufren un decremento en la precisión cuando se emplea el filtro de entrada. La pérdida de precisión de la TH y la THC se debe a que el filtro de suavizado, además de reducir el nivel de ruido de adquisición, también suaviza los bordes de la imagen bajo análisis, los cuales se emplean para realizar la detección. La pérdida de precisión del método PFF, se debe a que el filtro de suavizado, suaviza la región donde se encuentra la pupila, esto genera que la proyección de píxeles oscuros sea menos abrupta en la región de pupila, dificultando de esta manera, la tarea de reconocimiento de las redes neuronales.

TABLA III
PRECISIÓN DE CADA MÉTODO CON/SIN FILTRADO DE ENTRADA

MÉTODO	CON FILTRADO DE ENTRADA [%]	SIN FILTRADO DE ENTRADA [%]
THC	55	65
THC+F	70	80
PFF	60	70
ISOCURVAS	100	80

De los métodos analizados, solamente el método basado en isocurvas emplea un algoritmo de post-procesamiento. Dicho método emplea el kernel gaussiano que se describe en la sección II.E. En la Tabla IV se analiza la precisión promedio del método con y sin filtro de salida. El filtrado se aplica sobre el mapa de centros de isocurvas. La tabla se genera procesando la base de datos

completa, abarcando imágenes de las tres categorías. Como puede verse, cuando se utiliza el filtro de salida, la precisión aumenta un 9%. Esto sucede debido al alto nivel de discretización que posee el mapa. El filtro de suavizado permite considerar el aporte de todos los votos de la región y mejora considerablemente la precisión de la estimación.

TABLA IV

PRECISIÓN DEL MÉTODO DE ISOCURVAS CON/SIN FILTRADO DE SALIDA	
CON FILTRADO DE SALIDA	SIN FILTRADO DE SALIDA
[%]	[%]
97	88

H. Tasa de Detección de Pupila

En esta sección se analiza el costo de procesamiento asociado a cada método. Como medida se utiliza el número de cuadros de imagen de video procesados por segundo, expresado en FPS.

La Tabla V muestra el número promedio de FPS que se logran para cada método, empleando una computadora de escritorio con un procesador Intel Core i5-7400 CPU @ 3.00GHz, con 4 GB de memoria RAM. Como puede verse, el método basado en isocurvas es el más lento, pero igualmente la tasa de procesamiento es lo suficientemente buena como para realizar procesamiento en tiempo real. La THC y el método PFF son los más rápidos, operando a una tasa que oscila entre los 28 y 29 cuadros por segundo.

TABLA V

COSTO DE PROCESAMIENTO DE CADA MÉTODO CORRIENDO EN UNA COMPUTADORA DE ESCRITORIO				
	THC	THC+F	PFF	ISOCURVAS
FPS (promedio)	28	22	29	20

La Tabla VI muestra el desempeño de cada método empleando una placa de desarrollo STM32F4 Discovery, basada en un procesador Cortex M4, con 192KB de memoria RAM [16]. Nuevamente, el método basado en isocurvas es el método más lento, y la THC y el método PFF son los más rápidos. Si bien el método de isocurvas opera a una tasa de seis cuadros por segundo, este valor es significativo, considerando las prestaciones de la plataforma con la que se está trabajando.

TABLA VI

COSTO DE PROCESAMIENTO DE CADA MÉTODO CORRIENDO EN UNA PLACA STM32F4 DISCOVERY				
	THC	THC+F	PFF	ISOCURVAS
FPS (promedio)	12	8	15	6

IV. CONCLUSIONES

En términos de nivel de precisión, robustez a cambios de iluminación, tamaño de pupila y color de ojos, el método presentado supera significativamente a los métodos desarrollados anteriormente por los autores. Este nivel de precisión y robustez, permite que el mismo pueda ser empleado en aplicaciones complejas que requieren niveles altos de precisión, bajo condiciones ambientales variables, para usuarios con distintas características oculares. La ventaja principal del método es que puede ser aplicado en plataformas de bajo costo y recursos computacionales limitados.

Para contrastar resultados, se presentaron distintos métodos de detección de pupila desarrollados por los autores. En cuanto al costo de procesamiento, el método presentado obtuvo la menor tasa de procesamiento, operando a una tasa de seis cuadros por segundo, en una placa Discovery. Sin embargo, esta velocidad de procesamiento no resulta despreciable, considerando la plataforma con la que se está trabajando. Además, es posible incrementarla significativamente empleando técnicas de submuestreo, métodos de correspondencia determinística (velocidad, proximidad, aceleración, etc.), métodos de correspondencia estadística (filtros de Kalman, filtros de Partículas, etc.), entre otros. Por su parte, el aumento de costo computacional, respecto a los métodos presentados, se debe, en gran parte, al número de cálculos de derivadas parciales que requiere el algoritmo.

Como trabajo futuro se propone perfeccionar la última etapa del método, es decir, en lugar de realizar la detección a partir de la posición del máximo global del mapa de centros, desarrollar un algoritmo que emplee la información del mapa, combinada con algún otro conjunto de características propias de la pupila, para mejorar la precisión y exactitud de la detección. Además, se proyecta aplicar el método en el campo de asistencia de personas con discapacidad, permitiéndole a personas con movilidad reducida, poder controlar una silla de ruedas a través de la mirada.

REFERENCIAS

- [1] Jean-Michel Hoc (2000) From human – machine interaction to human – machine cooperation, *Ergonomics*, 43:7, 833-843, DOI: 10.1080/001401300409044.
- [2] Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. 2018a. PuRe: Robust pupil detection for real-time pervasive eye tracking. *Computer Vision and Image Understanding* (Feb 2018). <https://doi.org/10.1016/j.cviu.2018.02.002>.
- [3] Zimmermann, J., Vazquez, Y., Glimcher, P. W., Pesaran, B. & Louie, K. Oculomatic: high speed, reliable, and accurate open-source eye tracking for humans and non-human primates. *J. Neurosci. Methods* 270, 138–146 (2016).
- [4] C. Yan, Y. Wang, Z. Zhang Robust real-time multi-user pupil detection and tracking under various illumination and large-scale head motion *Comput Vision Image Understand*, 115 (8) (2011), pp. 1223-1238
- [5] Fabian Timm and Erhardt Barth. Accurate eye centre localisation by means of gradients. In *Proc. Computer Vision Theory and Applications*, 2011.
- [6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001*, pp. I-I. doi: 10.1109/CVPR.2001.990517
- [7] El Kaddouhi, S., Saaidi, A. & Abarcan, Eye detection based on the Viola-Jones method and corners points. *M. Multimed Tools Appl* (2017) 76: 23077. <https://doi.org/10.1007/s11042-017-4415-5>.
- [8] Zhang, Z., & Zhang, J. (2010). A strong tracking nonlinear robust filter for eye tracking. *Journal of Control Theory and Applications*, 8(4), 503–508. doi:10.1007/s11768-010-8063-9
- [9] Wu, J. & Trivedi, M.M. (2007) Simultaneous Eye Tracking and Blink Detection with Interactive Particle Filters. *EURASIP J. Adv. Signal Process.* (2007) 2008: 823695. <https://doi.org/10.1155/2008/823695>
- [10] R. Valenti and T. Gevers, "Accurate eye center location through invariant isocentric patterns." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.
- [11] A. Amir, L. Zimet, A.S. Vincentelli, S. Kao. An embedded system for an eye-detection sensor. *Computer Vision and Image Understanding*, 98 (2005), pp. 104-123.
- [12] Borsato, F. H., Diaz-Tula, A., & Morimoto, C. H. (2019). xSDL: stroboscopic differential lighting eye tracker with extended temporal support. *Machine Vision and Applications*. doi:10.1007/s00138-019-01022-y.
- [13] J. Kolodko, S. Suzuki and F. Harashima, "Eye-gaze tracking: an approach to pupil tracking targeted to FPGAs," 2005 IEEE/RSJ

- International Conference on Intelligent Robots and Systems, Edmonton, Alta., 2005, pp. 344-349. doi: 10.1109/IROS.2005.1545423
- [14] P. M. Corcoran, F. Nanu, S. Petrescu and P. Bigioi, "Real-time eye gaze tracking for gaming design and consumer electronics systems," in *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 347-355, May 2012. doi: 10.1109/TCE.2012.6227433.
- [15] Schmidt, A., Kasinski, A.: The performance of the haar cascade classifiers applied to the face and eyes detection. In: Springer (ed.) *Proceedings of the 5th International Conference on Computer Recognition Systems CORES2007*. Advances in Soft Computing, vol. 45. Wroclaw, Poland, pp. 816-823 (2007).
- [16] STM32F4DISCOVERY Expansion Boards Datasheets, www.element14.com/stm32f4-expansion.
- [17] Cristian E. Ordoñez; Juan I. Pastore; Virginia Ballarin; Eduardo L. Blotta, "Real-Time Iris-Tracking Embedded System" SASE 2015, FIUBA, Bs. As., Argentina, ISBN 978-987-45523-3-4
- [18] C. E. Ordonez, E. L. Blotta and J. I. Pastore, "Real-time gaze-tracking embedded-system," 2017 Eight Argentine Symposium and Conference on Embedded Systems (CASE), Buenos Aires, 2017, pp. 1-5. doi: 10.23919/SASE-CASE.2017.8115375
- [19] C. E. Ordoñez, L. Dai Pra, E. Blotta, J. I. Pastore and L. I. Passoni, "Real-Time Pupil-Tracking embedded-system based on neural networks," 2017 XVII Workshop on Information Processing and Control (RPIC), Mar del Plata, 2017, pp. 1-6. doi: 10.23919/RPIC.2017.8214334
- [20] M. Soltany, S. T. Zadeh, and H. R. Pourreza, "Fast and Accurate Pupil Positioning Algorithm using Circular Hough Transform and Gray Projection," *International Conference on Computer Communication and Management (icccm)*, vol.5, pp. 556-561, 2011.
- [21] J. Gao, S. Zhang and W. Lu, "Application of hough transform in eye tracking and targeting," 2009 9th International Conference on Electronic Measurement & Instruments, Beijing, 2009, pp. 3-751-3-754. doi: 10.1109/ICEMI.2009.5274208
- [22] K. Toennies, F. Behrens and M. Aurnhammer, "Feasibility of Hough-transform-based iris localisation for real-time-application," *Object recognition supported by user interaction for service robots*, Quebec City, Quebec, Canada, 2002, pp. 1053-1056 vol.2. doi: 10.1109/ICPR.2002.1048486
- [23] Qi-Chuan Tian, Quan Pan, Yong-Mei Cheng and Quan-Xue Gao, "Fast algorithm and application of Hough transform in iris segmentation," *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, Shanghai, China, 2004, pp. 3977-3980 vol.7. doi: 10.1109/ICMLC.2004.1384533
- [24] S. Haykin, *Neural Networks and Learning Machines*, MA, Cambridge:Prentice-Hall, 2008.
- [25] Lichtenauer, Jeroen & Hendriks, Emile & Reinders, Marcel. (2005). Isophote properties as features for object detection. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2. 649 - 654 vol. 2. 10.1109/CVPR.2005.196.
- [26] Doğan, Fatih & Yayli, Yusuf. (2015). On isophote curves and their characterizations. *Turkish Journal of Mathematics*. 39. 650-664. 10.3906/mat-1410-4.
- [27] Kervrann, C., Hoebcke, M. & Trubuil, Isophotes Selection and Reaction-Diffusion Model for Object Boundaries Estimation, *A. International Journal of Computer Vision* (2002) 50: 63. <https://doi.org/10.1023/A:1020276319925>.
- [28] J. Lichtenauer, E. Hendriks and M. Reinders, "Isophote properties as features for object detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 649-654 vol. 2. doi: 10.1109/CVPR.2005.196
- [29] P.W. Verbeek, A class of sampling-error free measures in oversampled band-limited images, *Pattern Recogn Lett*, 3 (4) (1985), pp. 287-292.
- [30] J. Koenderink and A. van Doorn. Surface shape and curvature scales. *Im. and Vis. Comp.*, pages 557-565, 1992.
- [31] H. Farid and E. P. Simoncelli, "Differentiation of discrete multidimensional signals," in *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 496-508, April 2004. doi: 10.1109/TIP.2004.823819
- [32] Base de datos generada con placa Discovery STM32F4, http://www3.fi.mdp.edu.ar/pdi/Database/PUPILAS_DISCOVERY.rar



Cristian E. Ordoñez recibió su título de Ingeniero Electrónico en la Facultad de Ingeniería perteneciente a la Universidad Nacional de Mar del Plata (UNMDP) en el año 2012, y actualmente, está completando su doctorado en Ingeniería Electrónica, en el Laboratorio de Procesamiento de Imágenes (LPI), perteneciente al Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE), en la misma facultad. Tiene una beca de doctorado otorgada por el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), agencia de la Secretaría de Ciencia, Tecnología e Innovación Productiva, Argentina. Su investigación se centra principalmente en el diseño de algoritmos de procesamiento de imágenes en tiempo real para sistemas embebidos.



Eduardo Luis Blotta nació en 1958, en Mar del Plata, Argentina. En 1988 recibió su título de grado en Ingeniería Electrónica y en 2010 recibió su título de Doctor en Ingeniería Electrónica del Departamento de Electrónica de la Facultad de Ingeniería, Universidad Nacional de Mar del Plata (UNMDP). Actualmente es Profesor Asociado,

responsable de asignaturas de grado y posgrado en el área de sistemas embebidos, y miembro del Laboratorio de Procesamiento de Imágenes en el Instituto de Ciencia y Tecnología en Electrónica (ICYTE), CONICET-UNMDP. Sus principales áreas de interés son procesamiento de imágenes, en particular imágenes bio-médicas, procesamiento de imágenes en sistemas embebidos y morfología matemática.



Juan Ignacio Pastore obtuvo sus títulos de Profesor en Matemática y Licenciado en Ciencias Matemáticas en la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de Mar del Plata, en los años 2001 y 2002, respectivamente. En año 2009 obtuvo el título de Doctor en Ingeniería orientación Electrónica. Desde el año

1998 es miembro del Laboratorio de Procesamiento de Imágenes, dependiente del Instituto en Ciencias y Tecnología en Electrónica (ICYTE). Su área de investigación se ha centrado en el Procesamiento de Imágenes aplicando técnicas de Morfología Matemática. Actualmente es Profesor en el Área de Matemática Aplicada y Co-Director del Laboratorio de Procesamiento de Imágenes. También se desempeña como investigador del Consejo Nacional de Investigaciones Científicas y Técnicas de la República Argentina.