

High-frequency event-based Eye Tracking

Nicolas Carrasco^{1,2}, Nicolas Mastropasqua^{1,2}, and Daniel Acevedo^{1,2}

¹ Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales. Dpto. de Computación. Buenos Aires, Argentina

² CONICET-UBA. Instituto de Ciencias de la Computación (ICC). Buenos Aires, Argentina
{ncarrasco, nmastropasqua, dacevedo}@dc.uba.ar

Resumen In recent years, event cameras have gained popularity in computer vision. Their high temporal resolution makes them a particularly useful tool for tracking eye movements, which can reach speeds of 700°/s, as in the case of saccades. The precise analysis of these movements has applications in AR/VR devices, Driving Monitor Systems, interactive systems, clinical diagnostics, and more. In this work, we propose a event-based blink detector that allows to precisely segment the intervals where blinks are produced. We implement an eye-tracking method based on previous work, which uses events and images. This method builds a parametric model of the pupil that updates as many events occur, achieving a high frequency that allows for smooth and continuous tracking of its position. It uses a computationally low-cost polynomial regressor to estimate the gaze point in real time. Preliminary evaluation results show mostly stable pupil tracking. We will use this as a baseline to implement an event-based method.

Keywords: eye tracking, event cameras

Seguimiento ocular de alta frecuencia basado en eventos

Resumen En los últimos años, las cámaras de eventos han ganado popularidad en el ámbito de la visión computacional. Su alta resolución temporal las convierten en una herramienta especialmente útil para el seguimiento de los movimientos oculares, los cuales pueden alcanzar velocidades de 700°/s, como en el caso de las sacadas. El análisis preciso de estos movimientos encuentra aplicaciones en dispositivos de AR/VR, *Driving Monitor Systems*, sistemas interactivos, diagnósticos clínicos, etc. En este trabajo proponemos un detector de parpadeos que requiere eventos únicamente, permitiendo la segmentación precisa de los intervalos donde se producen. Implementamos un método de seguimiento ocular basado en un trabajo previo, que utiliza eventos e imágenes, construyendo un modelo paramétrico de la pupila que se actualiza a medida que se producen varios eventos, logrando una alta frecuencia, lo que permite el seguimiento suave y continuo de su posición. Utiliza un regresor polinomial de bajo costo computacional para estimar el punto de mirada en tiempo real. Los resultados preliminares muestran un seguimiento de la pupila mayormente estable. Utilizaremos este método como línea de base para implementar un método solo basado en eventos.

Keywords: seguimiento ocular, cámaras de eventos

1. Introducción

El seguimiento ocular consiste en el monitoreo y procesamiento de los movimientos que realiza el ojo con el fin de determinar su posición y trayectoria (Hansen & Pece, 2005). A lo largo de los años, este problema ha sido objeto de estudio en múltiples disciplinas, despertando interés en campos como la neurología, la psicología y la computación (Jacob, 1991; Rayner, 1998). El seguimiento preciso de dichos movimientos habilita diversas aplicaciones tecnológicas y clínicas, tales como dispositivos de realidad virtual (Moreno-Arjonilla et al., 2024), sistemas de detección de somnolencia y atención en conductores (Kapitaniak et al., 2015), herramientas de apoyo al diagnóstico clínico (Opwonya et al., 2021), entre otras.

Los sistemas de seguimiento ocular comerciales están limitados en frecuencia (≈ 200 – 2000 Hz) (Pupil Labs, 2023; SR Research, 2022). Aunque estas tasas suelen resultar suficientes, ciertas aplicaciones requieren mayor resolución temporal para capturar dinámicas rápidas de la mirada, por ejemplo, en el análisis de latencias sacádicas en el Trastorno por Déficit de Atención e Hiperactividad. (Rommelse et al., 2008).

Las cámaras de eventos se convierten en una herramienta capaz de tratar algunos de estos problemas. Estos son dispositivos que, a diferencia de las cámaras tradicionales, capturan los cambios de intensidad de cada pixel individual, de forma independiente y asincrónica. Cuando el cambio supera un cierto umbral, se genera un evento, el cual se representa como una tupla (t, x, y, p) donde t es una marca temporal (*timestamp*), x e y es la posición del pixel afectado y p la polaridad del mismo, siendo 1 si es un cambio de intensidad positivo y -1 si es negativo. Este mecanismo de operación permite alcanzar una resolución temporal muy alta, baja latencia, un alto rango dinámico y bajo consumo de energía (Gallego et al., 2020).

Uno de los trabajos fundacionales para el seguimiento ocular con cámaras de eventos es el de Angelopoulos et al., 2021, donde se propuso un método híbrido basado en eventos e imágenes capaz de seguir la pupila y estimar la mirada a una frecuencia 10 kHz o mayor, reportando un rendimiento comparable al de los sistemas de seguimiento ocular comerciales. Su enfoque híbrido permite aplicar métodos de procesamiento de imágenes tradicionales a la vez que se aprovecha la alta resolución temporal de los eventos para conseguir un seguimiento continuo y en tiempo real. A pesar de su impacto, aún no existe una implementación completa y de libre acceso.

En este trabajo proponemos una implementación completa del método de Angelopoulos et al., 2021, y presentamos un método de detección de parpadeos utilizando únicamente eventos, el cual podría utilizarse como parte del preprocesamiento en *datasets* similares. Los parpadeos son movimientos rápidos donde el ojo se cierra, y generan una gran cantidad de eventos dificultando el seguimiento preciso de la pupila.

La estructura del trabajo se describe a continuación. La Sección 2 revisa la literatura relacionada para el problema del seguimiento ocular. En la Sección 3 describimos el *dataset* propuesto por Angelopoulos et al., 2021, que utilizaremos para los experimentos. La Sección 4 detalla los métodos utilizados y los resultados de su evaluación se detallan en la Sección 5. Finalmente, en la Sección 6 las conclusiones y pasos futuros.

El repositorio con la implementación del método propuesto se encuentra en github.com/nicocarrasco703/high_freq_event_eye_tracking.

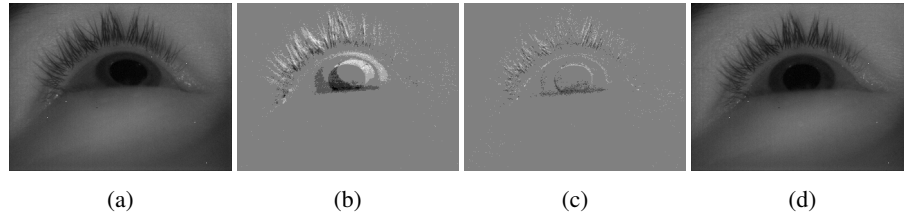


Figura 1: Sujeto de ejemplo durante una sacada. (a) La pupila en su posición inicial. (b, c) La acumulación de eventos durante el movimiento sacádico. (d) Posición final.

2. Trabajo relacionado

El seguimiento ocular mediante cámaras en visión artificial ha evolucionado desde métodos basados en modelos (Guestrin & Eizenman, 2006; Wang & Ji, 2017), hasta métodos basados en la apariencia y el aprendizaje profundo (Cheng et al., 2024). Sin embargo, estos traen consigo limitaciones que son inherentes al uso de cámaras convencionales, las cuales afectan su efectividad para capturar ciertos movimientos. Otra de las desventajas es la alta latencia, que dificulta el seguimiento en tiempo real.

Es por esto que en los últimos años, la comunidad académica ha incrementado su interés por el uso de las cámaras de eventos para el problema del seguimiento ocular, logrando avances significativos y consiguiendo un rendimiento similar a los métodos que utilizan cámaras convencionales (Iddrisu et al., 2024).

3. Dataset

El *dataset* propuesto por Angelopoulos et al., 2021³ está compuesto por grabaciones binoculares de movimientos oculares de 24 sujetos. Las capturas se obtuvieron con dos cámaras DAVIS346b colocadas a aproximadamente 25 cm del sujeto, las cuales permiten grabación simultánea con imágenes en escala de grises a 25 FPS y eventos. Los movimientos oculares incluyen fijaciones, donde los ojos permanecen relativamente estables; sacadas, movimientos rápidos que desplazan el punto de mirada de un lugar a otro, y movimientos de seguimiento suave, los cuales posibilitan que la mirada siga continuamente un objetivo. En la Figura 1 se pueden observar muestras de imágenes junto con los eventos correspondientes representados como imágenes.

Se realizan dos experimentos, uno para sacadas y fijaciones, donde se les pidió a los sujetos fijar la mirada en un estímulo aleatorio en una grilla de 11×11 en el monitor, mostrado por única vez con una duración de 1.5 segundos. Y otro para seguimiento suave, en el cual los sujetos debían seguir un estímulo que se movía suavemente a través del monitor, siguiendo una trayectoria predecible, empezando desde arriba y moviéndose para abajo ocupando el ancho del monitor. Para cada sujeto se obtuvo una única grabación por cada ojo y la duración promedio de las grabaciones fue de 4.8 minutos.

Los estímulos fueron mostrados en un monitor de 40 pulgadas con una resolución de 1080×1920 a 40 cm de distancia del sujeto.

³ Disponible en github.com/aangelopoulos/event_based_gaze_tracking

4. Metodología

Los eventos producidos por la cámara de eventos llegan como una secuencia de datos disperso en el tiempo $E = \{(x_k, y_k, t_k, p_k)\}_{k>0}$. Esta dispersión otorga varias ventajas, como la capacidad de procesarlos eficientemente a gran velocidad. No obstante, en esta forma no es posible aplicar métodos de seguimiento basados en imágenes.

El método propuesto utiliza la señal multimodal de la cámara de eventos: toma la ventaja de utilizar eventos para aprovechar su mayor resolución temporal a la vez que se apoya en el uso de técnicas de procesamiento de imágenes tradicionales, consiguiendo mayor robustez. En síntesis, primero se ajusta un modelo 2D del ojo a partir de imágenes y eventos, luego se construye un mapeo para calcular el punto de mirada o *point of gaze* que representa el punto en la pantalla donde el sujeto está mirando. En los intervalos entre imágenes se generan estimaciones de acuerdo a la llegada asincrónica de eventos generada por el movimiento de la pupila, permitiendo el seguimiento de la pupila a mayor frecuencia.

4.1. Preprocesamiento

Los parpadeos son movimientos rápidos donde el ojo se cierra, generando una gran cantidad de eventos alrededor del área de interés, esto dificulta el seguimiento preciso de la pupila causando perturbaciones y deformaciones en la elipse ajustada a la pupila. Desarrollamos un detector de parpadeos que permite segmentar los intervalos temporales donde se producen parpadeos y suprimirlos de la secuencia final para luego procesarlos con el algoritmo de ajuste de elipses.

Actividad temporal. Construimos un perfil de actividad (*activity profile*) que modela el comportamiento de los eventos en el tiempo (Lenz et al., 2020). Para cada evento entrante $e_i = (x_i, y_i, t_i, p_i)$, actualizamos la actividad temporal por polaridad $A_p(t_i)$ de la siguiente manera:

$$A_p(t_i) = A_p(t_{u_p}) e^{-\frac{t_i - t_{u_p}}{\tau}} + \frac{1}{\mu} \mathbb{I}[p_i = p] \quad (1)$$

donde t_{u_p} denota el timestamp del evento más reciente de polaridad $p \in \{+1, -1\}$ anterior a t_i , τ actúa como un factor de decaimiento temporal, μ es un factor que determina el aporte de cada evento y $\mathbb{I}$ es la función indicadora. Para cada polaridad p se computan dos señales A_{ON} y A_{OFF} respectivamente, definiendo 1 como el valor inicial de cada una.

Detección de parpadeos. El *activity profile* permite analizar la dinámica temporal de las distintas etapas del movimiento de apertura y clausura del ojo durante un parpadeo como se observa en la Figura 2. El punto más bajo entre ambos picos indica un estado transitorio del cierre a la apertura (Figura 2b).

Dado que la llegada de los eventos de polaridad positiva y negativa es asincrónica, discretizamos el tiempo en intervalos de 2 ms y computamos la actividad allí mediante interpolación lineal fragmentaria con τ en 15 ms y μ en 10000.

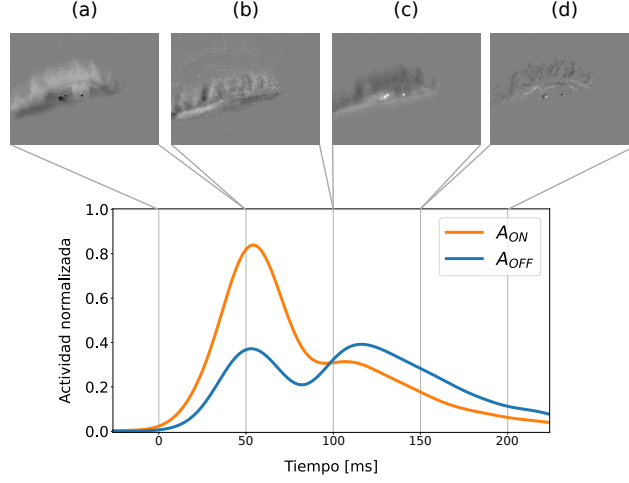


Figura 2: *Activity profile* de un parpadeo de ~ 200 ms con τ de 15 ms realizado por un sujeto. La actividad se normaliza y se le aplica un filtro gaussiano para suavizarla. (a) El cierre del ojo genera una gran cantidad de eventos positivos. (b) El ojo permanece cerrado, se estabiliza la actividad ON y OFF. (c) El ojo comienza a abrirse, se generan eventos negativos en su mayoría. (d) La actividad decae al estabilizarse el ojo.

Para detectar el patrón temporal de los parpadeos, nuestro método busca los cruces por cero de la señal $A_{ON} - A_{OFF}$ suavizada con filtro Gaussiano. De esta forma, los ceros representan candidatos de los momentos de reapertura del parpado. Luego, definimos una ventana temporal alrededor de cada candidato de acuerdo a la distribución de la duración de los parpadeos en el *dataset* y filtramos aquellos que contengan un pico a izquierda y derecha con suficiente prominencia. La estimación de la duración total del parpadeo resulta de la suma del ancho de los picos.

4.2. Modelo paramétrico

Definimos un modelo paramétrico $\mathcal{M} = \{\mathcal{E}, \mathcal{P}, \mathcal{C}\}$ que consiste en curvas paramétricas para modelar distintas partes del ojo (Tian et al., 2000). En este modelo, $\mathcal{E}$ es una elipse que representa la pupila, $\mathcal{P}$ es una parábola que modela el parpado y $\mathcal{C}$ es un círculo que describe el destello reflejado en el ojo, producto la luz IR utilizada en el entorno de grabación.

En nuestro método, solo consideramos la elipse que define la pupila ya que a partir de ésta es que realizaremos el seguimiento de la mirada. La elipse $\mathcal{E} = (a, h, b, g, f) \in \mathbb{R}^5$ está expresada por los parámetros de la ecuación cuadrática que define una elipse:

$$E(x, y) = ax^2 + hxy + by^2 + gx + fy + d \quad (2)$$

donde se fija $d = -1$ por conveniencia, al ser un factor de escala arbitrario que corresponde al desplazamiento del plano que corta la sección cónica definida por $\mathcal{E}$.

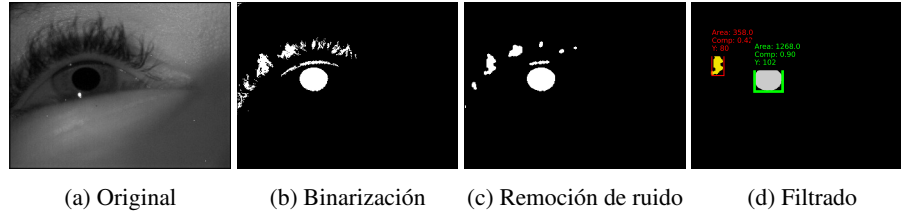


Figura 3: Secuencia de procesamiento de una imagen para obtener puntos candidatos. (a) La imagen original. (b) Resultado de binarización. (c) Eliminación de ruido mediante operaciones morfológicas. (d) Filtrado por área y *compactness*.

En el momento que llega una imagen o un evento (uno o muchos), los parámetros de la elipse son actualizados resolviendo un sistema de ecuaciones mediante cuadrados mínimos, que se detallará en la Sec. 4.3. Los puntos que forman parte de la elipse cumplen la ecuación 2. Previo a esto es necesario realizar una selección de puntos candidatos para realizar la actualización.

Filtrado de puntos de una imagen. Dada una imagen de entrada, la pupila suele observarse como una región aproximadamente circular y de baja intensidad (oscura), como se ve en la Figura 3a. Nuestro objetivo es aislar esa región y extraer solo los puntos de interés asociados. Para ello aplicamos la siguiente secuencia de procesamiento de imagen.

Primero realizamos una binarización utilizando un umbral inicial θ (Figura 3b). Luego aplicamos una operación morfológica de apertura (*opening*), empleando como elemento estructurante un círculo discretizado de radio σ . Esta operación permite eliminar pequeñas regiones aisladas producidas por ruido, preservando al mismo tiempo las estructuras de mayor tamaño (Figura 3c).

Como el resultado obtenido de los pasos previos no fue suficiente para segmentar correctamente la pupila, optamos por apartarnos del enfoque del trabajo original y utilizar un filtrado por área para las regiones detectadas mediante componentes conexas, conservando únicamente aquellas cuya área se encuentren dentro de un rango predefinido (entre 300 y 4000 píxeles). Este criterio descarta tanto regiones demasiado pequeñas (ruido residual) como regiones demasiado grandes que no corresponden a la pupila. El último criterio de selección es el valor de *compactness*, que dada una región R se define como $compactness(R) = \frac{4\pi \cdot area}{perimetro^2}$.

Esta medida toma valores cercanos a 1 para formas aproximadamente circulares, mientras que disminuye a medida que la forma se vuelve más irregular o alargada. Bajo la hipótesis de que la pupila es aproximadamente circular, esperamos que su valor de *compactness* sea mayor que el del resto de las regiones detectadas (Goni et al., 2004). Una vez obtenida la región de la pupila (Figura 3d), aplicamos un detector de bordes (Canny, 1986) para extraer su contorno, quedándonos finalmente con el conjunto de puntos C_{img} que describen su borde.

En algunos casos, debido a variaciones de contraste en la secuencia de grabación, el uso de un umbral de binarización fijo permite el paso de ruido que no puede ser eliminado completamente mediante operaciones morfológicas. En estos escenarios, resulta

Algorithm 1 Ajuste en línea de $\mathcal{E}$

```
1:  $\bar{A} \leftarrow \text{Id}_{5 \times 5}, \bar{b} \leftarrow 0_5$   $\triangleright$  Inicializar  $\bar{A}$  and  $\bar{b}$ 
2: for each entry in datastream do
3:   if entry is an event  $e$  then
4:      $C_{\text{evt}} \leftarrow C_{\text{evt}} \cup \text{process\_event}(e)$ 
5:   if entry is a frame  $f$  then
6:      $C_{\text{img}} \leftarrow \text{process\_frame}(f)$ 
7:   if  $|C_{\text{evt}}| = 20$  or  $|C_{\text{img}}| > 0$  then
8:      $C \leftarrow C_{\text{evt}} \cup C_{\text{img}}$ 
9:      $C_{\text{evt}}, C_{\text{img}} \leftarrow \emptyset$ 
10:     $A \leftarrow \text{Id}_{5 \times 5}, b \leftarrow 0_5$   $\triangleright$  Re-inicializar  $A$  y  $B$ 
11:    for all  $(x, y) \in C$  do
12:       $w_{x,y}^T \leftarrow (x^2, xy, y^2, x, y)$   $\triangleright$  Ec. (5)
13:       $A \leftarrow A + w_{x,y} w_{x,y}^T$   $\triangleright$  Ec. (5)
14:       $b \leftarrow b + w_{x,y}$   $\triangleright$  Ec. (5)
15:       $\bar{A} \leftarrow \gamma \bar{A} + (1 - \gamma) A$   $\triangleright$  Ec. (6)
16:       $\bar{b} \leftarrow \gamma \bar{b} + (1 - \gamma) b$   $\triangleright$  Ec. (6)
17:     $\mathcal{E} \leftarrow \text{solve}(\bar{A}, \bar{b})$ 
```

necesario recalcular el umbral de manera adaptativa para discriminar de forma más efectiva la pupila del resto de la imagen. Para ello, empleamos el método Multi-Otsu (Liao et al., 2001) aplicado sobre una región de interés que incluye tres clases: la pupila, el iris y, en ciertos casos, el reflejo de luz infrarroja. Seleccionamos el umbral correspondiente a la clase de menor intensidad. Este enfoque mostró un mejor desempeño en comparación con el uso de un umbral fijo en dichas condiciones.

Filtrado de puntos de eventos. Al recibir un evento $e = (x, y, t, p)$, el criterio de selección es la distancia entre la posición donde ocurre el evento y la última elipse estimada:

$$e \in C_{\text{evt}} \iff \|P_{\mathcal{E}_k}(x, y) - \mathcal{E}_k\|_2 < \delta \quad (3)$$

donde $\mathcal{E}_k$ es la elipse ajustada en el último *timestamp* $k < t$. δ es un umbral arbitrario. $P_{\mathcal{E}_k}(x, y)$ es la proyección ortogonal del punto (x, y) a la elipse. Esta proyección se puede calcular de distintas formas, nosotros implementamos un método iterativo usando multiplicadores de Lagrange.

4.3. Ajuste de elipse a la pupila

Utilizamos cuadrados mínimos para ajustar los puntos candidatos a la elipse de nuestro modelo. El ajuste de elipses se realiza por lotes de N eventos, siendo $N = 20$ el balance óptimo reportado por Angelopoulos et al., 2021 para conseguir estimaciones robustas. Buscamos minimizar el error cometido:

$$\mathcal{E}^* = \arg \min_{\mathcal{E} \in \mathbb{R}^5} \sum_{(x,y) \in C} E_{\mathcal{E}}(x, y)^2 \quad (4)$$

siendo $E_{\mathcal{E}}(x, y)$ el error cometido en el ajuste del punto (x, y) a la elipse de parámetros $\mathcal{E}$. Resolvemos las ecuaciones normales mediante el sistema $A\mathcal{E} = b$, con

$$A = \sum_{(x,y) \in C} w_{x,y} w_{x,y}^T, \quad b = \sum_{(x,y) \in C} w_{x,y}, \quad w_{x,y} = (x^2, xy, y^2, x, y)^T \quad (5)$$

Comenzamos con una estimación inicial $\bar{A}$ y $\bar{b}$. Al recibir datos del sensor, realizamos una actualización de los parámetros de la elipse a partir de actualizar la matriz $\bar{A}$ y el vector $\bar{b}$ en línea, lo que es posible gracias a que estos términos son sumatorias. Formalmente, construimos $\bar{A}_t$ y $\bar{b}_t$ a partir de los puntos candidatos recibidos en el tiempo t , como se indica en la Ecuación 5. Luego la actualización se produce de la siguiente manera:

$$\bar{A}_{t+1} = \gamma \bar{A}_t + (1 - \gamma) A_t, \quad \bar{b}_{t+1} = \gamma \bar{b}_t + (1 - \gamma) b_t, \quad \text{con } \gamma \in [0, 1] \quad (6)$$

El parámetro γ puede interpretarse como un factor de regularización espacio-temporal. Cuando γ toma valores cercanos a 0, el modelo prioriza la información reciente, atenuando la influencia de observaciones pasadas. Valores cercanos a 1 refuerzan la contribución de los estados previos, disminuyendo el impacto de nuevas mediciones. En el Algoritmo 1 se muestra un pseudo-código del método de ajuste de elipses completo.

4.4. Gaze mapper

En esta etapa, buscamos estimar el *point of gaze*, las coordenadas (x_s, y_s) de la pantalla donde el sujeto está mirando (ver Figura 4). Buscamos entonces una función $f: \mathcal{M} \rightarrow \mathbb{R}^2$, que será el *gaze mapper*.

Se adopta un enfoque común en la literatura (Morimoto & Mimica, 2005), que consiste en aproximar (x_s, y_s) a partir del centro de la pupila (x_e, y_e) mediante dos polinomios, uno por cada coordenada:

$$R_{\theta}(x_e, y_e) = \begin{pmatrix} x_s \\ y_s \end{pmatrix} = \begin{pmatrix} R_{\theta_x}(x_e, y_e) \\ R_{\theta_y}(x_e, y_e) \end{pmatrix} \quad (7)$$

Este enfoque se beneficia por su bajo requerimiento computacional, por lo que es ideal para mantener la frecuencia de actualización de los eventos.

Calibración. Para encontrar los coeficientes $\theta_{x,y}$ de estos polinomios, se requiere de un proceso de calibración que consiste en generar pares $\{(x_e, y_e), (x_s, y_s)\}$ con (x_e, y_e) el centro de la elipse estimada $\mathcal{E}$, y (x_s, y_s) el punto en la pantalla donde el sujeto está mirando, formalmente buscamos:

$$R_{\theta_{x,y}}(x_e, y_e) = \alpha_i x_e^2 + \gamma_i x_e y_e + \beta_i y_e^2 + \epsilon_i x_e + \zeta_i y_e + \eta_i \quad (8)$$

Podemos calcular $\theta_{x,y}$ resolviendo los siguientes cuadrados mínimos:

$$\arg \min_{\theta_x} \|R_{\theta_x}(x_e, y_e) - x_s\|^2 \quad \text{y} \quad \arg \min_{\theta_y} \|R_{\theta_y}(x_e, y_e) - y_s\|^2 \quad (9)$$

Los centros de las elipses (x_e, y_e) se pueden calcular a partir de los coeficientes de $\mathcal{E} = (a, h, b, g, f)$ de la siguiente manera.

$$x_e = \frac{2bg - hf}{h^2 - 4ab} \quad \text{y} \quad y_e = \frac{2af - hg}{h^2 - 4ab}$$

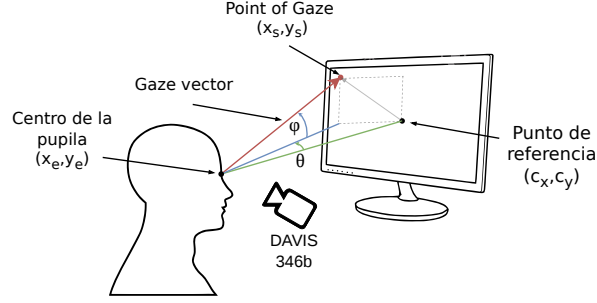


Figura 4: Esquema geométrico del sistema de seguimiento ocular. El *gaze vector* se origina en el centro de la pupila (x_e, y_e) y se proyecta hacia la pantalla, donde define el *point of gaze* (x_s, y_s) . Los ángulos θ (horizontal) y ϕ (vertical) parametrizan la dirección del *gaze vector* respecto al sistema de referencia centrado en (c_x, c_y) .

4.5. Métricas de evaluación

Accuracy: Refiere a el error angular promedio en la estimación de la mirada cuando un sujeto fija su vista en un punto conocido. Es decir:

$$\text{Accuracy} = \frac{1}{n} \sum_{i=1}^n \|\hat{d}_i - d_i\|_2$$

donde $\hat{d}_i = (\hat{\phi}_i, \hat{\theta}_i)$ es la estimación y $d_i = (\phi_i, \theta_i)$ es el verdadero ángulo de visión.

Precision: Se calcula como el desvío estándar empírico, en ángulo visual, entre las estimaciones angulares de todos los sujetos. Siendo $\tilde{d}$ la media de las estimaciones:

$$\text{Precision} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\hat{d}_i - \tilde{d})^2}$$

5. Resultados experimentales

En esta sección proponemos una serie de métricas para evaluar nuestro detector de parpadeos como herramienta de anotación. Para analizar su calidad como herramienta de anotación aproximada, validamos el método con un subconjunto de sujetos del *dataset*.

Describimos la metodología de evaluación utilizada en el trabajo de Angelopoulos et al., 2021 que ocuparemos para comparar nuestra implementación del método. Utilizamos un subconjunto preliminar de sujetos para evaluar el seguimiento de la pupila, así como el *gaze mapper*.

5.1. Detector de parpadeos

Para validar el método, anotamos manualmente los segmentos temporales de parpadeos para un subconjunto de 14 sujetos. Como ambas modalidades de la cámara de

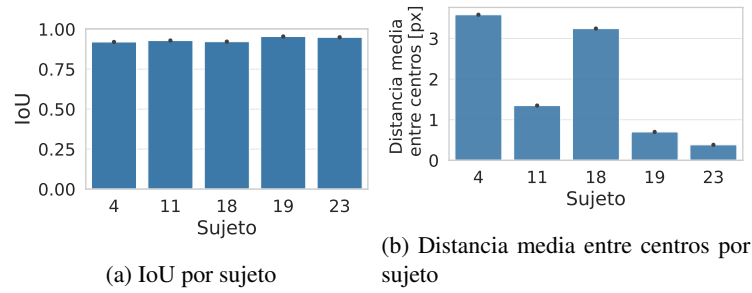


Figura 5: Métricas de evaluación para el ajuste de la pupila.

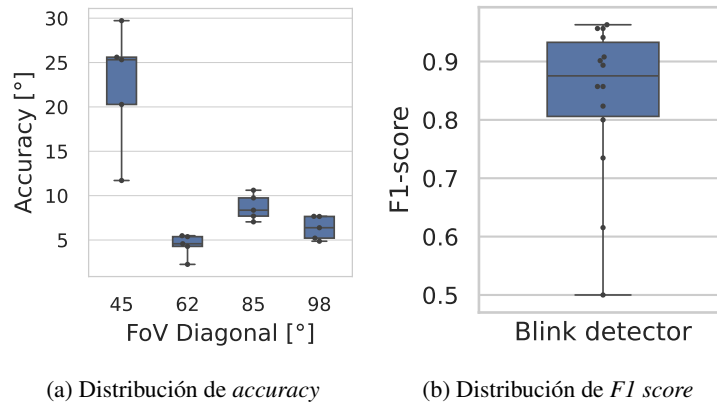


Figura 6: Métricas de evaluación. (a) Box-plot de *accuracy* para distintos FoV diagonales. (b) Distribución de *F1 score* para detección de parpadeos por sujeto.

eventos están sincronizadas espacio-temporalmente, los intervalos fueron delimitados a partir de la inspección visual de las imágenes. En consecuencia, no están estrictamente alineadas con el *onset* y *offset* de los parpadeos. Un total de 359 parpadeos fue obtenido a partir de las anotaciones de dos de los autores del trabajo.

Para garantizar la consistencia entre anotadores, propusimos un criterio de anotación preliminar para especificar casos borde y delimitar la duración de lo que consideramos un parpadeo. En la primera etapa, cada anotador etiquetó el mismo subconjunto de 6 sujetos y se evaluó el acuerdo entre anotadores, lo que permitió resolver los casos de discrepancia y refinar el criterio de anotación. A partir de estas consideraciones, cada anotador continuó el proceso de etiquetado de los 8 sujetos restantes, separados en dos conjuntos disjuntos.

La evaluación del acuerdo entre anotadores se realizó a partir de *agreement rate* y *Intersection-over-Union* (IoU) promedio. Para calcular el *agreement rate*, consideramos un *greedy matching* entre los segmentos de cada anotador con un IoU superior a 0.5. Los segmentos que no pertenecen al conjunto de coincidencias fueron considerados desacuerdos, y cada segmento fue emparejado a lo sumo con un único segmento del otro anotador. Finalmente, el *agreement rate* se definió como la proporción entre el

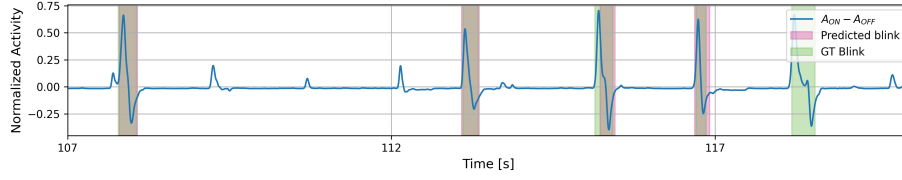


Figura 7: Recorte temporal de la señal $A_{ON} - A_{OFF}$ de un sujeto junto con los intervalos anotados (GT) y las predicciones del método. El patrón generado deviene de la firma temporal analizada en la Fig. 2, el cual puede ser detectado a partir de picos positivos y negativos aledaños a los cruces por cero en esta señal.

número de segmentos coincidentes y el número total de segmentos anotados por ambos anotadores. Por otra parte, el IoU promedio se calculó únicamente sobre los intervalos pertenecientes al conjunto de coincidencias.

El *agreement rate* obtenido al cabo de la primera etapa fue de 93.04% con un umbral de IoU de 0.5, y un IoU promedio de 90.06%. Estos resultados evidenciaron un nivel de acuerdo en la tarea de anotación más que aceptable y permitieron discutir y refinar los criterios de anotación previo a la siguiente etapa de anotación.

Para evaluar el método que permite segmentar temporalmente los parpadeos a partir de la señal $A_{ON} - A_{OFF}$, calculamos el *F1 score* por sujeto a partir del mismo esquema de *greedy matching* con IOU @ 0.5. Ajustamos los umbrales de prominencia mínima para detectar los picos positivos y negativos en $A_{ON} - A_{OFF}$ minimizando el *Micro F1* entre todos los sujetos. En la Figura 7 puede observarse la detección del algoritmo en un recorte temporal de dicha señal para un sujeto específico.

Los resultados de la evaluación se reportan en la Figura 6b, donde se observa la distribución del *F1 score* por sujeto para la mejor combinación de umbrales seleccionados. El *Micro F1* obtenido fue de 88.18% y el *Macro F1* de 83.63%, mostrando un desempeño aceptable dada la simpleza del método.

5.2. Ajuste de elipse

Para evaluar la efectividad del ajuste de la pupila, comparamos la última estimación de la elipse basada en eventos, $\mathcal{E}_{\text{evt}}$, obtenida inmediatamente antes de la llegada de una imagen, con la estimación de la elipse asociada a ese imagen, $\mathcal{E}_{\text{img}}$. Como métricas de comparación, utilizamos IoU (Figura 5a) y la distancia media entre centros de las elipses (Figura 5b). Las métricas de IoU muestran buen solapamiento entre las máscaras de la pupila, así como una distancia entre centros media de menos de 1.5 píxeles, llegando en peor caso hasta 3.5 píxeles, mostrando un seguimiento de la pupila mayormente estable y suave.

5.3. Gaze mapper

Para evaluar el *gaze mapper*, calculamos la relación entre el ángulo horizontal θ , el ángulo vertical ϕ y las coordenadas (x_s, y_s) de la pantalla donde el sujeto está mirando (ver Figura 4). Esta relación es trigonométrica ya que la pantalla se encuentra a una

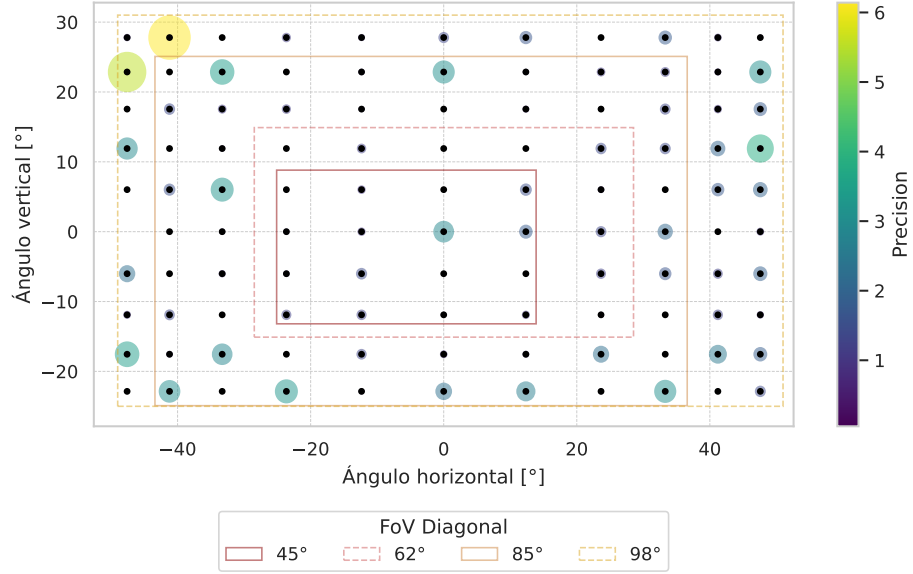


Figura 8: Precision para distintos FoV diagonales. Los puntos de color negro representan el estímulo de tamaño 1.3° , nuestro *ground truth*. El radio de los círculos centrados en los *ground truth* y divididos por un factor de escala de 2, representan *precision* como la desviación estándar recortada de la estimación promedio de todos los sujetos.

distancia fija (40 cm). Si (c_x, c_y) representa el centro de la pantalla a una distancia fija D , la conversión se calcula como:

$$\theta = \frac{180}{\pi} \tan^{-1}(|x_s - c_x|/D), \quad \phi = \frac{180}{\pi} \tan^{-1}(|y_s - c_y|/D)$$

Realizamos un proceso de calibración, como se detalla en la Sec. 4.4. Cada vez que se reporta una métrica (*accuracy* o *precision*) para un FoV (*Field of View*) particular, calibramos en los puntos de índice par y evaluamos en los impares, luego hacemos lo opuesto y promediamos los resultados. Debido a la ausencia de *ground truth* para las estimaciones basadas en eventos, no es posible evaluar directamente su exactitud. Por este motivo, evaluamos las estimaciones obtenidas a partir de imágenes empleando como referencia los puntos de *ground truth* asociados a la posición en pantalla donde el sujeto fijaba la mirada durante los movimientos sacádicos.

Los resultados preliminares muestran un error en grados de más de 20° en el FoV diagonal de 45° , mientras que en los FoV más grandes el error tiende a estabilizarse, bajando a menos de 10° (Figura 6a). Si bien en Angelopoulos et al., 2021 se reportan valores de *accuracy* más bajos, solo muestran las métricas de un mejor caso para un único sujeto (en un rango de $0.45 - 1.75^\circ$). Por otro lado, la Figura 8 se visualiza la *precision* como la desviación estándar (ver Sec. 4.5) con un recorte simétrico de 20% en cada extremo, los círculos centrados en los puntos negros (*ground truth*) representan la dispersión estadística de las predicciones entre los sujetos. Notamos que en los FoV más grandes, principalmente en los puntos en los bordes, esta dispersión incrementa hasta

6°. Esto se explica por la oclusión y también por el impacto que tienen los pequeños cambios en las predicciones del centro de la pupila.

6. Conclusiones y pasos futuros

El método propuesto por Angelopoulos et al., 2021 constituye una base sólida para la investigación en seguimiento ocular mediante cámaras de eventos. Su enfoque híbrido permite combinar ventajas de los sensores tradicionales, ofreciendo —de acuerdo con la literatura original— un compromiso adecuado entre robustez y precisión con bajos requerimientos de cómputo y consumo energético. No obstante, presenta las limitaciones propias de su diseño original: su dependencia estricta de imágenes (capacidad ausente en diversos sensores de eventos), su susceptibilidad potencial al *slippage*, y la ausencia de un *ground truth* preciso para el ajuste del modelo en el dominio de eventos.

En este trabajo se intentó reproducir la implementación original con la mayor fidelidad posible. Durante este proceso surgieron múltiples dificultades derivadas de la falta de detalles implementativos en la publicación original. Obligando a tomar decisiones de diseño que potencialmente difieren del método original.

Las métricas de *accuracy* y *precision* obtenidas no alcanzan los valores reportados en el trabajo original. Esta discrepancia puede explicarse por diversos factores, entre ellos diferencias en la selección de hiperparámetros, posibles variaciones en la implementación del módulo de *gaze mapping*. A pesar de ello, el sistema desarrollado logra un seguimiento de la pupila mayormente estable en el conjunto de evaluación preliminar, aprovechando los principios fundamentales introducidos en el trabajo original.

Por otra parte, el método de detección de parpadeos propuesto constituye una contribución relevante, dado el impacto negativo de los parpadeos en el seguimiento ocular. Si bien se obtiene una tasa de detección satisfactoria, el método resulta sensible al ruido y presenta falsos positivos, especialmente debido a la similitud entre los perfiles de actividad generados por movimientos sacádicos y aquellos generados por parpadeos o bien por movimientos contráctiles del párpado. No obstante, su formulación basada exclusivamente en eventos lo hace compatible con sistemas que prescinden completamente de información basada en imágenes.

Como línea de trabajo futuro, consideramos que el método de Angelopoulos et al., 2021 puede servir como un modelo de referencia adecuado para el desarrollo de enfoques de seguimiento ocular basados únicamente en eventos.

Referencias

- Angelopoulos, A. N., Martel, J. N., Kohli, A. P., Conradt, J., & Wetzstein, G. (2021). Event-Based Near-Eye Gaze Tracking Beyond 10,000 Hz. *IEEE Trans. Vis. Comput. Graph.*, 27(5), 2577-2586.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, (6), 679-698.
- Cheng, Y., Wang, H., Bao, Y., & Lu, F. (2024). Appearance-Based Gaze Estimation With Deep Learning: A Review and Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12), 7509-7528.

- Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., et al. (2020). Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(1), 154-180.
- Goni, S., Echeto, J., Villanueva, A., & Cabeza, R. (2004). Robust Algorithm for Pupil-Glint Vector Detection in a Video-Oculography Eyetracking System. *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, 941-944.
- Guestrin, E., & Eizenman, M. (2006). General Theory of Remote Gaze Estimation Using the Pupil Center and Corneal Reflections. *IEEE Transactions on Bio-medical Engineering*, 53(6), 1124-1133.
- Hansen, D. W., & Pece, A. E. (2005). Eye Tracking in the Wild. *Computer Vision and Image Understanding*, 98(1), 155-181.
- Iddrisu, K., Shariff, W., Corcoran, P., O'Connor, N. E., Lemley, J., & Little, S. (2024). Event Camera-Based Eye Motion Analysis: A Survey. *IEEE Access*, 12, 136783-136804.
- Jacob, R. J. K. (1991). The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Trans. Inf. Syst.*, 9(2), 152-169.
- Kapitaniak, B., Walczak, M., Kosobudzki, M., Jóźwiak, Z., & Bortkiewicz, A. (2015). Application of eye-tracking in drivers testing: A review of research. *Int. Journal of Occupational Medicine and Environmental Health*, 28, 941-954.
- Lenz, G., Ieng, S.-H., & Benosman, R. (2020). Event-Based Face Detection and Tracking Using the Dynamics of Eye Blinks. *Frontiers in Neuroscience*, 14.
- Liao, P.-S., Chen, T.-S., Chung, P.-C., et al. (2001). A fast algorithm for multilevel thresholding. *J. Inf. Sci. Eng.*, 17(5), 713-727.
- Moreno-Arjonilla, J., López-Ruiz, A., Jiménez-Pérez, J. R., Callejas-Aguilera, J. E., & Jurado, J. M. (2024). Eye-tracking on virtual reality: a survey. *Virtual Reality*, 28(1), 38.
- Morimoto, C. H., & Mimica, M. R. (2005). Eye Gaze Tracking Techniques for Interactive Applications. *Computer Vision and Image Understanding*, 98(1), 4-24.
- Opwonya, J., Doan, D., Kim, S. G., Kim, J., Ku, B., Kim, S., Park, S., & Kim, J. U. (2021). Saccadic Eye Movement in Mild Cognitive Impairment and Alzheimer's Disease: A Systematic Review and Meta-Analysis. *Neuropsychology Review*, 32, 193-227.
- Pupil Labs. (2023). *Neon Eye tracker*. Consultado el 13 de abril de 2026, desde <https://pupil-labs.com/products/neon>
- Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3), 372.
- Rommelse, N. N. J., Van der Stigchel, S., & Sergeant, J. A. (2008). A Review on Eye Movement Studies in Childhood and Adolescent Psychiatry. *Brain and Cognition*, 68(3), 391-414.
- SR Research. (2022). *EyeLink 1000 Plus Eye tracker*. Consultado el 13 de abril de 2026, desde <https://www.sr-research.com/eyelink-1000-plus>
- Tian, Y.-l., Kanade, T., & Cohn, J. (2000). Dual-State Parametric Eye Tracking. *Proc. 4th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, 110-115.
- Wang, K., & Ji, Q. (2017). Real time eye gaze tracking with 3d deformable eye-face model. *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 1003-1011.