

Sistema de Seguimiento de Pupilas de Bajo Costo Computacional

Bertoni Luciano, Dwek Matias, Iribarren Rodrigo, Laurutis Nicolás, Rondinella Rocco, Tachella Julián
 Contacto: madwek@itba.edu.ar, jtachell@itba.edu.ar

Abstract—Un nuevo método de detección y rastreo de pupilas en tiempo real es propuesto, diseñado para la utilización con cámaras de resolución básica como webcams. Este método se basó en una combinación de algoritmos de bajo costo ya existentes, con una nueva propuesta para la rápida detección de la pupila. Además se analizó la utilización de filtros de Kalman para acondicionar las mediciones de la posición de la pupila. Finalmente se analizaron los resultados obtenidos con la utilización del método propuesto.

Index Terms—Gaze Tracking, Image Signal processing, Pupil Detection, Kalman Filtering

I. INTRODUCCIÓN

LA DETECCIÓN y el seguimiento de los ojos se presenta como una interacción natural entre el hombre y la máquina. Las aplicaciones de este tipo de procesamiento de imágenes son numerosas, tanto en hardware como en software, con particular interés cuando se desarrollan en tiempo real. Entre otras se pueden mencionar la interacción con una computadora para personas con dificultades motrices [1], implementación en proyectos ligados con el entrenamiento [3], como los videojuegos o inclusive aplicaciones más específicas como la detección de un conductor distraído al volante [2].

Entonces la incógnita que se presenta es si es posible desarrollar un sistema preciso de seguimiento de ojos con un costo computacional lo suficientemente bajo como para ser implementado en tiempo real. En efecto, a continuación se presentan los resultados del desarrollo realizado.

II. ANTECEDENTES

Hasta la actualidad han sido propuestos numerosos métodos de rastreo del movimiento de pupilas, especialmente para sistemas de control mediante parpadeos del usuario[4]. También se han propuesto sistemas de seguimiento en vivo con una cámara a una distancia mayor a 30 cm, diseñados con el objetivo de controlar equipos electrónicos con la vista humana. La infinidad de posibles aplicaciones de dicho control hace que la investigación en este campo sea prometedora. Los sistemas con mayor fidelidad a los movimientos de la pupila fueron logrados mediante iluminación activa infrarroja[5]. Sin embargo, a pesar de haber varias propuestas[6], no se han logrado implementar eficientemente sistemas de control mediante vista para webcams de bajo costo computacional. En especial dichos sistemas son interesantes para implementar en smartphones, ya que en la actualidad todos los smartphones traen cámaras de alta calidad, pudiéndose implementar aplicaciones que usen las cámaras frontal y trasera.

III. DESCRIPCIÓN DEL ALGORITMO

El sistema desarrollado se basa en los siguientes pasos fundamentales, resumidos en la figura 1. En primera instancia se detectan los ojos, luego se buscan los autovalores[10] de los mismos para su posterior seguimiento. Una vez comenzado el seguimiento de los ojos, se recortan ambos ojos y se detecta la posición y el radio de ambas pupilas frame por frame en tiempo real. Luego la posición y el radio de las pupilas es filtrado, para el cálculo de la posición de la vista. En caso de perder los puntos seguidos de los ojos, se regresa al paso inicial.

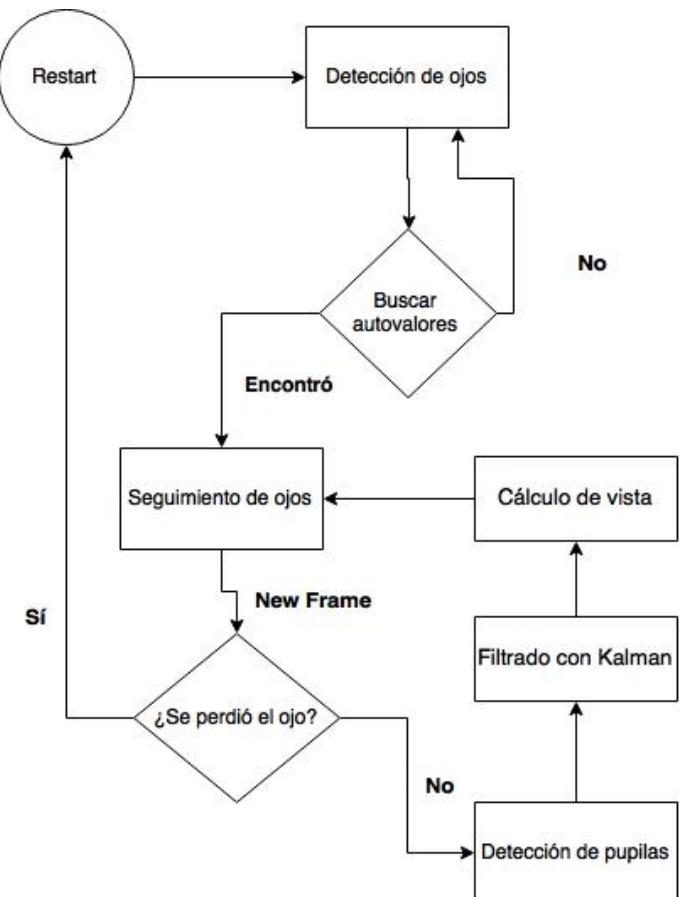


Fig. 1. Diagrama en bloques del funcionamiento del sistema

IV. DETECCIÓN DE OJOS

El primer paso en el seguimiento de las pupilas consiste en detectar la posición de los ojos de la persona. Para realizar

dicha tarea se utilizó el algoritmo propuesto por Viola y Jones en MATLAB[8], basado en la detección de características del tipo de Harr[8], debido a su bajo costo computacional. Teniendo en cuenta que los falsos positivos del algoritmo pueden representar un 40% de las detecciones, y siendo éstos inaceptables a la hora de encontrar los ojos para un seguimiento, se diseñó un algoritmo que redujo los falsos positivos a menos del 1%. Primero se busca la posición de la cara frontal en una captura de la webcam, filtrando según la cara más grande. Luego, se recorta dicha cara frontal y se busca el ojo derecho en el cuadrante izquierdo superior, también filtrando por el tamaño con el criterio del mayor tamaño. Para el ojo izquierdo se realiza el mismo procedimiento pero con el cuadrante superior derecho. De esta forma se reduce el error drásticamente, teniendo que correr el algoritmo solo tres veces, con tamaños reducidos en las últimas dos veces. En la última sección se mostrarán los resultados obtenidos en relación al tiempo de obtención de ambos ojos.

V. SEGUIMIENTO DE OJOS

Una vez detectada la posición de los ojos, es necesario seguir la posición de los mismos en la imagen, debido a que es muy costoso computacionalmente detectar los ojos en cada frame en real time. Por eso se investigaron las distintas técnicas de seguimiento, buscando la más adecuada para los ojos. Se utilizó el algoritmo basado en la publicación de Kanade [9], que propone calcular la transformación geométrica \bar{h} entre dos frames sucesivos que minimice el error cuadrático medio, mediante una búsqueda iterativa del tipo Newton-Raphson.

Además se tuvo que utilizar la técnica propuesta por Tomasi[10], para encontrar los puntos que presenten mejores características para el tracking, minimizando la necesidad de nuevas detecciones. Este algoritmo utiliza una técnica basada en la obtención de autovalores de la matriz de transformación entre frames (basada en \bar{h}) para hallar dichos puntos. Además se incorpora el peso de los errores entre el primer frame de detección y el n-ésimo frame de seguimiento, sin limitar el error a frames sucesivos, lo que otorga información fundamental acerca de la confianza en los puntos que se están siguiendo. En caso de que el error de un punto entre el primer frame y el último tome valores mayores a n pixels, se deja de seguir dicho punto. Este valor n debe ser ajustado según la práctica, y en la aplicación desarrollada se encontró el óptimo en $n = 4$.

Si se pierden la mayoría de los puntos, se debe volver a la etapa de detección y reiniciar el algoritmo de seguimiento. Cabe aclarar que la pérdida de los puntos de tracking es algo inevitable en un período de entre [20, 60]seg (según lo obtenido experimentalmente), con el movimiento de una persona promedio frente a la webcam. Este fenómeno es aprovechado beneficiamente por el sistema para realizar una nueva detección que obtenga puntos de seguimiento más confiables. Es necesario que dicho refresco se produzca en promedio cada intervalos mayores a 30seg, debido que el costo de la detección toma en un procesador actual típico aproximadamente 1seg, de lo que se obtiene que el 3.22% del tiempo el usuario pierde el control del sistema durante la detección y el 96.78% restante puede utilizar el sistema.

La pregunta que se debe estar haciendo el lector a esta altura es por qué no utilizar el mismo sistema de detección y seguimiento para las pupilas. La respuesta fue una de las principales motivaciones de este trabajo, ya que resulta ineficiente aplicar la misma técnica a las pupilas que a los ojos. La razón radica en el tamaño de las pupilas dentro de la imagen de una webcam: Mientras que el ojo ocupa, con la persona a unos 50 cm de la pantalla en resolución 800x600, aproximadamente unos 60x30 pixels, la pupila ocupa tiene un radio menor a los 5 pixels. Esto en conjunto con que el gradiente de colores en una pupila promedio es muy suave, hace que la detección de autovalores sea muy ineficiente y falle en la mayoría de los casos, y de no fallar, el seguimiento no supera los 5seg en el mejor de los casos. Por esto se investigó en una técnica de rápida detección de pupilas a partir del recorte del ojo, cuyo bajo costo computacional permita realizar dicha detección frame por frame.

VI. DETECCIÓN DE PUPILAS

Para poder determinar los movimientos de la pupila, en una primera instancia es necesario detectar donde se encuentra la misma. A partir de la imagen ya aislada de un ojo se implementó un algoritmo capaz de encontrar tanto el radio como la ubicación de la pupila. Aunque existen desarrollos que logran ubicar con gran presión el iris y la pupila [11], estos poseen dificultades en ser implementados en tiempo real.

En primer lugar, a la imagen del ojo se le aplica un filtro mediano bidimensional con el objetivo de remover el ruido de la captura inicial [7]. Este tipo de filtro no lineal reemplaza cada píxel por la media de los píxeles que lo rodean. El valor de la media será el valor del píxel que separe el resto de los valores en un conjunto que valores superiores al de la media y en uno con valores inferiores. En caso de que el total de píxeles que se toman sea par, se toma el promedio de los dos valores medios. Se pueden observar los resultados en la figura 2 , donde la subfigura 2a es la entrada y 2b es la salida. Cabe aclarar que el procesamiento de la imagen se realiza a partir de su escala de grises ya que esta sólo posee información sobre la intensidad de luz de la captura, teniendo un mejor SNR. Esto que mejora la detección de bordes y rasgos y, en este caso, de círculos.

Después, se realiza el ajuste de contraste de la imagen a partir de la ecualización del histograma de la misma. Esta transformación consiste en modificar la intensidad de los píxeles de tal manera que la distribución de grises resultante sea uniforme. Esta transformación otorga a la imagen un realce del contraste preservando su luminosidad, lo que permite una mejor detección de la pupila [12].

A continuación se convierte la imagen a binario con cierto umbral blanco/negro. Es decir que todo pixel que no posea una luminosidad mayor a este umbral será reemplazado por negro. Para condiciones medianas a altas de iluminación se encontraron mejores resultados con un umbral cercano a 0.17. En la figura 3 se muestran las imágenes obtenidas para distintos valores de umbral: 0.10, 0.15, 0.17 y 0.22.

Una vez filtrada la imagen original del ojo se procede a buscar el mejor círculo presente a partir de la transformación

circular de Hough con un valor de sensibilidad igual a 1 [13]. La transformación de Hough circular consiste en encontrar figuras que se asemejen a círculos imperfectos, donde los círculos se encuentran como máximos de un espacio llamado acumulador. Se acota la búsqueda de radios a partir de un proceso que se explicará en la sección VII. A partir del algoritmo de la transformación de Hough se entrega al resto del programa el radio y el centro de la pupila.

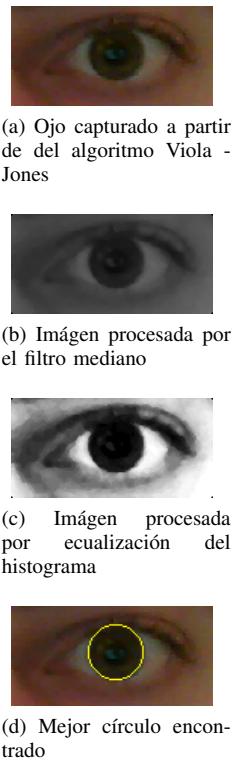


Fig. 2. Aplicación de distintos filtros

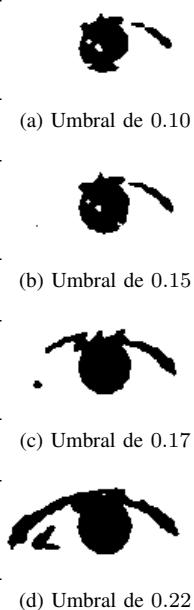


Fig. 3. Conversión de la imagen a binario

VII. FILTRADO DE LAS DETECCIONES DE PUPILA

El algoritmo de detección de pupilas debe perder precisión a cambio de ganar en menor costo computacional, permitiendo una ejecución en tiempo real. Esta pérdida en precisión genera que algunas detecciones tengan error y generen errores en el seguimiento de las pupilas. Dicho error puede representarse como ruido blanco con una distribución gaussiana alrededor de la media que es el centro de la pupila. Para mejorar la detección se buscó filtrar dicho ruido, reduciendo el *SNR* de la posición (x, y) de las pupilas.

Para lograr el filtrado óptimo de dicho ruido, se utilizaron filtros de Kalman[14], [15], aprovechando su condición de depender únicamente del último estado, ideal para implementación de un algoritmo en real time, que busca ocupar la menor cantidad de memoria posible. Los modelos típicos, utilizados en seguimiento de objetos, de velocidad constante o aceleración constante no aplican para el movimiento de la pupila, ya que se divide en movimientos rápidos sacádicos y movimientos de foco lento. Por esto, sabiendo que el error de la detección es mayor en el eje vertical, se filtraron las posiciones x e y de forma incorrelacionada, variando el modelo de ruido en la medición y modelo de estados. La obtención de los mejores modelos de Kalman para el caso se realizó de forma práctica, encontrando los valores óptimos de 4 pixels de ruido en el proceso y 10 pixels de ruido en la medición para la posición en el eje x, y 2 pixels de ruido en el proceso y 20 pixels de ruido en la medición para el eje y. También se realizó un filltrado de Kalman del radio detectado, para suavizar los cambios en las detecciones del radio y mejorar las proximas estimaciones utilizando como parámetro la predicción del filtro.

El filtrado no sólo otorga el beneficio de la reducción del ruido en la medición, sino que reduce el efecto de cuantización de la medición relacionado a el número entero de pixels de la pantalla. Es decir, el algoritmo de detección de pupila devuelve números redondeados y truncados en cuanto a la posición (x, y) de las pupilas, cuyo truncamiento puede resultar en errores considerables, en especial debido a que el rango dinámico de movimiento de la pupila se extiende tan solo de 1 a 6 pixels aproximadamente (al mirar desde una esquina de la pantalla hasta la otra). El filtro de Kalman devuelve números en punto flotante, reduciendo drásticamente dicho error de cuantización.

VIII. CÁLCULO DE LA POSICIÓN DE LA VISTA

Una vez obtenida la posición del ojo y la pupila dentro de cierto frame, es necesario realizar el cálculo de la posición de la mirada en la pantalla, el objetivo final del sistema de seguimiento. Este cálculo se basa en la relación que hay entre la posición relativa de la pupila dentro del ojo. Debido a que el seguimiento de los ojos puede perder puntos entre frames sucesivos sin perder el seguimiento del ojo, una aproximación de la posición del ojo únicamente a partir de los puntos seguidos genera mucho error. Para reducir dicho error de pérdida de puntos de tracking, se generó el recuadro de recubrimiento del ojo a partir del frame de la detección del ojo. De esta forma el recuadro queda fijo y otorga un marco de referencias fijo

para la posición de la pupila, reduciendo el error mencionado previamente. Para poder seguir eventuales movimientos de la cabeza, se mide la distancia entre el recuadro obtenido por los puntos en seguimiento y el recuadro fijo, estableciendo un límite de distancia de 10 pixels. Al superar dicho límite se vuelve al paso inicial, buscando nuevas detecciones de ambos ojos.

El cálculo de la posición de la vista con el sistema de referencias mencionado enfrenta dos problemas. Primero, se debe saber donde están apuntando los ojos en el instante inicial de detección, para obtener las siguientes mediciones en referencia a esa inicial. Por esto, se propuso suponer que el usuario en promedio mira el centro de la pantalla durante la reconfiguración, criterio que en la práctica dió buenos resultados, evitando la necesidad de calibración del sistema, siempre tediosa para los usuarios. Siguiendo esta linea de razonamiento, se calculó el offset estático entre la posición relativa de las pupilas dentro del ojo en el instante inicial como

$$\begin{aligned} offsetInit_X &= ojow/2 - pupila_X \\ offsetInit_Y &= ojoH/2 - pupila_Y \end{aligned}$$

donde $ojow$ y $ojoH$ son el ancho y alto en pixels del recuadro fijo del ojo respectivamente, y $pupila_X$ y $pupila_Y$ son la posición relativa de la pupila dentro del ojo en el eje x e y respectivamente (en este caso en el momento de detección del ojo). El segundo problema radica en el posible movimiento de la cabeza, dejando corrido el marco fijo de referencias. Esto se puede solucionar calculando un offset dinámico, es decir que se actualiza frame a frame, según:

$$\begin{aligned} offsetDinámico_X &= ojoX - ojotracking_X \\ offsetDinámico_Y &= ojoY - ojotracking_Y \end{aligned}$$

donde $ojoX$ y $ojoY$ son la posición (x, y) del marco de referencia fijo dentro del frame, y $ojotracking_X$ y $ojotracking_Y$ son la posición (x, y) del recuadro recubridor de los puntos en seguimiento del ojo, dentro del frame.

Finalmente la posición de la mirada en la pantalla es calculada en cada frame según

$$\begin{aligned} posX &= (ojow/2 - pupila_X - offsetInit_X - \\ &\quad - offsetDinámico_X) * Sensi_X \\ posY &= (ojow/2 - pupila_Y - offsetInit_Y - \\ &\quad - offsetDinámico_Y) * Sensi_Y \end{aligned}$$

Donde $Sensi_X$ y $Sensi_Y$ son constantes de sensibilidad que deben ajustarse según el tamaño de la pantalla, la resolución de la cámara y el tamaño de los ojos en el frame, es decir la distancia entre el usuario y la webcam. Cabe aclarar que los resultados obtenidos para ambos ojos deben ser promediados para reducir el error en la estimación de la posición de la vista.

IX. IMPLEMENTACIÓN Y RESULTADOS

Se implementó el sistema desarrollado en el entorno de MATLAB, en una computadora con un procesador Intel i7 de primera generación, memoria RAM de 8GB, placa de video AMD de 1GB y una cámara Logitech HD. La cámara fue utilizada en la resolución 800x600, con el usuario a unos 50cm de la misma. Dicha resolución fue encontrada como óptima en la relación calidad de video y complejidad de cálculo, dependiendo ésta de la computadora y cámara utilizadas. Fueron implementados dos modos de uso, el primero de seguimiento de pupilas únicamente, y el segundo de movimiento de un cursor mediante la mirada del usuario.

Se analizaron los datos de las posiciones de las pupilas detectadas, con el objetivo de evaluar las corrección de dichas estimaciones y del efecto de los filtros de Kalman utilizados. En la figura 4 se puede observar la posición del cursor que sigue la vista en el eje horizontal de la pantalla. Se puede observar como en los momentos de movimiento sacádico del ojo, la salida filtrada sigue a la salida sin filtrar, ya que las variaciones en las detecciones de la pupila son grandes. En cambio, en los momentos de fijación de la vista en un punto, el filtrado de la posición ofrece una mejor estimación del punto observado, debido a que reduce el ruido introducido por el error en la detección de la posición de la pupila. También se puede notar que mientras los valores sin filtrar estan cuantizados a números enteros, los valores filtrados no sufren dicha cuantización. En el eje horizontal los resultados fueron de alta fidelidad entre la vista del usuario y el cursor en la pantalla.

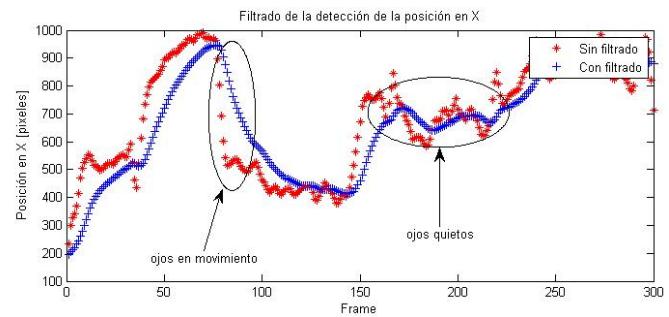


Fig. 4. Posición de la vista en la pantalla en el eje horizontal

En la figura 5, se puede observar el movimiento del cursor en el eje vertical. Como se observa en el gráfico, el ruido en la detección de la posición y de la pupila es mayor que el del eje x, habiendo grandes variaciones debido a malas estimaciones. Esto se debe a que el movimiento de la pupila del usuario para ver desde el límite inferior al límite superior es de rango dinámico considerablemente menor al de la posición horizontal, generando mayor error al momento de la estimación. Aunque el filtrado de Kalman mejora la estimación considerablemente, como se observa en los casos de ojos quietos y ojos en movimiento, hay ciertos casos donde el ruido de la estimación prohíbe obtener una estimación correcta del punto visto en el eje y por el usuario, es decir que la señal se encuentra por debajo del piso de ruido de la estimación. Es por esta razón que la fidelidad de seguimiento en el eje

vertical no fue la deseada en los experimentos realizados. Una forma de mejorar esto es optimizar la detección de pupilas en el eje horizontal, ya que los movimientos son captados por el sistema como puede observarse en la parte derecha de la figura, aunque en momentos de pequeñas variaciones la señal se pierde en el ruido.

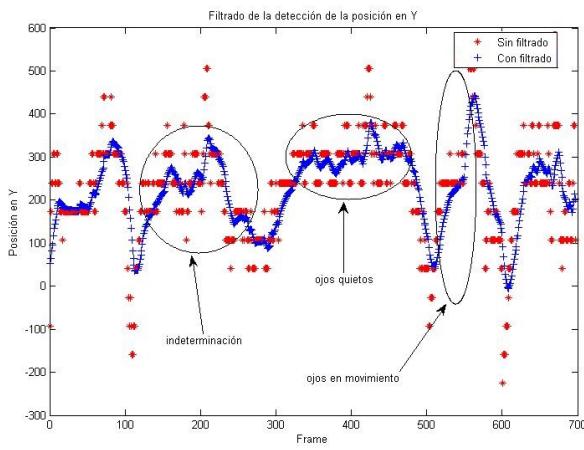


Fig. 5. Posición de la vista en la pantalla en el eje vertical

X. CONCLUSIONES

Fueron presentados los detalles de la implementación de la detección y del seguimiento pasivos de las pupilas a partir de una webcam y una computadora comercial. Para este fin se utilizaron algoritmos que involucran filtros Kalman, la transformación de Hough y el reconocimiento de características del tipo de Harr, entre otros.

Se encontraron resultados prometedores entre los que se pueden mencionar un fiel seguimiento de las pupilas en el eje x y un seguimiento aceptable en el eje y . El método desarrollado fue exitosamente ejecutado en una computadora personal con las características descriptas y en una webcam comercial de recursos medios, a partir del software MATLAB.

Trabajos futuros se centrarán en una optimización de la detección del movimiento en el eje y y una minimización del error de la posición del centro de la pupila. También resultaría interesante medir el tiempo de cómputo de este trabajo en un lenguaje de menor nivel a MATLAB, como por ejemplo una implementación en lenguaje C++. Con el objetivo de proveer una mayor exposición al público de este desarrollo, se proyecta una integración de este trabajo en una aplicación concreta, como lo puede ser una demostración focalizada al entretenimiento. Estas áreas serán dirigidas en futuras fases de este proyecto.

REFERENCES

- [1] Julian, David P. (Cupertino, CA), "Systems and methods for counteracting a perceptual fading of a movable indicator", United States Patent 8937591
- [2] James Russell Clarke, Sr., Phyllis Maurer Clarke, "Sleep detection and driver alert apparatus", United States Patent 5689241 A
- [3] Howard Cheng, Cid Newman, "Eye tracking enabling 3D viewing on conventional 2D display", United States Patent 8704879 B1
- [4] Aleksandra Królik, Paweł Strumiłło, "Eye-blink detection system for human-computer interaction" 2011
- [5] CH Morimoto, D Koons, A Amir, M Flickner, "Pupil detection and tracking using multiple light sources" Image and vision computing, 2000
- [6] Carlos H. Morimoto, Marcio R.M. Mimica, "Eye gaze tracking techniques for interactive applications"
- [7] E. Davies Machine Vision: Theory, Algorithms and Practicalities, Academic Press, 1990, Chap. 3.
- [8] P Viola, M Jones, "Rapid object detection using a boosted cascade of simple features" Vision and Pattern Recognition, 2001
- [9] BD Lucas, T Kanade, "An iterative image registration technique with an application to stereo vision." IJCAI, 1981
- [10] J Shi, C Tomasi, "Good features to track" Proceedings CVPR, 1994
- [11] <http://www.mathworks.com/matlabcentral/fileexchange/15652-iris-segmentation-using-daugman-s-integrodifferential-operator>
- [12] Omprakash Patel, Yogendra P. S. Maravi and Sanjeev Sharma, "A Comparative Study of Histogram Equalization Based Image Enhancement Techniques for Brightness Preservation and Contrast Enhancement", Signal & Image Processing : An International Journal (SIPIJ) Vol.4, No.5, October 2013.
- [13] Milad Soltany , Saeid Toosi Zadeh and Hamid-Reza Pourreza "Fast and Accurate Pupil Positioning Algorithm using Circular Hough Transform and Gray Projection", 2011 International Conference on Computer Communication and Management Proc .of CSIT vol.5 (2011)
- [14] Z Fu, Y Han, "Centroid weighted Kalman filter for visual object tracking" Measurement, 2012
- [15] Z Zhu, Q Ji, K Fujimura, K Lee, "Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination" Pattern Recognition, 2002



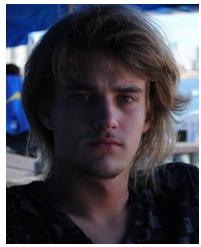
Matias Dwek es alumno de ingeniería electrónica en el ITBA, le gustan los videojuegos, ir al cine y el rock. En su tiempo libre juega al fútbol con amigos.



Rodrigo Iribarren es alumno de ingeniería electrónica en el ITBA, le gusta armar computadoras, los videojuegos y el fútbol. En su tiempo libre sale con amigos.



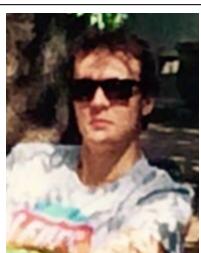
Rocco Rondinella es alumno de ingeniería electrónica en el ITBA, le gusta viajar, jugar a juegos móviles y salir con sus amigos. En su tiempo libre lee libros de aventuras.



Nicolás Laurutis es alumno de ingeniería electrónica en el ITBA, le gusta el rock, salir con amigos y navegar. En su tiempo libre lee libros.



Julián Tachella es alumno de ingeniería electrónica en el ITBA, le gusta la música, la síntesis de audio y el fútbol. En su tiempo libre es baterista en una banda, “Motoneta” (<http://motoneta.bandcamp.com/>).



Luciano Bertonatti es alumno de ingeniería electrónica en el ITBA, le gusta jugar al tennis, la pizza y las series de televisión. En su tiempo libre se dedica a la jardinería.