

Implementación del algoritmo de contornos activos para determinar la morfología de la pupila.

A. Zúñiga López¹, E. Suaste Gómez², G. M. Sánchez Sánchez³, L. A. Villamar Martínez.

¹Departamento de Electrónica, Universidad Autónoma Metropolitana - Azcapotzalco, Ciudad de México.

²Sección de Bioelectrónica, Departamento de Ingeniería Eléctrica, CINVESTAV-I.P.N., Ciudad de México.

³Unidad Académica Profesional-Acolman, Universidad Autónoma del Estado de México, Acolman, Edo. Méx.

⁴Departamento de Ingeniería, Academia de Electrónica, UPIITA- Instituto Politécnico Nacional, Ciudad de México.

Resumen— En este trabajo se presenta la implementación en Java y OpenCv del algoritmo de contornos activos (Snake), para determinar el contorno de la pupila en imágenes de baja resolución, teniendo como resultado una aproximación aceptable del borde de la pupila, con pocas iteraciones. Además se determinó, que la morfología de contorno de la pupila, tiene una similitud con una circunferencia y de los resultados obtenidos, se observó que hay un valor alto en la correlación que se implementó.

Palabras clave—Contornos Activos, OpenCv, Pupila.

I. INTRODUCCIÓN

La pupila es una parte fundamental del ojo, su función consiste en regular la cantidad de luz que pasa a la retina. Para cada persona existe una considerable variación en el comportamiento de la abertura, este tipo de característica hace que sea un rasgo muy particular para cada persona. Para la medicina y otras áreas de investigación, es importante estudiar el comportamiento que tiene la pupila y su morfología [3,7-11].

Para conocer la forma de la pupila, se requiere utilizar técnicas de procesamiento de imágenes, por lo que, se han utilizado diversos métodos, como son: detectores de borde, transformada de Hough y combinaciones de ambos [2,4]. Sin embargo, muchos de los algoritmos utilizados tienen los siguientes inconvenientes: considerar que el borde de la pupila es un círculo [2,4], que no es necesariamente cierto, y que varios métodos de detección de bordes requieren fijar un valor de umbral, causando que los bordes se eliminen.

Uno de los métodos que en recientes años ha sido aplicado con éxito para localizar la frontera de una pupila, es el método de contornos activos (o *Snake*) [1-5,7-8], desarrollado por Kass et al. [13] y posteriormente modificado por Williams y Shah [6], donde ellos proponen el algoritmo de Greedy.

En este trabajo se describe la implementación del algoritmo *Greedy Snake*, para utilizarlo en la estimación del borde de la pupila en imágenes de baja resolución y posteriormente realizar un pequeño análisis para estimar que tan confiable es aproximar una pupila a un circunferencia.

II. ALGORITMO GREEDY SNAKE

Los contornos activos se expresan como un proceso de minimización de la energía. Esta energía funcional incluye

además de un borde, las propiedades que controlan la forma en que el contorno se puede estirar y curvar. En consecuencia, la energía funcional es la suma de una función de la energía del contorno interior, su energía de restricción y la energía de la imagen: éstos se indican por E_{int} , E_{con} y E_{img} respectivamente. La energía funcional se define como la integral de estas funciones [4,12]:

$$E_{snake} = \int_{s=0}^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s)) ds \quad (1)$$

El algoritmo de Greedy Snake implementa el proceso de minimización de la energía como un desarrollo puramente discreto. Este proceso empieza con la especificación de un contorno inicial, que evoluciona de manera iterativa, buscando alrededor de un punto del contorno, un nuevo punto que contenga el menor valor de energía. En su forma matemática, esto se expresa como [12]:

$$E_{snake}(s) = \alpha(s) \left| \frac{dv_s}{ds} \right|^2 + \beta(s) \left| \frac{d^2v_s}{ds^2} \right|^2 + \gamma(s) E_{edge} \quad (2)$$

Las diferenciales de primer y segundo orden son aproximaciones para cada punto buscado en el vecindario local del punto de contorno seleccionado (v_s el conjunto de puntos de Snake). Los parámetros de ponderación α , β y γ , son las funciones del contorno. En esta implementación, el contorno se almacena como una matriz de vectores, donde cada vector tiene dos elementos: x y y , que son las coordenadas del contorno calculado. Por otra parte E_{edge} es una función de energía que puede ser implementado como las magnitudes de los bordes del operador de Sobel en el punto x y y .

En este trabajo se implementó el modelo del algoritmo de Greedy en Java utilizando para su desarrollo las librerías de OpenCv. La secuencia de pasos del algoritmo se muestra en la figura 1.

III. METODOLOGÍA

La metodología que se implementó en este trabajo fue la siguiente: se toma de una base de datos (ver referencia [14]) una imagen de forma aleatoria, del estudio de 5 personas; donde las características más relevantes de estas imágenes son: (1) se capturan con una cámara de alta

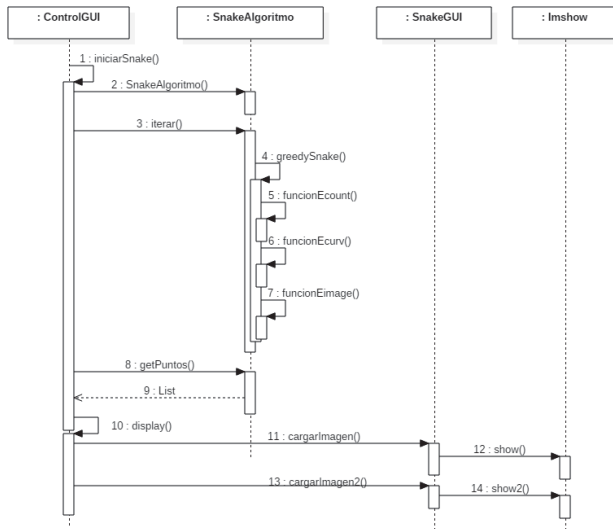


Fig. 1. Diagrama de secuencias para el algoritmo Greedy Snake.

velocidad, pero debido a la gran cantidad de cuadros por segundo adquiridos, las imágenes obtenidas tienen una baja resolución (260 x 260), (2) la pupila está iluminada con infrarrojo para tener un mejor contraste y (3) las imágenes tienen una corrección de distorsión.

La primera etapa consiste en colocar de forma manual un contorno inicial en la imagen como se muestra en la figura 2, éste se encuentra dentro de la región interna de la pupila, además, dicho contorno no debe estar muy cercano al borde que se va a encontrar. Posteriormente, se inicia el algoritmo de Snake, inicializando el programa con el número de iteraciones que se deseen realizar y los valores de α , β y γ , se tomaron siguiendo el criterio de la referencia [4]. Cabe aclarar, que las imágenes que se analizaron no pasan por alguna etapa de pre-procesamiento de imágenes, esta es una ventaja del método desarrollado.

Cuando el programa concluye con las iteraciones especificadas, se obtiene como salida, una imagen con el contorno que calculó (ver figura 3) y un archivo de texto con los puntos (x,y) que describen al mismo.

IV. RESULTADOS

Con el algoritmo implementado, se realizaron varias corridas con diferentes iteraciones para determinar cuál es la más adecuada, esto se muestra en la figura 4. De estos resultados preliminares se observó que los contornos extraídos (con 50 iteraciones) tiene una buena aproximación, ya que visualmente se observaba que el contorno extraído se ajustaba a la pupila, además el tener pocas iteraciones repercute en el costo computacional.

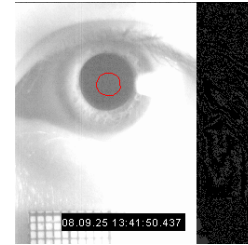


Fig. 2. Imagen a procesar, con el contorno con el que inicia el algoritmo.



Fig. 3. Contorno de la pupila, extraído por el algoritmo de contornos activos.

Como se mencionó anteriormente, una de las salidas del programa es un archivo de texto que reporta las coordenadas del contorno calculadas, de ese archivo se obtuvieron en promedio 146 puntos aproximadamente que son los que describen el contorno (vector de coordenadas, graficado en la figura 5). Antes de graficar estos valores, se procedió a normalizarlos con el objetivo de establecer una correlación con un patrón, es decir, el círculo unitario. También se trasladan los primeros puntos al final del vector de coordenadas, para que sus valores coincidan con el patrón de muestra. Así, se realizó un corrimiento de la información.

Por último, los valores del eje y de las coordenadas del contorno se comparan con una función sinusoidal que se generó, del tipo $\sin(\frac{2\pi}{T}n)$ donde T es el periodo, con una amplitud de uno y la misma cantidad de muestras, es decir, $T = 146$, como se observa en la gráfica de la figura 5. Los resultados de las correlaciones se muestran en la tabla 1.

IV. DISCUSIÓN DE RESULTADOS

De los resultados obtenidos observamos que en casi todas las imágenes analizadas, con pocas iteraciones, se empieza a distinguir un contorno aceptable, que en términos del rendimiento computacional es deseable. Sin embargo, notamos que algunas imágenes como en la figura 4c el contorno se empieza a deformar, aún cuando se realiza con 100 iteraciones; también comprobamos que con el algoritmo de contornos activos se obtiene una respuesta aceptable al ruido, como se muestra en la imagen 2, en donde la presencia de un punto luminoso no afecta demasiado en la búsqueda del borde de la pupila. Los resultados obtenidos comprueban que el algoritmo de Snake es una buena opción para el cálculo del contorno de la pupila, coincidiendo con lo reportado en algunas referencias.

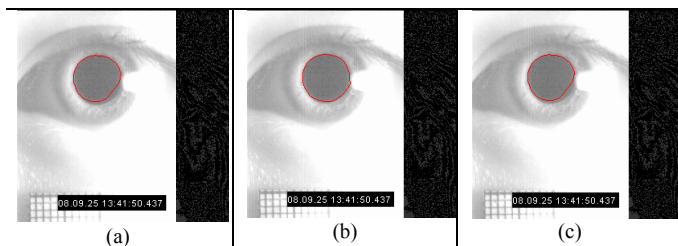


Fig. 4. Detección de contornos con a) 10 iteraciones, b) 25 iteraciones y c) con 100 iteraciones. Todos tiene el mismo contorno de inicio.

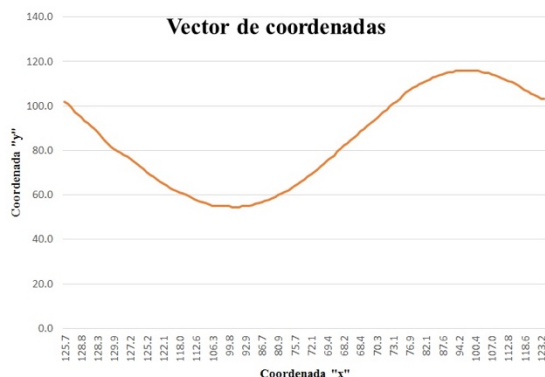


Fig. 5. Gráfica de las coordenadas del contorno de una pupila, obtenidas del programa implementado.

Tabla 1.
Correlación a un círculo

	Correlación
Sujeto 1	0.9659
Sujeto 2	0.9342
Sujeto 3	0.9210
Sujeto 4	0.8962
Sujeto 5	0.9686

Por otra parte, en la tabla 1 se observa que la mayoría de los sujetos a los cuales se determinó su correlación, presentan una gran similitud con una circunferencia, por lo que para este estudio preliminar, se considera que la pupila puede aproximarse a un círculo, ya que en muchos casos se obtiene una aproximación estadísticamente aceptable.

V. CONCLUSIÓN

En este trabajo previo se comprobó que con el método de Greedy Snake, se obtienen resultados significativos para la obtención del borde de la pupila, sin embargo, faltaría por realizar en un trabajo posterior, pruebas con una mayor cantidad de imágenes y que además presenten otro tipo de ruidos como son las pestañas. Una de las ventajas que presenta el programa que se desarrolló, es su diseño modular, así que se pueden realizar cambios, de una manera relativamente simple. El programa que se implementó no necesita un pre-procesamiento, lo que facilita el

procesamiento de las imágenes de la pupila. Por último, se pudo comprobar que en este pequeño espacio de muestras, la pupila se modela como un círculo, sin embargo, esta conclusión no puede ser determinante, ya que la muestra no es significativa; por lo que, uno de los trabajos proyectados a futuro es realizar un análisis con mayor detalle.

BIBLIOGRAFÍA

- [1] A. Hilal, P. Beuseroy and B. Daya, "Real Shape Inner Iris Boundary Segmentation Using Active Countour without Edges", in *International Conference on Audio, Language and Image Processing*, pp. 14-19, 2012.
- [2] Alaa Hilal, Bassam Daya and Pierre Beuseroy, "Hough Transform and Active Contour for Enhanced Iris", *International Journal of Computer Science Issues*, Vol. 9, Issue 6, No. 2, pp. 1-10, Nov. 2012.
- [3] Ann A. Jarjes, Kuanquan Wang and Ghassan J. Mohammed, "GVF Snake-based Method for Accurate Pupil Contour Detection", *Information Technology Journal*, 9(8), pp. 1653-1658, 2010.
- [4] Ann A. Jarjes, Kuanquan Wang and Ghassan J. Mohammed, "Improved Greedy Snake Model for Detecting Accurate Pupil Countour", in *3rd International Conference on Advances Computer Contro ICACC*, pp. 515-519, 2011.
- [5] Carlos A. C. M. Bastos, Tsang Ing Ren and George D. C. Calvancanti, "A Combined Pulling & Pushing Active Contour Method for Pupil Segmentation", in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 850-853, 2010.
- [6] Donna J. Williams and Mubarak Shah, "A Fast Algorithm for Active Contours and Curvature Estimation", *CVGIP: Image Understanding*, Vol. 55, No. 1, pp. 14-26, Jan., 1992.
- [7] Gökyay Akinci, Ediz Polat and Orhan Murat Kocak, "A video-based eye pupil detection system for diagnosing bipolar disorder", *Turkish Journal of Electrical Engineering & Computer Sciences*, 21, pp. 2367-2377, 2013.
- [8] Gökyay Akinci, Ediz Polat and Orhan Murat Kocak, "Neuropsychiatric Disorders Classification Using a Video Based Pupil Detection System", in *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1-5, 2012.
- [9] Jianzhong Wang, Guanyue Zhang and Jiadong Shi, "Pupil and Glint Detection Using Werable Camera Sensor and Near-Infrared LED Array" *Sensors*, 15, pp. 30126-30141, 2015.
- [10] John Daugman, "New Methods in Iris Recognition", *IEEE Transactions on Systems, Man, and Cybernetics--Part B: Cybernetics*, Vol. 37, No. 5, Oct., 2007.
- [11] Masaru Kiyama, Hitoshi Iyatomi and Koichi Ogawa, "Robust Video-oculography for Non-invasive Autonomic Nerve Quantification" in *33rd Annual International Conference of the IEEE EMBS*, pp. 494-497, 2011.
- [12] Mark S. Nixon, Alberto S. Aguado, *Feature Extraction and Image Processing*, Newnes, Great Britain, 2002, pp. 217-246.
- [13] Michael Kass. Andrew Witkin and Demetri Terzopoulos, "Snake: Active Contour Models", *International Journal of Computer Vision*, pp. 321-331, 1988.
- [14] Luis A. Villamar and Ernesto Suaste "Measurement of the pupillary response using high-velocity videooculography and conventional videooculography", *2009 Pan American Health care Exchanges*, pp. 9-10, 2009