

Control de Calidad en Imágenes de Iris mediante Razonamiento Ontológico

Alberto de Santos Sierra¹, Javier Guerra Casanova¹, Carmen Sánchez Ávila¹ y
Vicente Jara Vera²

¹ Grupo de Biometría y Tratamiento Numérico de la Información.
Centro de Domótica Integral (CeDInt)
<alberto@cedint.upm.es>
<jguerra@cedint.upm.es>
<csa@mat.upm.es>

² Dpto. de Matemática Aplicada a las Tecnologías de la Información.
ETSI de Telecomunicación. Universidad Politécnica de Madrid.
Ciudad Universitaria s/n, 28040 Madrid
<vjara@mat.upm.es>

Resumen La segmentación es uno de los pasos que consume más tiempo en el proceso de reconocimiento de Iris. Mediante Morfología Matemática y Redes Neuronales, este algoritmo consigue aislar el Iris dentro de la imagen del ojo, no sólo con una alta precisión, sino también con una gran reducción en el tiempo y en la carga computacional. Además, este innovador esquema presenta una ontología capaz de decidir si la extracción de características posterior a la segmentación puede llevarse a cabo. Los algoritmos de reconocimiento de Iris podrán beneficiarse de este nuevo enfoque, y además, la esencia del algoritmo puede ser extendida a otros procesos de segmentación biométrica.

Palabras clave: Biometría, Reconocimiento de Iris, Segmentación, Lógica Difusa, Morfología Matemática, Redes Neuronales, Razonamiento Ontológico.

1. Introducción

El reconocimiento de Iris engloba diferentes aspectos desde que se captura la imagen, hasta que el sistema es capaz de rechazar o aceptar al usuario que pretende acceder al sistema. Dentro de todos estos pasos, la segmentación es el que más tiempo consume en el proceso completo, siendo esta la razón por la cual se debe hacer especial esfuerzo en reducir no sólo el tiempo de computación sino también la complejidad del procedimiento.

Algunas soluciones han sido implementadas para resolver este problema. La mayoría de ellas se aprovechan de la geometría circular del Iris y de la Pupila, y de que ambos son casi concéntricos. La estrategia consiste en detectar el Iris, la Pupila y los párpados, [1], [2], [3], tomando como solución la intersección de

todos ellos. De este enfoque se derivan algunos problemas, ya que algunas veces el Iris en general está escondido en parte debajo del párpado, haciendo que aislar el Iris, sea algo bastante difícil para llevar a cabo.

El enfoque nuevo propuesto en este documento únicamente se centra en la parte del Iris que se puede ver, en otras palabras, qué sección de Iris no se encuentra escondida debajo del párpado, mediante Morfología Matemática, [4], y Redes Neuronales Artificiales, [5].

Una vez que la segmentación se ha llevado a cabo, un proceso de decisión basado en Ontologías ha sido implementado para dilucidar si de la imagen segmentada con anterioridad pueden ser extraídas las características necesarias para elaborar el patrón biométrico que posteriormente servirá para identificar al individuo, [3].

El documento continúa presentando primeramente el proceso de segmentación, y el control de calidad basado en razonamiento ontológico. Los resultados serán presentados tras estas secciones, considerando aspectos temporales y de precisión, terminando con las conclusiones y el trabajo futuro.

2. Segmentación de Iris

2.1. Detección de Pupila

El primer paso en el procedimiento de reconocimiento de Iris precisa de una correcta detección de la Pupila, [2], [3]. Considerando las características geométricas de la Pupila, (i.e., casi circular), es obviamente innecesario utilizar todo el contorno como resultado de la segmentación. Por el contrario, se propone el uso de únicamente cinco puntos: $pCenter$, $pVertical_{min}$, $pVertical_{max}$, $pHorizontal_{min}$ y $pHorizontal_{max}$.

Estos puntos, Fig. 1, engloban toda la información necesaria para afirmar no sólo dónde se encuentra la pupila sino también hasta qué punto se encuentra dilatada.

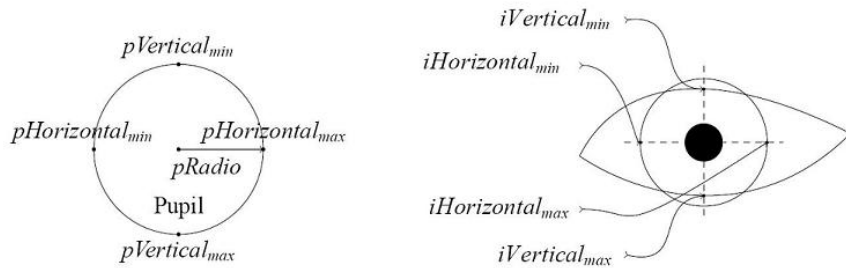


Figura 1. Puntos más importantes del Ojo: Puntos de Pupila (Izquierda), Puntos de Iris (Derecha).

Debe resaltarse que esta detección de Pupila únicamente considera la parte visible de la Pupila, es decir, si el párpado cubre parte de la Pupila, $pVertical_{min}$ sería más cercano a $pCenter$, haciendo más fácil la tarea de un Control de Calidad, tal y como se detallará en la Sección 2.4.

La segmentación podría incluso reducirse más aún y reducir el número de puntos requeridos para localizar la Pupila a únicamente dos: $pCenter$ y $pRadio$, ya que estos dos puntos pueden describir una circunferencia de forma precisa. Sin embargo, existen situaciones donde la Pupila puede estar desplazada en la imagen (debido a la concavidad ocular), perdiendo parte de sus propiedades circulares. Esto no quiere decir que la pupila deje de ser circular, sino que debido a la captura de la imagen, la perspectiva puede ofrecer una Pupila con cierta excentricidad, más similar a una elipse que a una circunferencia. En estas situaciones se requerirían más puntos para extraer toda la información de la Pupila, de ahí que se haya optado por la elección de los puntos anteriores.

Para llevar a cabo la segmentación, se precisa de operadores morfológicos de Apertura y Clausura, [4], y umbralización para detectar la parte visible de la Pupila dentro de una imagen.

Finalmente, en la Sección 3, se proporcionarán los resultados sobre la precisión de este algoritmo. Sin embargo, se puede adelantar que el procedimiento es capaz de detectar cada Pupila para todas las imágenes existentes en la base de datos CASIA (v1 y v3), [6], con ningún error y con un tiempo de computación despreciable.

El siguiente paso, la detección de Iris, precisará únicamente del punto $pCenter$, porque una de las grandes ventajas de este algoritmo se basa en el hecho de que el centro del Iris no es ni considerado ni detectado, decreciendo aún más el tiempo de computación del proceso completo.

2.2. Detección de Iris: Primera Aproximación

En este nuevo enfoque se detectará y aislará únicamente la parte visible del Iris dentro de una imagen. Se dividirá el proceso completo en dos aproximaciones: Primeramente, los puntos de Iris serán definidos y extraídos tal y como se hizo con la pupila anteriormente (Sección 2.1). Aunque esta aproximación pudiera poco precisa, proporciona una muy buena solución para la detección de Iris, debido al hecho de que los puntos de Iris encierra perfectamente el Iris completo.

En segundo lugar, una segunda aproximación aísla el Iris como en enfoques previos propuestos en la literatura, tomando la información extraída en la primera aproximación. Aunque este nuevo enfoque puede obtener el aislamiento del Iris completo (tal y como se verá en la siguiente sección), la primera aproximación proporciona al Control de Calidad suficiente información para llevar a cabo el rechazo o aceptación de la imagen capturada.

Una vez que se han presentado ambas aproximaciones, se explica la primera. En este caso, cuatro puntos para el Iris son definidos: $iVertical_{min}$, $iVertical_{max}$, $iHorizontal_{min}$ y $iHorizontal_{max}$. Estos puntos se corresponden con los límites del Iris visible en una imagen, tal y como puede verse en la Fig. 1.

En dicha Figura, se aprecia como el párpado cubre parte del Iris.

Para continuar con el algoritmo de detección de Iris, el lector debe ser consciente de ciertas propiedades geométricas en el Iris. Dos características han sido consideraras para este procedimiento:

- *pCenter* y el centro del Iris están suficientemente cerca como para ignorar la diferencia entre ambos
- El Iris y la Esclerótica³, son claramente diferente en términos de color.

Considerando la geometría circular del Iris, la idea de un filtro capaz de proporcionar valores altos (en términos de intensidad) cercanos al contorno de la pupila, y valores cada vez más bajos a medida que los píxeles se alejen del centro de la pupila, puede ofrecer un buen resultado a la hora de aislar el Iris. De hecho, dicho filtro toma *pCenter* como parámetro de entrada para obtener una mayor precisión a la hora de ubicar los filtros dentro de la imagen.

Se implementaron cuatro filtros para llevar a cabo tal idea. Tres de ellos dependen enormemente de la distribución elegida para aclarar las componentes de la imagen que se encuentran más alejadas de *pCenter*, dejando casi intactas las componentes más cercanas a la Pupila. Únicamente se diferencian en la distribución usada. En la Cuadro 1, se muestran los nombres de los filtros y sus expresiones matemáticas. Las coordenadas representadas por (x_0, y_0) se corresponden con *pCenter*.

Cuadro 1. Descripción de los Filtros.

Nombre	Expresión Matemática	Parámetros
Gaussian	$A_\gamma e^{-\sigma_x(x-x_0)^2} e^{-\sigma_y(y-y_0)^2}$	$A_\gamma, \sigma_x, \sigma_y, (x_0, y_0)$
Coseno I	$A_I \cos(x - x_0) \cos(y - y_0)$	$A_I, (x_0, y_0)$
Coseno II	$A_{II} \cos^2(x - x_0) \cos^2(y - y_0)$	$A_{II}, (x_0, y_0)$

Todas las amplitudes A_γ , A_I y A_{II} definen de qué forma el filtro difumina la imagen.

En la Fig. 2, se pueden ver las distribuciones de los filtros. El lector podrá notar ahora el efecto que tendría cada filtro en la imagen si el pico más alto de cada filtro es desplazado a *pCenter* de cada imagen. Esas componentes más alejadas de la Pupila serán atenuadas en mayor medida que aquellas más cercanas a la Pupila.

Una vez que los filtros previos han difuminado la imagen, se precisa de un filtro extra para eliminar esas componentes más claras dentro de la imagen. Este filtro está basado en Morfología Matemática y umbralización, y ofrece como resultado una máscara con las componentes más oscuras dentro de la imagen.

Aunque la ejecución del algoritmo no es tan precisa como en el caso de la detección de pupila, esta primera aproximación ofrece al Control de Calidad

³ Parte blanca del ojo

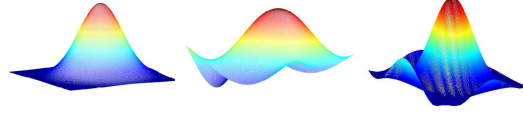


Figura 2. De izquierda a derecha: Gaussian, Coseno I y Coseno II.

posterior una información muy útil sobre el Iris, y de forma muy rápida, en otras palabras, puede indicar cómo de abierto está el párpado, si el Iris está desplazado en la imagen, si el Iris se encuentra dentro de la imagen, entre otros.

Finalmente, y acorde a lo sugerido en la sección anterior, los resultados de este algoritmo, así como de un ejemplo visual se proporcionarán en la Sección 3.

2.3. Detección de Iris: Segunda Aproximación

Como se vio con anterioridad, la primera aproximación proporciona al Control de calidad posterior la información suficiente para decidir la situación del Iris dentro de la imagen. Sin embargo, el enfoque ‘tradicional’ de la detección de Iris requiere un aislamiento completo del Iris, y algunas aplicaciones necesitan esta segmentación para poder funcionar correctamente. Para una mayor generalización de este algoritmo, se necesita la Segunda Aproximación para aislar completamente el Iris.

Este procedimiento es el paso final en el proceso completo de segmentación de Iris, aunque se ha sugerido en este documento que ese paso no es obligatorio para una segmentación precisa.

Para alcanzar este fin, es decir aislar completamente el Iris, se precisa de varios pasos. Primeramente, un Filtro con distribución Gaussiana es elegido para llevar a cabo la segmentación, ya que sus parámetros σ_x and σ_y , ver Cuadro 1, puede ser modificada y adaptada para delimitar el Iris.

Estos parámetros pueden ser fijados a mano. Sin embargo, esto es algo impensable para un sistema automático. El problema consiste, entonces, en relacionar estos parámetros con alguna medida ya realizada en etapas anteriores (por ejemplo, la distancia entre $pVertical_{min}$ y $iVertical_{min}$).

Algunos parámetros fueron propuestos y estudiados en este documento. Sin embargo, únicamente cuatro de estos parámetros fueron significativos en este estudio. Estos parámetros están recogidos en la Cuadro 2, y son definidos en términos de los puntos de Iris extraídos en la primera aproximación previa.

Esta distinción fue hecha en base a un estudio estadístico previo, [3], donde todas las variables fueron consideradas. Dichas variables relacionaban todas las posibles distancias entre los puntos de Iris, $pCenter$ y los puntos de Pupila. Además de esta conclusión, del estudio estadístico se puede obtener una conclusión mucho más importante: Las componentes horizontales de los parámetros en Cuadro 2, i.e. $i\Delta H_{max}$ y $i\Delta H_{min}$, estaban relacionadas fuertemente con σ_x y por otro lado, $i\Delta V_{max}$ y $i\Delta V_{min}$ estaba también relacionado con σ_y .

Para encontrar una función que relacionara los parámetros anteriores, dos redes neuronales, [5], [12], fueron implementadas. Esta red neuronal encontró la

Cuadro 2. Parámetros significativos relacionados con σ_x y σ_y .

Nombre	Definición
$i\Delta H_{max}$	Distancia entre $iHorizontal_{max}$ y $pCenter$
$i\Delta H_{min}$	Distancia entre $iHorizontal_{min}$ y $pCenter$
$i\Delta V_{max}$	Distancia entre $iVertical_{max}$ y $pCenter$
$i\Delta V_{min}$	Distancia entre $iVertical_{min}$ y $pCenter$

solución para ξ y ζ , cuya definición viene a continuación: $\sigma_x = \xi(i\Delta H_{max}, i\Delta H_{min})$ y $\sigma_y = \zeta(i\Delta V_{max}, i\Delta V_{min})$. De hecho, el error cuadrático medio (MSE, en inglés) para ambas redes es similar a $2 \cdot 10^{-3}$, algo insignificante si es comparado al error de precisión en las medidas de σ_x and σ_y , cuyos valores están cercanos a $\epsilon = \pm 0,01$.

Estas redes neuronales proporcionan una rápida computación de los valores de entrada σ_x y σ_y . Por lo tanto, la segunda aproximación únicamente necesita tiempo para computar los Filtros basados en distribuciones y el filtro encargado de eliminar las componentes más claras en la imagen. Un ejemplo visual de la ejecución de este algoritmo se presenta en la Sección 3, junto con un estudio temporal de dicho algoritmo.

2.4. Control de Calidad

El Control de Calidad, en el contexto de la segmentación de Iris, consiste en aceptar o rechazar una imagen de Iris en base a características extraídas de la imagen en sí misma, [3]. En este documento, un Control de Calidad basado en políticas se encargará de relacionar las características del Iris obtenidas en las secciones 2.1 y 2.2. Debido a este control, un programador (o usuario final) es capaz de definir diferentes políticas para aceptar o rechaza una imagen de Iris, o por el contrario, para tratar dicha imagen para poder mejorar su calidad. Estas políticas dependerán de los algoritmos usados para extraer información biométrica del Iris en etapas siguientes.

Consecuentemente, la ontología denominada OntIrisQualityControl ha sido desarrollada para almacenar la información sobre las políticas aplicadas a las imágenes de Iris. Cada vez que una imagen de iris es procesada, algunos parámetros importantes son almacenados en las instancias de la ontología. Entonces, el razonamiento sobre la ontología se realiza a través de reglas SWRL (Semantic Web Rule Language), [7], para obtener de forma automática el tipo de Control de Calidad que debe ser aplicado a la imagen de iris, dependiendo de las características del Iris y de la imagen procesada.

2.5. Ontología

La ontología OntIrisQualityControl para el Control de Calidad basado en políticas y en las imágenes de Iris, ha sido desarrollado con Protegé 3.4, [8]. Esta ontología esta compuesta por las siguientes clases:

- **ImageFeatures:** En esta clase, hay información sobre la calidad de la imagen. Cada vez que una imagen de iris es procesada, una nueva estancia ImageFeatures es creada, rellenando sus propiedades con valores obtenidos del estudio de algunas características de la imagen con diferentes algoritmos de tratamiento de imágenes, [4]. Estas características son seleccionadas puesto que son los factores más representativos en lo referente a la calidad de imágenes. Está compuesta además por las siguientes propiedades, que se expresan en porcentaje:
 - **Image_blur:** Medida de borrosidad de una imagen. Su rango está entre 0 y 100 %
 - **Image_distorsion:** Medida de distorsión de la imagen. Su rango está entre 0 y 100 %
 - **Image_occupied:** Medida de la porción ocupada por algunos objetos como sombras y párpados. Su rango está entre 0 y 100 %
 - **Image_noise:** Medida del ruido de la imagen.
 - **Image_outfocused:** Medida del desenfoque en la imagen.
- **IrisImage:** Clase con las características del iris bajo procesamiento. Cada vez que una imagen de iris es procesada, una nueva instancia de la clase IrisImage es generada, y sus propiedades son rellenadas con los valores calculados con los puntos obtenidos en las Secciones 2.1 y 2.2. Las propiedades de IrisImage son:
 - **Iris_height_min:** Es el menor valor entre $|iVertical_{min} - pVertical_{min}|$ y $|pVertical_{max} - iVertical_{max}|$. Representa la mínima apertura vertical del Iris.
 - **Iris_width_min:** Es el menor valor entre $|iHorizontal_{min} - pHorizontal_{min}|$ y $|pHorizontal_{max} - iHorizontal_{max}|$. Hace referencia a la mínima apertura horizontal del Iris.
 - **Iris_axis_difference:** Es el valor $|pHorizontal_{max} - pHorizontal_{min}| - |pVertical_{max} - pVertical_{min}|$. Representa el grado en el que el Iris está centrado.
 - **Iris_radius_360_max:** Es el radio máximo de la corona circular que se puede extraer en el Iris.
 - **Iris_radius_256_max:** Es el radio máximo de la corona circular de 256° que se puede extraer en el Iris.
- **QualityControl:** Esta clase almacena las diferentes posibilidades del Control de Calidad a la hora de reaccionar frente a los diferentes tipos de imágenes e iris preprocesados. Una imagen puede ser rechazada, aceptada o tratada para mejorar su calidad. Esta clase está compuesta por las siguientes propiedades:
 - **qualityControl_type:** El tipo de Control de Calidad. Puede tomar estos valores:
 - Aceptada: La imagen es aceptada.
 - Rechazada: La imagen es rechazada.
 - Tratamiento: Antes de extraer la información de la imagen, se debe tratar previamente con algoritmos almacenados en la propiedad qualityControl_mode, [4].
 La descripción de estos algoritmos está fuera del alcance de este documento.

- **qualityControl_mode**: Algoritmo para tratar la imagen y mejorar sus características.
- **Policy**: Cualquier instancia de esta clase es una política de reacción para el Control de Calidad. Está compuesto por los mínimo o máximo valores que deben tener los parámetros de las instancias IrisImage e ImageFeature para aplicar un Control de Calidad específico. Los programadores finales definirán unas políticas de Control de Calidad dependiendo de los requisitos de los algoritmos usados para extraer la información biométrica de las imágenes de Iris. Las propiedades de esta clase casan con las propiedades de IrisImage e ImageFeatures:
 - policy_iris_width_min
 - policy_iris_height_min
 - policy_iris_axis_difference_min
 - policy_iris_radius_256_min
 - policy_iris_radius_360_min
 - policy_image_blur_max
 - policy_image_distorsion_max
 - policy_image_noise_max
 - policy_image_occupied_max
 - policy_image_outfocused_max
 - policy_qualityControl
- **Result**: Esta clase almacena las instancias de qualityControl que han sido activadas en el proceso de razonamiento. Está compuesto por las siguientes propiedades:
 - result_qualityControl

2.6. Razonamiento y manejo de conflictos

Una vez que los puntos importantes y las características de la imagen del Iris han sido obtenidas y se han generado nuevas instancias de ImagesFeatures e IrisImage, la fase del razonamiento comienza. En esta fase, el Control de Calidad será activado acorde con las políticas definidas y las características de la imagen y el Iris obtenidas previamente. Como resultado de esto, habrá una instancia de Result, rellena con todas las políticas de control activadas, [9].

Es posible que de acuerdo con las características de la imagen de Iris y las políticas definidas, algunas contradicciones aparezcan en el Control de Calidad, haciendo un conflicto de resultados. Las reglas siguientes han sido definidas para manejar estos conflictos:

- Si hay al menos un Reject, entonces se rechaza.
- Si existe más de un posible Treatment, entonces la imagen es tratada con cada algoritmo propuesto por el razonamiento.
- Si se genera Accept y Treatment, entonces la imagen es tratada.

La fase de razonamiento ha sido desarrollada con el plug-in Jess 7, [10], para razonar sobre ontologías con reglas SWRL en Protegé.

3. Resultados

La intención de esta sección es doble: Por un lado, proporciona un estudio temporal de cada uno de las etapas en la segmentación de Iris aquí presentada, y por otro lado, presenta algunos ejemplos de la ejecución del procedimiento completo.

3.1. Estudio temporal

Primeramente, el entorno donde se ha llevado a cabo el experimento es presentado, pues los tiempos de computación dependen enormemente de ello. Un ordenador Core 2 Duo HP PC 1.86 GHz fue usado, con 2GB de memoria RAM, y con R14 Matlab 2007 para ejecutar los algoritmos programados.

Considerando esto, el tiempo promedio de cada proceso es mostrado para medir la ejecución temporal. La detección de Pupila conlleva 0,28 segundos, la primera aproximación 2,24 segundos y la segunda aproximación 1,21 segundos. Por lo tanto, únicamente casi 2,25 segundos serían necesarios para detectar el Iris en promedio (con la primera aproximación) en términos de requisitos de un buen Control de Calidad, tal y como se vio en la Sección 2.4.

Además, con casi un segundo más, el proceso completo sería llevado a cabo. Además, el tiempo para procesar la ontología es despreciable, con lo cual no se ha considerado en esta sección.

Esto es un resultado importante si es comparado con otros algoritmos y los entornos de experimentación, considerando el hecho de que la implementación fue hecha en Matlab y no ha sido depurada de forma detallada, ni optimizada. Una implementación en C o C++ será un trabajo futuro para así mejorar más aún los tiempos de computación.

3.2. Ejemplo Visual

En esta sección se presenta cómo actúan los Filtros basados en distribuciones anteriores y los resultados de las aproximaciones, primera y segunda.

Fig. 3 muestra los diferentes efectos de los filtros previos. El lector puede fácilmente observar como la imagen es difuminada en aquellas regiones más alejadas de la Pupila. Los efectos dependen no solamente de los parámetros mostrados en la Tabla 1, sino también de la intensidad de la imagen. En otras palabras, la respuesta del filtro para ojos oscuros es diferente que en ojos claros.

Además, Fig. 4 muestra dos ejemplos de la primera aproximación en el proceso de detección de Iris. Tal y como se dijo en las secciones anteriores, la información proporcionada por esta primera aproximación es más que suficiente para localizar el Iris y decidir sobre la calidad de la imagen adquirida.

También puede verse el resultado de la detección de Pupila en la misma imagen. En la segunda imagen, el Iris se encuentra parcialmente cubierto por las pestañas, siendo ésta una situación de gran dificultad para detectar el Iris.

En la imagen superior se aprecian los resultados de la detección de Pupila e Iris, tal y como se ha propuesto en este documento.

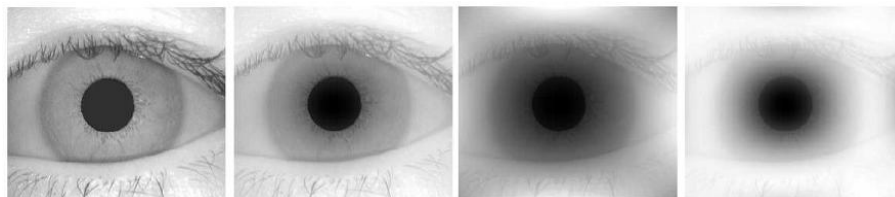


Figura 3. Efectos de los Filtros basados en distribuciones en imágenes de iris: Imagen Original, Gaussian, Coseno II, Coseno I.

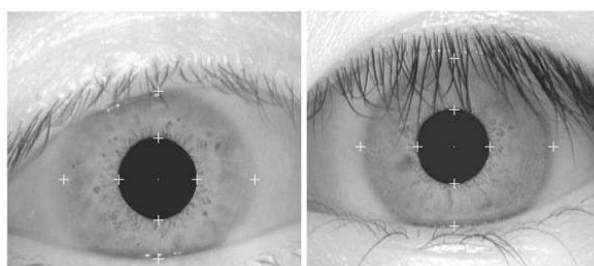


Figura 4. Dos ejemplos de la detección de Pupila y la primera aproximación en la detección de Iris.

Fig. 5 muestra el resultado de la segunda aproximación para dos imágenes diferentes. Los valores de los parámetros han sido obtenidos a partir de una Red Neuronal que fue entrenada en base a los valores extraídos de las imágenes. La frontera interna del iris coincide con el resultado de la frontera externa de la Pupila. A pesar de ser posible aislar completamente la pupila, en este documento se propone extraer únicamente cinco puntos que delimiten la pupila en vez de toda la frontera.

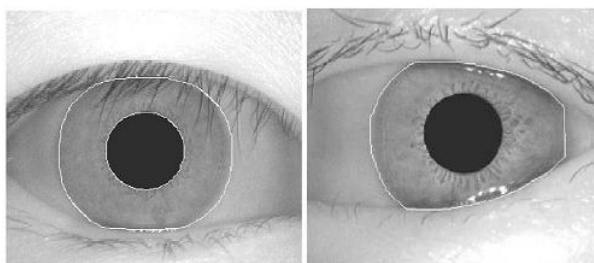


Figura 5. Resultados de la segunda aproximación en detección de Iris.

4. Conclusiones y Trabajo Futuro

4.1. Conclusiones

Esta detección de Iris tiene dos objetivos: primeramente, el algoritmo es capaz de aislar el Iris, mediante la obtención de ciertos puntos importantes (cuatro para el Iris, cinco para la Pupila). Esta propuesta ahorra coste computacional y tiempo en el proceso, a parte de espacio en memoria a la hora de almacenar la segmentación de la imagen.

Además, el proceso de detección de Iris contiene una segunda etapa, donde el Iris puede ser aislado completamente. Esto permite que este algoritmo pueda ser integrado en sistemas actuales basados en detección de Iris. Este segundo paso, no sólo realiza la segmentación de forma rápida sino que además es bastante preciso a la hora de aislar el Iris.

Por otro lado, el Control de Calidad basado en ontologías y razonamiento basado en reglas, es capaz de proporcionar al proceso de segmentación nuevas herramientas para decidir si una posterior extracción de características es factible con la imagen que se ha capturado o si, por el contrario, la imagen precisa de un tratamiento numérico o si el sistema requiere una nueva captura de imagen.

Finalmente, el proceso completo mejora a los algoritmos actuales de segmentación en diferentes aspectos, haciendo de este procedimiento un algoritmo fácilmente extensible a otras técnicas biométricas.

4.2. Trabajo Futuro

Como trabajo futuro, el algoritmo será utilizado con otras bases de datos para probar su eficiencia. Sin embargo, teniendo en cuenta que el algoritmo ha sido implementado de forma independiente de la base de datos, los resultados prometen ser interesantes.

Por otro lado, un algoritmo evolutivo, [11], podría ser utilizado para evitar el problema de ‘overfitting’ en la red neuronal, y además, se estudiará la posibilidad de utilizar Lógica Difusa, [12], junto con la red neuronal para hacer este algoritmo aún más preciso.

5. Agradecimientos

Los autores quieren agradecer al Proyecto CENIT Secur@: Seguridad y Confianza en la Sociedad de la Información, financiado por el Ministerio de Industria, Turismo y Comercio.

Referencias

1. J. Daugman, *High Confidence Visual Recognition of Persons by a Test of Statistical Independence*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No 11, Nov. 1993.

2. J. Daugman, 'How Iris Recognition Works', IEEE Transactions on Circuits and Systems For Video Technology, Vol. 14, n 1, January 2004.
3. A. de Santos Sierra, C. Sánchez Ávila, E. Marchiori, *Iris Recognition: Segmentation enhancement by using Morphological Operators*, Master Final Thesis, June, 2007. Amsterdam.
4. R. C. González, R. E. Woods, S. L. Eddins, *Digital Image Processing*, 2nd Edn, Prentice All, 2004.
5. J. R. Hilera, V. J. Martínez, *Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones*, Ra-ma, 1995
6. CASIA Iris Image Database. <http://www.sinobiometrics.com>
7. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, M. Dean, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, W3C Member Submission 21 May 2004. <http://www.w3.org/Submission/SWRL/>
8. Ontology editor and knowledge-base framework Protegé. <http://protege.stanford.edu/>
9. J. Guerra Casanova, *Diseño y desarrollo de un sistema de reacción ante ataques en una red basada en ontologías*, Master Final Thesis, July, 2008. Madrid.
10. Rule Engine plugin Jess. <http://herzberg.ca.sandia.gov/>
11. A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Berlin, Germany: Springer, 2003.
12. P. Stavroulakis, *Neuro-Fuzzy and Fuzzy-Neural Applications in Telecommunications*, Springer, 2004, Berlin.