2009

Análisis Comparativo de Algoritmos en Segmentación de Iris

Universidad Carlos III de Madrid

Jesús Tejedor Gómez
Supervisor: Miguel A. Patricio Guisado
01/06/2009

En agradecimiento a todos aquellos que lo saben, por su interés, paciencia (mucha paciencia) y que aportaron libremente su tiempo para escucharme y sugerirme ideas sobre cómo enfocar este proyecto.

También agradecer a mi tutor Miguel Ángel Patricio por su dedicación y extensa aportación.

Índice de Contenidos

Índice de Figuras

Índice de Tablas

1. Introducción

La biometría, el estudio de métodos para el reconocimiento único de personas a través de sus facultades físicas o comportamientos, ha ido ganando adeptos desde el inicio de la computación, debido a la continua proliferación de avances tecnológicos que permiten el uso de automatismos para el reconocimiento de personas.

Este interesante campo de investigación, de gran amplitud, ha sido, es y será utilizado para realizar labores de seguridad, de distinta índole y grado, despertando el interés en el ámbito público y privado, aunque con ciertas reticencias y críticas relacionadas con la privacidad personal.

Dentro del gran ámbito de la biometría donde se puede destacar, el reconocimiento de huellas dactilares, de retina y de voz, entre otros, se puede resaltar el reconocimiento de iris como una herramienta biométrica para el reconocimiento de personas de manera unívoca y de suma precisión.

En la siguiente tabla (Tabla 1 - Comparativa de técnicas biométricas) se puede encontrar una comparativa entre los más importantes sistemas biométricos actualmente en vigor, donde se percibe claramente como el reconocimiento de iris ofrece una de las mejores prestaciones.

	Fiabilidad	Facilidad de Uso	Prevención de Ataques	Aceptación	Estabilidad	Promedio
Reconocimiento de Iris	Muy alta	Media	Muy alta	Media	Alta	**Alto**
Reconocimiento retinal	Muy alta	Baja	Muy alta	Baja	Alta	**Medio**
Reconocimiento dactilar	Alta	Alta	Alta	Alta	Alta	**Alto**
Reconocimiento geom. mano	Media	Alta	Alta	Alta	Media	**Medio**
Reconocimiento de escritura	Media	Alta	Media	Muy alta	Baja	**Medio**
Reconocimiento de voz	Alta	Alta	Media	Alta	Media	**Medio**
Reconocimiento facial	Media	Alta	Media	Muy alta	media	**Medio**

Tabla 1 - Comparativa de técnicas biométricas

El reconocimiento de iris es uno de los métodos biométricos más confiables que existen actualmente, sus soluciones comerciales están basadas en la toma de imágenes de alta resolución y potentes herramientas que se ejecutan en computadores de excelentes prestaciones, ofreciendo una óptima garantía de fiabilidad en entornos condicionados. Estudios

han demostrado una fiabilidad de 1 entre 16 millones de encontrar dos iris exactamente iguales (1), como consecuencia, el reconocimiento goza de una gran precisión.

Este sistema biométrico está compuesto de un conjunto de fases o etapas, al igual que el resto de métodos, que se pueden apreciar en la imagen siguiente (Figura 1 - Diagrama de bloques de un sistema biométrico). El estudio a realizar, dentro del ámbito del reconocimiento de iris se centrará en la extracción de las características tomadas del sensor, en el diagrama, el bloque remarcado en rojo, que consistirá en la segmentación de la imagen para la obtención de aquella sub-imagen que sea únicamente iris.

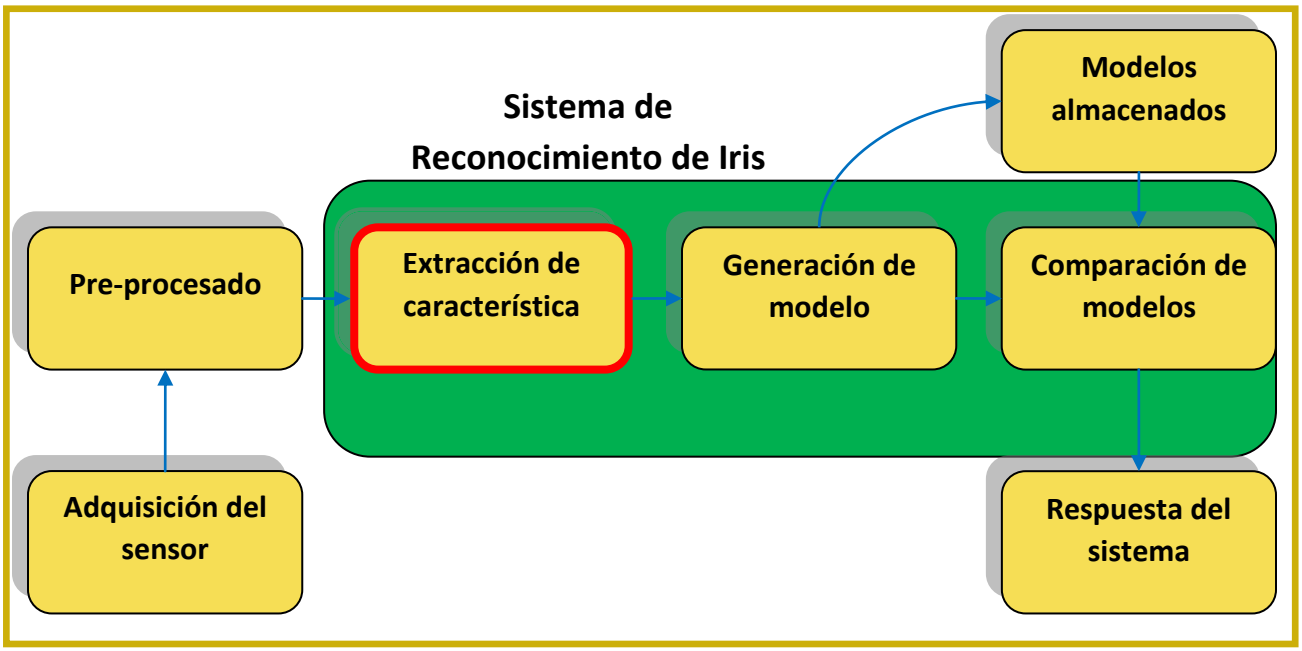


Figura 1 - Diagrama de bloques de un sistema biométrico

La extracción de características, y especialmente en el ámbito del reconocimiento de iris, la segmentación de iris, revierte un especial interés debido a la importancia de optimizar el sistema para una correcta localización y aproximación de la característica, a ser tomada en la identificación o verificación de personas. Disminuye el tiempo de computación, mejora eficientemente el sistema para su uso en tiempo real, evita la adquisición de componentes no deseadas en el modelo y facilita la labor de las sucesivas fases.

Los sistemas de reconocimiento de iris tienen segmentaciones basadas en condiciones muy favorables e imágenes de muy buena calidad. Estas condiciones no son fáciles de obtener y requieren la activa cooperación de la personas a identificar, estando sujeto a un proceso lento e inconfortable para el usuario por ello este documento se centrará a su vez en segmentaciones que no ofrezcan este condicionamiento y sean preparadas para entornos no invasivos.

1.1 Conceptos básicos

Para la comprensión de este importante campo de investigación, la segmentación de iris, es necesario entender y conocer ciertos términos que guardan relación con el contexto biométrico y que a continuación se detallan.

1.1.1 Anatomía del iris

Citando textualmente el iris es "un tejido de colores en forma de anillo alrededor de la pupila, a través de la cual entra la luz…penetrando en el interior del ojo" (2). El dilatador y el esfínter son dos músculos que controlan el tamaño del iris para ajustar la cantidad de luz que es permitida entrar en la pupila. La Figura 2 - Imagen C36_S2_I13 de la base de datos UBIRIS v2 muestra un ejemplo de imagen adquirida de una reconocida base de datos para investigación, posteriormente explicada en este documento. En ella se pueden observar las distintas partes del ojo a tener en consideración para la segmentación.

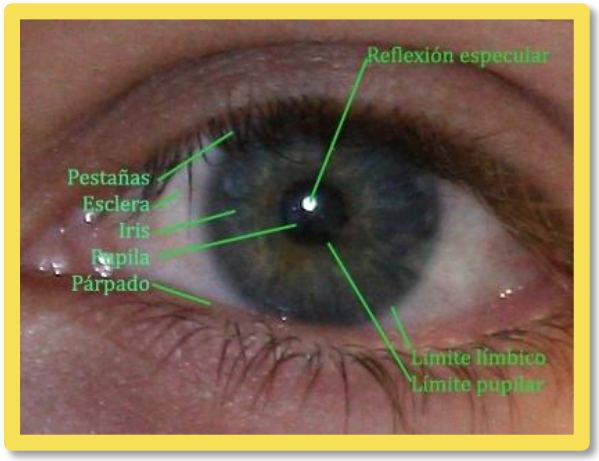


Figura 2 - Imagen C36_S2_I13 de la base de datos UBIRIS v2

El iris contiene un rico patrón de surcos, crestas y tramas pigmentadas. La superficie del iris está compuesta por dos regiones, la parte central denominada *zona pupilar* y la exterior denominada *zona ciliar*, delimitados a su vez por el *collarette*[1] que es el borde entre ambas regiones.

Aunque es un detalle pequeño, revierte especial importancia, la creencia de que la textura pigmentada del iris es aleatoriamente determinada durante el desarrollo fetal del ojo. Al igual, que son diferentes entre personas (con cualquier grado de parentesco) e incluso, entre los ojos de una misma persona (3).

[1] Collar en italiano

El color del iris puede cambiar en la tonalidad del color a lo largo de la niñez. Sin embargo, para la mayor parte de la vida humana, la apariencia del iris es relativamente constante aunque sujeta a enfermedades que pueden cambiarla.

1.1.2 Verificación

El proceso de aseverar si una entidad concuerda con la cual se compara (prueba de si X es el mismo que Y). En segmentación de iris, es la comprobación de que partes de la imagen son iris. En el reconocimiento de iris es contrastar las características obtenidas (iris segmentado) con la almacenada y comprobar si coincide.

En este escenario, en la segmentación de iris, se puede dar por cada componente (píxel) de la imagen cuatro determinadas salidas.

- *Verdadera Aceptación:* Sucede cuando el sistema determina que el píxel es formante del iris y realmente es.
- *Falsa Aceptación:* Sucede cuando el sistema determina que el píxel es formante del iris y realmente no es.
- *Verdadera Negación:* Sucede cuando el sistema determina que el píxel no es formante del iris y realmente no es.
- *Falsa Negación:* Sucede cuando el sistema determina que el píxel no es formante del iris y realmente es.

Estas definiciones, a menudo son representadas en una función curva denominada receiver operating characteristic (ROC) o también denominada directamente curva ROC. Siendo los ejes las dos aceptación o bien las dos negaciones. Un ejemplo de esta gráfica puede ser observada en la figura siguiente (Figura 3 - Ejemplo de curva ROC).

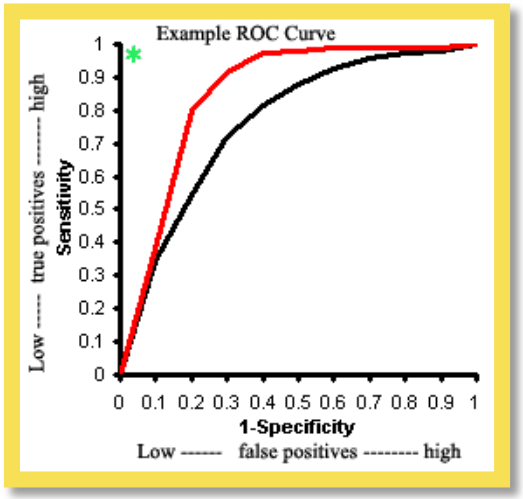


Figura 3 - Ejemplo de curva ROC

Un dato importante a extraer de esta gráfica y a menudo utilizado como métrica de fiabilidad es el Equal-Error Rate (EER), es una cantidad numérica que especifica cuando es idéntica la relación entre la falsa negación y la falsa aceptación.

1.1.3 Identificación

El proceso de determinar cuál es la identidad de una entidad contra una base de datos contenedoras de entidades conocidas (realizar una comprobación entre una colección de entidades {A, B, C,..., Z} para saber si esta X). Esta parte no pertenece a la segmentación de iris pero si es incorporada en el reconocimiento de iris, donde la segmentación es importante, ya que su salida determina la característica (sub-imagen) a comparar.

1.2 Actualidad e historia

La investigación biométrica del iris, se ha visto acelerada y ampliada dramáticamente a partir del 2001, ya que a partir de este año hubo una explosión de publicaciones, incluyendo patentes acerca de este campo.

Pero si se ha de remontar a alguna fecha inicial para el planteamiento del uso del iris como parámetro biométrico se puede superar el siglo fácilmente como sucede en el trabajo de A. Bertillon (4), donde se sugiere por primera vez el uso del iris como una biometría. Un dato curioso es que ha sido tomado a su vez por la ciencia ficción, visto en películas de James Bond en los años 80, al igual que en otras películas más recientes.

Pero la idea de la automatización del proceso de reconocimiento, es mucho más reciente. En 1987, Flom y Safir obtienen una patente para el diseño conceptual y no implementado de un sistema biométrico (5). Su descripción sugería un entorno altamente controlado, para obtener siempre el mismo diámetro de iris, proponen condiciones de iluminación constantes para obligar a la pupila a tener un constante tamaño pero existen condiciones impuestas en esta investigación que no son prácticas, ni tampoco posibles.

No será hasta 1994 cuando John Daugman obtenga la patente (6) con su algoritmo de reconocimiento de iris y promueva por primera vez la segmentación de la imagen para la obtención únicamente del iris. Pero es en el año 2003 cuando Daugman publica sus estudios referentes al reconocimiento del iris (7). Su trabajo, con el transcurso de los años, ha sido considerado finalmente un estándar y un punto de referencia para el resto de investigaciones.

Al poco tiempo, Richard P. Wildes propone en su publicación (8), una nueva investigación con un enfoque totalmente distinto, cimentando el reconocimiento de iris para futuras investigaciones. Ya que el trabajo de Daugman, como el de Wildes, sientan las bases de este campo biométrico.

Un gran número de publicaciones posteriores han retomado y ampliado el estudio a partir de los cimientos ya mencionados, pero actualmente la investigación de este campo ha tomado un nuevo enfoque muy interesante, la segmentación de iris en condiciones no cooperativas, tales como gafas, lentillas (transparentes y pigmentadas) o adquisición fotográfica en ángulos forzados, ya que anteriormente la toma de imágenes estaba altamente condicionada (iluminación constante, posicionamiento de la cabeza, distancia de la adquisición entre la cámara y la entidad….etc.). Aproximando así, el sistema biométrico a un entorno más real, en el cual se pueda identificar a personas sin tanto condicionamiento con garantías de exactitud.

1.3 Medios y herramientas del proyecto

Para la consecución del presente proyecto se han utilizado distintas herramientas y medios. Estos medios pueden ser separados en dos principales vertientes, hardware y software. A continuación se detalla en tablas los medios utilizados respectivamente.

SOFTWARE		
Herramienta	**Versión**	**Función**
Microsoft Office Word	2007	Edición del presente documento
Microsoft Office Excel	2007	Realización de cálculos
Visual Studio C++	2005	Implementación de algoritmos
Ubiris Dataset	2	Testeo y experimentación de la implementación
NICE Evaluator	1	Balance de error

Tabla 2 - Herramientas Software

HARDWARE		
Herramienta	**Versión**	**Función**
Intel Core Quad	E6600	Equipo de pruebas
Canon EOS 5D	N/A	Realización de las fotografías para UBIRIS v2.

Tabla 3 - Herramientas Hardware

1.4 Objetivos del proyecto

El objetivo principal de este proyecto es la consecución de un análisis comparativo, realizado a partir de un conjunto de algoritmos implementados expresamente para su comparación y extracción de conclusiones a partir de los resultados obtenidos. Para ello, el objetivo principal debe cumplir otros objetivos secundarios que permitan la consecución del primero, estos serán:

- Realizar un diseño software que facilite la implementación de los algoritmos
- Implementar los distintos algoritmos propuestos
- Comprobar la fiabilidad y calidad de cada algoritmo.
- Extraer conclusiones a partir de los datos obtenidos de la evaluación de cada algoritmo.

1.5 Acerca de este proyecto

El proyecto se centrará en la segmentación de iris, resaltando la importancia de esta fase en el reconocimiento de iris y explicando la necesidad de su utilización. Tomará un conjunto de algoritmos y los descompondrá, realizando un estudio pormenorizado de cada uno de ellos, comparándolos para resaltar sus principales virtudes y defectos.

El apartado 2 consiste en un amplio resumen de las técnicas más óptimas y vanguardistas existentes en la segmentación actualmente. En ella se deshilvanará un conjunto de estudios explicando su contenido.

A su vez este apartado recoge las principales bases de datos existentes usadas en la segmentación para testear la fiabilidad y robustez de los distintos algoritmos que se pueden utilizar.

El apartado 0 recoge un estudio minucioso de los distintos algoritmos a implementar y comparar, detallando su funcionamiento y características principales, explicando sus etapas y componentes.

En el apartado 4 se profundiza en el desarrollo de la aplicación para la comprobación y testeo de los algoritmos a estudiar desde una óptica relacionada con la ingeniería del software.

El apartado 5 se centra en la experimentación de los distintos algoritmos, explicando mediante gráficas y resultados obtenidos, un balance de cada uno de los métodos propuestos para la consecución y captura de la sub-imagen que conforma el iris.

También se explicará que base de datos se ha utilizado para realizar el análisis comparativo, las razones y virtudes por las que se ha tomado y como está compuesta.

En el apartado 6 se explora las distintas conclusiones que se pueden extraer del estudio realizado y se propone potenciales estudios futuros que se pueden realizar a partir de este proyecto.

Finalmente el apartado 7 recoge todas las referencias a estudios, investigaciones, artículos y patentes citadas en el documento y en el que buenamente está basado este proyecto pero que contienen investigaciones utilizadas en más campos de investigación, es decir, son investigaciones de carácter global.

2. Estado del arte

La finalidad de este apartado es explicar minuciosamente aquellas técnicas que reviertan mayor interés – por su precisión, por sus algoritmos utilizados – para la realización de la fase de segmentación de iris. No se abarcará el amplio espectro que contempla este abundante campo de investigación para poder centrarse únicamente en aquellos aspectos más destacables.

Para tener un concepto global del estado actual del arte, se anima al lector a profundizar en este campo mediante la publicación de Kevin W. Bowyer (9) donde se realiza un estudio hasta mediados del 2008, bastante exhaustivo, del reconocimiento de iris, especialmente el apartado 5 donde se recoge la segmentación de iris.

A continuación se expondrán distintos criterios para poder realizar una clasificación de los algoritmos más vanguardistas, según la cual, cada uno quedará enmarcado en un contexto propio.

2.1 Clasificación de las técnicas de segmentación

Es difícil encontrar una taxonomía que relacione los algoritmos propuestos en la actualidad, el amplio marco de algoritmos de diversa complejidad y orígenes, son el factor principal que determina este campo.

Aún así se mostrarán un conjunto de criterios que establecen unos principios según los cuales las distintas técnicas puedan ser clasificadas. Esto criterios ya están subyacentes en las distintos algoritmos, como consecuencia, se percibe un panorama global del enfoque realizado por los investigadores en este campo.

2.1.1 Base de Datos

En los trabajos iniciales, debido al comienzo en un campo inexplorado y virgen, no existen estándares, ni bases de datos con carácter específico, dejando al investigador, la labor, de realizar su propio banco de pruebas para la exploración. En este apartado entran una gran cantidad de estudios que adolecen de este problema por falta de rigurosidad en los resultados experimentales, ya que no poseen un baremo sólido que proporcione credibilidad y fiabilidad al estudio.

Aun así, en la actualidad, aún perviven ciertas investigaciones que, o bien no ofrecen resultados experimentales, o bien utilizan un dataset propio que no ofrece contraste. En ambos casos, es difícil medir la calidad de los algoritmos si no se propone un medio de certificada garantía, posponiendo estos estudios en preferencia de aquellos que si han sido testados y comprobados.

En definitiva, las investigaciones pueden ser clasificadas mediante sus estudios experimentales y la calidad de los mismos, en función de la certificación inherente existente, que consiste en usar una base de datos con un buen prestigio y unas altas prestaciones.

A continuación se realiza un pequeño resumen de las distintas bases de datos utilizadas en la segmentación de iris y facilitar así, la comprensión de cómo este criterio establece una correcta clasificación.

2.1.1.1 CASIA dataset

La primera base de datos que ofreció un entorno para la realización de pruebas fue CASIA (10), aunque en su primera versión no ofreció un correcto Ground Truth[2] que permitiese una comparativa y evaluación del proceso, ni unas imagen de máxima fiabilidad pues, para proteger la patente de adquisición de imágenes (y que no se apreciase la reflexión de los nodos infrarrojos) estaba tratada para que la pupila ofreciese un homogéneo color negro.

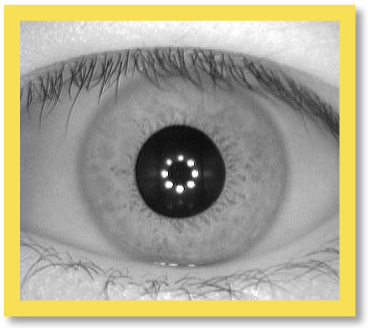


Figura 4 - Ejemplo de CASIA v3.

Posteriores versiones de esta extensa base de datos palian todos sus problemas, ofreciendo tanto posibilidades de testeo como imágenes con reflejos especulares en la pupila.

Esta base de datos en su versión tres cuenta con 22051 imágenes, de 700 sujetos y 1500 ojos, las imágenes tienen una profundidad 8-bit en nivel de gris, y cuenta con tres colecciones, en interior y auto-realizada la imagen, en interior y con/sin iluminación y en exteriores.

2.1.1.2 NIST ICE dataset

Propuesta para el certamen norteamericano NIST ICE (11), una competición para promover la creación de algoritmos robustos y eficientes de segmentación y reconocimiento, fue recopilada por la Universidad de Notre Dame y responde a imágenes tomadas en 250 sujetos y más de 30000 fotografías para cada ojo.

[2] Ground Truth: Se hace referencia a una colección que permite determinar en una clasificación supervisada una verificación de los valores clasificados. Es decir, es una colección que determina correctamente la clasificación y puede ser comparada para determinar la efectividad de los algoritmos.

Figura 5 - Ejemplo de NIST ICE

La base de datos es de la modalidad "one-to-one"[3] para la correspondiente ejecución del algoritmo y provee de un Ground Truth donde comparar las imágenes y poder comprobar la calidad del algoritmo.

2.1.1.3 UBIRIS dataset

UBIRIS: A Noisy Iris Image Database (12), es una base de datos mucho más cercana a la realidad, ofrece un conjunto de imágenes caracterizadas por sus condiciones ruidosas y su acercamiento a entornos no cooperativos. Es una base de datos preparada para ser "one-to-one".

Es la que ofrece una mayor calidad, con imágenes a color y realizadas a distintas distancias, está realizada por Hugo Proença y el laboratorio Socialabs.

Figura 6 - Ejemplo en condiciones ruidosas de UBIRIS

Esta base de datos será ampliamente explicada en el apartado de experimentación pues es la base de datos utilizada para el testeo y comparación de los distintos algoritmos utilizados.

[3] One-to-one: responde a un tipo de ejecución en la cual se especifica que cada imagen es tomada una a una para su tratamiento e identificación.

2.1.2 Aprendizaje Automático vs. Iteración

Aunque guarda relación con la naturaleza del algoritmo, este apartado especifica, como los algoritmos pueden ser clasificados en función de las necesidades que presenten.

Un algoritmo de aprendizaje es un algoritmo que necesita una fase previa a la ejecución y testeo, en esta fase se pretende ajustar un conjunto de parámetros (diferentes en las distintas técnicas) para adaptarse al problema. Es decir, mediante la propuesta de información no estructurada en forma de ejemplos el algoritmo infiere una solución. Es un proceso de inducción del conocimiento y son procesos de caja negra[4].

Un algoritmo iterativo busca resolver un problema mediante sucesivas aproximaciones que se acerquen a la solución del problema comenzando a partir de una estimación inicial. Muy útiles para la resolución de problemas con un gran número de variables (como es el caso de la segmentación de iris).

Como consecuencia, los distintos algoritmos propuestos para la solución del problema de segmentación de iris pueden ser clasificados en función de su carácter iterativo o bien de aprendizaje, ya que sus fases contemplan únicamente una de estas dos vertientes.

2.1.3 Cooperativo vs. No Cooperativo

Los últimos estudios relacionados con la biometría del iris, están enfocados para entornos donde no sea necesaria la cooperación del sujeto a identificar o verificar.

Aunque esta clasificación hace referencia a la adquisición del sensor, está íntimamente ligado a la segmentación de iris, pues existe una mayor complejidad en el tratamiento de imágenes donde el sujeto no coopere, ya que no se puede tomar muchas hipótesis que faciliten la creación del algoritmo (tales como, la pupila está en el centro de la imagen).

En sistemas preparados para la no cooperación, la importancia del proceso de segmentación es vital, ya que es necesario capturar únicamente el iris y poder así mantener un reconocimiento invariante y sin ruido.

Por ello, y aunque este enfoque es relativamente nuevo, los algoritmo se pueden clasificar en algoritmos para la adquisición de imágenes no cooperativas y cooperativas.

Dentro de esta clasificación entraría a su vez, si el sistema es invasivo o no invasivo. Un sistema no cooperativo, debe ser no invasivo a su vez ya que el sujeto en la adquisición de la imagen no debe favorecer para la extracción de características.

[4] Proceso de Caja Negra: se denomina así a aquel proceso que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno

2.1.4 Naturaleza del Algoritmo

Finalmente, los algoritmos pueden ser clasificados mediante la naturaleza matemática que utilicen para la resolución de la segmentación. Este será el criterio establecido para clasificar los distintos algoritmos propuestos a continuación.

La segmentación del iris puede ser clasificada en función de las herramientas matemáticas de las que usa y a la familia que estas mismas pertenezcan. Para ello, es necesario, remontarse a los trabajos cimentadores de este campo de investigación, Daugman y Wildes.

La primera vertiente de naturaleza algorítmica la establecerá el trabajo efectuado por Daugman y los consecutivos estudios realizados por otros investigadores, el principal problema de esta investigación estriba en la complejidad computacional.

Wildes, con su proposición distinta, establecerá otro perfil genérico donde los investigadores exploran para la resolución del problema. Esta rama de investigación ha producido importantes logros en este campo.

Después este estudio se centrará en aquellas técnicas heredadas de la inteligencia artificial, especialmente del aprendizaje automático, ya explicado con anterioridad. Algoritmos de aprendizaje que ofrecen resultados óptimos a partir de estas herramientas matemáticas.

Finalmente y para concluir el estado técnico actual, se establecerá un apartado donde se reúnan algoritmos que recogen otras modalidades matemáticas y que ofrecen una perspectiva interesante, arrojando resultados de calidad.

2.2 Operador Integro Diferencial

Esta herramienta matemática fue utilizada por primera vez en la investigación que aborda el sistema biométrico completo del reconocimiento de iris de John Daugman (7). Como, una vez obtenida y pre-procesada la imagen, se obtiene aquella sub-imagen que conforma el iris.

Figura 7 - Segmentación y Codificación de iris por Daugman

Para ello, en sus trabajos más tempranos, Daugman asume que el iris es un anillo circular, aunque esta afirmación a menudo no es exacta pues los límites del iris pueden no ser perfectamente circulares. Para el proceso de detección utiliza un operador integro-diferencial basado en tres parámetros (radio, y coordenadas del centro), buscando en el espacio paramétrico de tal manera que:

$$\max(r, x_0, y_0) \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right|$$

Donde:

$G_\sigma(r)$	Es una función de suavizado.
$I(x,y)$	Es la imagen del ojo
r	Es el radio del límite límbico del ojo
x_0	es la coordenada x en la imagen del centro del límite límbico
y_0	es la coordenada y en la imagen del centro del límite límbico

Recientemente, Daugman ha realizado investigaciones donde la asunción circular total no está incluida, y consigue mejorar el modelo de límites del iris. Incluso si están parcialmente ocultos por pestañas o párpados son encontrados los límites (13).

Únicamente este operador no garantiza la búsqueda exitosa del iris por ello, distintos investigadores han completado este operador con un proceso más laborioso y exacto que será explicado a continuación.

2.2.1 Regularización de límites y eliminación de reflexiones

Esta mejora de la segmentación basada en el operador integro-diferencial es propuesta por Labati en (14). La finalidad de Labati en (14). La finalidad de esta mejora es obtener un algoritmo de una mayor robustez frente a entornos menos frente a entornos menos cooperativos, y tolerante a ruidos. Basado en imágenes con formato de escala de grises, conlleva un de escala de grises, conlleva un proceso que se puede observar en la

Figura 8 - Diagrama del algoritmo de mejora del operador integro-diferencial.

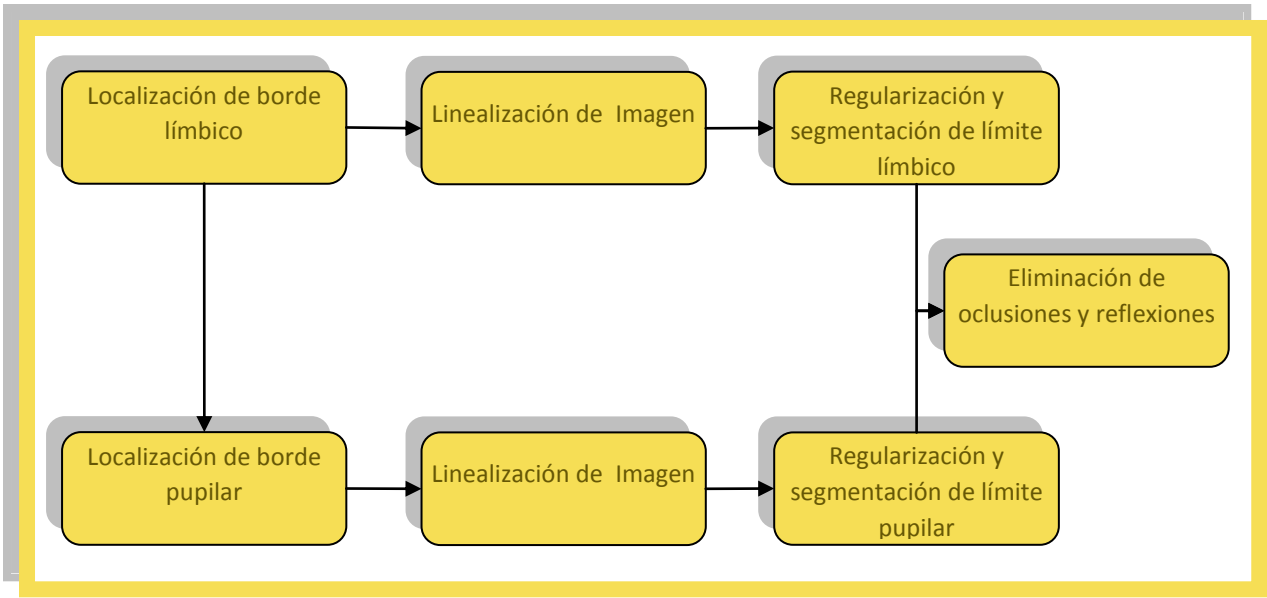


Figura 8 - Diagrama del algoritmo de mejora del operador integro-diferencial

La mayoría de los parámetros existentes en el operador integro diferencial son hallados experimentalmente, σ para suavizar el resultado y los radios de los bordes límbico y pupilar están acotados en un intervalo cerrado empírico.

$$Sea\ R_l\ el\ radio\ límbico\ \Rightarrow R_l \in [R_{l,max}, R_{l,min}]$$

$$Sea\ R_p\ el\ radio\ límbico\ \Rightarrow R_l \in [R_{p,max}, R_{p,min}]$$

Mientras que para determinar los centros de ambos límites (las coordenadas en la imagen del pixel central de cada límite) no es posible introducir conocimiento *a priori* debido a las características propias de la base de datos, donde el ojo puede estar situado en cualquier espacio de la imagen y ninguna hipótesis, con el fin de hacer más eficiente al algoritmo, puede ser introducida (como por ejemplo, los centros circulares pueden estar situados por el centro de la imagen).

Mucho más interesante, son las fases siguientes que gestionan y mejoran el operador en sí. Existen tres fases, la primera consiste en la linearización de los límites circulares, es decir un aplanamiento de los anillos con el fin de realizar una localización de carácter más fino, la segunda es la regularización de las franjas obtenidas en el apartado anterior con el fin de eliminar contornos no deseados, mientras que la tercera es suavizar el contenido eliminando así cualquier tipo de oclusión o reflejo especular.

La fase de linearización consiste en:

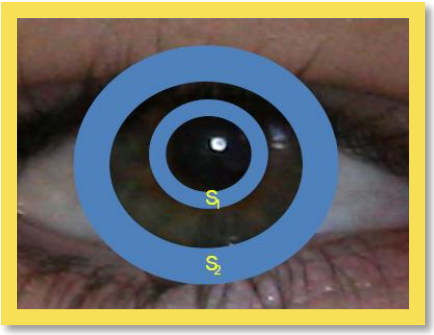


Figura 9 - Extracción de franjas en el límite límbico y pupilar

- Extraer dos franjas de la imagen original, como se observa en Figura 9 - Extracción de franjas en el límite límbico y pupilarmediante el uso del gradiente del radio obteniendo la imagen R(x, y) (para la franja límbica y franja pupilar).

1) Obtención de imágenes convolucionadas con un clásico Sobel Kernel 3x3

$$G_x(x,y) = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * I(x,y) \quad G_y(x,y) = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

2) Obtención de la magnitud $G(x,y)$ y la fase de la señal bidimensional $\theta(x,y)$

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad \theta(x,y) = \frac{1}{\tan\frac{G_x(x,y)}{G_y(x,y)}}$$

3) Obtención de la imagen ángulo mediante el procesamiento del ángulo de la línea recta que pasa a través del punto $\{x_0, y_0\}$ candidato para cada pixel $\{x, y\}$

$$\Omega(x,y,x_0,y_0) = \frac{1}{\tan\frac{y - x_0}{x - y_0}}$$

4) Quedando la imagen R(x,y)

$$R(x,y) = G(x,y) \cdot \cos\big(\theta(x,y) - \Omega(x,y,x_0,y_0)\big)$$

5) Finalmente esta transformación no lineal es aplicada a un intervalo de los centros obtenidos x_c, y_c con el radio correspondiente en cada caso r_c.

$$R_C = \begin{cases} R(x,y) & si\ r_c(1-\alpha) \le \sqrt{(x-x_c)^2 + (y-y_c)^2} \le r_c(1+\alpha) \\ 0 & sino \end{cases}$$

- Transformar las sub-imágenes obtenidas $R_l(x,y)$ y $R_p(x,y)$ de coordenadas cartesianas truncadas en coordenadas polares $b_l(\theta)$ y $b_p(\theta)$ mediante el uso de la conversión explicada en (15).

La fase de regularización consiste en:

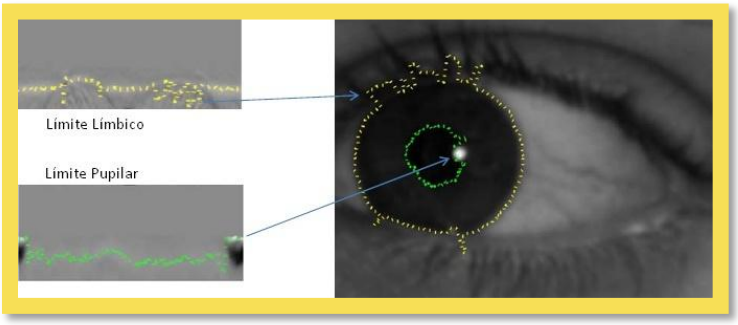


Figura 10 - Limites límbico y pupilar en coordenadas polares

- Partiendo de la asunción de que las discontinuidades percibidas en estas franjas son oclusiones, se realiza un algoritmo para que sean ambos límites curvas cerradas, regularizándolos. Para ello se realizan los siguientes pasos:

a) Se identifica el segmento continuo
b) Se controla la continuidad polar
c) Evaluación de la longitud y almacenamiento de los nuevos resultados obtenidos
d) Mientras alguno de los valores permanezcan discontinuos se procede al paso a).

- Finalmente se aplica un filtro paso bajo para suavizar la rápida transición obtenida en el anterior proceso garantizando que el límite final obtenido sea un camino cerrado.

La fase de eliminación consiste en tres sub-fases diferentes:

- Eliminación de las pestañas individuales:
    - Se aplica un filtro Gabor sobre una imagen umbralizada debido a que las pestañas presentan un patrón perceptible gracias a unas características específicas en frecuencia y angulosidad.

Filtro Gabor:

$$g(x,y) = K^{-\pi\cdot(a^2\cdot(x-x_0)^2+b^2\cdot(y-y_0)^2))^{j(2\pi F_0\cdot(x\cdot\cos\ \omega_0+y\cdot\sin\ \omega_0)+P)}}$$

Operación umbral:

$$L_1(x,y) = \begin{cases} 1 & si\ |I(x,y) * Re(g(x,y))| > t_3 \\ 0 & sino \end{cases}$$

$$t_3 = k\ \cdot max|I_r(x,y) * g(x,y)|$$

Donde:

- K es la escala de la magnitud de la dotación gausiana.
- $a\ y\ b$ son los dos ejes de la dotación gausiana.
- $x_0\ e\ y_0$ son las coordenadas del pico gausiano.
- F_0 es la frecuencia espacial de la onda sinusoide.
- ω_0 es la orientación de la onda.
- P es el desplazamiento de fase.

  - Se aplica la operación morfológica de cierre [5](con un elemento estructural de 3x3) para obtener el resultado final para eliminar el patrón percibido, es decir, las pestañas.

- Eliminación de reflexiones y conjunto de pestañas no distinguibles, basada en grandes contrastes en la varianza local, consiste en:
  - Realizar una operación umbral.

$$L_2(x,y) = \begin{cases} 1 & si\ var\big(I(x,y)\big) > t_4 \\ 0 & sino \end{cases}$$

Siendo:

- Var la varianza local en una matriz 3x3, cuyo elemento central es el pixel actual
- t_4 la máxima varianza de intensidad encontrado en la imagen

[5] La operación de cierre de carácter global esta explicada en la sección Anexo, operaciones morfológicas

- Aplicar la operación morfológica[6] con un elemento estructural acorde a las necesidades de la base de datos para obtener donde está situado las masas compactas de pestañas y las bajas reflexiones (L_3).
- Aplicar una operación umbral sobre la imagen para eliminar las fuertes reflexiones.

$$L_4(x,y) = \begin{cases} 1 & si\ I(x,y) > t_5 \\ 0 & sino \end{cases}$$

Siendo:

- t_4 el máximo valor de intensidad encontrado en la imagen

- Fusionar las distintas imágenes para eliminar todos los elementos no deseados con la imagen donde se encuentran los límites del iris. Se comienza por un conjunto de puntos iniciales para que, a continuación, se aplique una técnica de crecimiento espacial.

Puntos iniciales

$$L_5 = L_3\ OR\ L_4$$

Técnica de crecimiento

$$\mu + \beta \cdot \sigma < I(x,y)$$

Donde:

- μ Desviación local en una matriz de 3x3.
- β es el factor peso que debe ser encontrado empíricamente.
- σ Desviación estándar en una matriz de 3x3.
- $I(x,y)$ píxel a analizar

Esta mejora del operador, de gran complejidad computacional, pues el tiempo de procesado de cada imagen es bastantes considerable, es ante todo, de gran fiabilidad, ya que cuenta con una tasa de acierto cercana al 98% y ha sido testeada en más de una base de datos, tales como UBIRIS y CASIA, que cuentan con una gran rigurosidad, demostrando una gran robustez y una gran aproximación en la segmentación de iris.

[6] Esta operación de carácter global esta explicada en la sección Anexo

2.2.2 Localización de párpados y reflejos especulares

En este caso, Wojciech Sankowski, propone para mejorar el operador integro-diferencial, un algoritmo para la eliminación explícita de los párpados y los reflejos especulares existentes. Es un estudio interesante, pues su propuesta para solventar el problema de los párpados y pestañas es tanto eficiente como eficaz. El diagrama del proceso completo puede observarse en la Figura 11 - Esquema de algoritmo para la mejora del operador integro-diferencial.

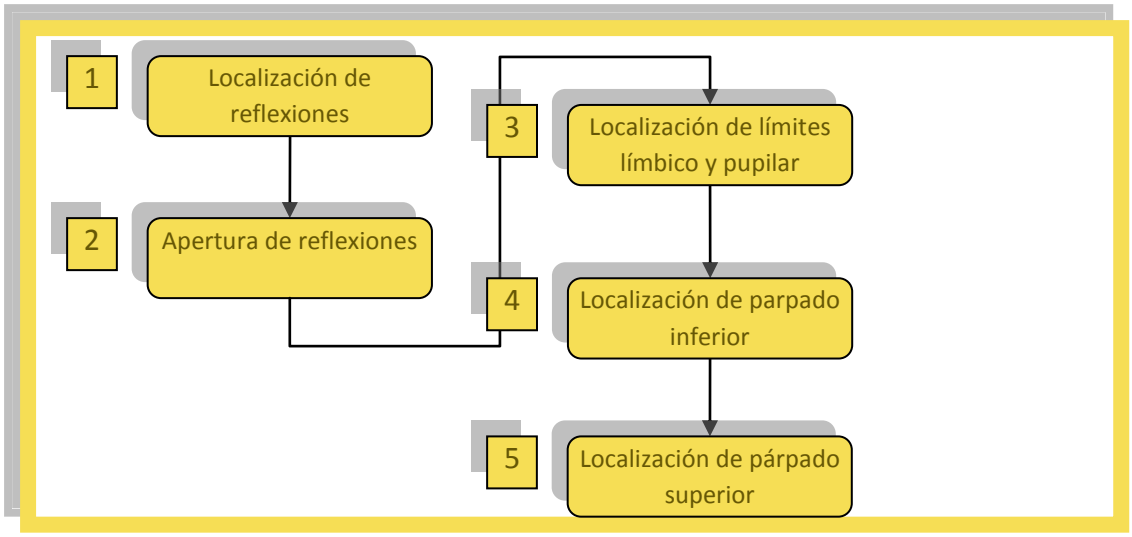


Figura 11 - Esquema de algoritmo para la mejora del operador integro-diferencial

El operador integro diferencial se aplica en la fase dos pero con una variación con respecto al original. Consiste en la incorporación de un modelo de iris en círculos no concéntricos pero tampoco completos. Como se puede apreciar en la Figura 12 - Patrón de búsqueda para el límite límbico el patrón que se propone para soslayar el problema de los párpados es un límite límbico no completo. Para ello, este estudio se apoya en la revisión de Daugman sobre el operador que él propuso (16).

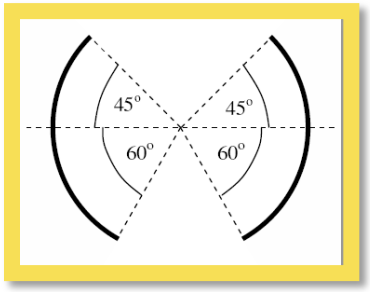


Figura 12 - Patrón de búsqueda para el límite límbico

A continuación se detallará cada fase de la que se compone la mejora del algoritmo, explicando las razones de su inclusión y las herramientas matemáticas utilizadas.

- La fase de localización de reflexiones tiene el cometido de encontrar aquellas secciones de la imagen que debido a razones de iluminación no pueden aportar información sobre el iris. Esta fase consiste en:
    - Transformar el espacio de color RGB de entrada en el espacio YIQ[7]
    - Realizar una operación de umbral para determinar aquellas zonas que sean reflexiones.

$$U_{ref} = I_{ave} + P \cdot (I_{max} - I_{ave})$$

$$U(x,y) = \begin{cases} 1 & si\ I(x,y) > U_{ref} \\ 0 & sino \end{cases}$$

Siendo:

- U_{ref} es el umbral de referencia para realizar la operación
- I_{ave} es la intensidad media de la imagen
- P es un valor comprendido entre 0 y 1
- $U(x,y)$ es la imagen resultante de la operación umbral

    - A la imagen resultante se le aplica un conjunto de operaciones morfológicas – dilatación y cierre[8] – para aumentar las secciones de iluminación y unir las secciones contiguas respectivamente
- La apertura de las secciones de reflexión esta realizada mediante un estudio de vecindad para aumentar los valores de las componentes de cada pixel. La finalidad de esta fase es incrementar el rendimiento del algoritmo y que sean más perceptibles las zonas de reflexión.

$$R_f = \frac{\sum_{i=1}^{4} w_i R_i}{\sum_{i=1}^{4} w_i} \qquad G_f = \frac{\sum_{i=1}^{4} w_i G_i}{\sum_{i=1}^{4} w_i} \qquad B_f = \frac{\sum_{i=1}^{4} w_i B_i}{\sum_{i=1}^{4} w_i}$$

$$siendo\ w_i = \frac{1}{d_i}$$

Siendo:

- R_f, G_f, B_f Las componentes del pixel analizado
- d_i la distancia correspondiente entre el pixel vecino y el pixel a analizar
- $i \in N$ y $i \in [0,4]$.

[7] El espacio de color RGB e YIQ están explicados en la sección de anexo, espacios de color.
[8] Las operaciones morfológicas en este capítulo son explicadas en la sección de anexo, operaciones morfológicas.

- Para la localización de los límites límbico y pupilar, este algoritmo ejecuta sobre la imagen el operador integro-diferencial. Pero especifica que es necesario encontrar los parpados que ocluyen parte del iris para la obtención de una segmentación óptima.

- Con el fin de localizar el límite inferior de párpado con el límite límbico se somete a la imagen a:

    - Pre-procesado para eliminar el ruido resultante por el hardware y por la iluminación. Para ello se somete a la imagen a un filtro gausiano en el dominio espacial con el fin de suavizar la imagen.
    - Detección de bordes horizontal para localizar el párpado aplicando un filtro Sobel[9].
    - Una aplicación de una operación de umbral

$$U_{grad} = K \cdot \frac{1}{W \cdot} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} G(x,y)$$

$$U(x,y) = \begin{cases} 1 & si\ I(x,y) > U_{grad} \\ 0 & sino \end{cases}$$

Siendo:

- U_{grad} es el umbral de referencia para realizar la operación, media ponderada de la intensidad del gradiente.
- W Ancho de la imagen
- H Altura de la imagen
- $G(x,y)$ la intensidad del gradiente de la imagen en el punto x, y
- $U(x,y)$ es la imagen resultante de la operación umbral

    - Procesado del mapa de bordes mediante la eliminación provisional del límite pupilar – así el algoritmo no se confundirá de límite a buscar – y la ampliación de los bordes obtenidos ya que también contiene información útil la vecindad del borde – mediante un filtro para suavizar –.
    - Una modelización del párpado inferior como un arco.
        - En primer lugar, operando sobre la imagen suavizada de bordes obtenida, se le superponen arcos de distinta amplitud y centros variables. Para saber que arco modeliza mejor el borde encontrado, se realiza la media ponderada de la intensidad de los pixeles del arco sobre la imagen.

[9] Citada y expuesto anteriormente, esta explicado en el apartado Anexo, Detección de bordes

$$g_{l1} = \frac{1}{I(n)} \sum_{i=0}^{I(n)-1} E(x_a\,(n,i), y_a(n,i))$$

Siendo:

- $E(\dots)$ La imagen obtenida en el procesamiento de bordes
- $I(n)$ El número de pixeles que contiene el arco en la imagen
- $x_a(n,i), y_a(n,i)$ son las coordenadas del pixel iésimo del arco enésimo
- $n \in Naturales\ y\ n \in [1, N]$ siendo N el número de arcos.

- Después, con el fin de ajustar mejor el arco al párpado, se utiliza la imagen original convertida a escala de grises y se realiza un procedimiento parecido al anterior, basándose en el principio de que la esclera contiene un valor en intensidad medio muy alto. La finalidad buscada es obtener cuando el arco termina.

$$g_{l2} = \frac{1}{J(n)} \sum_{j=0}^{J(n)-1} P(x_{end}\,(n,j), y_{end}\,(n,j) - y_{offset}\,)$$

Siendo:

- $P(\dots)$ Imagen pre-procesada en escala de colores de la imagen original
- $J(n)$ El número de pixeles que contiene el final del arco en la imagen
- $x_a(n,j), y_a(n,j)$ son las coordenadas del pixel final iésimo del arco enésimo
- y_{offset} viene determinado $y_a(n,j)$
- $n \in Naturales\ y\ n \in [1, N]$ siendo N el número de arcos.

- Finalmente, para juntar ambos criterios, se realiza la multiplicación de ambos valores obtenidos para cada arco y se escoge el de mayor cuantía.

$$g_l(n) = g_{l1}(n) \cdot g_{l2}(n)$$

Siendo:

- $g_l(n)$ Es el valor obtenido para detectar el párpado inferior
- $g_{l1}(n)$ Es la representación matemática de los bordes
- $g_{l2}(n)$ Es la representación matemática de la intensidad de la esclera
- $n \in Naturales\ y\ n \in [1, N]$ siendo N el número de arcos.

- Localización del límite del párpado superior. Este proceso es muy distinto al párpado inferior pues hay que tratar con las pestañas que también ocluyen el iris. Para ello se realizará un barrido con dos tipos de ventanas. Las ventanas estarán implementadas para representar las dos partes con la que colinda el límite del párpado. La primera será la del párpado y la segunda la de la esclera. A su vez se asume tres propiedades observadas:
  a. La piel y las pestañas presentas bajos valores en los canales verde y azul.
  b. La esclera tiene un alto valor medio RGB.
  c. Entre la esclera y el párpado se perciben grandes diferencias en el canal verde y azul.

  Para representar estas tres propiedades, se utilizan tres medidas que se explican a continuación.

Primera Propiedad:

$$g_{u1} = \frac{1}{1 + \frac{G_{ave\ ey} + B_{ave\ ey}}{2}}$$

Siendo:

- $G_{ave\ ey}, B_{ave\ ey}$ son las medias de los canales verde y azul respectivamente en la ventana párpado
- g_{u1} es la expresión matemática de la primera propiedad asumida.

Segunda Propiedad:

$$R_{ave\ sc} = \frac{1}{W' \cdot H'} \sum\sum R_{sc}(x,y) \qquad G_{ave\ sc} = \frac{1}{W' \cdot H'} \sum\sum G_{sc}(x,y) \qquad B_{ave\ sc} = \frac{1}{W' \cdot H'} \sum\sum B_{sc}(x,y)$$

$$g_{u2} = 1 + \frac{R_{ave\ sc} + G_{ave\ sc} + B_{ave\ sc}}{3}$$

Siendo:

- $R_{ave\ sc}, G_{ave\ sc}, B_{ave\ sc}$ son las medias de los canales rojo, verde y azul respectivamente en la ventana esclera
- $R_{sc}(x,y)$, $G_{sc}(x,y)$, $B_{sc}(x,y)$ son el valor del canal rojo, verde y azul respectivamente en el pixel cuyas coordenadas son x, y.
- W' es la anchura de la ventana esclera.
- H' es la altura de la ventana esclera
- g_{u2} es la expresión matemática de la segunda propiedad asumida.

Tercera Propiedad:

$$g_{u1} = \begin{cases} 1 + \Delta_{GB} & si\ \Delta_{GB} \geq 0 \\ 1 & sino \end{cases} \qquad \Delta_{GB} = \frac{G_{ave\ sc} - G_{ave\ ey} + B_{ave\ sc} - B_{ave\ ey}}{2}$$

Siendo:

- $G_{ave\ ey}, B_{ave\ ey}$ son las medias de los canales verde y azul respectivamente en la ventana párpado
- $G_{ave\ sc}, B_{ave\ sc}$ son las medias de los canales verde y azul respectivamente en la ventana esclera
- Δ_{GB} es el incremento producido en ambas ventanas sobre los canales verde y azul.
- g_{u3} es la expresión matemática de la tercera propiedad asumida.

Estas tres propiedades son fusionadas para ser integradas en ambas ventanas, es decir, para ventana párpado existirá una ventana esclera y ambas se complementarán para formar la medida que determinará si existe o no el borde del párpado. La fusión se realizará de la siguiente manera.

$$g_u = g_{u1} \cdot g_{u2} \cdot g_{u3}$$

Siendo:

- g_u la medida para indicar en qué grado es el límite del párpado
- g_{u1} la medida de la primera propiedad asumida
- g_{u2} la medida de la segunda propiedad asumida
- g_{u3} la medida de la tercera propiedad asumida

Pero una vez obtenida, una medida según la cual se pueda determinar el límite del párpado superior, se debe encontrar situando un gran número de pares de ventana para la búsqueda, como se aprecia en la Figura 13 - Búsqueda del límite límbico en su parte superior. Este proceso está compuesto de dos etapas.

- La primera de ellas, muy ruda, es para determinar con un ancho y altura de ventana un aproximación tosca de donde está situado el límite.
- La segunda, es sobre las ventanas que ya han sido determinadas como límites, en una ventana de dimensiones más reducidas, para realizar una búsqueda mucho más fina.

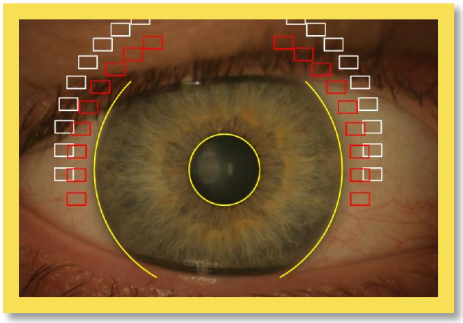


Figura 13 - Búsqueda del límite límbico en su parte superior

Este método de mejora del operador propuesto por Daugman ha resultado ser de suma precisión, según estudios experimentales sobre distintas bases de datos, ha obtenido una tasa de éxito en más del 98,4% para determinar correctamente que partes de la imagen son iris. Las pruebas realizadas han sido sobre UBIRIS, ISEP (17) y BATH (18), bases de datos de gran rigurosidad. Como consecuencia, se puede percibir este estudio, como sobresaliente por su gran precisión.

2.3 Detección de Bordes y Transformada de Hough

Wildes (8), en sus trabajos más tempranos, describe el desarrollo de su sistema de reconocimiento de iris para Sarnoff Labs con un enfoque técnico en la segmentación de iris muy

distinto al propuesto por Daugman y su operador integro-diferencial. Esta vía de investigación representa la otra gran vertiente existente en los estudios sobre el reconocimiento de iris.

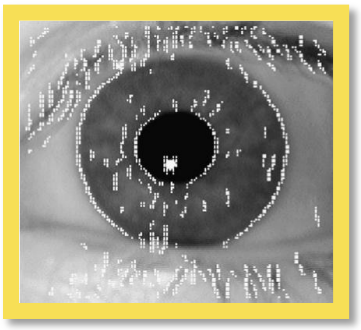


Figura 14 - Segmentación del ojo por mapa de bordes y transformada de Hough

En la segmentación de iris, Wildes propone realizar un mapa de bordes binario, y sobre ese mapa aplicar una transformada de Hough para la detección de círculos. Este método es considerablemente más estable frente a perturbaciones ruidosas que el propuesto por Daugman pero también contiene la premisa de la circularidad del iris y presenta una menor precisión – aunque muy leve –.

Para profundizar en todas las técnicas de segmentación y reconocimiento, Wildes, en 2005, realiza un capitulo de libro muy exhaustivo, donde detalla con suma precisión, todas las componentes de sus sistema e incorpora una gran cantidad de nuevos avances (19), mejorándolo considerablemente.

Se invita al lector a profundizar en el estudio de detección de bordes y en la transformada de Hough situado en el apartado

Anexos, especialmente los puntos 1 y 2, donde es explicado en detalle estas herramientas de carácter global.

La siguiente sección del este punto, es un amplio resumen de las distintas mejoras efectuadas por varios investigadores sobre esta técnica de segmentación y como ha repercutido en la evolución de esta herramienta.

2.3.1 Mejoras del algoritmo

A lo largo del tiempo que lleva esta técnica en vigor, muchos autores han propuesto mejoras considerables para este algoritmo con el fin de aumentar la precisión, fiabilidad, velocidad u otras características del mismo.

Huang en (20) propone un método interesante, sugiere la modificación de una primera localización en una imagen re-escalada para usar esta información como guía en la imagen original. Ellos presentan una única idea para realizar el paso de comparación con invariante rotacional. Usando una imagen de ambos ojos, utilizar el izquierdo para la segmentación y el derecho para la toma de la orientación.

Liu utiliza un detector Canny[10] y la transformada de Hough pero intenta simplificar los métodos para aumentar la velocidad de procesado (21). Los límites pupilar y límbico son modelados como circunferencias concéntricas. Imágenes de ejemplo son mostradas en las cuales la asunción parece plausible, pero la idea es probada únicamente en 5 personas y esa falta de experimentación es negativa.

Otros estudios, tales como Sung (22), además de aplicar las herramientas tradicionales, intentan encontrar el límite collarette mediante ecualización por histograma y filtros de paso alto.

El sistema "ND_IRIS" propuesto por Liu en (23), propone cuatro importantes mejoras al método tradicional. Los puntos del borde alrededor de reflexiones especulares son eliminados gracias a que son ignorados aquellos pixeles con gran intensidad en la detección Canny. Adicionalmente, es introducida una mejora considerable en la transformada de Hough. Ellos introducen una fase de suposición-verificación para capturar falsos candidatos de localización de iris y un método para la detección de oclusiones producidas por el párpado. Experimentos comparan los resultados obtenidos por Masek (24) y con la informe de localización del LG 2200 – es un sistema biométrico para el iris – La localización de Masek tiene una tasa de acierto del 90.9%, mientras que el LG 2200 obtiene 96.6 y finalmente el ND_IRIS supera a los otros dos estudios en la localización de iris con un 97.1%.

Algunos grupos siguieron la idea general de Wildes pero adicionalmente propusieron un método para encontrar una vasta localización de la pupila para guiar en fases siguientes la búsqueda de los límites del iris.

Lili y Mei (25) encuentran una localización del iris inicial vasta, basados en la asunción de que existen tres grandes secciones en el histograma de la imagen, correspondientes a las regiones de la pupila, el iris y la esclera respectivamente. Ellos también utilizan una detección de bordes

[10] Explicado en profundidad en el apartado Anexo, detección de bordes

y entonces ajustan círculos para los límites límbico y pupilar. La calidad de la imagen es evaluada en términos de nitidez, oclusiones de párpado y pestañas, y dilatación de pupila.

En el artículo de He y Shi (26), la imagen es transformada en binario para localizar la pupila, y posteriormente se aplica la detección de bordes y la transformada para encontrar el límite límbico.

Feng en (27) usa una estrategia "vasto-a-fino" para encontrar los limites aproximadamente como arcos de circunferencias. Una de las mejoras sugeridas es usar los bajos contornos de la pupila para estimar los parámetros del límite pupilar porque "es estable incluso cuando la imagen del iris está seriamente ocluida".

2.4 Aprendizaje Automático

Esta rama de inteligencia artificial, es una modalidad que encierra un gran conjunto de herramientas matemáticas. Este documento se centrará únicamente en aquellas técnicas utilizadas en el ámbito de la segmentación de iris.

Como se ha explicado previamente, el aprendizaje automático es una herramienta de resolución de problemas mediante la inferencia a partir de un conjunto de ejemplos, información no estructurada. Puede ser utilizado en una gran cantidad de campos de investigación y en sistemas biométricos siempre ha tenido un gran uso.

En este apartado se recogerán las principales investigaciones realizadas a partir del conjunto de herramientas que engloba el aprendizaje automático. Supone la tercera vertiente principal en la segmentación de iris, y los estudios presentan gran interés debido a la problemática de introducir esta rama en el procesado de imágenes.

2.4.1 Redes Neuronales

Los conceptos matemáticos asociados a este tipo de aprendizaje automático pueden ser estudiados en sección anexo V, así se puede centrar el discurso del documento en el problema de la segmentación de iris

Esta herramienta en especial ha sido poco utilizada en la segmentación de iris pero H. M. Al-Bakry propone una red neuronal cooperativa para la detección y localización de iris en imágenes (28). El mismo autor ha realizado estudios sobre otros sistemas biométricos tales como la detección de caras aplicando a su vez, redes neuronales también (29).

El algoritmo propuesto opera sobre imágenes en escala de grises para la localización del iris, el sistema determina si la imagen contiene iris con una salida binaria.

Contiene una pre-procesado de las imágenes que consiste en la ecualización de los histogramas de las imágenes con el fin de reducir la variabilidad en la iluminación y aumentar el contraste.

La entrada al sistema consistirá en imágenes de 20x20 píxeles y debe ser invariable para la posición, la rotación, y el escalado del iris. Para ello, se reduce el tamaño de la imagen hasta cumplir la condición de 20x20 con un factor de escalado de 1.2, consiguiendo así, la invariabilidad en translación y posición. Con respecto a la rotación, se consigue que el sistema sea invariable gracias a la alimentación a la red neuronal con las mismas imágenes con una rotación de 5º hasta describir 360º.

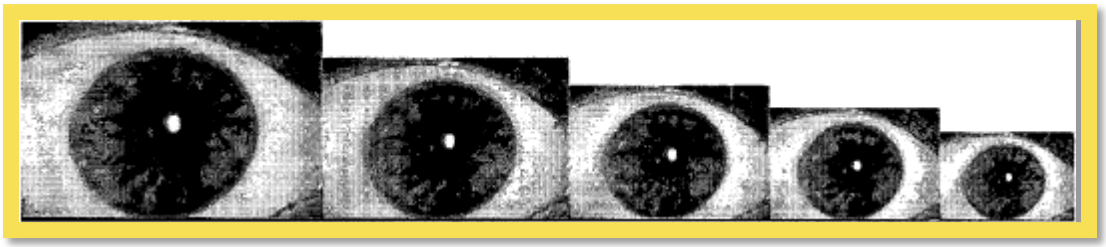


Figura 15 - Reducimiento recursivo de imagen en el estudio de El-Barky

Aun así, el estudio de Al-Barky no cuenta con una amplia, ni respaldada base de datos, si no que prepara su propia base de datos restando credibilidad a la investigación. Son 400 imágenes, las cuales son 200 con iris y 200 sin iris.

La red neuronal consiste en una capa de entrada (20 neuronas), una capa oculta (13 neuronas) y una capa de salida (una única neurona), basadas en una retroalimentación mediante la correlación cruzada en el dominio de la frecuencia.

Con el fin de mejorar el sistema se utilizan ventanas vecinas para revelar características predominantes de imágenes que no contengan iris y así utilizar dichas ventanas para la obtención de resultados más óptimos.

Los valores de cada neurona de la capa oculta quedará configurado para una parte de la imagen como:

$$h_i = g\left(\sum_{j=1}^{m}\sum_{k=1}^{n} x_j(j,k)\cdot I(j,k) + b_i\right)$$

Siendo:

- h_i la salida de la neurona oculta iésima
- $g(\cdots)$ función de activación de la neurona
- $x_j\ (j,k)$peso de la neurona de entrada j-ésima
- $I(j,k)$ pixel de la sub-imagen en la posición j-ésima y k-ésima
- m,n dimensiones de la sub-imagen a procesar.
- b_i umbral de la neurona oculta iésima.

Con la ecuación antes descrita se puede obtener el valor de cada neurona de la capa oculta con respecto a toda la imagen de la siguiente manera:

$$h_i(u,v) = g\left(\sum_{j=-\frac{m}{2}}^{\frac{m}{2}}\sum_{k=-\frac{n}{2}}^{\frac{n}{2}} x_j(j,k)\cdot Z(u+j,v+k) + b_i\right)$$

Siendo:

- $h_i(u,v)$ la salida de la neurona oculta iésima para toda la imagen Z
- $g(\cdots)$ función de activación de la neurona
- $x_j\ (j,k)$peso de la neurona de entrada j-ésima
- $Z(u,v)$ pixel de la imagen en la posición u-ésima y v-ésima
- m,n dimensiones de la imagen a procesar.
- b_i umbral de la neurona oculta iésima.

De esta manera se obtiene la correlación cruzada en cada neurona de la capa oculta. Dadas dos funciones cualquiera $f(x,y)$ y $d(x,y)$, su correlación cruzada puede ser expresada como:

$$f(x,y) \otimes d(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(x,y) \cdot d(x+m, y+n)$$

Siendo:

- $f(x,y) \otimes d(x,y)$ es la correlación entre las funciones f y d.
- $f(x,y)$ valor de la función f en el punto x, y.
- $d(x,y)$ valor de la función d en el punto x, y.
- m, n dimensiones del espacio

De esta manera, la salida de cada neurona en la capa oculta puede ser expresada como:

$$h_i = g(X_i \otimes Z + b_i)$$

Siendo:

- h_i la salida de la neurona oculta iésima para toda la imagen Z
- $g(\cdots)$ función de activación de la neurona
- X_i Es el vector de pesos de la neurona de entrada j-ésima
- Z la imagen a procesar.
- b_i umbral de la neurona oculta iésima.

Gracias a esta última fórmula, la correlación cruzada, que es la salida de cada neurona en la capa oculta, puede ser expresada en el dominio de la frecuencia gracias a la transformada de Fourier:

$$X_i \otimes Z = F^{-1}(F(Z) \cdot F^*(X_i))$$

Siendo:

- $X_i \otimes Z$ es la correlación cruzada en el dominio de la frecuencia
- F^{-1} denota la transformada de Fourier
- F denota el paso al dominio de la frecuencia de una función.
- X_i Es el vector de pesos de la neurona de entrada j-ésima
- Z la imagen a procesar.

Finalmente se obtiene la salida de la red neuronal como:

$$O(u,v) = g\left(\sum_{i=1}^{q} w_0(i) \cdot h_i(u,v) + b_0\right)$$

Siendo:

- $O(u,v)$ es la salida de la red neuronal cuando la ventana deslizante esta localizada en la posición u, v
- $g(\cdots)$ función de activación de la neurona
- $w_0(i)$ es el peso de la neurona iésima de la capa oculta para la neurona de la capa de salida
- q dimensión de la capa oculta
- $h_i(u,v)$ valor de la salida de la neurona iésima de la capa oculta
- b_0 umbral de la única neurona de la capa de salida.

El estudio realizado con la red neuronal acaba realizando una demostración, acerca de la menor complejidad computacional teórica, que tiene el uso de los parámetros internos de la red, en el dominio de la frecuencia con respecto a si se hubiese realizado en el dominio espacial.

Como conclusión, se puede observar como este estudio, *a priori* de gran interés, va perdiendo rigurosidad y calidad debido a la falta de base de datos con la cual realizar un correcto test, a la gran tosquedad en la localización del iris y a el uso de un simple perceptrón multicapa en vez de redes neuronales mucho más sofisticadas y precisas tales como ART 1 (30) o Kohonen (31).

2.4.2 Support Vector Machine (SVM)

Las máquinas de soporte vectorial o máquinas de vectores de soporte (Support Vector Machines, SVM)[11] son un conjunto de algoritmos desarrollados recientemente por Vladimir Vapnik y su equipo en los laboratorios AT&T. Pertenecen a la familia de los clasificadores lineales puesto que inducen separadores lineales o hiperplanos en espacios de características de muy alta dimensionalidad (introducidos por funciones núcleo o kernel) con un sesgo inductivo muy particular (maximación del margen).

Inicialmente se usaron para problemas de clasificación binaria, pero después se ha extendido su uso a problemas de regresión, agrupamiento, clasificación multi-clase, regresión ordinal, y se está trabajando en la búsqueda de resolver problemas más complejos (árboles y grafos).

[11] Esta técnica es explicada ampliamente en la sección anexo, Support Vector Machine

En los sistemas biométricos se ha usado tanto para el reconocimiento de caras, como el de huellas dactilares e incluso en el reconocimiento de iris.

Especialmente, en el reconocimiento de iris, es usado para la segmentación de iris, y es, el estudio que se va explicar, debido al gran interés que tiene, ya sea por la rigurosidad del estudio, como por el uso de esta nueva técnica, o como la gran robustez y precisión de la que hace gala.

Jiali Cui en (32) propone un método basado en tres fases:

- *Una detección de pupila, mediante el análisis de masas de color negro con la técnica de máquina de vectores de soporte:*

  La pupila es un disco circular negro. Para evaluar la circularidad de las regiones negras de la imagen introducida se utiliza conocimiento geométrico.

$$I_c = \frac{S}{R^2}$$

Siendo:

- $R = max\{dist(p, p_c) | p \in region\}$
- S indica el área de la región
- I_c Intensidad circular del área
- p, p_c es el punto más alejado del centro y el punto central de la masa respectivamente.

Así se ha conseguido un vector de características bidimensional (circularidad, área), que es utilizado en SVM para obtener un correcto hiperplano. La función entonces de decisión queda de la siguiente manera:

$$e_{SVM} = sgn\left(\sum_{i=1}^{n} \alpha_i^* \cdot y_i \cdot K \cdot (x_i, x) + b^*\right)$$

Siendo:

- e_{SVM} la salida de decisión
- α_i^* , b^* los factores de maximación obtenidos
- $K \cdot (x_i, x)$ La función de distancia radial por el kernel escogido (RBF).
- sgn signo de la función

Para eliminar el ruido resultante de la salida se somete a la imagen a dos operaciones morfológicas, cierre y apertura para eliminar dicho ruido.

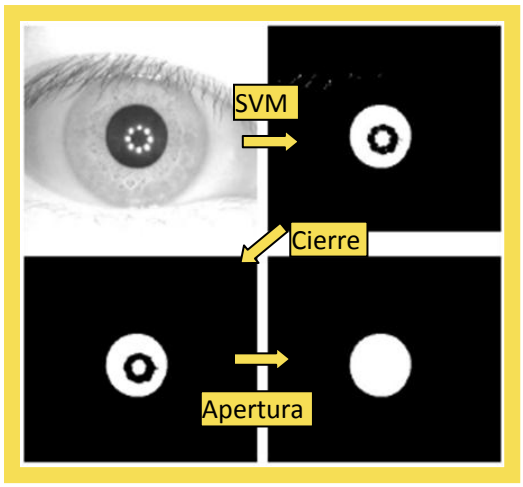


Figura 16 - Ejemplo de clasificación de Pupila con SVM

- *Una detección de iris, mediante los algoritmos LDA12 en coordenadas polares.*

LDA es el clasificador lineal escogido ya que optimiza la separabilidad de las diferentes clases, disminuye la dimensionalidad de la búsqueda y ofrece una importante mejora la velocidad de procesado con respecto a otros clasificadores lineales. Será usado con un conjunto de muestras de ejemplo que permitan realizar distinción de clases.

$$e_{LDA} = w^{*T} \cdot x$$

Siendo:

- e_{LDA} la salida de decisión
- x vector de entrada
- w^{*T} el factor optimizado para minimizar el error de clasificación.

Las clases a distinguir serán dos tipos de entrada diferente, la primera una región sin iris y la segunda una región con iris

Figura 17 - Ejemplo de imágenes en coordenadas polares para LDA

[12] Este algoritmo secundario puede ser estudiado según el artículo del creado Ronald A. Fisher (48).

- *Un sistema de fusión, para combinar de manera satisfactoria el uso de ambas técnicas mediante un mecanismo de votación.*

  Finalmente hay que fusionar la información obtenida de ambos clasificadores para la obtención de una salida que determine en una imagen que parte es iris y cual no. En definitiva, el problema consiste en reunir la información para poder separar en la imagen cada pixel en dos posibilidades, cual píxel es iris y cual no.

$$Umbral_{SVM}(x \in Clase_{iris}) = \begin{cases} 1 & e_{SVM}(x) \geq 0 \\ 0 & sino \end{cases} \qquad Umbral_{LDA}(x \in Clase_{iris}) = \begin{cases} 1 & e_{LDA}(x) \geq 0 \\ 0 & sino \end{cases}$$

$$Umbral_{fusión}(x \in Clase_{iris}) = R_{SVM} Umbral_{SVM} + R_{LDA} Umbral_{LDA}$$

$$Decision(x) = \begin{cases} Clase_{iris} & Umbral_{fusión}(x \in Clase_{iris}) \geq Umbral_{fusión}(x \in Clase_{no\ iris}) \\ Clase_{no\ iris} & sino \end{cases}$$

Siendo:

- $R_{genérico}$ la relación entre el verdadero positivo y el verdadero falso del clasificador genérico
- x píxel de la imagen de entrada.

Este método basado sobre todo en el análisis de figuras, está sujeto al posible ruido existente tales como reflexiones especulares, pestañas o gafas. La base de datos en la que ha sido probado, CASIA, presenta un conjunto de prueba exento de cualquier tipo de ruido. En consecuencia, y a pesar de los buenos resultados obtenidos – una detección por encima del 98% de precisión – no puede ser utilizado en entornos no invasivos o no cooperativos.

2.5 Otros Algoritmos

Además de las ramas ya mencionadas, existen estudios, fuera de esta clasificación, que provienen de distintos tipos de técnicas relacionados con el tratamiento de imágenes y de carácter iterativo-lógico que no pueden ser clasificadas en las familias ya mencionadas.

Sin embargo, no se debe restar importancia a estos algoritmos porque hayan caído en esta categoría. A continuación se introduce las investigaciones más interesantes que existen en la literatura relacionada.

2.5.1 Sistema lineal-iterativo

La evolución de los sistemas biométricos avanza hacia una eficiencia temporal, con la finalidad de poder ser usados en entornos de tiempo real, sin constricciones al usuarios y en entornos nos invasivos y poco, o nada, cooperativos.

Para ello, se deben encontrar rápidos algoritmos con una fiabilidad óptima, Topi Mäenpää propone en (33) un proceso iterativo para la segmentación de una imagen para localizar el iris. En la Figura 18 - Diagrama del proceso iterativose puede encontrar un diagrama del proceso.

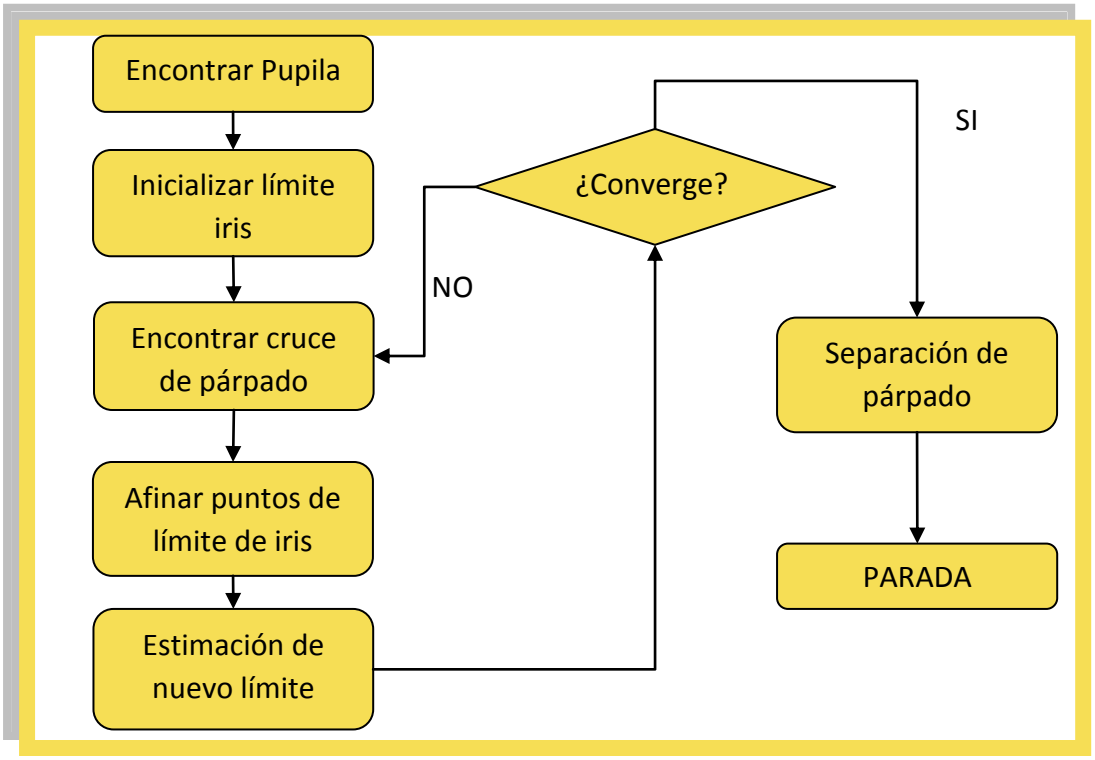


Figura 18 - Diagrama del proceso iterativo

A continuación se procede a explicar las fases que conforman el proceso iterativo junto con una breve conclusión para determinar la impresión final del estudio.

- *Encontrar Pupila:*

  Se utiliza imágenes que contienen un ojo, en escala de grises. Esta etapa contiene un pre-proceso de la imagen, consistente en realizar un umbralizado y una operación morfológica de apertura[13] para eliminar ruido.

  A la masa compacta y negra obtenida se le somete a la búsqueda del centro de la pupila.

[13] Explicado en la sección anexo, operaciones morfológicas.

$$(x_c y_c) = \frac{1}{N}\sum_{i=1}^{N}(x_i, y_i) \ \ \forall i \in Masa\ compacta$$

$$Radio = \sqrt{\frac{N}{\pi}}$$

Donde:

- $x_c y_c$ son las coordenadas del centro de la pupila
- x_i, y_i son los puntos contenidos en la pupila detectada.
- N es el cardinal de la masa obtenida.

- *Inicializar límite de iris:*

Esta fase inicializa puntos sobre una circunferencia con radio algo más grande que el obtenido para la pupila, el tamaño de este nuevo radio solo repercute en la velocidad de la convergencia.

$$(x_i y_i) = (x_c y_c) + r \cdot (\cos\alpha \cdot \sin\alpha)$$

$$\alpha = \begin{cases} -\pi/4 + \frac{i \cdot \pi}{2 \cdot \left(\frac{N}{2} - 1\right)} \\ 3 \cdot \pi/4 + \frac{\left(i - \frac{N}{2}\right) \cdot \pi}{2 \cdot \left(\frac{N}{2} - 1\right)} \end{cases}$$

Donde:

- $x_c y_c$ son las coordenadas del centro obtenidos previamente
- x_i, y_i son los puntos contenidos para la inicialización del límite.
- N es el cardinal de los puntos creados.
- α es el ángulo de inicialización en la sucesión de puntos.

- *Afinar puntos de límite de iris:*

Para guiar en la selección de puntos a obtener, este estudio se basa en dos propiedades:

  - El iris es (en promedio) más oscuro que la esclera.
  - La transición existente de oscuro a claro estima la localización exacta del límite.

Cuando cada punto ha sido previamente inicializado se le aplica una fuerza como se verá a continuación, basada en las dos propiedades antes mencionadas.

$$(x_i y_i) \longleftarrow (x_i y_i) + f$$

- Los puntos escogidos son actualizados en cada iteración mediante la fuerza

$$f = \mu_1 \cdot f_{i,1} + \mu_2 \cdot f_{i,2}$$

- La fuerza es la conjunción de otras fuerzas parametrizadas por μ_1 y μ_2

$$f_{i,1} = -G(x_i\ , y_i) = -\nabla(K \cdot I_{entrada}\ )$$

- La primera fuerza es el resultado del gradiente de la imagen de entrada $I_{entrada}$ tratada con un filtro gausiano K

$$f_{i,2} = |\nabla\ |G|(x_i\ , y_i)$$

- La segunda fuerza es un gradiente de segundo orden, un gradiente más profundo.

- *Estimar el nuevo límite límbico*

Una vez aplicada la fuerza en todos los puntos, una nueva localización para el centro del iris es estimada. Mediante la ecuación del circulo como comienzo en la construcción de un modelo de estimación lineal.

$$\sqrt{(x_c - x_i)^2 + (y_c - y_i)^2} = r$$

- Ecuación de la circunferencia

$$\begin{pmatrix} x_0^2+y_0^2 \\ \vdots \\ x_{N-1}^2+y_{N-1}^2 \end{pmatrix} = \begin{pmatrix} 2x_0 & 2y_0 & 1 & -1 \\ \vdots & \vdots & \vdots & \vdots \\ 2y_{N-1} & 2x_{N-1} & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} x_c \\ y_c \\ r^2 \\ x_c^2+y_c^2 \end{pmatrix}$$

- Modelo de estimación lineal, los términos elevados al cuadrado son omitidos del resultado por no ser lineales.

$$Z = H \cdot \Theta$$

- La ecuación anterior expresada en calculo matricial

$$\widehat{\Theta} = (\mathrm{H}^{\mathrm{T}} \cdot \mathrm{H})^{-1} \cdot \mathrm{H}^{\mathrm{T}} \cdot \mathrm{Z}$$

- Estimación de la solución utilizando la pseudo-inversa de Moore-Penrose. Para estimar el centro (x_c, y_c)

$$\hat{r} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(\hat{x}_c - x_i)^2 + (\hat{y}_c - y_i)^2}$$

- Para estimar el nuevo radio a calcular basta con coger el centro ya estimado y recalcular el radio

El algoritmo parará cuando la diferencia entre los nuevos puntos y los que había sea inferior a un píxel.

- *Encontrar cruce de párpados:*

Es un proceso paralelo e incorporado en el algoritmo general, situado después de la afinación de los nuevos puntos. Para detectar los párpados, solo es necesario inspeccionar el perfil de grises en la actual estimación de la localización de iris. El perfil es obtenido a partir de una imagen suavizada.

$$g(\gamma) = I_{suave}\,(x_c + r \cdot \cos\gamma\,,\quad y_c + r \cdot \sin\gamma)$$

Donde:

- $\gamma \in [0, 2\pi)$ es el ángulo de rotación
- $I_{suave}\,(x, y)$ es la imagen suavizada

Si el perfil de grises cruza el límite de cualquier párpado, existe un gran cambio percibido en la imagen. La exacta localización del límite del párpado es asumida a estar en el punto de mayor descenso en el ascenso.

Por lo tanto para considerar el cruce del límite del iris se debe satisfacer las siguientes condiciones:

1. $\frac{d}{d\gamma} g_s(\gamma)$ debe ser mayor que un umbral determinado experimentalmente.
2. $\frac{d^2}{d\gamma^2} g_s(\gamma)$ debe ser cero. Esta restricción existe para encontrar minimos/maximos locales.
3. $\frac{d}{d\gamma} g_s(\gamma)$ debe ser positiva para el límite superior y negativa para el límite inferior.

- *Corte de los párpados:*

Una vez la localización de iris ha concluido en su estado final, una última estimación es realizada para eliminar los párpado es usada. Se realiza mediante una estimación lineal acerca de todos los puntos obtenidos en todas las iteraciones contrastándola con la función de una parábola.

$$\begin{pmatrix} y_0 \\ \vdots \\ y_{M-1} \end{pmatrix} = \begin{pmatrix} x_0^2 & x_0 & 1 \\ \vdots & \vdots & \vdots \\ x_{M-1}^2 & x_{M-1} & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

- Estimación lineal de los puntos estimados con respecto a la ecuación de la parábola $y = a \cdot x^2 + b \cdot x + c$

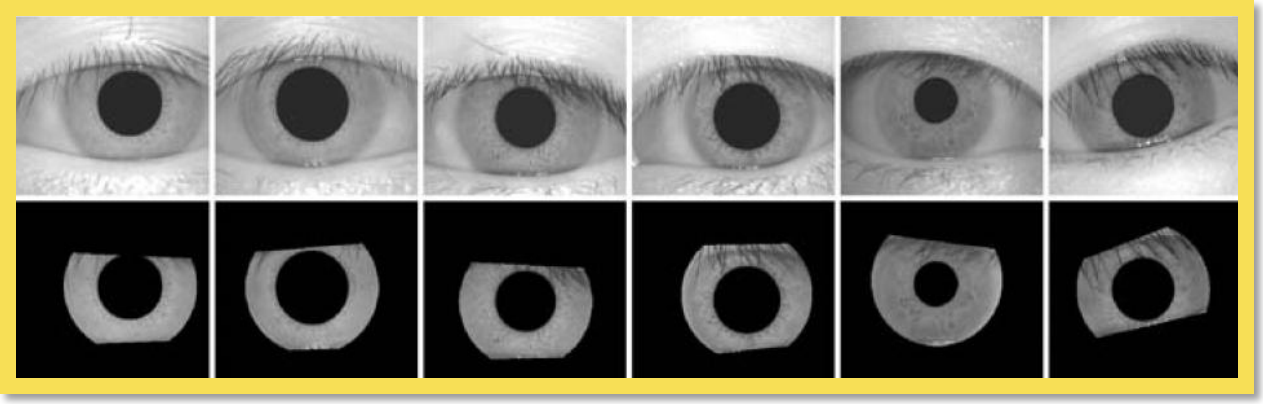


Figura 19 - Resultados obtenidos en el proceso iterativo

Este estudio presenta muchas ventajas, tales como la utilización del algoritmo en hardware modesto por su poca complejidad, inserción en casi cualquier aplicación y la rapidez con la que converge el algoritmo con resultados factibles. Pero no ha sido testeado en una base de datos óptima, pues, se ha probado únicamente en CASIA v1, la cual no contenía un entorno de pruebas. En consecuencia, la investigación de bajo coste computacional y alta eficiencia temporal, debe ser probada en una base de datos óptima para una comparativa interesante de sus resultados y comprobar, que eficiencia y eficacia estén equilibradas y sean óptimas.

En estudios futuros se podría contemplar la eliminación de las pestañas que se perciben en las imágenes finales pues hacen que el iris este sujeto a variaciones y no es posible entonces realizar técnicas de reconocimiento de iris sólidas y robustas.

2.5.2 Sistema Experto basado en reglas

La investigación explicada a continuación presenta un enfoque muy curioso debido a la aplicación de un "sistema experto" al procesamiento de imágenes. En la segmentación de iris, el sistema experto intenta emular aquellos principios intuitivos que la percepción humana realiza de forma innata.

Almeida en (34) propone una aproximación de la localización del iris basada en el paradigma de los sistemas expertos, mediante la definición de reglas. Existen algunas excepciones pues se debe adecuar los sistemas expertos al procesamiento de imágenes y como consecuencia se debe realizar algunos sacrificios sobre ciertas metodologías y conceptos propios de este paradigma.

El planteamiento inicial está basado en las reglas:

a) La pupila debe ser un circulo pequeño muy oscuro, localizado en la región central de la imagen
b) El centro de la pupila y el del iris deben ser muy cercano entre sí.

c) Los centros de la pupila y el iris deben ser más oscuros que la media de la imagen.
d) Alrededor de la pupila debe existir un pigmentación (marrón, azul, verde…).
e) El iris debe tener esencialmente una forma circular, eventualmente truncada en los límites superior e inferior.

Estas reglas no pueden ser aplicadas así, algunas han tenido que ser reestructuradas, otras eliminadas y otras simplificadas en reglas más simples.

Una vez realizada todas las reglas, específicas tanto para la detección, omisión y localización de regiones de las imágenes (detección de iris, localización de pupila y omisión de piel por ejemplo), se han dividido para forman un conjunto de tareas complementario. Entre estas tareas puede ser necesario la inclusión o entrada de imágenes ya tratadas desde el inicio.

Las tareas en las que se apoya esta investigación son:

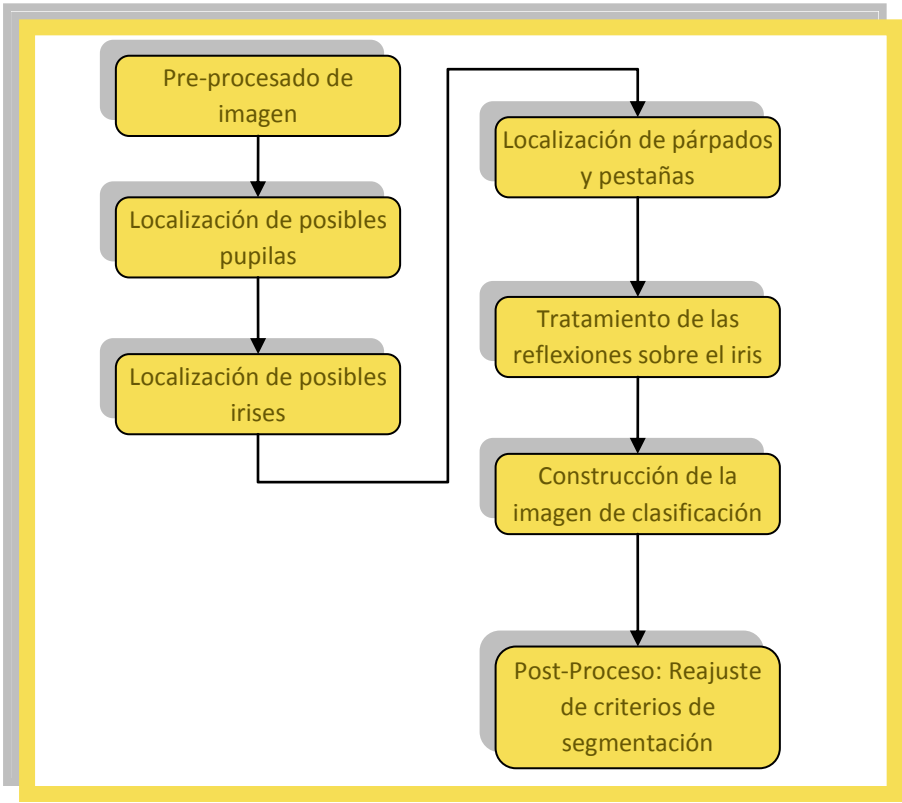


Figura 20 - Tareas del Sistema Experto

- *Tarea de Pre-procesado:*

  La primera de las sub-fases que componen esta tarea consiste en una primaria reducción de reflexiones. Para ello se somete a la imagen al análisis basado en las reglas:

  - En el espacio de color RGB[14] si se consideran los valores de cada canal deben ser muy altos para ser una reflexión de luz
  - Para no ser confundida estas regiones con otras también claras (tales como la esclera), se establece un tamaño máximo de región especular.

  Se sustituye estas regiones encontradas con la media de la mitad de los píxeles más oscuros localizados alrededor de esta misma región, para optimizar la búsqueda en tareas posteriores.

  La segunda de las sub-fases que componen esta tarea es la construcción de una imagen parcial del color. Esta parte consiste en la deformación del espacio de color, mediante la extracción en imágenes de la media de color azul y verde, al igual que la del canal rojo en su totalidad (admitiendo todo el rango de color).

  La tercera de las sub-fases es incrementar el contraste de la imagen original pasada a escala de grises, para facilitar la labor de la detección de la pupila. El resalte del contraste ser realiza mediante las reglas:

  - Las pupilas son negras, o en su defecto, muy oscuras.
  - Los pixeles formantes de la pupila, no existen grandes diferencias entre los canales en el espacio RGB.

  Para expresar matemáticamente estos dos conceptos se realiza un procesado en dos pasos:

Regla 1

$$I_{intermediate} = I + a \cdot (I - D_{oscuro})$$

Regla 2

$$I_{final} = I_{intermediate} + |R_{canal} - (G_{canal} + B_{canal})|$$

  La cuarta de las sub-fases consiste en la introducción de una máscara probabilística. Consistente en reducir la oscuridad de las zonas donde la presencia de la pupila y el iris tiende a ser menos frecuente. Esta mascara es una directa correlación entre el centro y cualquier pixel del centro en distancia euclídea).

[14] Este espacio es ampliamente explicado en Anexo, Espacios de Color.

- *Localización de la pupila:*

  La entrada de esta tarea es la imagen con el contraste aumentado. Está basada en tres métodos distintos.

  El primer método consiste en identificar el rectángulo vertical más oscuro de la anchura de la pupila. Experimentalmente se ha decidido utilizar 50 píxeles, como consecuencia el rectángulo tiene las dimensiones de 200x50 encontrando con una tasa de éxito del 96%, las pupilas.

  El segundo método consiste en encontrar las esquinas más oscuras en varias dimensiones, buscando sobre toda la imagen. La menor dimensión para este lado de dichas escuadras será el diámetro de la pupila más pequeña en las imágenes entrenadas, análogamente sucede lo mismo con la mayor.

  El tercer método utilizado es usado para crear semillas que serán añadidas más tarde. Mediante la predefinición de una rejilla lógica, se intenta ampliar el espacio de búsqueda para no caer en regiones prometedoras pero incorrectas. Es decir, se amplía todo el espacio de la imagen la búsqueda de la pupila.

- *Localización de los posibles irises:*

  La entrada de esta tarea es la imagen pre-procesada del canal rojo, para la correcta discriminación de la piel que contiene una alta intensidad.

  El sistema es análogo a la localización de pupilas, basada en tres métodos. El primero identificación de rectángulos que puedan contener el iris con una búsqueda. Un segundo método que determine la variación de color rojo en los irises de todas las imágenes de ejemplo para finalizar una variación de los posibles radios desde un mínimo y un máximo encontrado empíricamente.

  Se crean 10 candidatos como máximo, para la selección del iris y existe la asunción de circularidad del iris. El factor de calidad del iris queda:

$$E_{iris} = \alpha(V_{exit} - V_{ring}) + \beta(V_{exit} - V_{inside}) + \delta(255 - V_{ring})$$

Donde:

- E_{iris} es la evaluación de iris
- V_{exit} es el valor medio de los tres pixeles inmediatamente fuera del limite
- V_{inside} es el valor medio de los tres pixeles inmed. dentro del límite
- V_{ring} es el valor medio de los tres pixeles inmed. sobre el límite

$$\alpha + \beta + \delta = 1$$

Donde:

- α, β, δ son tres factores calculados empíricamente

- *Combinación del iris y la pupila:*

  En este apartado se realiza una comparativa de todas las localizaciones de iris y pupila. Utilizando una búsqueda heurística, se utilizan todos los parámetros previamente mencionados para encontrar la mejor solución conjunta mediante un sistema de votación.

- *Localización de párpados:*

  Para realizar este apartado se utilizaron criterios inspirados en la percepción humana. Estos criterios pueden ser transformados en un conjunto de reglas que, a su vez, se pueden reunir en tres principales aproximaciones:

  - Una positiva detección de los párpados a través de la identificación del color de la piel
  - Una negativa detección para los parpados a partir de un análisis del color del iris y los límites de la zona de color
  - Un análisis global más sofisticado de la imagen, para determinar la configuración de las características tales como pestañas, zonas mostrando esclera, cejas… y la posibilidad de que la detección previa del iris y la pupila estén interferidas en sus límites por los párpados.

  Aún así estas ideas no fueron implementadas en su totalidad y esta tarea únicamente describe un arco donde se supone que esta el párpado mediante un análisis de la piel únicamente.

- *Tratamiento de las reflexiones en el iris:*

  Esta tarea se centrará únicamente en la parte ya determinada como iris, despreciando el resto. El primer tratamiento es realizado en la fase de pre-procesado, pero es de cierta tosquedad y aplicado a toda la imagen. En este caso, se

asume que las partes con reflexiones tienen una mayor intensidad. Por ello se realiza una operación de umbralizado sobre la sub-imagen del iris para retirar aquello que se determina como reflexión. Se utiliza una imagen en escala de grises transformando directamente la imagen original del plano RGB a escala de grises.

- *Tratamiento de las reflexiones en el iris:*

  Para realizar esta tarea y hacer coincidir los correctos resultados que tenia del Ground Truth con respecto a la solución obtenida por su algoritmo, se baso en ajuste de post-proceso que consistía en un potente ajuste automático de los parámetros tratados tales como el grado de intensidad en zonas de transición, las reflexiones, el radio de la pupila y el iris gracias a las premisas obtenidas a partir de la comparativa de resultados.

Este algoritmo tiene dos evaluaciones distintas, la primera, con la misma colección con la que trabajó el autor y una segunda que es una ampliación de la primera. Ambas colecciones provienen de la base de datos UBIRIS y muestran un gran contraste que hay que analizar.

El ratio de éxito en la primera colección es del 98,2% mientras que en la segunda colección (aquella con la que el autor no estudió para el algoritmo) es 91.3% de éxito. Este descenso tan radical existente entre ambos ratios, es debido a un sobre-entrenamiento en las reglas, especificándolas excesivamente para una única determinada sub-colección de imágenes.

El algoritmo pierde una gran capacidad de generalización debido a la sobre-adecuación del estudio para la primera colección de imágenes.

El uso del espacio RGB, que por definición, se creó para el almacenamiento digital, es de carácter disperso, siendo muy difícil definir colores como sub-espacios algebraicos. Como consecuencia, cuando realiza asunciones relacionadas con el color (tales como la eliminación de la piel) el algoritmo pierde eficacia por usar este espacio y no otros como el espacio HSV[15] (y todas sus variantes) que ofrecen mejores características para la definición de sub-espacios.

El tiempo de procesado de una imagen supera ampliamente los 3 minutos en el tratamiento de todas las imágenes, haciendo este algoritmo deficiente para ser utilizado en ámbitos de tiempo real, tales como la video-vigilancia.

Un número excesivo de parámetros en el algoritmo, son absolutos y adecuados únicamente a un porcentaje de las imágenes específico. Esto merma la calidad de fiabilidad del algoritmo, reduciendo drásticamente sus prestaciones. Se debería haber utilizado parámetros relativos a cada imagen, adecuándose a ella.

[15] Ambos espacios de color, RGB y HSV son explicados en el apartado Anexo, espacios de color.

Por último, destacar el inmenso esfuerzo que ha debido ser la creación de un algoritmo tan complejo, intentando fusionar dos ramas de la computación, como son la ingeniería del conocimiento y la percepción computacional.

3. Algoritmos propuestos

En este apartado se explicará detalladamente los distintos algoritmos que se desean analizar en el presente documento. Los algoritmos a explicar tratarán sobre la segmentación de iris en entornos no-cooperativos, pues revierte especial interés, debido a que es una vertiente que aún queda por mejorar, porque en entornos invasivos, el estado del arte está muy avanzando y arroja unos resultados casi perfectos.

El primer algoritmo a analizar, es del autor Hugo Proença, propuesto en (35), apuesta por un algoritmo no-cooperativo, basado en *fuzzy c-means* y detección de bordes, que es comparado con otros trabajos encontrados en la literatura y que eran (en el 2003), los métodos más utilizados y elaborados.

Para el segundo algoritmo, se ha procedido a utilizar un estudio de T. Zan y Z. He, miembros del instituto "Laboratorio Nacional del instituto de reconocimiento de patrones de la academia china de automatización y ciencia" con sede en Beijing. Esté método, muy interesante, fue ganador del último certamen de NICE Iris y está basado en una compleja variación del operador integro diferencial con sutiles y óptimas mejoras.

Finalmente, se explicará un algoritmo de propia producción, una vasta aproximación, que se encuentra en fases iniciales basado en *k-means,* proyecciones en 1-D y un gran estudio del color en el espacio HSV.

3.1 Algoritmo propuesto I: Metodología para segmentaciones no cooperativas

En la literatura se puede encontrar un gran número de estudios en la biometría del iris por el autor Hugo Proença, miembro activo de la comunidad científica, ha aportado un gran número de mejoras y una perspectiva para enfocar el reconocimiento comparable con los padres (Daugman y Wildes) de este campo de investigación. En especial, hay que resaltar el sobresaliente trabajo realizado sobre la segmentación de iris en condiciones no favorables.

El algoritmo a estudiar, aborda la segmentación del iris desde una perspectiva no-cooperativa y no invasiva (36). Trabaja sobre distintos algoritmos pero propone una metodología propia que guarda gran interés, gracias a la gran robustez que presenta frente a condiciones ruidosas (tales como gafas, lentillas, imágenes mal enfocadas,…etc.). La finalidad del trabajo de Proença fue crear un mapa de bordes menos dependiente de las características específicas de la imagen. Como se puede apreciar en la Figura 21 - Diagrama de flujo en el Algoritmo 1el algoritmo está compuesto por 4 procesos. Inicialmente se definen la extracción de características que guardan interés para la segmentación. Después se establece un sistema de agrupamiento para determinar que partes son candidatas de ser iris y cuáles no. La tercera fase consiste en crear

un mapa de bordes a partir del agrupamiento realizado, para finalizar con una transformada de Hough que determine los límites límbico y pupilar.

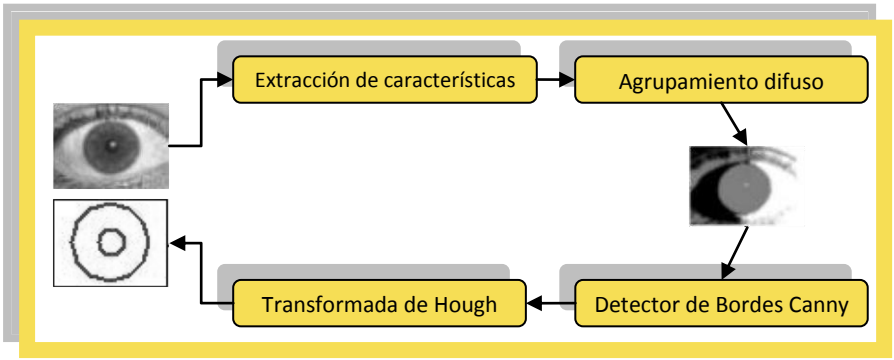


Figura 21 - Diagrama de flujo en el Algoritmo 1

Este estudio realiza una gran comparativa sobre distintos algoritmos pero en el presente documento nos centraremos únicamente en el algoritmo propuesto por Proença de propia elaboración pues es el que revierte un mayor interés en el enfoque no cooperativo. A continuación una detallada explicación de las distintas fases de las que está compuesto este algoritmo.

3.1.1 Extracción de características

Se realizan varias comprobaciones para seleccionar las mejores selecciones de características y se evalúan cuál de ellas ofrece simultánea y claramente la identificación del las regiones del iris minimizando la información ruidosa relacionada con los párpados y las pestañas.

Se concluye que la colección de tres parámetros (x, y, z), siendo x e y las coordenadas de la posición del pixel a estudiar y z la intensidad del pixel en la imagen, puede caracterizar cada píxel y permitir una correcta segmentación. Aunque se realizan pruebas con otros conjuntos de parámetros tales como momentos.

En la Figura 22 - Colección de características comprobadas puede comprobar un conjunto de características comprobadas, en las cuales se puede apreciar como posición + intensidad píxel es la que mejor agrupamiento consigue.

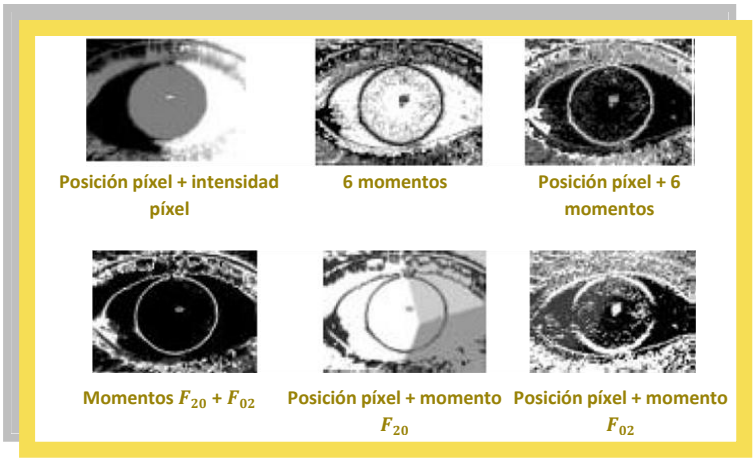


Figura 22 - Colección de características comprobadas

3.1.2 Agrupamiento difuso

En el agrupamiento (Clasificación), la características más importante del método a utilizar, debe ser su capacidad de clasificación, en la misma clase, todos los píxeles que pertenezcan al iris y a todos los remanentes en una clase distinta.

Se realiza un buen estudio sobre distintos algoritmos de agrupamiento, explicando su funcionamiento, pero en este documento, gracias a la evaluación percibida en el artículo, explicaremos y profundizaremos en aquel que provee una mayor tasa de acierto, el algoritmo *fuzzy c-means*, explicado ampliamente en la sección de anexo, herramientas matemáticas del algoritmo propuesto I.

Es una variante del *k-means*, difiere del mismo, en la premisa de que cada elemento pertenece únicamente a una clase. En el caso difuso, todas las semillas son actualizadas a partir de todos los puntos de la imagen con el fin de proveer un contexto en el límite no tan segmentado. En la Figura 23 - Ejemplo de agrupamiento con el algoritmo *fuzzy c-means*se puede apreciar un ejemplo de agrupamiento mediante el algoritmo *fuzzy c-means*. Se nota que los límites están conformados ambiguamente, determinando grados de pertenencia entre distintas clases.

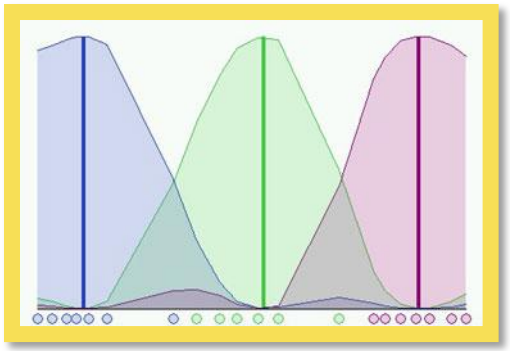


Figura 23 - Ejemplo de agrupamiento con el algoritmo *fuzzy c-means*

3.1.3 Detección de Bordes Canny

Para extraer un mapa de bordes, a partir del agrupamiento realizado, el autor propone utilizar el detector de bordes más preciso, el detector canny[16]. Gracias a este detector se toma una imagen que contiene únicamente los bordes encontrados en el agrupamiento. En la FIGURA se puede observar un ejemplo del resultado de usar este detector.

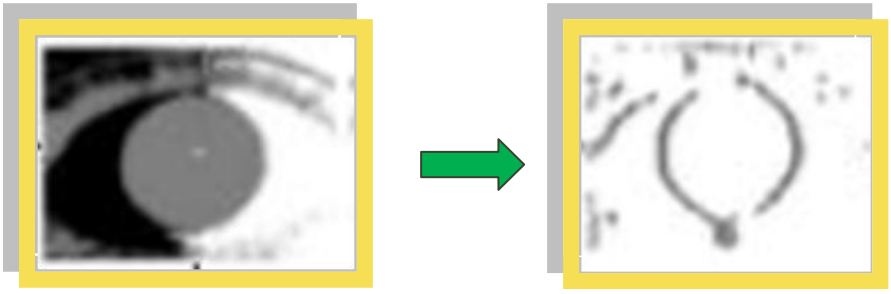


Figura 24 - Ejemplo de extracción de mapa de bordes mediante Canny

3.1.4 Transformada de Hough

Finalmente, y con la premisa de que, tanto la pupila como el iris, son perfectamente circulares se aplica el proceso conocido como Transformada de Hough para círculos[17], la cual extrae los iris encontrados. Como se verá en el apartado Experimentación, este proceso, no es tan eficiente como debiera, debido a que la premisa de circularidad no es cumplida siempre, la oclusión de los parpados y las pestañas pueden producir desvíos en la conducta del proceso.

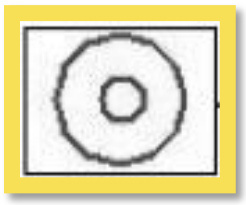


Figura 25 - Ejemplo de detección de círculos mediante transformada de Hough

3.2 Algoritmo propuesto II: Segmentación eficiente y robusta en iris ruidosos.

El centro de biometría e investigación en seguridad es un departamento del instituto de automatización chino, con sede en Beijing. Es una activa comunidad científica que ha intervenido en multitud de conferencias acerca del reconocimiento de iris.

[16] Este detector de bordes esta explicado en anexo, detección de bordes.
[17] Esta técnica de detección de círculos es explicada en anexo, transformada de Hough.

La última intervención fue realizada en el certamen NICE Iris, su trabajo, segmentación de iris en condiciones desfavorables obtuvo el primer puesto gracias a la robustez y eficiencia del algoritmo presentando una tasa de acierto muy desbordante.

Tienu Tan, Zhaofeng He y Zhenam Sum son los coautores de este algoritmo, explicado ampliamente pero sin detallar en el trabajo (37) proponen un enfoque totalmente nuevo y original con una gran resistencia a condiciones ruidosas. Llevan aportando considerables mejoras a la segmentación de iris desde el 2004.

La Figura 26 - Diagrama de bloques del algoritmo propuesto 2muestra en un diagrama de bloques las fases de la que se compone el algoritmo de segmentación, en los siguientes puntos se explicará en profundidad el funcionamiento de cada fase.

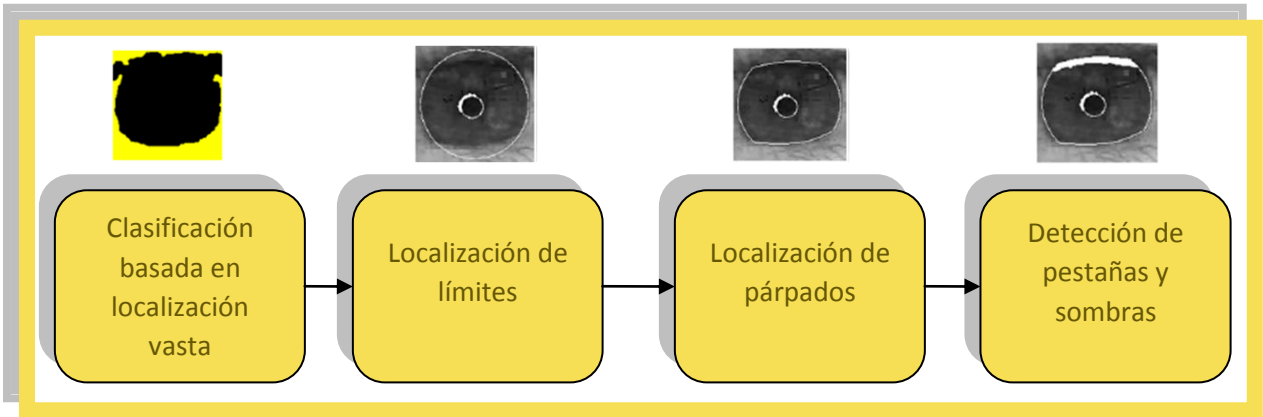


Figura 26 - Diagrama de bloques del algoritmo propuesto 2

3.2.1 Clasificación basada en localización basta

Está demostrado que los fallos de localización más habituales son producidos por el alto contraste local. Una sencilla idea para subsanar ese error es excluir las regiones que no conforman el iris mediante una localización inicial basta. Como consecuencia, se separará la imagen en distintas regiones, candidatos de iris, cejas, formas (tales como gafas) y piel.

Este proceso está compuesto de 2 fases consistentes en la inicialización y el agrupamiento de puntos no agrupados.

No obstante antes de realizar este agrupamiento es necesario, para no producir errores triviales, la eliminación de las reflexiones especulares existentes en el iris.

A continuación se explicará cómo se compone cada fase de este primero proceso.

3.2.1.1 Eliminación de reflexiones especulares

Esta fase esta basados en la realización de una función de umbralizado mediante un umbral adaptativo y una interpolación bilineal.

El umbral adaptativo T_{ref} se obtiene mediante el estudio del histograma de la imagen y la captura del 5% de los píxeles más brillantes encontrados.

Para no dejar un "hueco" en estos puntos y poder evaluar la imagen sin la pérdida de formas, se realiza una interpolación de los puntos señalados como reflexiones basada en la vecindad de la región analizada. De esta manera:

Para realizar la interpolación de $P_0(x_0, y_{0)}$, sean $P_l(x_l, y_0), P_r(x_r, y_0), P_t(x_0, y_t), P_b(x_0, y_b)$ las cuatro regiones/puntos vecinas de P_0 y $R(x, y)$ la máscara obtenida de la función umbral. Entonces se definen las nuevas coordenadas como:

$$x_l = \max_x \left\{ x: \sum_{i=0}^{L-1} R(x+i,,y_0) = 0, R(x+L,,y_0) = 1, x < x_0 \right\}$$

$$x_r = \min_x \left\{ x: \sum_{i=0}^{L-1} R(x-i,,y_0) = 0, R(x-L,,y_0) = 1, x > x_0 \right\}$$

$$x_l = \max_x \left\{ x: \sum_{i=0}^{L-1} R(x,,y_0+i) = 0, R(x,,y_0+L) = 1, y < y_0 \right\}$$

$$x_l = \min_x \left\{ x: \sum_{i=0}^{L-1} R(x,,y_0-i) = 0, R(x,,y_0-L) = 1, y > y_0 \right\}$$

En la Figura 27 - Ejemplo de eliminación especularse puede observar un ejemplo de cómo se realiza este procedimiento, la imagen a es la original, la imagen b es la máscara obtenida de realizar la función de umbral, junto con una mera explicación sobre los puntos de vecindad y la c el resultado de la interpolación.

a) Imagen Original b) Mascara de umbral c) Resultado final

Figura 27 - Ejemplo de eliminación especular

3.2.1.2 Inicialización de la vasta localización

Para la inicialización de las regiones se ha establecido empíricamente que la región de piel es más brillante que la conformada por el iris. Entonces se procede, con el fin de mejorar la eficiencia del algoritmo, pues considerar toda la imagen tendría un gran coste computacional, a determinar dos regiones iniciales.

La primera, denominada p_1, será considerada como piel y estará formada por el 30% de los pixeles más brillantes, mientras que, la región p_2 , constará del 20% de los píxeles más oscuros.

Nótese que las cejas pueden pertenecer a la región p_2, ya que también presentan unos valores de intensidad bajos, este error será subsanado en la etiquetación de las zonas.

3.2.1.3 Agrupamiento de los puntos no agrupados

Una vez inicializadas las regiones, el siguiente paso es determinar a qué regiones pertenecen aquellos puntos aun no clasificados. Para ello se procederá a seguir estos pasos:

1- Se calcula la intensidad media y la desviación estándar para cada región R.
2- Se escoge aleatoria un candidato de región, y se calcula la distancia de cada punto no agrupado mediante la distancia punto a región siguiente:

$$D(P,R) = \frac{|g_p - g_r|}{d_r}$$

Donde:
- g_p es la intensidad del punto
- g_r es la intensidad de la región
- d_r es la desviación de la región

3- Se evalúa que puntos son aptos para esa región, siempre y cuando satisfaga dos condiciones:
   a. $D(P,R) > T_{p2r} = 2.5$ Encontrado experimentalmente.
   b. Debe existir una conexión válida (vecindad 8) entre el punto y la región.
4- Si existe algún punto sin clasificar, se pasa al punto 2, si no, fin del agrupamiento.

3.2.1.4 Etiquetación semántica de las regiones

Después de la clasificación realizada de la imagen, queda identificar aquellos grupos que conforman el iris de los que no lo conforman. Para ello se partirá de las siguientes hipótesis.

- La región de la piel es identificable gracias a la intensidad más elevada que presenta.
- El genuino candidato de región de iris presenta habitualmente una forma más densa en el centro que en los límites horizontales.
- La ceja es una franja oscura horizontal encima de la región de iris.
- Las gafas presentan un marco que usualmente es más oscuro que la intensidad media y presenta una forma aproximadamente rectangular.

Para esta fase, la forma de la región, la intensidad y la posición relativa de cada grupo serán adoptadas para extraer la información semántica. Mientras que la posición e intensidad media de los grupos son auto-explicativas se debe razonar sobre el uso de la forma. Cada región será representada por mediante la relación entre anchura y altura.

Con la conclusión de esta fase, se tiene una la imagen dividida en grupos y una vasta aproximación sobre el iris que en siguientes fases será refinada. El proceso completo puede ser observado en la Figura 28 - Proceso de localización. La imagen a) es la imagen original tomada, la imagen b) es después de haber retirado las reflexiones especulares dejando la interpolación. La imagen c) es la inicialización de las zonas donde se aprecia que, aquellos puntos cuadrados son aquellos que superan los dos criterios para entrar en la región candidato de iris y las cruces sería aquellos puntos que solo cumplirían el criterio de intensidad. En la imagen d) se puede apreciar la finalización del agrupamiento de los puntos no agrupados y en la última imagen, la e), se puede apreciar el proceso completo de localización vasta.

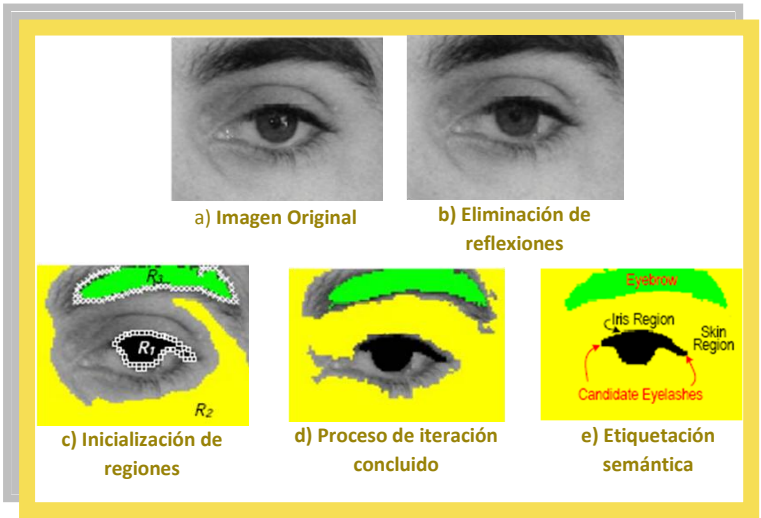


Figura 28 - Proceso de localización vasta

3.2.2 Localización de límites límbico y pupilar

Se necesita realizar una localización de mayor precisión, para ello, los limites serán modelados como dos círculos no concéntricos y encontrados mediante el operador integro diferencial[18]. Este operador será modificado para aumentar su eficiencia, la modificación consiste en un proceso iterativo para encontrar el camino de maximización más corto (es decir, más óptimo).

3.2.2.1 Anillo integro diferencial

Se construirá un anillo integro diferencial. La idea básica es empezar en un punto aleatorio p_0 de la región considerada como iris. Con radio 1 se calcula el valor del operador integro diferencial en la 8-vecindad de p_0, el de mayor cuantía es considerado, entonces, el siguiente punto en la iteración. El proceso iterativo parará cuando no encuentre ningún punto que supere el valor actual, considerando el punto como centro.

El paso inmediato al siguiente punto con mayor cuantía es una técnica voraz que puede no llevar al camino óptimo, denominada "parada al primero", pero paliar esto y aprovecharse de la eficiencia técnica que provee este sistema se ha procedido a introducir una mejora, con el fin de evitar caminos sub-óptimos.

3.2.2.2 Constelación integro diferencial

Para evitar caminos no deseados, se introduce para el mismo punto, una búsqueda 8-vecindad con distintos radios y se comparan entre si hasta encontrar el mejor resultado, donde se iniciará en la siguiente iteración.

Claramente, una constelación más grande facilita que la búsqueda integro diferencia converja en el óptimo global. No obstante, constelaciones más grandes significa mayor computación. Para este trabajo se ha considerado entonces, utilizar tres radios distintos, de valores 1, 3 y 6 para cada punto a explorar.

La introducción de la constelación durante la búsqueda del operador integro diferencial cambia el problema en su recorrido. Esto es debido a que la dirección de transición puede parecer al principio óptima pero se puede demostrar que sea un camino sub-óptimo con respecto a un anillo diferencial más grande. Esto permite no gastar excesivo esfuerzo en la búsqueda de una dirección de transición óptima sobre la propia constelación, ya que se consigue resultados moderadamente buenos.

En la Figura 29 - Operador integro diferencial en constelaciónse puede apreciar un conjunto de imágenes que explican el funcionamiento del anillo y la constelación. En la imagen a) se percibe el operador anillo de radio 1. En la imagen b) se percibe el proceso de la constelación. En la

[18] Visto y explicado en la sección 2 estado del arte.

imagen c) se percibe un recorrido de demostración en la imagen, donde en verde están los punto evaluados, lo rojos son los puntos donde se calcula el operador. Nótese la rápida convergencia hacia el centro óptimo y como únicamente una parte de los puntos es calculada con la técnica “parada al primero”.

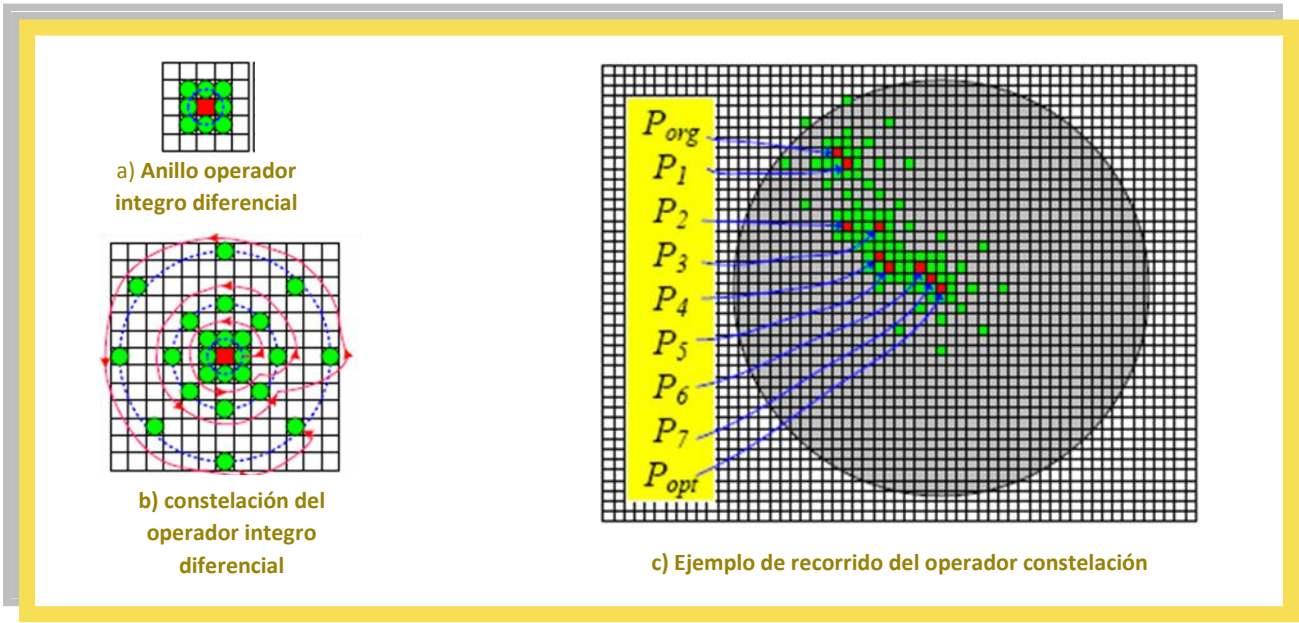


Figura 29 - Operador integro diferencial en constelación

3.2.2.3 Refinamiento de límites

En la segmentación de iris no-cooperativa, los límites límbico y pupilar tienden a no ser circulares o estar ocluidos. Por ello es necesario un refinamiento más preciso de los resultados mediante la eliminación de imprecisiones de localización debido al simple modelo circular usado. Para ello se utilizará la distribución de dos consecutivos y anulares anillos en la imagen. La intersección entre dichos histogramas será utilizado como umbral adaptativo para determinar a qué zona pertenece (como ejemplo se calcula los histogramas de las franjas referidas a $[R_{pupilar} - D, R_{pupilar}]$ y $[R_{pupilar}, R_{pupilar} + D]$). Los píxeles más brillantes de la franja interior serán considerados de la región interior y los píxeles más oscuros de la franja exterior serán considerados de la región exterior. Después se aplican operaciones morfológicas para considerar y evaluar la conexión espacial de los puntos refinados.

3.2.3 Localización de párpados

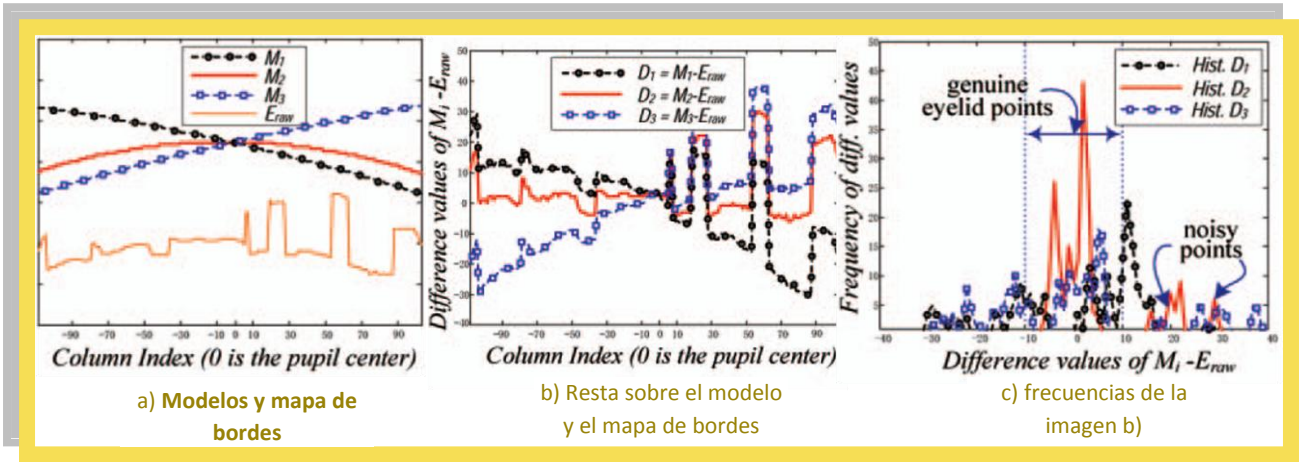
Basándose en la observación de que las pestañas son mayormente verticales, delgadas y oscuras, se adoptará un filtro de rangos de 1 Dimensión horizontal para la eliminación de las mismas. De esta manera las mayoría de las pestañas son debilitadas e incluso eliminadas.

A continuación se aplica un detector canny sobre la parte de la imagen que repercute. Con la intención de eliminar ruido se aplica un único punto de borde por columna. Como consecuencia se obtiene un mapa de bordes E_{raw} .

Se compara este mapa de bordes con unos modelos hallados experimentalmente y son descripción parabólica con distintas inclinaciones. E_{raw} y los modelos son restados para finalmente comprobar cual tiene una frecuencia óptima.

Una vez obtenido el modelo, es aplicado sobre la imagen para determinar el límite. Este proceso se aplica tanto para el párpado superior, como para el inferior.

En la Figura 30 - Modelos parabólicos insertados en el mapa de bordes se puede apreciar el proceso de sustracción entre modelo y mapa de bordes. La imagen a) es una representación de los tres posibles modelos que se aplican experimentalmente. La imagen b) es una representación de los modelos donde se ha sustraído el mapa de bordes. En la imagen c) se puede observar en la frecuencia del histograma cual es el modelo que mejor se aplica.



a) **Modelos y mapa de bordes**

b) Resta sobre el modelo y el mapa de bordes

c) frecuencias de la imagen b)

Figura 30 - Modelos parabólicos insertados en el mapa de bordes

3.2.4 Detección de pestañas y sombras

Pestañas y sombras (*ES*) son otro origen de posibles oclusiones en la segmentación de iris. Por ello se propone un método según el cual se retirará. La idea básica es extraer un umbral apropiado para la detección ES vía un modelo de predicción establecido estáticamente.

La propiedad más visible *ES* consiste en que son generalmente más oscuros que su entorno (párpados e iris), y la estrategia de detección más sencilla es encontrar un umbral que separe. No obstante es muy difícil encontrar un correcto umbral debido a la variación de intensidad y cantidad de ES entre las imágenes, especialmente en aquellas ruidosas. Los histogramas entre la región $ES_{candidata}$ y la región ES_{libre} difieren bastante, esto es debido a la oclusión producida por sombras y pestañas. Además, es sencillo asumir que a más oclusión, mayor diferencia entre

regiones. Como consecuencia, se puede predecir el nivel de oclusión de acuerdo al nivel de diferencia entre ambos histogramas.

Para validar esta asunción, se calcula la relación entre la cantidad de oclusión ES y la diferencia entre las regiones $ES_{candidata}$ y ES_{libre} sobre un conjunto de muestral. Si expresamos la diferencia entre histogramas mediante la distancia Chi, se puede encontrar una relación entre ambas regiones que defina el nivel de oclusión:

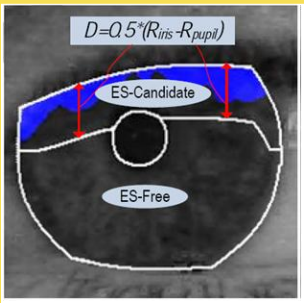
$$X^2 = \sum_{i}^{N} \frac{(h_{1i} - h_{2i})^2}{h_{1i} + h_{2i}}$$

Donde:

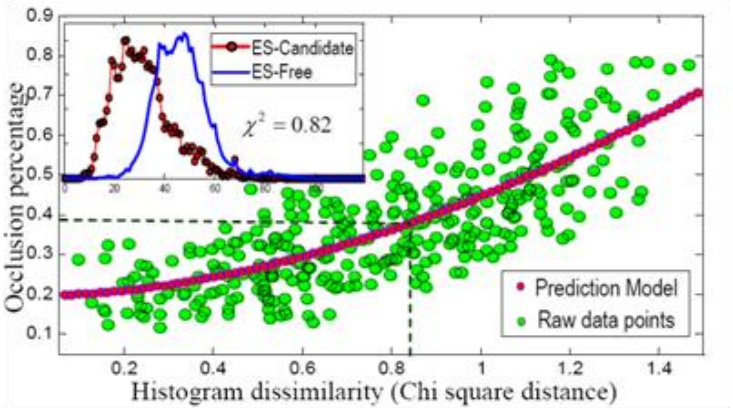
- X^2 es la distancia chi
- h_{1i} es el valor del histograma 1 en el punto iésimo
- h_{2i} es el valor del histograma 2 en el punto iésimo

Con la asunción demostrada positivamente, se puede establecer un modelo estático de predicción obtenido a partir de los datos, este modelo será una curva cúbica polinómica.

Como se puede apreciar en la Figura 31 - Localización ESb), gracias la asistencia del modelo de predicción se puede calcular el porcentaje de oclusión existente en la imagen mediante el uso de la distancia Chi. Gracias a este porcentaje se considera ES aquellos pixeles que tengan una intensidad inferior al porcentaje mostrado.



a) Estimación de la zona ES candidata



b) El modelo de predicción para la detección de las zonas ES. Este modelo de predicción provee de una estimación basta de un apropiado umbral de detección

Figura 31 - Localización ES

Finalmente un refinamiento es necesario para contemplar la 8 vecindad existente en la región ES. La mayoría de las oclusiones provienen desde el parpado superior, asumiendo esto, se

establece que entre todos los pixeles implicados exista una conexión con el párpado superior para ser considerados finalmente región ES.

3.3 Algoritmo propuesto III: Segmentación en condiciones ruidosas

Este algoritmo fue propuesto para el certamen NICE Iris por la universidad Carlos III. En él, esta vasta aproximación inicial, alcanzó la posición 14 según la relación de error, demostrando que, con una mayor elaboración, se puede alcanzar resultados más óptimos. El siguiente diagrama ilustra acerca del flujo de datos que se realiza en las distintas etapas del algoritmo.

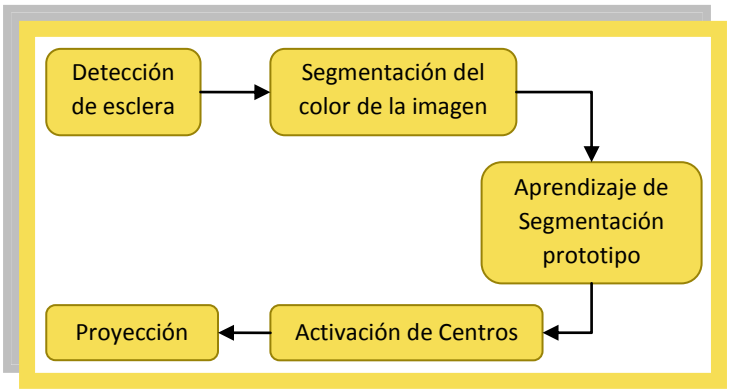


Figura 32 - Esquema de bloques del algoritmo 3 propuesto

La primera fase, es una detección de esclera mediante la extracción de características en el espacio HSV, la segunda fase realiza una segmentación del color mediante el algoritmo *k-means,* la salida obtenida de esta segmentación alimentará al aprendizaje de segmentaciones prototipo que conforma la tercera fase. Una vez obtenido las segmentaciones prototipo se procederá a realizar una activación de los centroides o semillas de la segmentación obtenidos en la fase 2. Finalmente se realizará una proyección vertical para eliminar el ruido resultante. A continuación se procede a explicar cada fase de manera detallada con el fin de aumentar la comprensión sobre el algoritmo.

3.3.1 Detección de Esclera

Esta parte del algoritmo consiste en la detección del blanco existente expuesto de un ojo. Su importancia radica en que permite determinar en primera instancia si el ojo está abierto y también realiza una primera acotación de la imagen a analizar en consecutivos procesos.

El proceso de detección de esclera consiste en:

- Transformar la imagen en formato RGB a un formato HSV.
- Realizar una proyección vertical del canal de saturación.

3.3.1.1 Transformación RGB a HSV

El objetivo de esta transformación es la obtención de la saturación existente en la imagen. La causa es la detección del color blanco. El color blanco en HSV tiene la particularidad de estar en un sub-espacio de fácil determinación, debido a que no contiene apenas saturación.

3.3.1.2 Proyección vertical de saturación

El objetivo de la proyección es determinar dos índices que acoten la situación del blanco. Para ello se realizarán tres procesos sobre el canal Saturación de la imagen transformada al espacio HSV.

El primer proceso consiste en:

- Sea I(x, y) una imagen de un solo canal, es decir, todos sus pixeles contienen información en escala de grises (de 0 a 255)$\Rightarrow$ se define $f(I)$ como:

$$f(I)(x,y) = \begin{cases} MAX & I(x,y) \leq LIMITE \\ 0 & sino \end{cases}$$

- El segundo proceso consiste en realizar una proyección vertical de la imagen $P_v(A)$[19], con el fin de acotar la imagen a analizar.

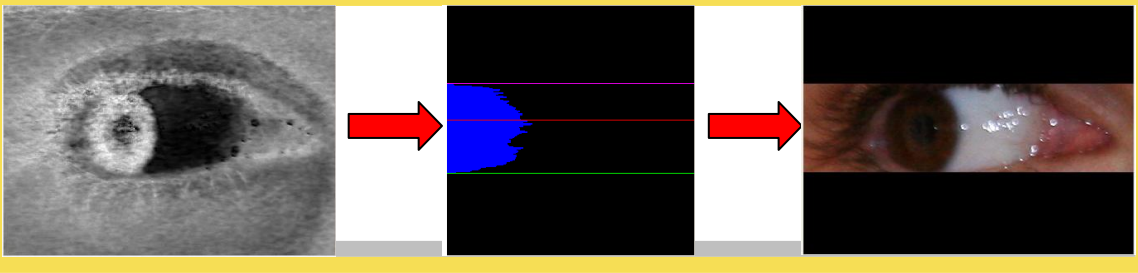


Figura 33 - Proyección vertical en el espacio HSV sobre el canal de saturación

El acotamiento se realiza mediante la determinación de máximos gradientes. Para ello, se busca el valor máximo de la proyección y se comprueba el máximo gradiente a ambos lados del valor máximo. En la imagen de la proyección se puede observar este comportamiento, la línea roja determina el máximo, la violeta y la verde determinan los máximos gradientes en sus respectivos lados.

[19] Explicado en la sección anexo, Herramientas matemáticas del Algoritmo 3 propuesto.

3.3.2 Segmentación del color de la imagen

Una vez limitada la imagen, se procede a realizar una clasificación del color. Para ello se utilizará el algoritmo *k-means* explicado en la sección anexo.

El concepto distancia en el algoritmo de *k-means* es muy importante, pues determina qué centro debe readaptarse en el entorno. Para la presente segmentación se ha utilizado el concepto de media sobre los canales R, G y B del píxel. Y con la finalidad de suavizar el resultado y aproximarlo a su entorno se ha creado una ventana que también recoja la media de los pixeles de su entorno.

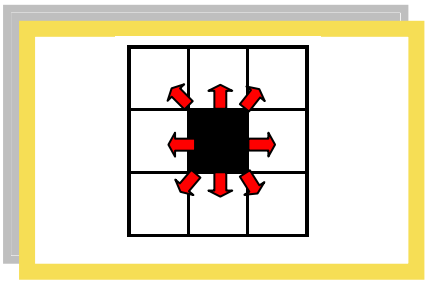


Figura 34 - Concepto de vecindad en Algoritmo 3

El color a pesar de que aporta una información relevante, no es suficiente, por ello se ha optado por conjuntarlo con la distancia respecto a posiciones. Es decir, la distancia usada para conocer que semilla es la más cercana al píxel a analizar es una combinación lineal del color y de la distancia existente entre la posición ocupada por el píxel y la semilla.

En consecuencia, cada centroide o semilla almacena información del color que representa y la posición en la que se encuentra.

Cada segmentación contiene 10 semillas. La determinación del numero ha sido estudiada mediante experimentación, había que escoger una clasificación lo suficientemente amplia como para realizar una segmentación apropiada pero manteniendo un equilibrio con el fin de evitar pérdida de información. Como se puede observar en las siguientes imágenes, la selección de diez semillas llega al compendio oportuno que determina el equilibrio entre procesamiento y segmentación.

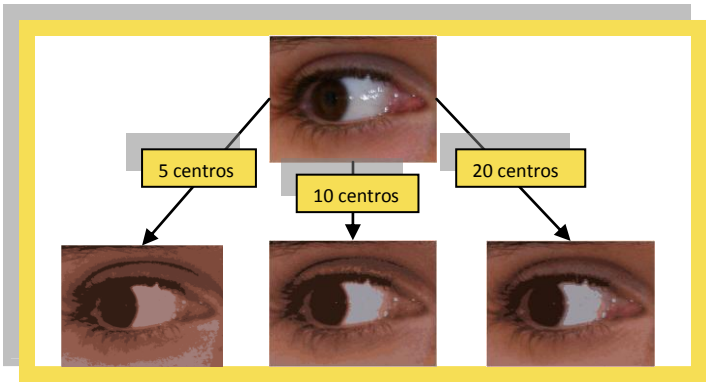


Figura 35 - Determinación de semillas en algoritmo 3

3.3.3 Aprendizaje de Segmentaciones prototipo

Una vez apreciado la segmentación de color y el acotamiento realizado de la proyección, se plantea el dilema de que centros escoger para definir aquel color con una densidad compacta que conforma el iris. Para ello se ha realizado un aprendizaje por refuerzo. Este aprendizaje consta de dos partes:

- **Proceso 1 –** La segmentación del color de cada imagen es distinta. Se realiza por separado, así no se puede generalizar para concretar el patrón común que caracteriza al ojo, una región de la imagen con un mismo color compacto. Para ello se debe conseguir una abstracción consistente. Esta abstracción estará basada en concretar un conjunto de segmentaciones que generalicen al mayor espectro posible de todas las segmentaciones existentes.

  En consecuencia, se ha creado un conjunto de segmentaciones a partir del resto existentes que irán adaptándose mediante el algoritmo *k-medias*. Es decir, sobre el espacio muestral de todas las segmentaciones posibles se ha escogido un conjunto aleatorio. Este conjunto será sometido al algoritmo *k-medias* con el fin de que se adapte y generalice el espacio muestral.

  El concepto distancia revierte interés también en este proceso. Para determinar la distancia entre el prototipo y la segmentación obtenida, se ha procedido a realizar una media del error que existe entre las semillas de la segmentación y del prototipo. La medida de error es la distancia entre la media del color entre las semillas del prototipo y la segmentación a estudiar.

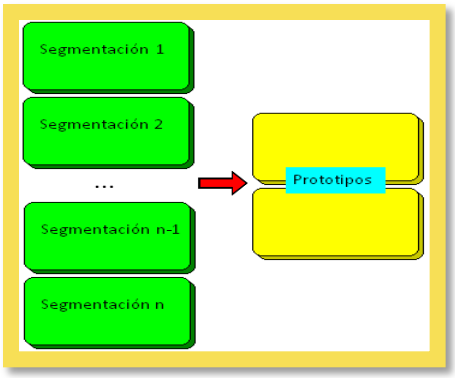


Figura 36 - Selección de prototipos en Algoritmo 3

- **Proceso 2 –** Una vez determinado un conjunto prototipo de segmentaciones se debe realizar un estudio para cuantificar cuales son aquellas semillas, dentro de las segmentaciones, que contiene un mayor número de pixeles determinados como iris.

  Una vez obtenido un conjunto de prototipo que se adapten al espacio muestral de las segmentaciones se debe proceder a establecer que semillas contienen un número mayor de píxeles clasificados como iris.

  Para realizarlo se procederá a utilizar el Ground Truth que se provee para NICE. Determinando que semillas son aquellas que contienen un porcentaje mayor de acierto.

  El algoritmo para cada imagen en estudio, es como prosigue:

  - Se obtiene la detección de esclera y la segmentación de la imagen.
  - Se comprueba que prototipo se asemeja más a la segmentación obtenida. Es decir, se examina cual es el prototipo con menor distancia respecto a la segmentación.
  - Una vez obtenido el prototipo que más cercano esta de la segmentación realizada se procede a comprobar con el Ground Truth que semillas tienen un mayor índice de clasificación del iris.
  - Se premia aquellas semillas con un mayor índice. Para comprobar cuales son aquellas semillas premiadas, se compara con todas las semillas. Aquellas ganadoras son reforzadas mientras que las perdedoras son devaluadas.

  Este proceso de refuerzo es realizado durante varios ciclos con el fin de profundizar en la clasificación correcta de las imágenes.

3.3.4 Activación de Centros

Una vez realizado el aprendizaje de segmentaciones y las semillas que deben activarse se procede a instaurar la aplicación para un estudio de fiabilidad. Se realizan prueban para comprobar que porcentaje de acierto plantea el estudio realizado.

Para ello se realiza el siguiente algoritmo sobre las imágenes:

1. Detección de esclera
2. Segmentación del color
3. Se realiza la función I(x, y) $\Rightarrow$ F(x, y)

$$F(x,y) = \begin{cases} 1 & \exists s_{i-ganadora} \ \ \| I(x,y), s_{i-ganadora} \ \| > MIN \\ 0 & sino \end{cases}$$

Donde S_i es la semilla iésima ganadora dentro del prototipo escogido y MIN la mínima distancia entre el píxel a estudiar y las semillas del prototipo.

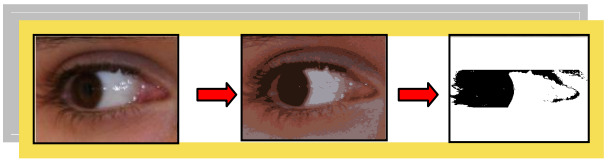


Figura 37 - Proceso de detección, segmentación y activación

3.3.5 Proyección

Esta etapa final tiene como objetivo eliminar un gran porcentaje de ruido. Para ello se utiliza una proyección horizontal del resultado obtenido en el anterior apartado. Quedándose así el resultado final.

Este proceso consiste en realizar una proyección vertical de la imagen $P_h(A)$, con el fin de acotar la imagen a analizar.

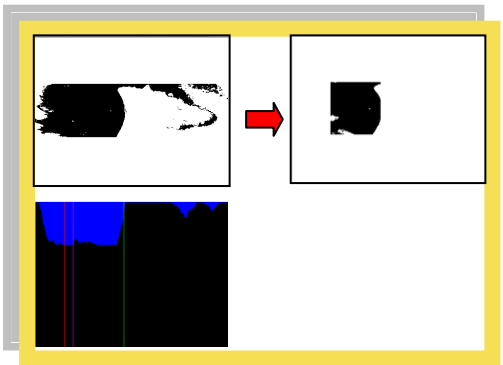


Figura 38 - Proyección Horizontal en el algoritmo 3

4. Metodología de Desarrollo

Este capítulo esta introducido para encuadrar el proyecto como un desarrollo software pues se ha debido de implementar los algoritmos para su consecuente comprobación. Abordará el proyecto desde una óptica de ingeniería del software para formalizar la planificación, el posible presupuesto que tendrá y finalmente un diseño software para indicar las directrices en el desarrollo propiamente dicho.

4.1 Planificación

La importancia de una buena planificación estriba en la necesidad de ajustar el tiempo para aumentar al máximo la productividad y tener consciencia de la magnitud del proyecto. Para ello expondrá la planificación inicial del proyecto, utilizando un archivo importado de Microsoft Office Proyect para visualizar como se compone esta planificación.

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Inicialización del Proyecto	0 días	lun 22/06/09	lun 22/06/09	
2	Recopilacion de Información	5 días	mar 23/06/09	lun 29/06/09	1
3	Reunion con tutor 1	0 días	mar 30/06/09	mar 30/06/09	2
4	Redaccion de Apartado 1	2 días	mié 01/07/09	jue 02/07/09	3
5	Redaccion de Apartado 2	4 días	vie 03/07/09	mié 08/07/09	4
6	Redaccion de Apartado 3	5 días	jue 09/07/09	mié 15/07/09	5
7	Diseño Software	10 días	mié 15/07/09	mar 28/07/09	6
8	Implementacion de Algoritmo 1	12 días	jue 30/07/09	vie 14/08/09	7
9	Implementacion de Algoritmo 2	17 días	vie 31/07/09	lun 24/08/09	7
10	Implementacion de Algoritmo 3	20 días	sáb 01/08/09	vie 28/08/09	7
11	Desarrollo Concluido	0 días	vie 28/08/09	vie 28/08/09	8;9;10
12	Redaccion de Apartado 4	4 días	vie 28/08/09	mié 02/09/09	11
13	Redaccion de Apartado 5	3 días	jue 03/09/09	lun 07/09/09	12
14	Redaccion de Apartado 6	2 días	mar 08/09/09	mié 09/09/09	13
15	Redaccion de Apartado 7	3 días	jue 10/09/09	lun 14/09/09	14
16	Reunion con tutor 2	0 días	mar 15/09/09	mar 15/09/09	15
17	Correción del Proyecto	2 días	mié 16/09/09	jue 17/09/09	16
18	Finalización del Proyecto	0 días	jue 17/09/09	jue 17/09/09	17

Como se puede observar todas las tareas han sido enumeradas y catalogadas para obtener de una manera precisa la duración, importancia, relación y dependencia entre las tareas a realizar.

En el siguiente diagrama se puede observar la planificación total realizada y como se ha procedido a realizar de una manera serial y consecutiva todas las tareas.

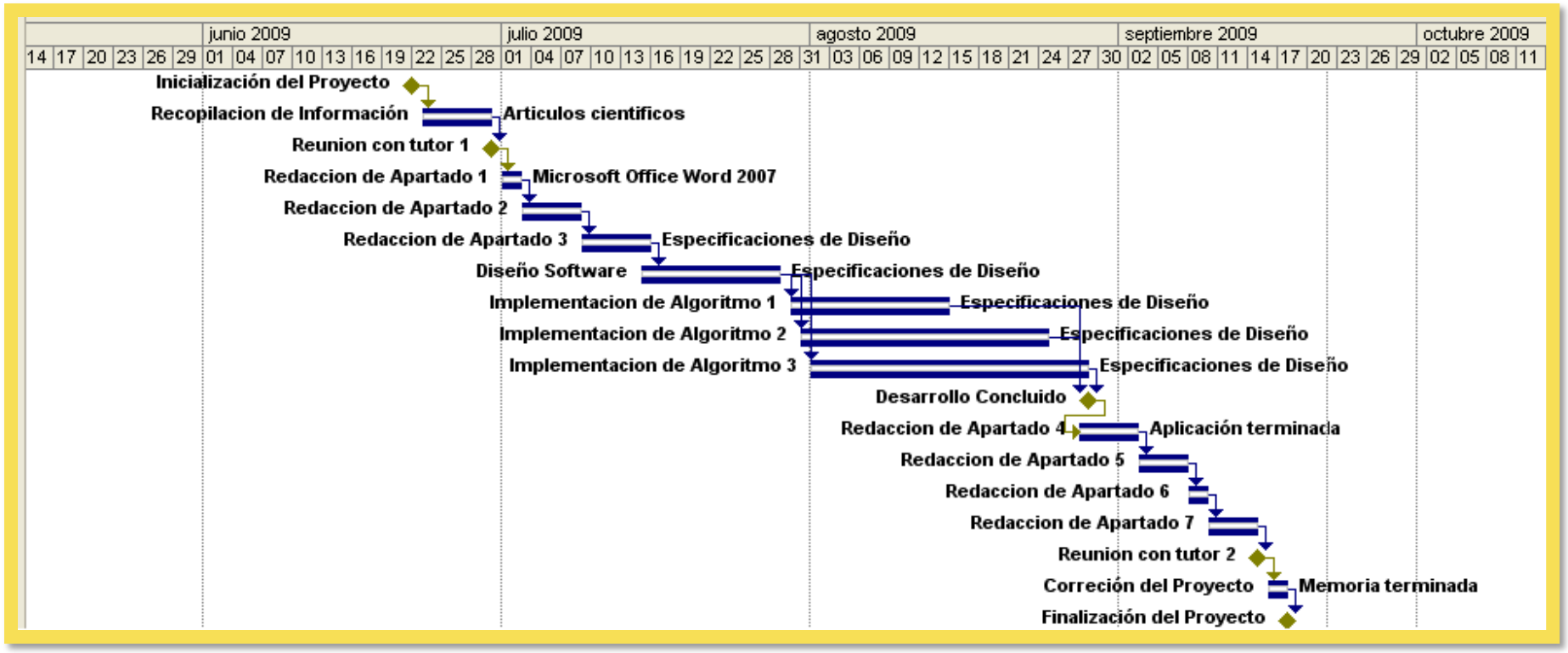


Figura 39 - Esquema de Grant para planificación

4.2 Presupuesto

Teniendo en cuenta la planificación realizada, el presupuesto del proyecto de investigación para el análisis comparativo de algoritmos en segmentación de iris queda de la siguiente manera:

Presupuesto basado en recursos humanos			
Tareas	Días	Horas	Coste
Recopilación de Información Científica	5 días	40 horas	1200 €
Redacción de Apartado 1	2 días	16 horas	480 €
Redacción de Apartado 2	4 días	32 horas	960 €
Redacción de Apartado 3	5 días	40 horas	1200 €
Diseño Software	10 días	80 horas	240 €
Implementación Algoritmo 1	12 días	96 horas	2880 €
Implementación Algoritmo 2	17 días	0 horas	0 €
Implementación Algoritmo 3	20 días	24 horas	720 €
Redacción de Apartado 4	4 días	32 horas	960 €
Redacción de Apartado 5	3 días	24 horas	720 €
Redacción de Apartado 6	2 días	16 horas	480 €
Redacción de Apartado 7	3 días	24 horas	720 €
Correción del Proyecto	2 días	16 horas	480 €
TOTAL	89 dias	440 horas	13200 €

Tabla 4 - Presupuesto de recursos humanos

Por cada día se ha trabajado 8 horas, excepto en la implementación completa que se procedió a realizar una realización en paralelo debido a que ciertos elementos eran comunes. También está incluido en coste temporal, el tiempo de aprendizaje de nuevas librerías y lenguaje de implementación que es coincidente con los puntos comunes con respecto a la implementación de los distintos algoritmos.

Se ha considerado que para introducir un coste monetario se establece el sueldo por hora del ingeniero que realiza las labores de investigación a 30 € la hora. Como consecuencia el coste total del proyecto contando únicamente recursos humanos asciende a 13200 €.

A este coste se le debe sumar todo aquel que forme parte de recursos físicos y no humanos. Descritos en la siguiente tabla.

Presupuesto basado en recursos físicos	
Material	Coste
Microsoft Office	165 €
Microsoft Visual Studio	540 €
Inter Core Quad E8800 (PC)	2000 €
TOTAL	2705 €

Tabla 5 - Presupuesto de recursos físicos

Los precios que figuran en la tabla anterior son aquellos estimados a partir de la compra de licencias y hardware necesario para el desarrollo y correcta implementación de los algoritmos. El coste de recursos materiales queda entonces como 2705 €.

El coste final del proyecto queda englobado tanto por los costes derivados de los recursos humanos, un único ingeniero y los recursos materiales derivados del desarrollo.

Presupuesto Global para el Proyecto	
	Coste
Presupuesto de recursos materiales	13200 €
Presupuesto de recursos humanos	2705 €
TOTAL	15905 €

Tabla 6 - Presupuesto Global

4.3 Diseño del Desarrollo Software

Para poder realizar estos algoritmos hay que pensar en aquellos aspectos físicos y prácticos que conlleva su correcto desarrollo. Como consecuencia, se debe planificar y proyectar el diseño de una aplicación que recoja la implementación inspirada en los algoritmos mencionados en este apartado.

Para ello se procede a realizar un único componente de alto nivel en la producción, fuertemente basado en la librería de libre distribución y código abierto denominada OpenCV (Open Source Computer Vision). El núcleo algebraico es altamente eficiente y fue liberado su código por la empresa Intel. En el apartado Anexo IX - OpenCV (Open Source Computer Vision) se recoge un pequeño resumen de su historia y sus principales aplicaciones.

El Algoritmo 1 será el algoritmo inspirado en el algoritmo 3, basado en K-medias, en apartados anteriores. El algoritmo 2 será el algoritmo inspirado en el algoritmo 2, basado C-medias difusas, en apartados anteriores. El algoritmo 3 será el algoritmo inspirado en el algoritmo 1, basado en el operador integro diferencial, en apartados anteriores.

A continuación se procede a mostrar un conjunto de diagramas con su consiguiente explicación del componente y las clases que lo conforman para obtener un enfoque global del diseño software inspirado en los algoritmos antes descritos.

4.3.1 Componentes de Alto Nivel

En primer lugar, para el diseño software, se debe cimentar el proyecto sobre una arquitectura que se amolde a las necesidades de poder implementar tres algoritmos distintos en una única aplicación. En este caso, la descripción por componentes del diseño en alto nivel, marca claramente como se aborda el problema de cómo conseguir la ejecución de cada algoritmo sin necesidad de cambiar su implementación.

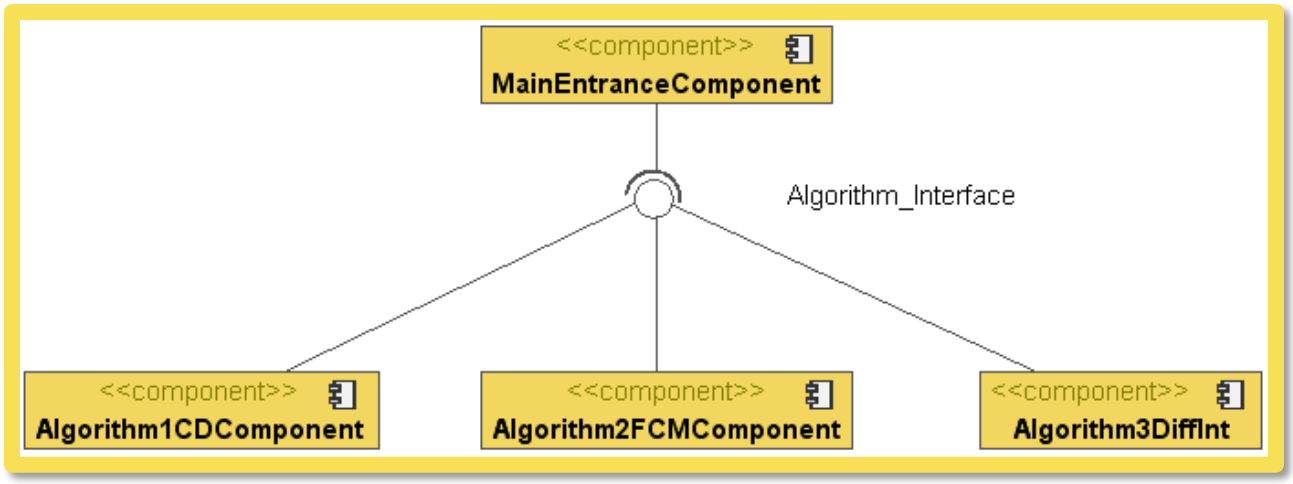


Figura 40 - Diagrama de Alto Nivel - Componentes del Diseño

Como se puede observar en la Figura 40 se ha procedido a separar cada algoritmo inspirado, en componentes distintos y como nexo común el componente MainEntrance que es el que procederá a conmutar el algoritmo deseado en función de las necesidades introducidas por parámetro a la aplicación central.

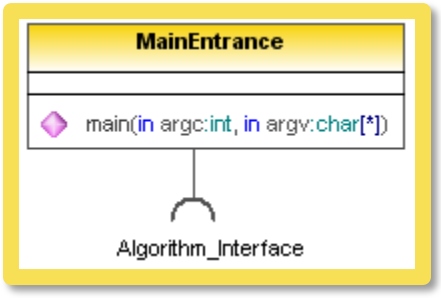


Figura 41 - Diagrama de Nivel 2: Componente MainEntrance - Clase MainEntrance

Las clases que componen el componente de entrada principal se pueden apreciar en el diagrama de la Figura 41, también se realiza una descripción de las mismas dejando ampliamente explicado el diseño de este componente.

MainEntrance			
Interfaz Requerido	Algorithm_Interface	**Interfaz Implementado**	N/A
Responsabilidades:		Ejecución de la aplicación y gestión de los algoritmos en tiempo real para conmutarlos. Los argumentos de la aplicación deben ser • Argumento 1 – Nombre del fichero que contiene la imagen en formato tiff. • Argumento 2 - Nombre del fichero de salida en formato bmp para guardar la imagen binarizada de salida • Argumento 3 – Opcional – Indicación del algoritmo que se desea ejecutar.	
Atributos:		N/A	
Operaciones:		**main:** función principal de la aplicación que conlleva la ejecución en el sistema de la aplicación para la realización de los algoritmos **Argumentos:** • Argc – entero que indica el numero de parámetros • Argv – contenedor de los argumentos en formato char []. **Retorno:** • Int – Entero que indica como ha procedido la ejecución. Si es negativo la ejecución ha sido mal realizada, en otro caso, la ejecución ha sido bien realizada.	

Tabla 7 - Clase MainEntrance

4.3.2 Componente 1: Algoritmo Cool Detector

Este componente esta dividido en dos diagramas, el primero consta de una descripción de las clases que conforman el elemento detector basado en K-medias y sus asociaciones. Mientras que el segundo aborda el resto del algoritmo.

4.3.2.1 Detector K-medias

En este apartado se recogerá el diagrama de clases del sub-componente Detector K-medias y una explicación detallada de sus clases.

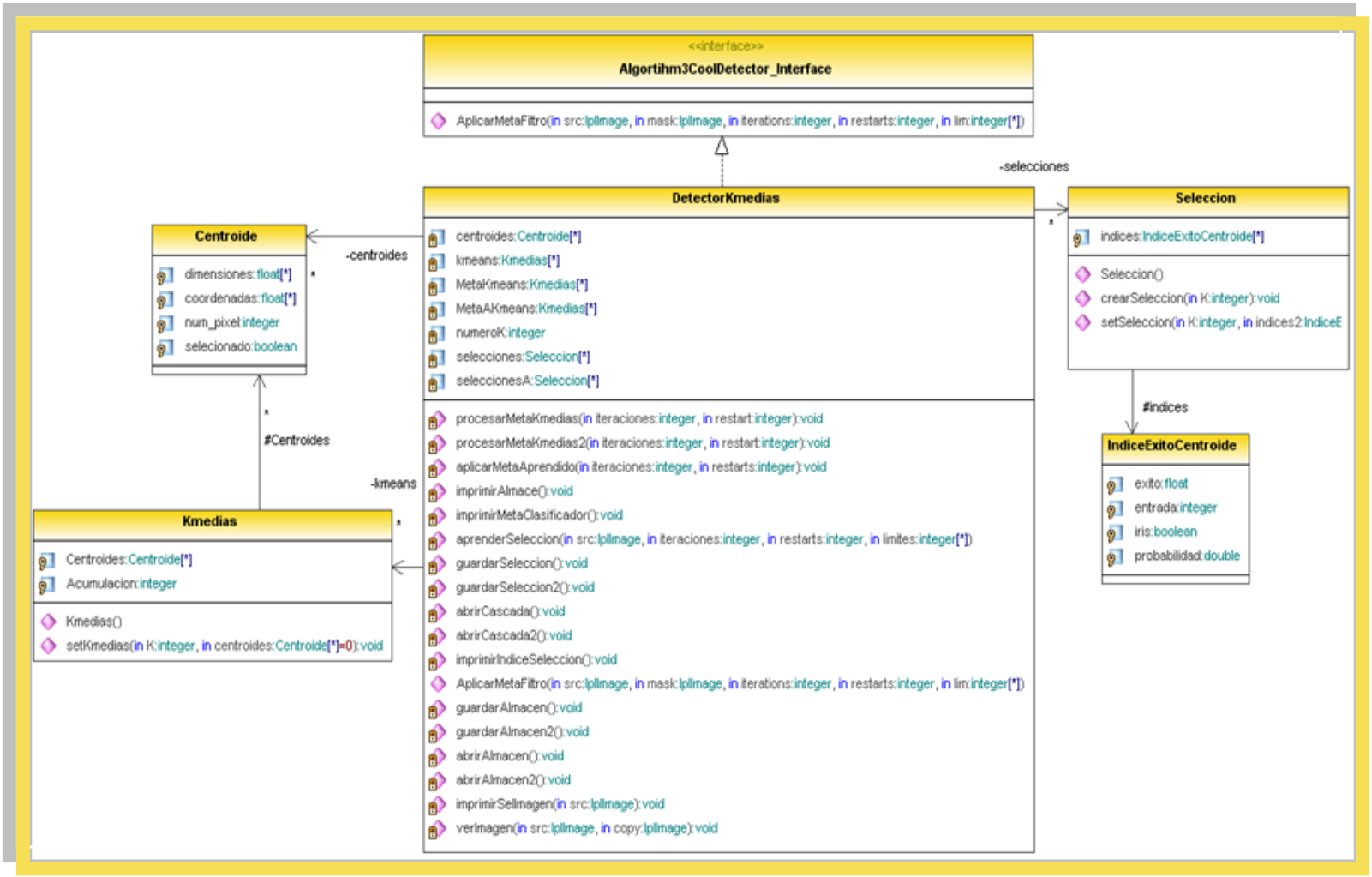


Figura 42 - Diagrama de Nivel 2: Componente Algoritmo 1 - Detector Kmedias

A continuación una explicación exhaustiva de todas las clases, interfaces y estructuras del diagrama antes expuesto en Figura 42.

AlgorithmCoolDetector - Interfaz			
Interfaz Requerido	N/A	**Interfaz Implementado**	N/A
Responsabilidades:		Ofrecer un nexo común entre el algoritmo principal y el detector K-medias	
Atributos:		N/A	
Operaciones:		**aplicarMetaFiltro:** comunicación entre el algoritmo general y el detector k-medias para la ejecución del algoritmo K-medias sobre la imagen bajo unos determinados límites, previamente calculados. Argumentos: • Src – IplImage contenedor de la imagen de entrada • Mask – IplImage contenedor de la imagen de salida • Iterations – entero que indica el número de iteraciones que se quieren realizar como máximo en el algoritmo k-medias. • Restarts – entero que indica el número de reinicios que se requiere realizar como máximo en el algoritmo k-medias. • Lim – limites sobre la imagen donde se requiere realizar el algoritmo k-medias en formato int[]. Retorno: • Vacío	

Tabla 8 - Interfaz AlgorithmCoolDetector

Centroide - Estructura			
Interfaz Requerido	N/A	**Interfaz Implementado**	N/A
Responsabilidades:		Ofrecer un almacenamiento para los datos de cada semilla, centro o centroide del algoritmo Kmedias.	
Atributos:		**Dimensiones:** contenedor de la información del color, está en formato int[], cada posición del vector indica el valor en un canal. **Coordenadas:** Indicación de la posición del centroide dentro de la imagen. Esta en formato int[2], cada posición guarda una dimensión espacial. **Num_pixel:** entero que indica el número de pixeles asociados a esta semilla. **Seleccionado:** Indicación si el centroide es en mayoría iris o no. En formato boolean.	
Operaciones:		N/A	

Tabla 9 - Estructura Centroide

Kmedias - Clase			
Interfaz Requerido	N/A	**Interfaz Implementado**	N/A
Responsabilidades:		Ofrece la estructura básica para la ejecución del algoritmo Kmedias, preparando distintos aspectos del algoritmos en su creación	
Atributos:		**Dimensiones:** Centroides que conforman la estructura de datos del algoritmo kmedias. **Acumulación:** Indicación del grado de acumulación de los centroides que reflejan el numero de pixeles asociados a cada centroide.	
Operaciones:		**Kmedias:** Constructor de la clase, inicializa las clases para su correcta ejecución. **setKmedias:** método para el cambio de cualquier atributo dentro de la clase. Argumentos**:** • K: indicación del número de centroides que contiene el algoritmo • Centroides: nueva estructura de almacenamiento para el algoritmo, puede ser configurado como nulo indicando que no debe ser cambiado Retorno**:** • Vacío	

Tabla 10 - Clase K-Medias

IndiceExitoCentroide - Estructura			
Interfaz Requerido	N/A	**Interfaz Implementado**	N/A
Responsabilidades:		Estructura de datos asociado a un centroide que indica mediante sus atributos si el centroide es o no conformante de iris	
Atributos:		**Éxito**: número en coma flotante que indica el éxito de que sea iris o no. **Entrada**: Indica el centroide asociado como indicador único dentro de la asociación Iris: indica mediante boolean si es o no iris. **Probabilidad**: indicación con una probabilidad de 0 a 1 si el centroide asociado a este índice es o no conformante a iris.	
Operaciones:		N/A	

Tabla 11 - Estructura IndiceExitoCentroide

Selección – Clase			
Interfaz Requerido	N/A	**Interfaz Implementado**	N/A
Responsabilidades:		Ofrece la estructura básica que conforma la indicación de todos índices de éxito de iris asociados a los centroides	
Atributos:		**Índices:** contenedor de datos de cada centroide en formato IndiceExitoCentroide	
Operaciones:		**Selección:** Constructor de la clase para la inicialización del objeto. **CrearSelección:** Crea una nueva selección en el caso de que no se haya realizado. Encapsula parte de la inicialización de la estructura básica. Argumentos: • K: entero que indica el numero de índices existentes Retorno: • Vacío **setSeleccion:** Cambio de la selección de índicesExitoCentroide Argumentos: • K: entero que indica el nuevo número de índices • Indices2: nueva estructura de índices a cambiar. Retorno: • Vacío	

Tabla 12 - Clase Selección

DetectorKmedias - Clase			
Interfaz Requerido	N/A	**Interfaz Implementado**	N/A
Responsabilidades:		Realizar la ejecución del algoritmo k-medias sobre la imagen bajo determinados límites introducidos como parámetro. Determinar que centroides contienen en su mayoría iris e indicarlo mediante la salida	
Atributos:		**Centroides**: Almacenamiento de datos para la meta-clasificación a realizar sobre la imagen. **Kmeans**: Estructura de datos que guarda la segmentación primaria de la imagen. **MetaKmeans**: Estructura de datos para la segmentación secundaria de la imagen. **MetaAKmeans**: Estructura de datos para la segmentación terciaria de la imagen. **numeroK:** Indicación del numero de centroides contenidos en el atributo centroide. **Selecciones**: Índices que indican que centroides son conformantes de iris en la segmentación secundaria **SeleccionesA**: Índices que indican que centroides son conformantes de iris en la segmentación terciaria	
Operaciones:		**aplicarMetaFiltro:** comunicación entre el algoritmo general y el detector k-medias para la ejecución del algoritmo K-medias sobre la imagen bajo unos determinados límites, previamente calculados.	

	Argumentos: • Src – IplImage contenedor de la imagen de entrada • Mask – IplImage contenedor de la imagen de salida • Iterations – entero que indica el número de iteraciones que se quieren realizar como máximo en el algoritmo k-medias. • Restarts – entero que indica el número de reinicios que se requiere realizar como máximo en el algoritmo k-medias. • Lim – limites sobre la imagen donde se requiere realizar el algoritmo k-medias en formato array de enteros. Retorno: • Vacío **procesarMetaKmedias:** Realiza el aprendizaje para la realización del algoritmo y guardado de los datos necesarios del meta-clasificador. Argumentos: • Iteraciones: entero que indica el número de iteraciones máximo para la ejecución del meta-clasificador. • Restarts: entero que indica el número de reinicios del algoritmo para la ejecución del meta-clasificador. Retorno: • void **procesarMetaKmedias2:** Selección de los centroides aprendidos más probables de ser iris, y guardado de dichas probabilidades para su posterior uso. Argumentos: • Iteraciones: entero que indica el número de iteraciones máximo para la ejecución del meta-clasificador. • Restarts: entero que indica el número de reinicios del algoritmo para la ejecución del meta-clasificador. Retorno: • Vacío **aplicarMetaAprendido:** Comparación con los meta-modelos y asignación de selecciones para saber que centroides deben activarse para la localización del iris. Argumentos: • Iteraciones: entero que indica el numero de iteraciones máximo para la ejecución del meta-clasificador. • Restarts: entero que indica el número de reinicios del algoritmo para la ejecución del meta-clasificador. Retorno: • Vacío **imprimirAlmacen:** Fase de depuración. Impresión de los datos almacenados en el almacén de modelos. Argumentos: • N/A Retorno: • Vacío **imprimirMetaClasificador:** Fase de depuración. Impresión de los

	datos del meta-clasificador para subsanar posibles errores. Argumentos: • N/A Retorno: • Vacío **aprenderSeleccion:** Un vez obtenidos los meta-modelos se procede a explorar cuales centroides de estos meta-modelos son más probables que sean iris. Argumentos: • src: IpImage. Contenedor de la imagen de entrada • iteraciones: numerode iteraciones que se utilizarán para el aprendizaje de la selección mediante el algoritmo k-medias. • Restarts: número de reinicios necesarios para el aprendizaje. • Limites: indicación de los limites que deben ser explorados dentro de la imagen. Formato array de enteros. Retorno: • Vacío **guardarSeleccion:** Guarda la selección aprendida previamente de los meta-modelos. Argumentos: • N/A Retorno: • Vacío **guardarSeleccion2:** Guarda la selección aprendida previamente de las probabilidades de iris Argumentos: • N/A Retorno: • Vacío **abrirCascada:** Apertura de los fichero de entrada necesarios que contienen la información acerca de los meta-modelos. Argumentos: • N/A Retorno: • Vacío **abrirCascada2:** Apertura de los ficheros de entrada necesarios que contienen la información acerca de la selección de centroides en cada meta-modelo. Argumentos: • N/A Retorno: • Vacío **imprimirIndiceSeleccion:** Fase de Depuración. Visualización del almacén de selecciones de meta-modelos. Argumentos:

	• N/A Retorno: • Vacío **guardarAlmacen**: Almacena los meta-modelos que estén en ese momento guardados en memoria dinámica en el disco duro. Argumentos: • N/A Retorno: • Vacío guardarAlmacen2: Almacena las selecciones que estén en ese momento guardados en memoria dinámica en el disco duro. Argumentos: • N/A Retorno: • Vacío **abrirAlmacen:** Abre el almacén que contiene los meta-modelos. Argumentos: • N/A Retorno: • Vacío **abrirAlmacen2:** Abre el almacén que contiene las selecciones de los meta-modelos. Argumentos: • N/A Retorno: • Vacío **imprimirSelImagen**: Fase de Depuración. Método que realiza una visualización de la activación de los centroides previamente seleccionados sobre la imagen original. Argumentos: • src: IplImage. Contenedor de la imagen de entrada para observar la activación Retorno: • Vacío **verImagen**: Fase de Depuración. Función según la cual se puede observar los cambios efectuados en la imagen desde su inicio hasta el procesado completo. Argumentos: • N/A Retorno: • Vacío

Tabla 13 - Clase Detector K-medias

4.3.2.2 Algoritmo Cool Detector

En este apartado se recogerá el diagrama de clases del sub-componente principal y una explicación detallada de sus clases. Esta parte del diseño es el enlace entre la aplicación central y el algoritmo CoolDetector.

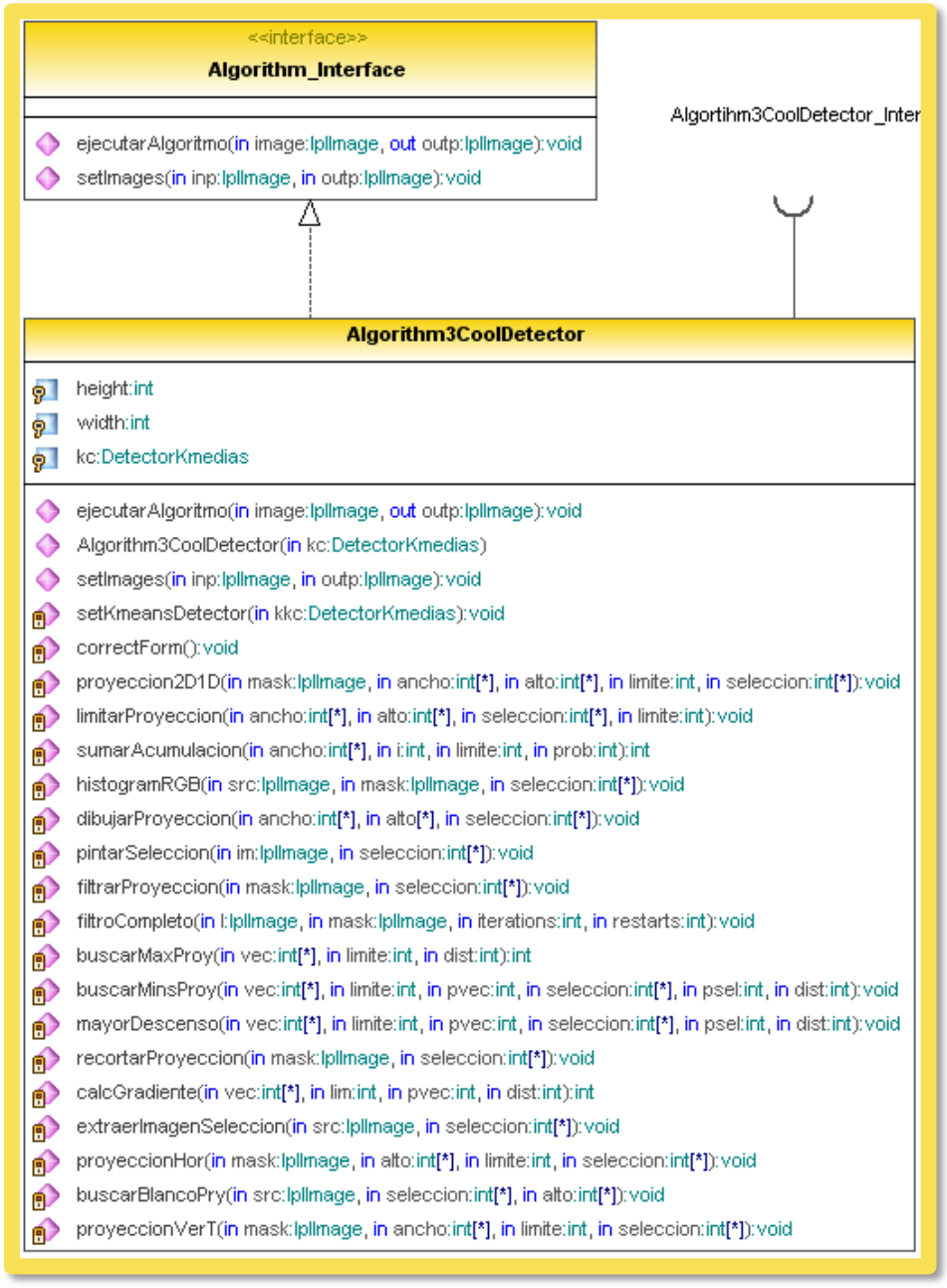


Figura 43 - Diagrama de Nivel 2: Componente Algoritmo 1 - Clase Algorithm1

A continuación una explicación exhaustiva de todas las clases, interfaces y estructuras del diagrama antes expuesto en Figura 43.

Algorithm- Interfaz			
Interfaz Requerido	N/A	**Interfaz Implementado**	N/A
Responsabilidades:		Ofrecer un nexo común entre la entrada principal y los algoritmo	
Atributos:		N/A	
Operaciones:		**ejecutarAlgoritmo:** Ejecución en tiempo real del algoritmo selecciónado. En función de cuál sea, variará las formas de hacerlo. Argumentos: • Image – IplImage contenedor de la imagen de entrada • Outp – IplImage contenedor de la imagen de salida Retorno: • Vacío **setImages:** Cambio de las imágenes tanto de salida como de entrada. Argumentos: • Image – IplImage contenedor de la imagen de entrada • Outp – IplImage contenedor de la imagen de salida Retorno: • Vacío	

Tabla 14 – Interfaz Algorithm

Algorithm- Interfaz			
Interfaz Requerido	N/A	**Interfaz Implementado**	Algorithm- Interfaz
Responsabilidades:		Realización del Algoritmo inspirado en el algoritmo 3 descrito en el apartado 3.3	
Atributos:		**Height**: entero que indica la altura de la imagen **Width**: entero que indica el ancho de la imagen **Kc**: Instancia de la clase DetectorKmedias para la ejecución posterior del mismo.	
Operaciones:		**ejecutarAlgoritmo:** Ejecución en tiempo real del algoritmo selecciónado en este caso el algoritmo basado en condiciones ruidosas. Argumentos: • Image – IplImage contenedor de la imagen de entrada • Outp – IplImage contenedor de la imagen de salida Retorno: • Vacío **setImages:** Cambio de las imágenes tanto de salida como de entrada. Argumentos: • Image – IplImage contenedor de la imagen de entrada • Outp – IplImage contenedor de la imagen de salida Retorno: • Vacío **AlgorithmCoolDetector**: Constructor de la clase para instantación de objetos que realicen este algoritmo. Argumentos:	

- Kkc: DetectorKmedias. Nueva asignación para el atributo kc.

Retorno:

- Vacío

setKmeansDetector: Nueva asignación para el atributo kc.

Argumentos:

- Kkc: DetectorKmedias. Nuevo valor de kc.

Retorno:

- Vacío

correctForm: Función que asegura la introducción de los parámetros en orden correcto y avisa al usuario de la nomenclatura necesaria.

Argumentos:

- N/A

Retorno:

- Vacío

proyeccion2D1D: Realiza una o las dos proyecciones sobre una determinada imagen ya binarizada y con una profundidad de 8 bits

Argumentos:

- Mask: IplImage. Imagen en profundidad 8 bits que contiene la información binarizada a proyectar
- Ancho. Array de enteros. Si no está vacío, realiza la proyección horizontal de la imagen y la situa en ancho.
- Alto: Array de enteros. Si no está vacío, realiza la proyección vertical de la imagen y la situa en alto.
- Selección: array de enteros. Indica donde están situados los máximos gradientes en ambas proyecciones.
- Lim: entero que indica el límite de la proyección.

Retorno:

- Vacío

limitarProyeccion: Limite la proyección realizada a partir de unos límites sobre una proyección ya realizada sobre una imagen.

Argumentos:

- Ancho: Array de enteros. Contenido de la proyección horizontal.
- Alto: Array de enteros. Contenido de la proyección vertical.
- Lim: entero que indica el límite de la proyección.
- Selección: array de enteros. Indica donde están situados los máximos gradiente en ambas proyecciones.

Retorno:

- Vacío

sumarAcumulacion: Realiza la acumulación bajo un determinado límite para el cálculo del gradiente

Argumentos:

- Ancho. Array de enteros que contiene la información de la proyección a calcular la acumulación
- I: entero que indica donde esta situado el punto según el cual se procede a sumar
- Limite: limite máximo dentro de ancho donde puede realizar

	la suma. • Prob: Indicación de que tipo de proyección es, si es una proyección horizontal o vertical. Retorno: • Vacío **histogramRGB:** *Deprecated.* Método obsoleto para la contrucción de un histograma de la imagen para la extracción de características finalmente no utilizadas. Argumentos: • Src: IplImage. Contenedor de la imagen de la cual se desea el histograma. • Mask: IplImage. Contenedor de la salida que contiene la extracción de las características obsoletas. • Selección: array de enteros que contiene la información sobre proyección y donde comenzar el histograma. Retorno: • Vacío **dibujarProyeccion:** Fase de Depuración. Visualización de las proyecciones para el análisis del algoritmo. Argumentos: • Ancho: array de enteros que contiene la información de la proyección horizontal. • Alto: array de enteros que contiene la información de la proyección vertical. • Selección:array de enteros que contiene los límites de las proyecciones. El máximo gradiente en ambas direcciones. Retorno: • Vacío **pintarSeleccion:** Recorta la imagen en función de la realización de las proyección para acotar la imagen. Argumentos: • Im: IplImage. Contenedor de la imagen de entrada que será recortada • Selección: array de enteros que contiene la información de los limites para recortar la imagen Retorno: • Vacío **filtrarProyeccion:** Realiza el filtro de proyección horizontal con su respectivo recorte de la imagen. Argumentos: • Mask: IplImage. Contenedor de la imagen a recortar • Selección: array de enteros que contiene la información de los límites para recortar la imagen. Retorno: • Vacío **filtroCompleto**: Realización de todas las fases del algoritmo contemplado en el apartado 3.3.

	Argumentos: • I:IplImage. Contenedor de la imagen de entrada sin tratar. • Mask:IplImage. Contenedor de la imagen de salida, debe estar inicializada y debe tener una profundidad de 8 bits. • Iterations: entero que indica el número de iteraciones que tiene el algoritmo k-medias. • Restarts: entero que indica el número de iteraciones que ejecuta el algoritmo k-medias. Retorno: • Vacío **buscarMaxProy:** Busca el máximo gradiente en la proyección introducida por parámetro. Argumentos: • Vec: Array de enteros que contiene la proyección a estudiar. • Limite: entero que indica el máximo margen de suavizado para el cálculo del máximo. • Dist: entero que indica el tipo de proyección. Retorno: • Vacío **buscarMinsProy:** Busca el mínimo gradiente en la proyección introducida por parámetro. Argumentos: • Vec: Array de enteros que contiene la proyección a estudiar. • Limite: entero que indica el máximo margen de suavizado para el cálculo del máximo. • Pvec: entero que indica el tamaño del vector • Selección: array de enteros donde se introducirán los valores de las mínimas proyecciones. • Psel: entero que indica el tamaño de selección • Dist: entero que es la indicación del tipo de proyección Retorno: • Vacío **mayorDescenso:** Buscar el mínimo gradiente en toda la proyección y no se concentra en mínimos locales. Argumentos: • Vec: Array de enteros que contiene la proyección a estudiar. • Limite: entero que indica el máximo margen de suavizado para el cálculo del máximo. • Pvec: entero que indica el tamaño del vector • Selección: array de enteros donde se introducirán los valores de las mínimas proyecciones. • Psel: entero que indica el tamaño de selección • Dist: entero que es la indicación del tipo de proyección Retorno: • Vacío **recortarProyeccion:** Recorta la proyección sobre una determinada

	imagen introducida por parámetro. Argumentos: • Mask: IplImage. Contenedor con profundidad de 8 bits que recortará la imagen. • Selección: Array de enteros que contiene los valores donde se debe recortar la imagen tanto vertical como horizontal. Retorno: • Vacío **calcGradiente:** Calcula ambos gradiente, el máximo y el mínimo posibles dentro de una determinada imagen. Argumentos: • Vec: Array de enteros que contiene la proyección donde se calculará su gradientes. • Lim: máximo suavizado para el cálculo del gradiente. • Pvec: entero que indica el tamaño del vector. • Dist: Entero que es la Indicación del tipo de proyección. Retorno: • Vacío **extraerImagenSeleccion:** Extrae las selecciones de la imagen para un posterior uso del mismo Argumentos: • Src: IplImage. Contenedor de la imagen de entrada que será tratada • Selección: Array de enteros donde se introducirán los valores oportunos que contienen las posiciones de las selecciones. Retorno: • Vacío **proyeccionHor:** Realiza la proyección horizontal de una determinada imagen introducida por parámetro. Argumentos: • Mask: IplImage. Contenedor de la imagen a ser proyectada • Alto: array de enteros que contendrá la proyección • Lim: Entero que indicará el máximo suavizado deseado en la proyección. • Selección: Array de enteros que indicará el máximo y mínimo gradiente. Retorno: • Vacío buscarBlancoPry Realiza la proyección horizontal de una determinada imagen introducida por parámetro al principio del algoritmo. Argumentos: • Src:IplImage. Contenedor de la imagen de entrada para la realización de la proyección. • Selección: Array de enteros que contienen los máximos y mínimos gradientes encontrados. Retorno: • Vacío

	proyeccionVerT: Realiza la proyección vertical de una determinada imagen introducida por parámetro. Argumentos: • Mask: IplImage. Contenedor de la imagen a ser proyectada • Alto: array de enteros que contendrá la proyección • Lim: Entero que indicará el máximo suavizado deseado en la proyección. • Selección: Array de enteros que indicará el máximo y mínimo gradiente. Retorno: • Vacío

Tabla 15 - Algorithm Cool Detector

4.3.3 Componente 2: Algoritmo FCM

Este componente realizará la implementación del algoritmo inspirado en el apartado 3.2. Basado en c-medias difusas y en la transformada de Hough. El diseño se descompondrá en dos apartados. El primero estará compuesto por el diagrama de clases de este componente y el segundo una explicación exhaustiva de las clases formantes.

4.3.3.1 Diagrama de Clases Componente 2

Figura 44 - Diagrama de Nivel 2: Componente Algoritmo 2 - Clase AlgorithmFCM

4.3.3.2 Descripción de Clases Componente 2

AlgorithmFCM - Clase			
Interfaz Requerido	N/A	**Interfaz Implementado**	Algorithm-Interfaz
Responsabilidades:		Realización el algoritmo inspirado en el propuesto en el apartado 3.2.	
Atributos:			
Operaciones:		**ejecutarAlgoritmo:** Ejecución en tiempo real del algoritmo selecciónado. Argumentos: • Image – IplImage contenedor de la imagen de entrada • Outp – IplImage contenedor de la imagen de salida Retorno: • Vacío	

	setImages: Cambio de las imágenes tanto de salida como de entrada. Argumentos: • Image – IplImage contenedor de la imagen de entrada • Outp – IplImage contenedor de la imagen de salida Retorno: • Vacío **Algorithm1FCM:** Constructor de la clase, inicializa los valores iniciales. Argumentos: • N/A Retorno: • Vacío **insertarMaximaVotación:** Inserta la máxima votación de la transformada de Hough en la imagen que luego se tratará. Argumentos: • Resultado: array de enteros que contiene las distintas votaciones. • Result: array de enteros que contiene la información global de votaciones. • Size_resul: entero que indica el tamaño de resul. • Size: entero que indica el tamaño de resultado. • Min_Radius: indicación del minimo radio a buscar. • X: posición en el eje x de la imagen del centro de la circuferencia • Y: posición en el eje y de la imagen del centro de la circunferencia Retorno: • Vacío **inicializarAgrupacionDifusa**: Inicialización aleatoria de los pesos asociados a cada centro teniendo en cuenta la restricción que los pesos asociados a un pixel debe tener una probabilidad de 1 para pertenecer a todo el conjunto. Argumentos: • Ini: matriz bidimensional en coma flotante que contiene la estructura de datos que será utilizada para inicializar los valores Retorno: • Vacío **setAll**: Modifica los valores de los atributos iniciales Argumentos: • C: entero que indica el numero de centros. • Tam: tamaño de la matriz de pesos. • Coef: coeficiente al que se potenciará los pesos y los centros Retorno: • Vacío **iteración**: Produce una nueva iteración en el algoritmo c-medias difuso que consta de la actualización de centros y pesos Argumentos:

	• N/A Retorno: • Vacío **agrupaciónDifusa:**Realiza el algoritmo completo de c-medias difuso con todas las iteraciones y el control de igualdad Argumentos: • N/A Retorno: • Vacío **comprobación:** Testea si los centros han cambiado bajo un determinado épsilon lo suficiente. Argumentos: • Uanterior: matriz bidimensional en coma flotante con la cual se comparará los nuevos pesos obtenidos y que contiene los viejos valores de los pesos. Retorno: • True si no se cumple false en caso contrario. **aleatorio:** Devuelve un numero aleatorio entre los valores que se han dado. Argumentos: • Num: Entero que indica cual es el máximo valor por el cual se debe dividir el número aleatorio. Retorno: • Vacío **actualizarcentros:** Una vez realizado una iteración realiza una actualización de los centros en función de los pesos según el algoritmo c-medias difuso. Argumentos: • N/A Retorno: • Vacío **actualizarPesos:** Una vez realizada una iteración realiza una actualización de los pesos en función de los centros según el algoritmo c-medias difuso. Argumentos: • N/A Retorno: • Vacío **escribirImagen:** Argumentos: • Outp: Escritura de la imagen de salida con los valores obtenidos. Retorno: • Vacío **máximo:** Devuelve el máximo valor entero en una colección bidimensional de números en coma flotante. Argumentos:

	• Pesos: matriz bidimensional con los pesos a estudiar. Retorno: • Vacío **HoughCirclesT:** Realiza la transformada de Hough para círculos siguiendo el algoritmo explicado en el anexo 7.1. Argumentos: • Edges: IplImage. Contenedor de la imagen que contiene los bordes obtenidos en la fase anterior o en la fase del algoritmo c-medias difuso. • Resultados: array de enteros contenedor de las votaciones efectuadas. • Size: entero que indica el tamaño de resultados. • Rad_min: Entero que indica mínimo radio a buscar en el mapa de bordes • Rad_max: Entero que indica máximo radio a buscar en el mapa de bordes Retorno: • Vacío

Tabla 16 Clase Algoritmo FCM

4.3.4 Componente 3: Algoritmo IntDiff

Este componente realizará la implementación del algoritmo inspirado en el apartado 3.23.2. Basado en la eliminación de reflejos, basto agrupamiento y el operador integro diferencial. El diseño se descompondrá en dos apartados. El primero estará compuesto por el diagrama de clases de este componente y el segundo una explicación exhaustiva de las clases formantes.

4.3.4.1 Diagrama de Clases Componente 3

Figura 45 - Diagrama de Nivel 2: Componente Algoritmo 3 - Clase AlgorithmIntDiff

4.3.4.2 Descripción de Clases Componente 3

<table>
<tr><th colspan="4">AlgorithmIntDiff</th></tr>
<tr><td>Interfaz Requerido</td><td>N/A</td><td>Interfaz Implementado</td><td>Algortihm_Interfaz</td></tr>
<tr><td>Responsabilidades:</td><td colspan="3">Realización del algoritmo basado en la proposición del apartado 3.2. Procederá a realizar distintas fases para la correcta segmentación del iris.</td></tr>
<tr><td>Atributos:</td><td colspan="3">Preprocesada: IplImage, contenedor de la imagen de la primera fase para la eliminación de los reflejos especulares en la imagen.
Entrada: IplImage, contenedor de la imagen de entrada sin alterar.
Salida: IplImage, contenedor de la imagen de salida binarizada del algoritmo, y obtenida tras la finalización de la fase de eliminación de sombras.
grupoBasto: IplImage, contenedor de la imagen obtenidas tras la segunda fase consistente en la agrupación basta.
intDiff: IplImage, contenedor de la imagen obtenida tras la tercera fase consistente en la constelación del anillo para operadores integro-diferenciales.
sinParpados: IplImage, contenedor de la imagen obtenida tras la cuarta fase consistente en la eliminación de parte del iris por detección de párpados.</td></tr>
<tr><td>Operaciones:</td><td colspan="3">ejecutarAlgoritmo: Ejecución en tiempo real del algoritmo selecciónado.
Argumentos:
• Image – IplImage contenedor de la imagen de entrada
• Outp – IplImage contenedor de la imagen de salida
Retorno:
• Vacío
setImages: Cambio de las imágenes tanto de salida como de entrada.
Argumentos:
• Image – IplImage contenedor de la imagen de entrada
• Outp – IplImage contenedor de la imagen de salida
Retorno:
• Vacío
AlgorithmIntDiff: Constructor de la clase inicializa todos los atributos a valores iniciales.
Argumentos:
• N/A
Retorno:
• Vacío.
eliminacionReflejos: Elimina los reflejos especulares debido a proyecciones de luz en el iris y otras partes de ojo.
Argumentos:
• Image – IplImage contenedor de la imagen de entrada.
• Outp – IplImage contenedor de la imagen de salida.
Retorno:
• Vacío.</td></tr>
</table>

	agrupamientoBasto: Realiza el agrupamiento basto para tener localizada la zona del iris. Argumentos: • Image – IplImage contenedor de la imagen de entrada. • Outp – IplImage contenedor de la imagen de salida. Retorno: • Vacío. **InicializaciónVasta:** Inicialización de los valores y determinación a priori de distintas regiones. Argumentos: • Image – IplImage contenedor de la imagen de entrada. • Outp – IplImage contenedor de la imagen de salida. Retorno: • Vacío. **etiquetacionVasta:** Un vez realizado el agrupamiento se procede a etiquetar cada parte para obtener la zona del iris. Argumentos: • Image – IplImage contenedor de la imagen de entrada. • Outp – IplImage contenedor de la imagen de salida. Retorno: • Vacío. **Anillar:** Comprobación del valor según el operador máximo diferencial y devolución del mismo. Argumentos: • Image – IplImage contenedor de la imagen de entrada. • Outp – IplImage contenedor de la imagen de salida. Retorno: • Vacío. **Costelar:** Operación completa de la constelación anillar basada en el operador integro diferencial. Argumentos: • Image – IplImage contenedor de la imagen de entrada. • Outp – IplImage contenedor de la imagen de salida. Retorno: • Vacío. **refinarLimite:** Refinamiento de los limites obtenidos descartando partes oscuras. Argumentos: • Image – IplImage contenedor de la imagen de entrada. • Outp – IplImage contenedor de la imagen de salida. Retorno: • Vacío. **localizarParpados:** Modelización de los párpados para ir refinando la zona del iris. Argumentos: • Image – IplImage contenedor de la imagen de entrada. • Outp – IplImage contenedor de la imagen de salida.

	Retorno: • Vacío. **eliminarSombras:** Eliminación de las distintas sombras existentes en la región del iris y de las pestañas que puedan ser encontradas. Argumentos: • Image – IplImage contenedor de la imagen de entrada. • Outp – IplImage contenedor de la imagen de salida. Retorno: • Vacío.

Tabla 17 Clase Algoritmo Integro Diferencial

5. Experimentación

Este apartado recaba la información necesaria para la realización de una exhaustiva comparativa entre la implementación inspirada en los algoritmos propuestos. Para ello se realizará un conjunto de pruebas con un ground truth y un evaluador que permita tener medidas cuantitativas en un esfuerzo de poder extraer conclusiones acerca de la eficacia de los algoritmos.

La base de datos, junto con el ground truth que incorpora, utilizada para la realización de las pruebas será UBIRIS en su versión 2, previamente explicada escuetamente y en apartados siguientes tratada en profundidad.

El evaluador a utilizar es una aplicación propuesta por el departamento SOCIA Lab – Soft Computing and Image Analysis Group – con sede en la Universidad de Beira para un certamen denominado NICE (Noisy Iris Challenge Evaluation) propuesto para la comunidad científica mundial con el fin de encontrar el algoritmo que realice más eficazmente la segmentación de iris.

A continuación se procederá a explicar la evaluación de los algoritmos, el evaluador, y la base de datos utilizada para la realización de las pruebas.

5.1 Base de Datos UBIRIS versión 2

En el año 2004 el departamento del laboratorio SOCIA construyó la base de datos UBIRIS v1. Su propósito fue simular un proceso de adquisición de imágenes menos restrictivo e incluir un distinto tipo de datos que ocluyen el iris. Una gran cantidad de experimentos fueron realizados sobre esta base de datos y reportados a la literatura científica, aunque el realismo de los factores ruidosos recibió algunas críticas. Esto indujo en una mayor motivación para el desarrollo de una nueva versión de la base de datos (UBIRIS v2) en la cual las imágenes son actualmente capturadas en condiciones no restrictivas (distintas distancias, en movimiento, oclusiones…), con sus correspondientes factores ruidosos mucho más realistas.

La fotografía original contiene las siguientes características:

Formato de Imagen Ubiris	
Herramienta	**Función**
Image Format	tiff
Image Dimensions	400 x 300 (ancho x alto)
Horizontal Image Resolution	72 dpi
Vertical Image Resolution	72 dpi
Image Bit Depth	24 bit
Image Capturing	On the move and at-a distance (between 3 and 7meters) image capturing.
Image Capturing Framework Details	Camera = Canon EOS 5D, Color Representation= sRGB, Shutter Speed = 1/197 sec. Lens Aperture = F/6, 4. Focal Length = 400 mm. F-Number = F/6,3. Exposure Time = 1/200 sec. ISO Speed = ISO-1600

Tabla 18 - Formato de la Imagen de UBIRIS v. 2

El propósito general de UBIRIS v2 es constituir una nueva herramienta para evaluar la eficacia del reconocimiento de iris alejándose de la captura de condiciones ideales. En este ámbito, varios tipos de imágenes no ideales, distancias de imagen, perspectiva del sujeto y condiciones de iluminación existente en la base de datos puede ser una fuerte utilidad para la especificación de la viabilidad y constricciones en el reconocimiento de iris.

Figura 46 - Adquisición de imágenes para UBIRIS v2 – Adquisición de las imágenes (A,B), fuentes de luz (C,D) y localización de los sujetos (E).

La configuración del entramado de adquisición está ilustrada en la Figura 46 - Adquisición de imágenes para UBIRIS v2 – Adquisición de las imágenes (A,B), fuentes de luz (C,D) y localización de los sujetos (E)., cuenta tanto con iluminación artificial y natural. Se han puesto diferentes marcas sobre el suelo para obtener las distintas distancias donde se sitúan los sujetos. Dos sesiones de adquisición fueron realizadas, cada una separadas una semana. Para ambas sesiones la localización y la orientación del dispositivo de adquisición y las fuentes de iluminación artificial ha sido cambiada, con la finalidad de incrementar la heterogeneidad.

Se solicitó a los sujetos andar algo más despacio que la velocidad normal y mirar a diferentes marcas laterales que les obliga a rotar tanto la cabeza como los ojos, activando la captura manual de tres imágenes por metro, entre los ocho y cuatro metros, dando un total de quince imágenes por ojo y sesión, para la mayoría de los individuos. Puede ser estresante que se requiera un comportamiento cooperativo pero tiene como único propósito normalizar el número de imágenes usables por sujeto en una sesión de imagen.

Existe en esta adquisición un problema de pigmentación en el iris, este problema estriba en una sesgo en el color del mismo. Debido al carácter latino-mediterráneo de los sujetos, la pigmentación tiende a ser generalmente muy oscura u oscura, habiendo únicamente un 13% de imágenes con iris de pigmentación suave.

El Ground Truth consiste en una colección de imágenes binarizada, asociadas uno a uno con una imagen original, cada fotografía tomada tiene una imagen donde es segmentado el iris. Las imágenes son del mismo tamaño que la imagen original y tienen valores binarios que determina que es iris y que no. Esta colección ha sido realizada manualmente para determinar la segmentación con el trabajo que acarreó la realización de la binarización.

A continuación se muestran un conjunto de ejemplos de interés que muestran algunos casos donde las imágenes están con sus respectivas clasificaciones. Se pueden apreciar oclusiones por pelo, por gafas e incluso ojos cerrados. Los reflejos a su vez también son un problema importante que debe ser tratado pues ocluyen bastante parte del ojo.

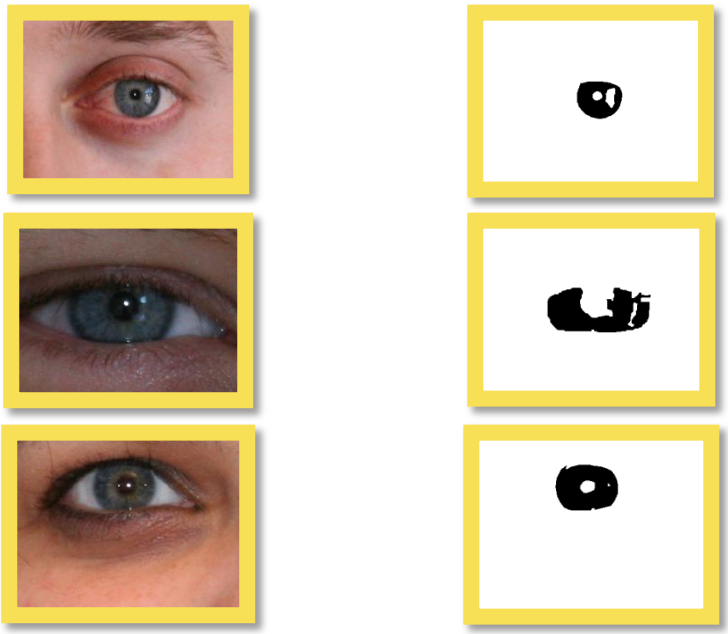


Figura 47 - Imágenes de entrada y salida con ojos de pigmentación clara en Ubiris

Estas imágenes muestran un conjunto de ejemplos de ojos de pigmentación clara, nótese como el color azul puede enturbiar e incluso hacer fracasar a la detección de iris debido a la fuerte pigmentación. Es un factor de ruido añadido para la detección.

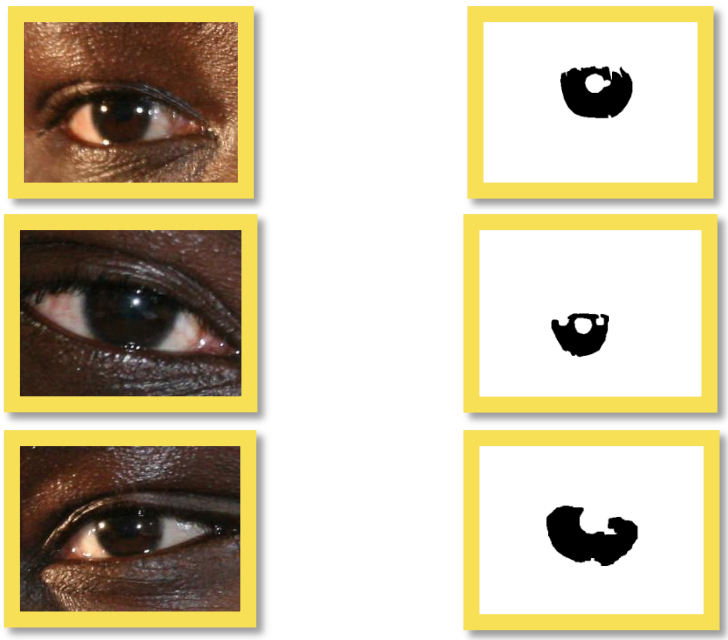


Figura 48 - Imágenes de entrada y salida con pigmentación de piel oscura.

Estos ejemplos muestran imágenes de personas de coloración oscura en su piel que también puede ser considerado un factor de ruido en las imágenes para una correcta segmentación del iris.

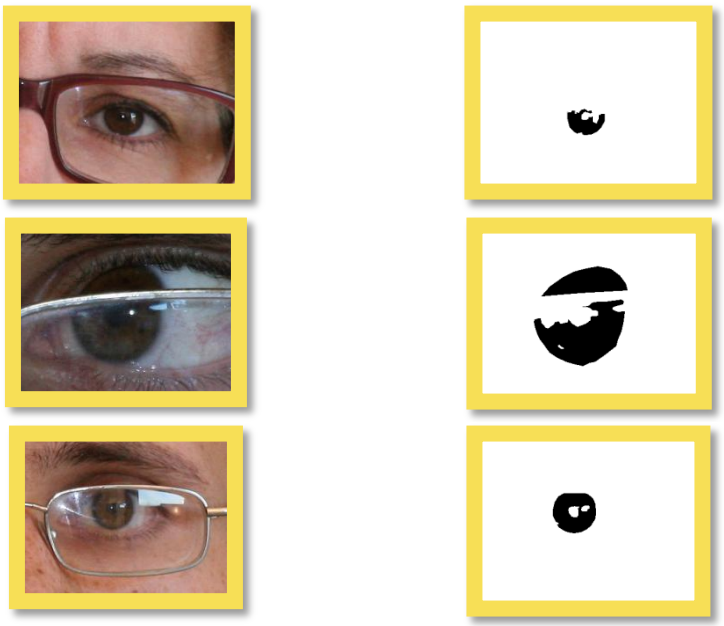


Figura 49 - Imágenes de entrada y salida con gafas que ocluyen el iris

Estos ejemplos muestras iris ocluidos en parte por gafas y reflejos producidos por las gafas siendo un factor de ruido a tener en cuenta para la segmentación correcta del iris en la base de datos Ubiris.

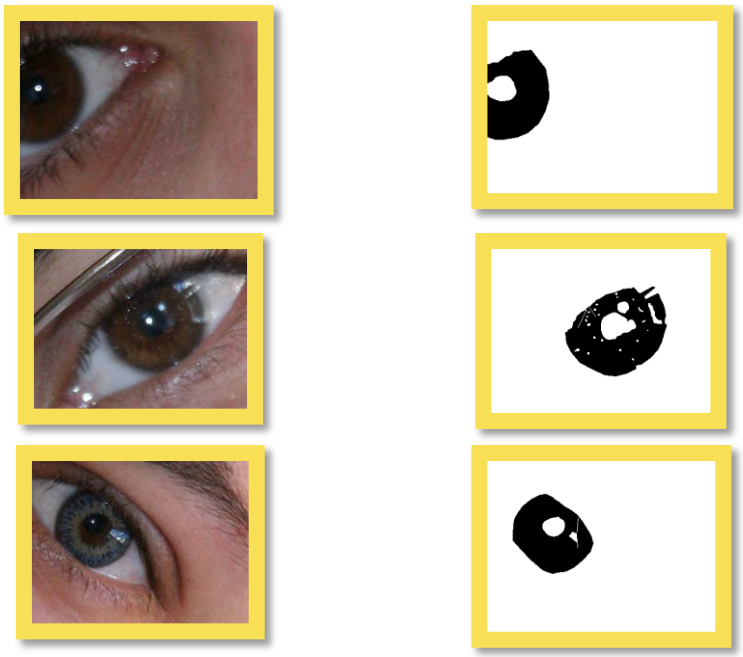


Figura 50 - Imágenes de entrada y salida con otras condiciones de ruido

En esta colección de ejemplo se puede percibir una gran disparidad de condiciones ruidosas que pueden darse en entornos reales, distintas distancias, ojos cerrados, ángulos forzados en la adquisición de imágenes, lentillas de contacto e incluso iris parciales debido a una toma forzada.

5.2 Evaluador NICE

En abril del 2008 se lanzó un certamen con el fin de determinar los mejores algoritmos sobre segmentación de iris usando la base de datos UBIRIS. Para ello se creó una aplicación que pudiese evaluar de manera cuantitativa el error cometido.

Con este fin, se creó la aplicación que toma como entrada un ejecutable (el algoritmo implementado en el) y comprueba teniendo como entrada una imagen de la base de datos que coincida la salida del algoritmo con el su clasificación manual. En la imagen siguiente se muestra como ejemplo la pantalla del evaluador.

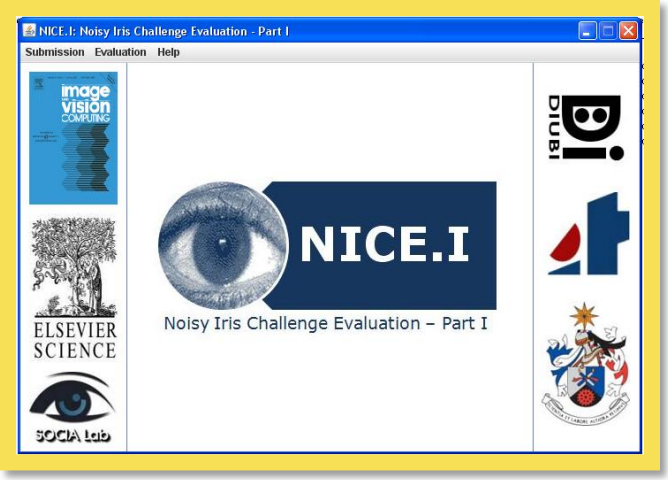


Figura 51 - Evaluador de NICE

Sea Alg el algoritmo que evaluar, el cual realiza la segmentación de las regiones libres de ruido del iris.

Sea $I = \{I_1, I_2, \dots, I_n\}$ es la colección de datos que contiene las imágenes de entrada del iris.

Sea $O = \{O_1, O_2, \dots, O_n\}$ es la colección de imágenes de salida correspondientes a las entradas descritas, tal que $Alg(I_i) = O_i$.

Sea $C = \{C_1, C_2, \dots, C_n\}$ la colección de las imágenes clasificadas manualmente (o lo que es lo mismo el Ground Truth), dadas por SOCIA Lab y contenidas en su base de datos UBIRIS. Se asume que cada C_i contiene una segmentación perfecta del iris y detección de ruido de la imagen I_i.

Todas las imágenes de I, O y C tienen las mismas dimensiones de ancho y alto.

Two measures of evaluation will be used:

El ratio de clasificación de Error (E_i) del Alg sobre la imagen de entrada I_i es dado por la proporción de los correspondiente pixeles no coincidentes sobre la imagen:

$$E_i = \frac{1}{c \cdot r} \sum_{c'} \sum_{r'} O(c', r') \otimes C(c', r')$$

Donde:

- $O(c', r')$ Pixel de imagen de salida
- $C(c', r')$ Pixel de la imagen de clasificación manual.
- c', r' son la anchura y la altura de la imagen

El ratio de error de clasificación E^1 del Alg es dado por la media de los errores en las imágenes de entrada E_i:

$$E^1 = \frac{1}{n}\sum_i E_i$$

Donde:

- n es el número de imágenes
- E_i es el error cometido en la imagen i
- E^1 es el error global cometido

La segunda medida de error es dada para compensar la desproporción entre las probabilidades *a priori* de los pixeles en la imagen no-iris e iris. E^2 Es por tanto la media entre los ratios de falsos positivos y de falsos negativos.

$$E_i^2 = \frac{FPR + FNR}{2}$$

$$E^2 = \frac{1}{n}\sum_i E_i$$

Donde:

- n es el número de imágenes
- E_i es el error cometido en la imagen i
- E^1 es el error global cometido
- FPR es el falso positivo
- FNR es el falso negativo

5.3 Evaluación de Algoritmos

Para la realización de las pruebas se ha procedido a utilizar 1000 imágenes de la base de datos Ubiris v2 con su respectivo ground truth. Estas imágenes recogen las cualidades que se han explicado previamente y que hace, que este estudio, revierta gran interés. Estas cualidades son las condiciones ruidosas en las que fueron tomadas las fotos.

La evaluación se ha realizado mediante la aplicación, previamente explicada, en soporte java, proporcionada por NICE para la evaluación de los algoritmos.

	Falso Positivo	Falso Negativo	Error[2]	Error[1]
Algoritmo Integro Diferencial	0,05516	0,18549	0,120325	0,062
Algoritmo c-medias difuso	0,18834	0,3536	0,27097	0,1248
Algoritmo k-medias de color	0,22516	0,33549	0,280325	0,15369

Tabla 19 -Resultados de los Algoritmos

En la tabla se puede apreciar el falso positivo, el falso negativo, la medida de error 2 que consiste en la media ponderada de ambos falsos y la medida del error 1 que es la media ponderada de los errores cometidos por el algoritmo, siempre y cuando no coincidan como se ha explicado previamente.

El algoritmo basado en el operador integro diferencial claramente se ha desmarcado en el éxito de sus resultados. En el grafico siguiente se percibe lo anterior expuesto y los resultados se presentan como los más óptimos dentro de la segmentación de iris no cooperativa.

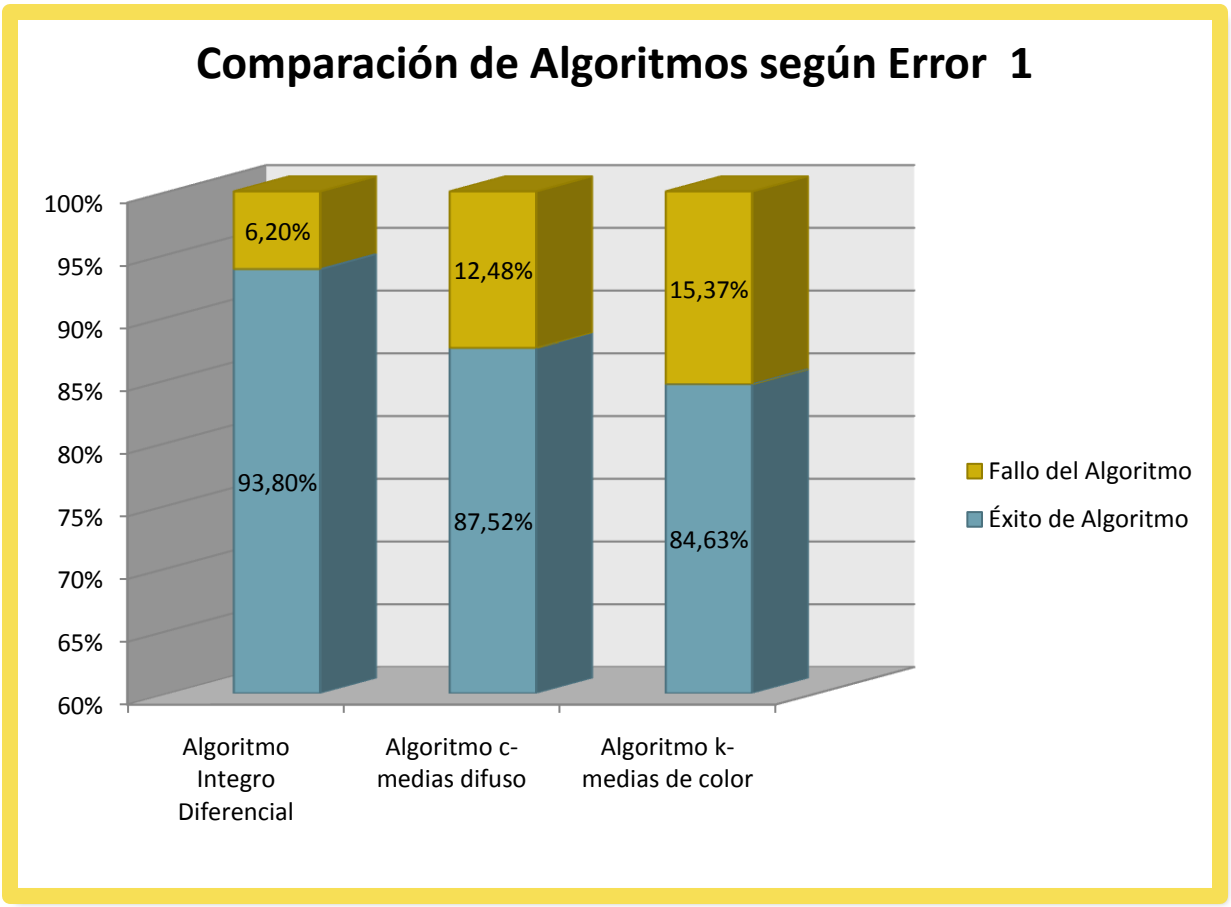


Figura 52 - Comparativa de algoritmos según tasa de error 1

En esta medida se puede apreciar como el algoritmo mas solido frente a las perturbaciones encontradas en la base de datos UBIRIS, y por lo tanto preparado para entornos más reales es el algoritmo basado en el operador integro-diferencial. Con un error del 6.2 %. Mientras que el algoritmo c-medias difuso presenta un desgaste debido a la modelización de dos círculos no concéntricos perfectos. No obstante el algoritmo de aproximación basta, basado en K-medias y el color presenta cierta robustez teniendo un error del 15%, aunque, no obstante tiene el desgaste mayor.

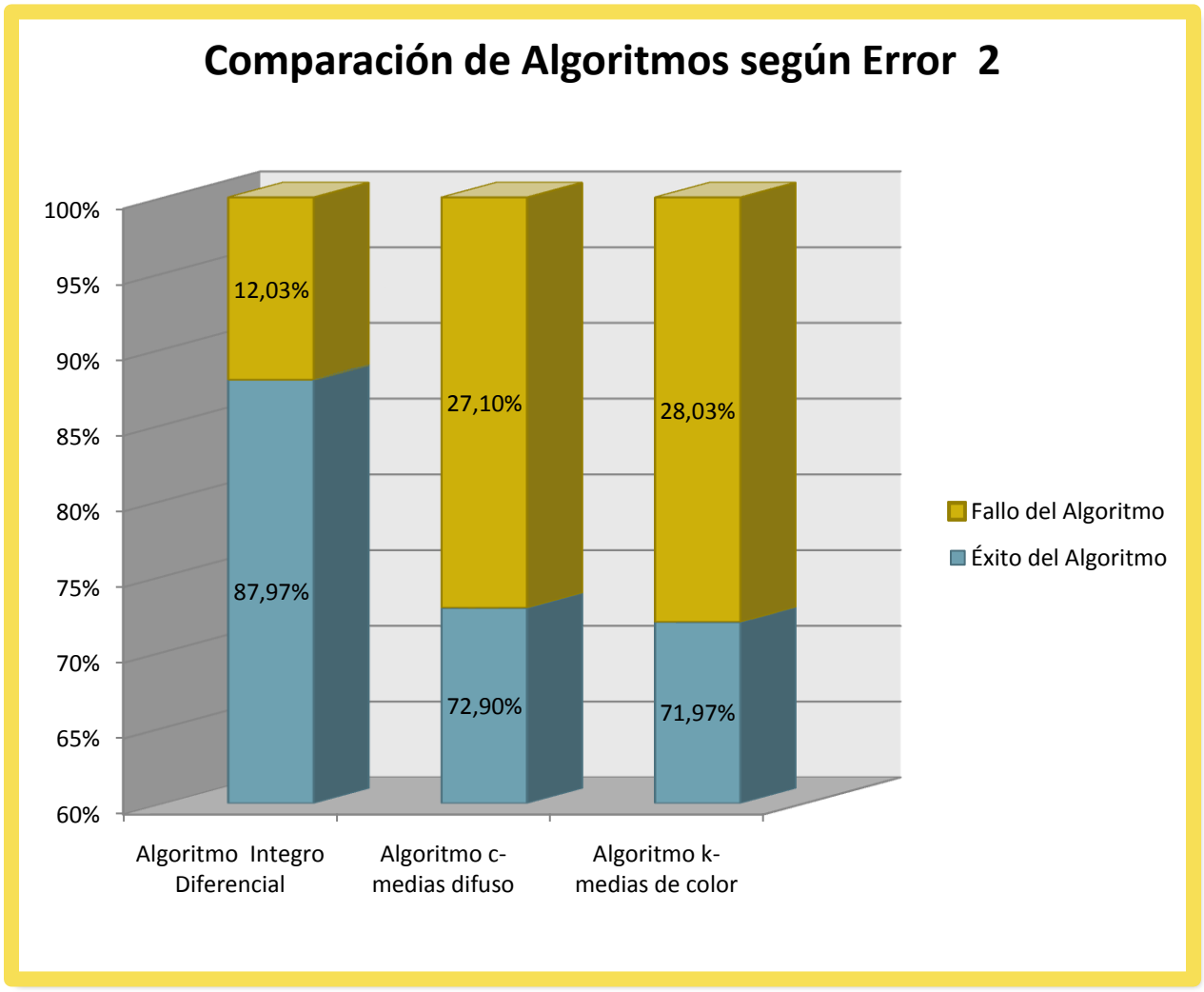


Figura 53 - Comparativa de algoritmos según tasa de error 2

Aquí se puede apreciar un cambio significativo en los errores basados en los falsos positivos y los falsos negativos. El algoritmo k-medias presenta un importante choque y aumenta drásticamente sus medias subiendo en el falso negativo de manera significativa con más del 20%. En cambio el algoritmo integro diferencial a presentado una gran robustez y es el mejor por diferencia con algo más del 12%. Por último, se aprecia como el algoritmo c-medias ha perdido parte de su eficacia y se aproxima en el error al algoritmo de basta aproximación.

Por último se mostrarán distintas imágenes que reflejen y aporten consistencia a los resultados numéricos obtenidos. Estas imágenes presenta la salida superpuesta a la imagen de entrada, de tal manera que se pueda percibir claramente cuál es el éxito de los algoritmos gracias a una inspección meramente visual.

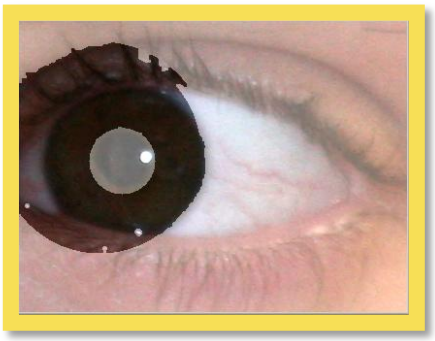


Figura 54 - Imagen 1 Algoritmo Integro Diferencial

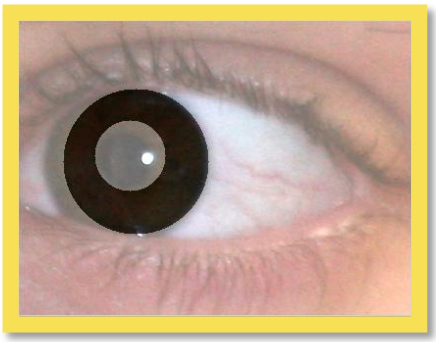


Figura 55 - Imagen 1 Algoritmo difuso c medias

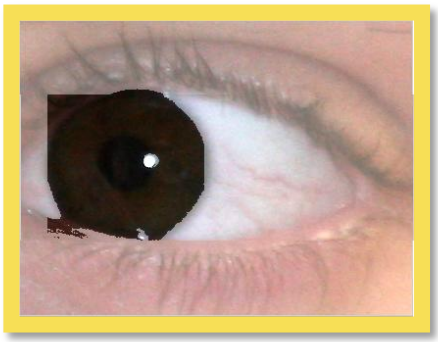


Figura 56 - Imagen 1 Algoritmo K - medias

En la Figura 54 se puede apreciar un amplio marco de error en el algoritmo integro diferencial debido a que no detecta el borde superior ni inferior del párpado (consecuencia de realizar únicamente el corte de párpados bajo fuertes restricciones). En cambio los resultados obtenidos en los otros dos algoritmos es bastante más perfecto (Figura 56 - Imagen 1 Algoritmo K - medias y Figura 55 - Imagen 1 Algoritmo difuso c medias), aunque se percibe una gran cantidad de falsos negativos en el algoritmo c-medias difuso (perfil izquierdo del ojo y amplitud de pupila) mientras que el algoritmo k-medias absorbe la pupila pero no los reflejos como falsos positivos.

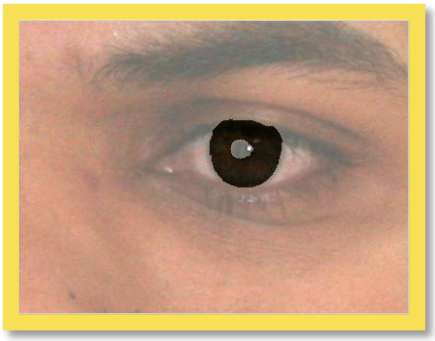


Figura 57 - Imagen 2 Algoritmo Integro Diferencial

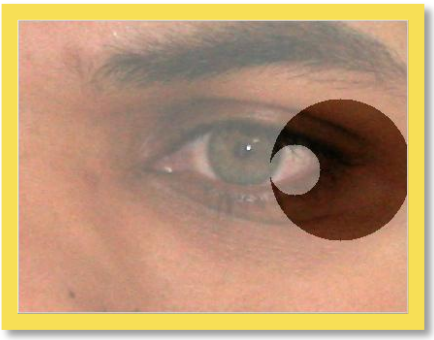


Figura 58 - Imagen 2 Algoritmo difuso c medias

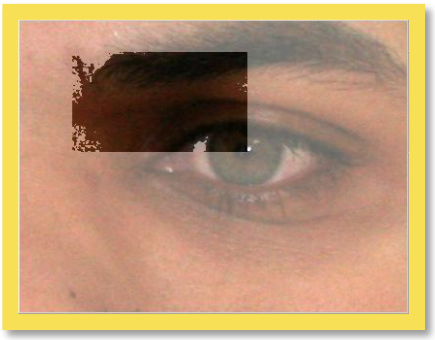


Figura 59 - Imagen 2 Algoritmo K - medias

En este caso, el algoritmo integro diferencial en la Figura 57 - Imagen 2 Algoritmo Integro Diferencial presenta un resultado óptimo, rozando la perfección sobre la segmentación del iris en la imagen tomada sobre un ojo de raza india y colores claros. En cambio, los otros dos

algoritmos presentan un error sustancial en las Figura 58 - Imagen 2 Algoritmo difuso c medias y Figura 59 - Imagen 2 Algoritmo K - medias, pues no aciertan que parte es la formante del iris y equivocan en su mayoría la localización del iris. Dando un resultado pésimo y superando el 80% de error en ambas imágenes.

Figura 60 - Imagen 3 Algoritmo Integro Diferencial

Figura 61 - Imagen 3 Algoritmo difuso c medias

Figura 62 - Imagen 3 Algoritmo K - medias

Este ejemplo presenta en las tres figuras una gran precisión gracias a la clara percepción de la circularidad y el color, pero existen matices que se deben explicar. El primer algoritmo en la Figura 60 - Imagen 3 Algoritmo Integro Diferencial presenta el mejor resultado gracias a que rechaza las partes brillantes de la imagen, elimina la pupila y corta el párpado superior. En la Figura 61 - Imagen 3 Algoritmo difuso c medias se modeliza correctamente el iris y la pupila pero se absorbe como falso positivo todos los brillos y reflexiones existentes en la imagen y situados dentro del iris (debido a que no tiene mecanismos de refinamiento para eliminar los reflejos). Por último en la Figura 62 - Imagen 3 Algoritmo K - mediasse eliminan correctamente las reflexiones pero la pupila es tomada como iris también, debido, como en el anterior algoritmo, a la falta de refinamiento para eliminar la pupila del iris.

Figura 63 - Imagen 4 Algoritmo Integro Diferencial

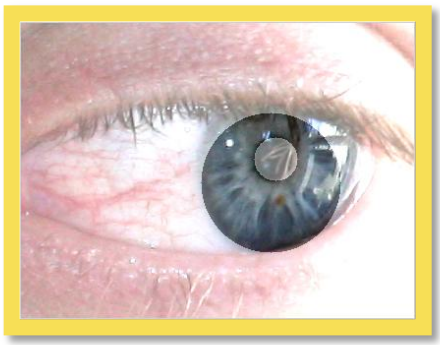


Figura 64 - Imagen 4 Algoritmo difuso c medias

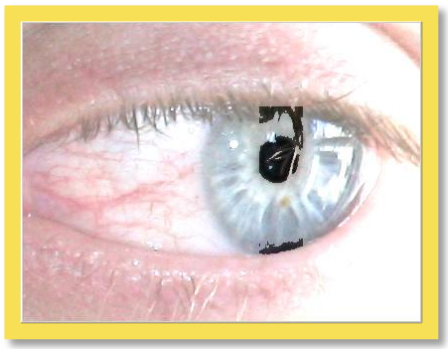


Figura 65 - Imagen 4 Algoritmo K - medias

En este último ejemplo, el algoritmo integro diferencial en la Figura 63 - Imagen 4 Algoritmo Integro Diferencial, es bastante mejor que en el resto de algoritmos en las Figura 64 - Imagen 4 Algoritmo difuso c medias y Figura 65 - Imagen 4 Algoritmo K - medias, a pesar de la inexactitud en uno de los márgenes (izquierdo para ser exactos), perfila correctamente los párpados y elimina regiones de sombra. Las otras dos imágenes son testigo de errores importantes, pues el algoritmo de c-medias difuso ofrece una circularidad demasiado escasa y no elimina los reflejos mientras que en el algoritmo k-medias debido a la claridad del color del iris es incapaz de detectar el iris y solo toma la pupila y parte de las pestañas como iris, incurriendo en un gran error.

6. Conclusiones

En este proyecto se ha presentado un análisis cuantitativo sobre distintos algoritmos que abordan la segmentación de iris en entornos no cooperativos, este entorno ha sido conseguido gracias a una base de datos preparada específicamente para ofrecer condiciones más reales en la adquisición de imágenes.

Resumiendo el proyecto se ha situado primeramente el estudio dentro de un ámbito de biometría, dentro de este ámbito se ha especificado el reconocimiento de iris como el campo de investigación que revierte interés para este documento. Finalmente se ha procedido a concretar la segmentación de la imagen para la extracción de características como el foco del análisis. Se ha explicado rigurosamente el estado del arte actual y como las nuevas tendencias marcan un camino para la adquisición, segmentación y reconocimiento de iris en entornos no cooperativos y en condiciones ruidosas o muy ruidosas.

Para el análisis de algoritmos se ha procedido a realizar la implementación inspirada en buena parte en la literatura actual. Teniendo como referencia los algoritmos de difusas c- medias, el operador integro diferencial modificado y el algoritmo propuesto de basta aproximación que se ha presentado con una mayor resistencia que *a priori* de la que se podía concebir.

También se ha aportado a este proyecto el rigor de un correcto diseño software que facilite el desarrollo de los algoritmos y permita marcar los fundamentos del mismo.

Con referencia a los algoritmos se puede observar como el algoritmo del operador integro diferencial ofrece unos resultados que se acercan a la perfección y por consiguiente se puede decir que es el más óptimo para una correcta segmentación del iris. Mientras que el algoritmo c-medias y transformada de Hough aunque presentase unos resultados óptimos en otras bases de datos tales como Ubiris v1 o bien CASIA, se aprecia que en condiciones ruidosas pierde eficacia, teniendo que aportar una mejoría que se puede traducir como un mayor refinamiento. Además este algoritmo fue únicamente probado mediante inspección visual en la literatura, perdiendo rigor. Finalmente el algoritmo basado en k-medias y estudio del color presenta una buena aproximación pero que necesita un refinamiento muy importante para ser competitivo.

Para futuros estudios, se debe conseguir que el algoritmo k-medias tenga mayor consistencia gracias a refinamientos tales como algoritmos de crecimiento semántico, linearización de los límites o aplicando tanto la transformada de Hough como el operador integro diferencial en la aproximación realizada para la obtención de un correcto resultado.

El algoritmo de c-medias debe realizar un estudio de las partes ocluidas por las pestañas y el ruido ocasionado por ojos cerrados, ya que en este estado aún no es contemplado y por consiguiente experimenta un grave deterioro en sus resultados.

El algoritmo del operador integro diferencial presenta unos resultados que poco pueden mejorar, aun así se pueden ajustar mediante un algoritmo genético la gran cantidad de parámetros a tomar para obtener una sustancial mejora gracias a la optimización del algoritmo en sí y no de sus conceptos. No obstante, surge la necesidad de eficiencia para poder ser utilizado en tiempo real (aunque el algoritmo más eficaz tarda únicamente algunos segundos). Se debería intentar optimizar el tiempo de ejecución para intentar reducirlo.

Gracias a la extensa base de datos que fue ofrecida por el laboratorio SOCIA Labs, los algoritmos pueden ser mejorados gracias al gran banco de pruebas que facilita, y la labor en la construcción de esta base de datos, o en consecutivas, no debe ser descartada pues ofrece un gran rigor y calidad en los estudios derivados. En definitiva, la investigación de bases de datos que ofrezcan un buen marco de pruebas debe ser de carácter primario para la continua mejora en este campo.

Es necesario destacar las inmensas complicaciones que se tuvo para la implementación del algoritmos inspirado en el operador integro diferencial pues se hubo de reiniciar la implementación un número considerable de veces debido a la falta de información y la necesidad de mejorar un algoritmo que, a veces, parecía imposible.

Todos los algoritmos están sujetos a la desventaja de la complejidad computacional, siendo bastante importante el tiempo de cómputo en las maquinas convencionales. Como consecuencia, se debe conseguir o bien una mayor capacidad computacional o bien una mejora en la complejidad de los algoritmos.

Finalmente, el proyecto realizado ha cumplido ampliamente todos los objetivos marcados en la introducción, pues se ha podido evaluar, comparar y extraer útiles conclusiones a partir del estudio realizado.

7. Anexos

En este apartado se recogen aquellos aspectos del proyecto que por su carácter genérico es introducido en este apartado. Las herramientas matemáticas y espacios algebraicos, entre otros, explicados en este anexo forman parte del proyecto pero se ha preferido recogerlos en un único apartado, este, para poder enfocar el documento únicamente a la segmentación del iris.

7.1 Anexo I – Transformada de Hough para círculos

La transformada de Hough, pensada en un principio para la detección de rectas es también aplicable a cualquier función de la forma $g(v, c) = 0$ donde v es un vector de coordenadas y c es un vector de coeficientes. Por ejemplo, puntos que caen en el círculo:

$$(x - c_1)^2 + (y - c_{12})^2 = c_3^2$$

Se pueden detectar empleando también la transformada de Hough. En este caso tenemos tres parámetros c_1, c_2 – que son las coordenadas del centro de la circunferencia – y c_3 – que es el radio de la circunferencia – lo que dará lugar a un espacio de parámetros de tres dimensiones, con celdas con forma de cubo y acumuladores – que son sub-divisiones del espacio paramétrico – de la forma $A(i, j, k)$. El procedimiento en este caso es para cada punto del plano formado por los ejes x e y, para cada c_1 y para cada c_2, calcular el valor de c_3 y actualizar el acumulador correspondiente a (c_1, c_2, c_3). La complejidad de la transformada de Hough es claramente dependiente del tamaño del espacio de parámetros.

Para percibir de mejor manera el concepto de transformada, se muestra el ejemplo siguiente, donde las imágenes siguientes muestran un conjunto de círculos. En la imagen de la izquierda hay tres círculos con tres radios diferentes. En este ejemplo se le pasa el parámetro radio a la transformada de hough para que busque círculos de radio 35. En la imagen de la derecha se observa que el círculo 2 todas la curvas pasan por un mismo punto indicando que el círculo 2 tiene un radio de 35 píxeles.

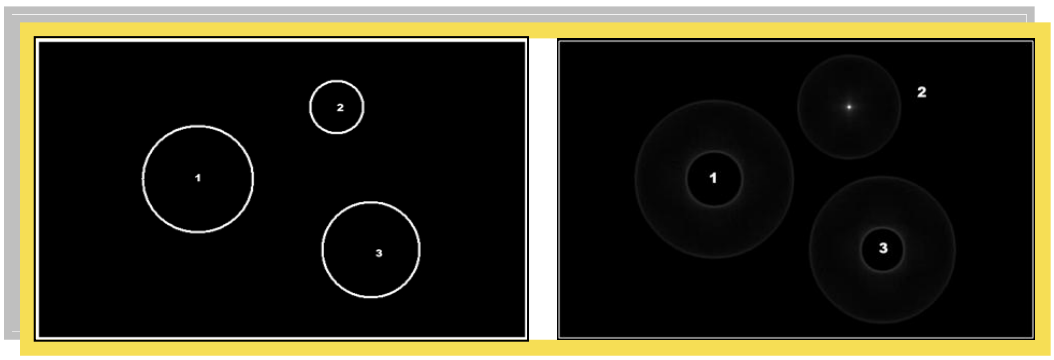


Figura 66 - Ejemplo de transformada de Hough para círculos

7.2 Anexo II – Técnicas para detección de bordes

Los bordes de una imagen contienen mucha de la información de la imagen. Los bordes cuentan donde están los objetos, su forma, su tamaño, y también sobre su textura. Los ejes o bordes se encuentran en zonas de una imagen donde el nivel de intensidad fluctúa bruscamente, cuanto más rápido se produce el cambio de intensidad, el eje o borde es más fuerte. Un buen proceso de detección de bordes facilita la elaboración de las fronteras de objetos con lo que, el proceso de reconocimiento de objetos se simplifica. Para poder detectar los bordes de los objetos, debemos de detectar aquellos puntos borde que los forman.

En general, los bordes de objetos en una imagen los podemos distinguir por los cambios más o menos bruscos de valor entre dos o más píxeles adyacentes. Podemos realizar una clasificación general de los bordes según sea su dirección en:

- Bordes verticales, cuando píxeles conectados verticalmente tienen valores diferentes respecto de los anteriores o posteriores.
- Bordes horizontales, cuando tenemos píxeles conectados horizontalmente, y estos tienen distintos valores respecto de los anteriores o posteriores.
- Bordes oblicuos, cuando tenemos una combinación de las componentes horizontales y verticales.

La diferencia entre los valores de los píxeles nos indica lo acentuado del borde, de forma que a mayores diferencias tenemos bordes más marcados y a menores tenemos unos bordes suavizados.

Los filtros utilizados para la detección de bordes son filtros diferenciales, que se basan en la derivación o diferenciación. Dado que el promediado de los píxeles de una región tiende a difuminar o suavizar los detalles y bordes de la imagen, y esta operación es análoga a la integración, es de esperar que la diferenciación tenga el efecto contrario, el de aumentar la nitidez de la imagen, resaltando los bordes.

7.2.1 Detección mediante derivada de primer orden

Muchas técnicas basadas en la utilización de máscaras para la detección de bordes utilizan elementos estructurales de distinto tamaño y forma. La ventaja de utilizar elementos estructurales grandes es que los errores producidos por efectos del ruido son reducidos mediante medias locales tomadas en los puntos en donde se superpone la máscara. Por otro lado, los elementos estructurales normalmente tienen tamaños impares, de forma que los operadores se encuentran centrados sobre los puntos en donde se calculan los gradientes.

$$\frac{\partial f(x,y)}{\partial x} = \Delta_x = \frac{f(x+d_x,y) - f(x,y)}{d_x} \qquad \Delta_x = f(i+1,j) - f(i,j)$$
$$\frac{\partial f(x,y)}{\partial y} = \Delta_x = \frac{f(x,y+d_y) - f(x,y)}{d_y} \qquad \Delta_y = f(i,j+1) - f(i,j)$$

Los operadores de gradiente común (o gradiente ortogonal) encuentran bordes horizontales y verticales. Estos operadores trabajan mediante convolución. Los operadores de Prewitt, Sobel, Roberts y Frei-Chen son operadores dobles o de dos etapas. La detección de bordes se realiza en dos pasos, en el primero se aplica una máscara para buscar bordes horizontales, y en el segundo paso buscamos los verticales, el resultado final es la suma de ambos. Se muestran algunas máscaras de convolución comunes a continuación. Los detectores de fila (horizontales) son $\mathrm{H_h}$ y los detectores de columna (verticales) son H_v.

	Roberts	Prewitt	Sobel	Frei-Chen
$\mathbf{H_h}$	$\begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & 1 \end{pmatrix}$
$\boldsymbol{H_v}$	$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$	$\begin{pmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -2 & -\sqrt{2} & -1 \end{pmatrix}$

7.2.2 Detección mediante derivada de segundo orden

Los operadores diferenciales del gradiente discutidos en la sección anterior producen una respuesta grande a través de un área donde un borde está presente, esto es especialmente cierto para los bordes de poca inclinación. En cambio los detectores de bordes de la derivada de segundo orden proporcionan una localización mejor del borde. Otra ventaja de los operadores de la derivada de segundo orden es que los contornos del borde detectados son curvas cerradas. Esto es muy importante en la segmentación de imagen. También, no hay respuesta a las áreas de variaciones lineales lisas en intensidad.

El *operador Laplaciano* se define como una *derivada de segundo orden*, por lo cual obtiene resultados superiores a los anteriores y puede trabajar con imágenes donde las variaciones de intensidad no sean suficientemente abruptas para ellos. No obstante presenta una sensibilidad más grande frente al ruido y una ligera incapacidad para determinar la dirección de los bordes.

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

Aunque el Laplaciano responde a las transiciones de intensidad, rara vez se utiliza en la práctica para la detección de bordes, pues tiene los siguientes inconvenientes:

- Los operadores basados en la primera derivada son sensibles al ruido en imágenes. El Laplaciano aún lo es más.
- Genera bordes dobles.
- No existe información direccional de los ejes detectados.

El Laplaciano es un buen ejemplo de un operador de derivada de segundo orden, se distingue de los otros operadores porque es omnidireccional, es decir, destacará los bordes en todas las direcciones. El operador Laplaciano producirá bordes más agudos que la mayoría de las otras técnicas, estos toques de luz incluyen pendientes positivas y negativas de la intensidad.

El borde Laplaciano de una imagen puede ser encontrado convolucionando con elementos estructurales tales como:

Laplaciano 1

$$\begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

Laplaciano 2

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Laplaciano 3

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

La imagen resultante exhibe un cambio del signo en los bordes de la imagen. Estos cambios de signo son referidos como pasos cero. Después del operador de convolución, la imagen se debe procesar para encontrar que estos pasos cero y para fijar, por consiguiente, los píxeles de la salida.

Como se puede ver, los elementos estructurales del operador Laplaciano coinciden con los elementos estructurales de filtro paso alto vistas anteriormente. Esto se debe a que el Laplaciano detecta los bordes, es decir, las altas frecuencias de la imagen, sin considerar la orientación, por lo que además de utilizarse para la detección de bordes sirve también para el filtrado paso alto de imágenes.

7.2.3 Detección mediante Canny

El operador de detección de bordes de Canny fue desarrollado por un catedrático de la universidad de Berkeley (EEUU) en 1986 y se basa en un algoritmo de múltiples fases para detectar un amplio rango de bordes. Es sin duda el operador más utilizado en la detección de bordes.

Este detector cuenta con cuatro fases que se detallan a continuación

7.2.3.1 Obtención del gradiente

Esta fase compuesta de dos etapas, consiste en realizar un suavizado basado en un filtro gausiano y la obtención del gradiente de la imagen suavizada.

- Para calcular el kernel Gausiano se utiliza

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Para calcular el gradiente sobre cada pixel se utiliza

$$e_s(i,j) = \sqrt{J_x^2(i,j) + J_y^2(i,j)}$$

- Se estima también la orientación de la normal de los bordes

$$e_0(i,j) = arctan\frac{J_y}{J_X}$$

7.2.3.2 Supresión no máxima al resultado del gradiente

El objetivo de este paso es Obtener bordes de 1 pixel de grosor al considerar únicamente pixels cuya magnitud es máxima en bordes gruesos y descartar aquellos cuyas magnitudes no alcancen ese máximo.

a) Para todo punto se obtiene la dirección más cercana d_k a 0, $\pi/4$, $\pi/2$ y $3\pi/4$ en $E_a(i,j)$.
b) Si $E_m(i,j)$ es menor que uno de sus dos vecinos en la dirección d_k, $IN(i,j) = 0$ Si no $IN(i,j) = Em(i,j)$.

7.2.3.3 Histéresis de umbral a la supresión no máxima

Permite eliminar máximos procedentes de ruido, etc.

- Entrada IN, E_a, y dos umbrales T_1 y T_2 ($T_1 < T_2$).
- Para todo punto en IN, y explorando en un orden:
  - Localizar el siguiente punto tal que $IN(i,j) > T_2$
  - Seguir las cadenas de máximos locales a partir de $IN(i,j)$ en ambas direcciones perpendiculares a la normal al borde siempre que $IN > T_1$. Marcar los puntos explorados.
- La salida es un conjunto de bordes conectados de contornos de la imagen, así como la magnitud y orientación.

7.2.3.4 Cierre de contornos abiertos (Algoritmo de Deriche y Cocquerez)

a) La imagen de entrada es una imagen de contornos binarizada (1= borde; 0=no borde)

b) Para cada punto de borde de un extremo abierto se le asigna un código que determina las direcciones de búsqueda para el cierre del contorno
c) Para los píxeles marcados con este código se marca como pixel de borde el de máximo gradiente en las tres direcciones posibles.
d) Se repiten los pasos hasta que se cierren todos los contornos.

7.3 Anexo III – Espacios de Color

Un espacio de color es un modelo para definir digitalmente el color percibido desde un punto de vista algebraico. Para cada unidad primaria (generalmente píxel) existe un conjunto de valores, que definen su color.

La forma de definir estos valores y lo que representan es lo que se denomina un espacio de color, que puede ser variable en función de las necesidades computacionales o bien respondiendo a estándares generales.

Existen un gran número de espacios de color, en este apartado únicamente se definirán aquellos que sean necesario para la comprensión del documento.

7.3.1 Espacio RGB – Red, Green, Blue –

La descripción RGB (del inglés Red, Green, Blue; "rojo, verde, azul") de un color hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: el rojo, el verde y el azul. Es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores luz primarios. Indicar que el modelo de color RGB no define por sí mismo lo que significa exactamente rojo, verde o azul, razón por la cual los mismos valores RGB pueden mostrar colores notablemente diferentes en diferentes dispositivos que usen este modelo de color. Aunque utilicen un mismo modelo de color, sus espacios de color pueden variar considerablemente.

Para indicar con qué proporción mezclamos cada color, se asigna un valor a cada uno de los colores primarios, de manera, por ejemplo, que el valor 0 significa que no interviene en la mezcla y, a medida que ese valor aumenta, se entiende que aporta más intensidad a la mezcla. Aunque el intervalo de valores podría ser cualquiera (valores reales entre 0 y 1, valores enteros entre 0 y 37, etc.), es frecuente que cada color primario se codifique con un byte (8 bits). Así, de manera usual, la intensidad de cada una de las componentes se mide según una escala que va del 0 al 255.

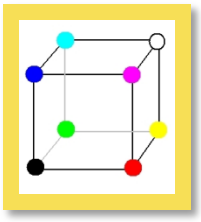


Figura 67 - Modelo RGB de color

7.3.2 Espacio HSV – Hue, Saturation, Value –

El modelo HSV (del inglés Hue, Saturation, Value – Tonalidad, Saturación, Valor), también llamado HSB (Hue, Saturation, Brightness – Tonalidad, Saturación, Brillo), define un modelo de color en términos de sus componentes constituyentes en coordenadas cilíndricas:

- Tonalidad, el tipo de color (como rojo, azul o amarillo). Se representa como un grado de ángulo cuyos valores posibles van de 0 a 360° (aunque para algunas aplicaciones se normalizan del 0 al 100%). Cada valor corresponde a un color. Ejemplos: 0 es rojo, 60 es amarillo y 120 es verde.

- Saturación. Se representa como la distancia al eje de brillo negro-blanco. Los valores posibles van del 0 al 100%. A este parámetro también se le suele llamar "pureza" por la analogía con la pureza de excitación y la pureza colorimétrica de la colorimetría. Cuanto menor sea la saturación de un color, mayor tonalidad grisácea habrá y más decolorado estará. Por eso es útil definir la no saturación como la inversa cualitativa de la saturación.

- Valor del color, el brillo del color. Representa la altura en el eje blanco-negro. Los valores posibles van del 0 al 100%. 0 siempre es negro. Dependiendo de la saturación, 100 podría ser blanco o un color más o menos saturado.

El modelo HSV fue creado en 1978 por Alvy Ray Smith (38). Se trata de una transformación no lineal del espacio de color RGB, y se puede usar en progresiones de color. Nótese que HSV es lo mismo que HSB pero no que HSL o HSI.

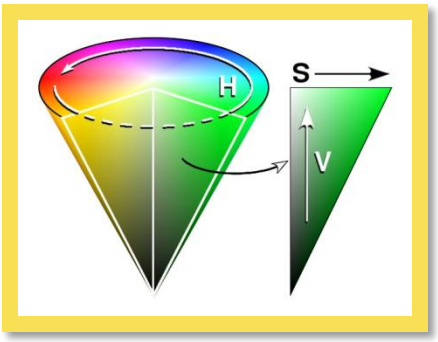


Figura 68 - Modelo HSV de color

A diferencia del espacio de color RGB este espacio presenta un conjunto de características útiles sobre todo relacionadas con la segmentación del color. Esto es debido a que es más fácil representar matemáticamente un sub-espacio (es decir, un color determinado) que en el espacio RGB donde la dispersión del conjunto sería mucho mayor (y por ende, de una menor precisión).

Sobre todo la particularidad del color blanco, determinado por la saturación que posea el píxel. Una saturación alta indica un color indeterminado y una alta probabilidad de que la percepción humana conciba ese píxel como blanco.

7.3.3 Espacio YIQ – Yluminance in phase Quadrature –

El modelo YIQ define un espacio de color, usado antiguamente por el estándar de televisión NTSC. I significa en fase (en inglés: in-fase), mientras que Q significa cuadratura (en inglés: quadrature). NTSC ahora utiliza el espacio de color YUV, que es también utilizado por otros sistemas como PAL

La componente Y representa la información de luminancia y es el único componente utilizado por los televisores de blanco y negro. I y Q representan la información de cromaticidad. En YUV, las componente U y V son las coordenadas X e Y en el espacio de color. I y Q son un segundo par de ejes en el mismo gráfico, rotados 33º, así IQ y UV representan diferentes sistemas de coordenadas en el mismo plano.

El sistema YIQ tiene la ventaja de utilizar las características de la respuesta humana al color. El ojo es más sensible a los cambios en el rango naranja-azul (I) que en el rango púrpura-verde (Q), así se requiere menos ancho de banda para Q que para I. La retransmisión de NTSC limita a I a 1.3 MHz y Q a 0.4 MHz I y Q son barajados en frecuencia con la señal Y de 4 MHz, que mantiene el ancho de banda de la suma de la señal por debajo de 4.2 MHz En sistemas YUV, como U y V contienen información del rango naranja-azul, ambos componentes tienen que tener la misma cantidad de ancho de banda para conseguir la misma fidelidad de color.

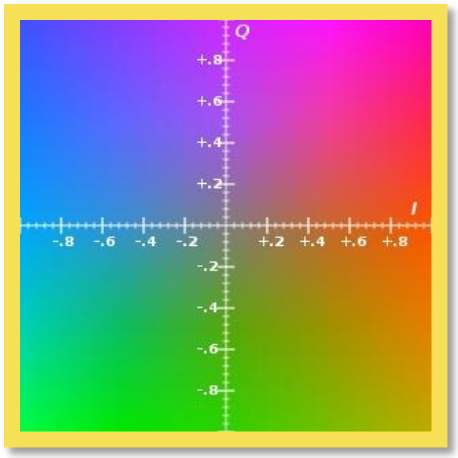


Figura 69 - Espacio YIQ de color

Muy pocos televisores realizan decodificación I-Q, debido a los grandes costes de la implementación. El Rockwell Modular Digital Radio (MDR) fue uno, que en 1997 operó en modo fotograma en tiempo real con un Procesador Rápido IQ (FIQP: Fast IQ Processor).

La representación YIQ se emplea a veces en transformaciones de procesamiento digital de imágenes en color. Por ejemplo, aplicando una ecualización del histograma directamente a los canales en una imagen RGB se alterarían los colores unos en relación con otro, resultando una imagen con colores que no tienen sentido. En vez de ello, si la ecualización del histograma es aplicada al canal Y de la representación YIQ de la imagen, sólo se normalizan los niveles de brillo de la imagen.

7.4 Anexo IV – Operaciones morfológicas

Las operaciones morfológicas aplican un elemento estructural a la imagen de entrada, sin cambiar el tamaño de la imagen de salida. Las operaciones morfológicas más comunes son la dilatación y la erosión. En una operación morfológica, el valor de cada píxel en la imagen de salida depende del valor de ese píxel en la imagen de entrada y su relación con la vecindad. Seleccionando el tamaño y forma de la vecindad (definido a través de un elemento estructural) se puede crear una operación morfológica, que altera el valor del píxel en la imagen de salida.

En este anexo solo se recogerán aquellas operaciones que guarden relación con el proyecto que se muestra, el resto de operaciones no serán contempladas en este documento.

7.4.1 Dilatación y erosión

La dilatación se basa en aumentar el nivel de los valores de los píxeles en el entorno de los objetos presentes en la imagen. La erosión se basa en reducir el nivel de los píxeles del entorno de un objeto. El número de píxeles a los que se aumenta o reduce el nivel depende del tamaño

y forma del elemento estructural usado para procesar la imagen. La dilatación y la erosión, expande y contrae la imagen respectivamente.

Para calcular la dilatación se superpone el píxel central del elemento estructural a cada píxel de la imagen de entrada, entonces el píxel de la imagen de entrada se altera en función de los valores de los píxeles del entorno, definidos por el elemento estructural. El valor del píxel de salida será el máximo entre todos los píxeles presentes en la vecindad.

Un ejemplo de aplicación de un elemento estructural se muestra a continuación:

Entrada → **Elemento Estructural** → **Salida**

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Los pixeles del contorno de la imagen son tratados de forma semejante a los filtros espaciales. En el caso de la dilatación, el valor de los píxeles de relleno, se definen como el mínimo valor que admite el formato de dato para representar los píxeles de la imagen original, con lo que se garantiza que se sustituyan los píxeles del contorno con el máximo valor de la vecindad definida por el elemento estructural.

Como se ha expuesto previamente, el elemento estructural define la forma y el tamaño de la vecindad del píxel que será analizado, para posteriormente alterar su valor. Está definida por una matriz formada por ceros y unos de forma y tamaño arbitrario en la cual las posiciones donde está el uno define la vecindad. La matriz que define el elemento estructural tiene un tamaño muy inferior al tamaño de la matriz original que define la imagen a la que modificará.

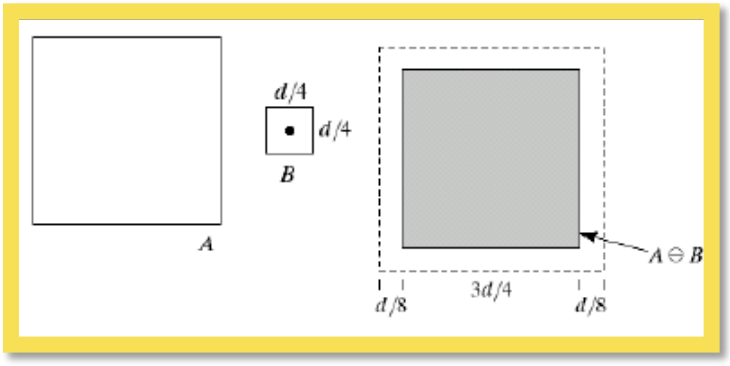


Figura 70 - Ejemplo de Erosión

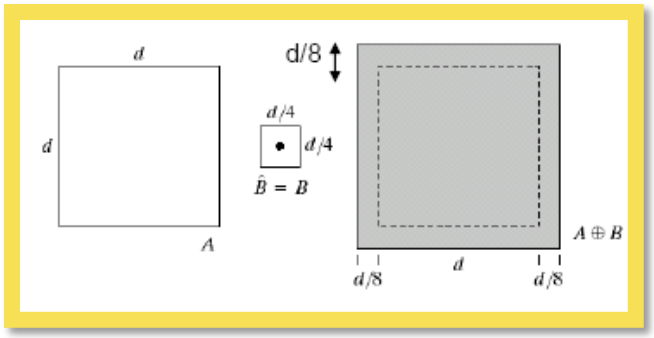


Figura 71 -Ejemplo de dilatación

7.4.2 Apertura y cierre

Las operaciones de dilatación y erosión se combinan para formar diferentes métodos de procesar la imagen. Por ejemplo, uno de ellos es la apertura de una imagen, que es la realización de una erosión seguida de una dilatación utilizando el mismo elemento estructural en ambas operaciones. Este método se aplica cuando se desea eliminar los pequeños objetos y mantener el tamaño en los grandes (eliminar ruído).

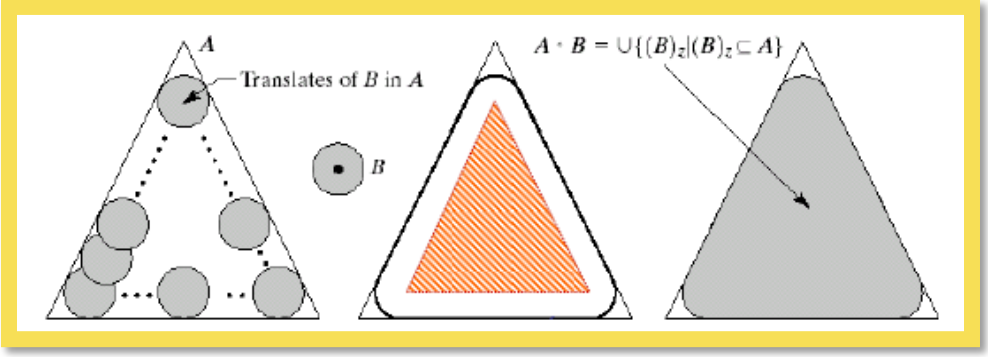


Figura 72 -Ejemplo de Apertura

La operación de cierre de una imagen se realiza cuando se aplica la dilatación y posteriormente la erosión (contrario a la apertura). Este proceso se caracteriza por rellenar huecos y conectar objetos que están próximos entre sí.

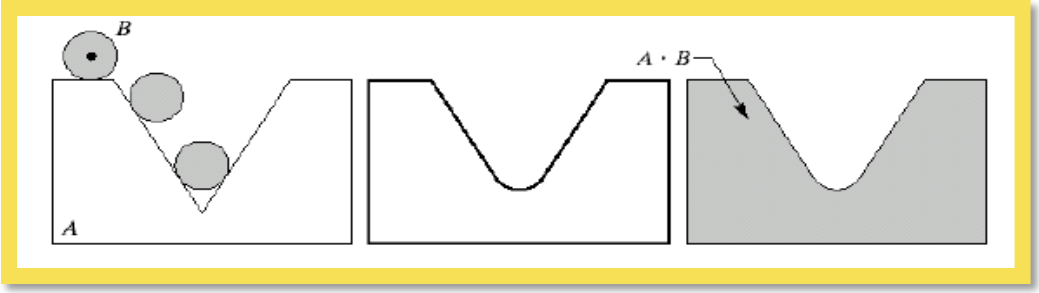


Figura 73 - Ejemplo de cierre

7.5 Anexo V - Redes de Neuronas, Un Visión General

Una red neuronal artificial (RNA) es un modelo computacional inspirado en redes neuronales biológicas que puede ser consideradas como un sistema de procesamiento de información con características como aprendizaje a través de ejemplos adaptabilidad, robustez, capacidad de generalización y tolerancia a fallas.

La RNA puede ser definida como una estructura distribuida, de procesamiento paralelo, formada de neuronas artificiales (llamados también elementos de procesamiento), interconectados por un gran número de conexiones (sinapsis), los cuales son usados para almacenar conocimiento que está disponible para poder ser usado.

7.5.1 Estructura de la Neurona Artificial

Una neurona artificial es una unidad de procesamiento de información de redes neuronales. El modelo de neurona más conocido es de McCulloch-Pitts.

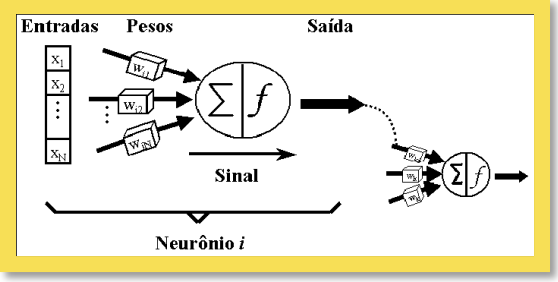


Figura 74 - Estructura de una neurona artificial McCulloch-Pitts

Puede observarse que N señales de entrada son representadas por las variables x_1, x_2 $\cdots x_N$ las cuales están asociadas a pesos que son representados por las variables w_{ij} los cuales determinan el nivel de influencia de la neurona j para la neurona i.

Existen dos estados de procesamiento para cada neurona: suma y activación.

En la primera etapa:

$$y_i = \sum_{j=1}^{N} w_{ij}\, x_j$$

Siendo:

- x_j es la entrada j-ésima
- w_{ij} el peso de la neurona iésima sobre la entrada j-ésima
- y_i es el estado interno de la neurona iésima.

En la segunda etapa, la salida de la neurona es generada a través de la aplicación de una función llamada función de activación.

$$x_i = f(y_j)$$

Siendo:

- x_i es la salida de la neurona iésima
- $f(y_j)$ es la función de activación aplicada al estado interno de la neurona.

La función f tiene como objetivo limitar el nivel de activación de entre [-1 ,1] o [0 1], en el caso de sea un valor continuo y si x_i es discreto entonces limitarlo a {-1,1} ó {0,1}

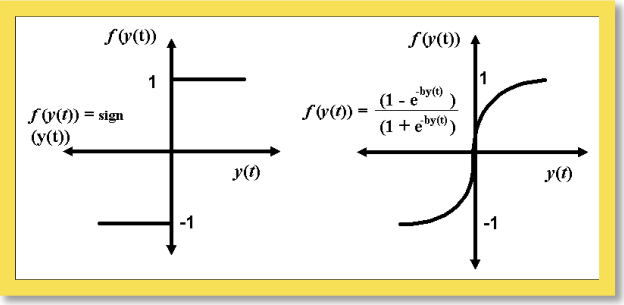


Figura 75 - Las Funciones comunes de activación en redes neuronales

Existen varios tipos de función de activación. La Figura 75 - Las Funciones comunes de activaciónmuestra dos funciones de activación más usadas: la función de grado y la tangente hiperbólica. Como se vio en la primera figura la salida de una neurona puede ser la entrada de otra. Generalmente, una red neuronal se forma por muchas neuronas de alguna forma acoplados.

7.5.2 Arquitectura de Red

La definición de arquitectura es un punto importante en el modelaje de una red neuronal, porque ella restringe un tipo de problema que puede ser tratado. Por ejemplo las redes de una capa. Una red también puede estar formada por múltiples capas, las que pueden ser clasificadas en tres grupos: capa de entrada, capas intermedias u ocultas y capas de salida.

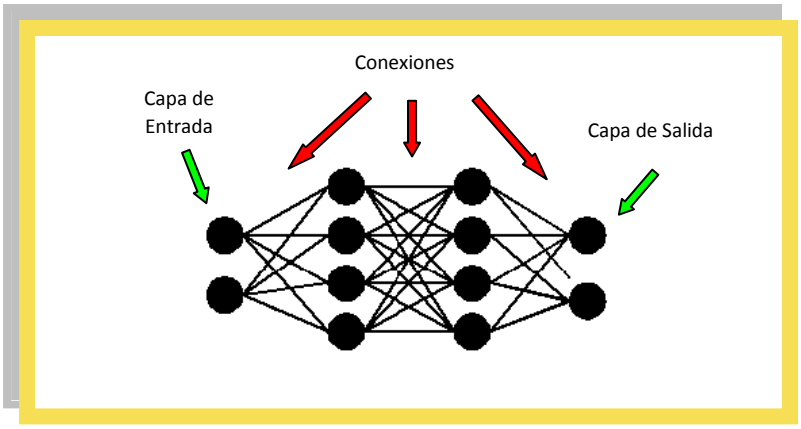


Figura 76 - Arquitectura de Red Neuronal

Basado en flujo de las señales, las redes neuronales también pueden ser clasificadas en dos tipos: *FeedForward* y *Redes Recurrentes*.

7.5.3 Redes Feed-Forward

Como podemos ver la estructura de una red *Feed-Forward* consiste en capas de neuronas donde la salida de una neurona de una capa, alimenta todas las neuronas de la capa siguiente. El aspecto fundamental de esta estructura es que no existen las uniones de retroalimentación. La red *Muiti-Layer Perceptron* (MLP) de un tipo de red *Feed-Forward* (39).

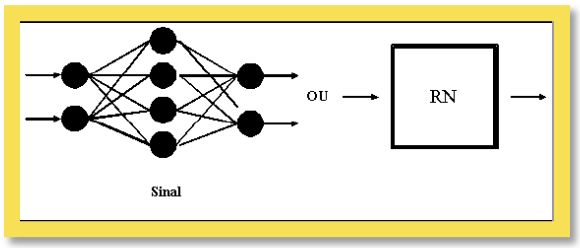


Figura 77 - Esquema de Red Feed-Forward

7.5.4 Redes Recurrentes

Redes recurrentes son aquellas que poseen conexiones de realimentación, como es visto en la figura, las cuales proporcionan un comportamiento dinámico. El modelo de Hopfield es un ejemplo de red neuronal recurrente y será presentado más adelante.

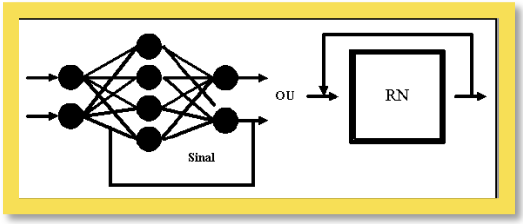


Figura 78 - Esquema de red recurrente

En general los siguientes parámetros son importantes para definir la arquitectura de una red neural: número de capas, número de neuronas en cada capa y tipo de conexión entre dos neuronas, que definen la red de feed-forward o Recurrentes.

7.5.5 Algoritmos de Aprendizaje de un RNA

Una propiedad importante de las redes neuronales es la habilidad de aprender a partir de su ambiente. Eso es realizado a través de un proceso interactivo de ajustes aplicado a sus pesos de conexión entre dos neuronas, denominados entrenamiento. Existen muchos algoritmos de aprendizaje. Cada uno sirve para determinar redes neuronales. Entre los principales se tienen:

7.5.5.1 Aprendizaje por Corrección de Error

Algoritmo muy conocido basado en la regla Delta, que busca minimizar la función de error usando un gradiente descendente. Este es el principio usado en el algoritmo BackPropagation, muy utilizado para el entrenamiento de redes de multiples capas como la Multilayer-Perceptron (MPL) (40).

7.5.5.2 Aprendizaje Competitivo

La cual dos neuronas de una capa compiten entre sí por el privilegio de permanecer activos, tal que una neurona con mayor actividad será el único que participará del proceso de aprendizaje. Es usado en mapas de Kohonen (31) y en redes ART (30).

7.5.5.3 Aprendizaje Hebbiano

Son dos neuronas que están simultáneamente activos a conexiones entre ellos que debe ser fortalecida caso contrario será debilitada (41) utilizada en el Modelo de Hopfield (42).

7.5.5.4 Aprendizaje de Boltzmann

Es una regla de aprendizaje estocástico obtenido a partir de principios de teórico de información y de termodinámica. El objetivo de aprendizaje de Boltzmann es ajustar los pasos de conexión de tal forma que el estado de las unidades visibles satisfaga una distribución de probabilidades deseada en particular.

Otro factor importante es la manera por la cual una red neuronal se relaciona con el ambiente.

A partir de ese concepto existen los siguientes paradigmas de aprendizaje:

- **Aprendizaje Supervisado:** Se utiliza un agente externo que indica a la red la respuesta deseada para el patrón de entrada.
- **Refuerzo:** Es una variante de aprendizaje supervisado a la cual se informa a la red solamente una crítica de corrección de salida de red y no la respuesta correcta en sí.
- **Aprendizaje No Supervisado (auto-organización):** No existe un agente externo indicando la respuesta deseada para los patrones de entrada. Este tipo de aprendizaje es utilizado en los modelos de Mapas de Kohonen (31), redes ART1 (30).

7.6 Anexo VI – Support Vector Machine

Clasificar datos en una tarea muy común en maquinas de aprendizaje. Supóngase que se tiene un conjunto de datos y cada uno pertenece a una de dos clases posibles, siendo el objetivo decidir a qué clase pertenece un nuevo dato y se quiere conocer cómo se pueden separar estos datos con un hiperplano de número de datos menos uno dimensiones. Esto es llamado un clasificador lineal. Existen muchos hiperplanos que pueden clasificar correctamente los datos. No obstante, la máxima separación entre dos clases es usualmente deseada. Así que se debe escoger que hiperplanos ofrecen una mayor distancia entre los datos de distinta clase. Esto clasificadores son denominados clasificadores de máximo margen y es donde encaja la máquina de vectores de soporte.

A continuación se procederá a una formalización del concepto arriba explicado y el funcionamiento en sí, de los SVM.

7.6.1 Formalización

Dado un conjunto datos de entrenamiento, pueden ser expresados como:

$$D = \{(x_i, c_i) | x_i \in \mathbb{R}^p, c_i \in \{-1,1\}\}_{i=1}^{n}$$

Donde:

- D es el conjunto de datos de entrenamiento
- c_i es la clase a la que pertenece el dato
- x_i es el vector real p-dimensional.

Cualquier hiperplano puede ser escrito como un sub-conjunto de los punt*os* x_i satisfaciendo:

$$W \cdot X - b = 0$$

Donde:

- W es el vector normal
- X es el vector muestral
- $\frac{b}{\|w\|}$ es el parámetro que determina el límite del hiperplano desde el origen a lo largo del vector normal.

Por lo tanto se desea escoger un W y un b que maximicen el margen, o la distancia entre los hiperplanos paralelos que estén tan lejos como sea posible para separar los datos. Estos hiperplanos puede ser descritos como:

$$W \cdot X - b = 1$$

$$W \cdot X - b = -1$$

Donde:

- W es el vector normal
- X es el vector muestral
- $\frac{b}{\|w\|}$ es el parámetro que determina el límite del hiperplano desde el origen a lo largo del vector normal.

Nótese que si los datos de entrenamiento son linealmente independientes, se puede seleccionar dos hiperplanos del margen que no contenga puntos entre ellos y entonces se puede intentar maximizar dicha distancia. Mediante el uso de la geometría, se encuentra la distancia entre estos dos hiperplanos como $\frac{2}{\|W\|}$, asi que se desea minimizar $\|W\|$. Para prevenir que los datos estén en el margen, se añade la siguiente condición $\forall i$

$$W \cdot X_i - b \geq 1$$

$$W \cdot X_i - b \leq -1$$

Donde:

- W es el vector normal
- X es el vector muestral

Esto puede ser expresado de otra manera con el fin de juntar ambas ecuaciones y así minimizar $\|W\|$ de manera conjunta.

$$C_i \cdot (W \cdot X_i - b) \geq 1 \; \forall i$$

Donde:

- W es el vector normal
- X es el vector muestral
- C_i es la clase a la que pertenece el punto

Existen diversas formas de expresar esta maximación y expresiones de núcleo, para tener un mayor contacto, se invita a la lectura de (43).

7.7 Anexo VII Herramientas M. del algoritmo propuesto 1

En este apartado se recogerán aquellas herramientas matemáticas de carácter general que se utilizan para el algoritmo 1 – Metodología para segmentaciones no cooperativas.

7.7.1 Fuzzy c – means

Fuzzy c-means es un método de agrupamiento, el cual permite que un mismo segmento de datos pertenezca a mas de una clase. Este método mejorado por Bezdek en (44) es frecuentemente usado en el reconocimiento de patrones. Está basado en la minimización de la siguiente función objetivo:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \cdot \|x_i - c_j\|^2$$

Donde:

- u_{ij} es el grado de pertenencia a la clase del dato iésimo con respecto al centro j-ésimo.
- m es un coeficiente mayor de 1 para cambiar el comportamiento del algoritmo
- x_i es el dato a clasificar
- c_j es el centro de cada clasificación
- J_m es la función a minimizar.

La partición difusa es aproximada a partir de una optimización iterativa de la función objetivo mostrada arriba, con la actualización de la pertenencia u_{ij} y los centros cada grupo c_j mediante:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

Donde:

- u_{ij} es el grado de pertenencia a la clase del dato iésimo con respecto al centro j-ésimo.
- m es un coeficiente mayor de 1 para cambiar el comportamiento del algoritmo
- x_i es el dato a clasificar
- c_j es el centro del grado de pertenencia
- c_k son el resto de centros de clasificación

$$c_j = \frac{\sum_{i=1}^{N} u_{ij} \cdot x_i}{\sum_{i=1}^{N} u_{ij}}$$

Donde:

- u_{ij} es el grado de pertenencia a la clase del dato iésimo con respecto al centro j-ésimo.
- x_i es el dato a clasificar
- c_j es el centro de cada clasificación

Esta iteración terminará cuando $\max_{ij} \{ |{u_{ij}}^{k+1} - {u_{ij}}^{k}| \} < \varepsilon$ siendo épsilon el criterio de término entre 0 y 1 y donde k son pasos de iteración. Este proceso converge a un mínimo local, es un problema que también acarrea el algoritmo *k-means*.

Finalmente el algoritmo está compuesto de los siguientes pasos:

1. Inicialización aleatoria de u_{ij} siendo $u_i = 1.\{u_{ij}\}^0$
2. A cada paso k realizado: Se calculan los centros a partir $\{u_{ij}\}^k$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij} \cdot x_i}{\sum_{i=1}^{N} u_{ij}}$$

3. Actualizar los valores $\{u_{ij}\}^k$ a $\{u_{ij}\}^{k+1}$:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

4. Si $\max_{ij}\{|{u_{ij}}^{k+1} - {u_{ij}}^{k}|\} < \varepsilon$ entonces parar si no, volver al paso 1.

Como ya se ha comentado, los datos son limitados en cada grupo mediante la media de su función de pertenencia, la cual representa el comportamiento difuso de este algoritmo. Para hacer esto, se debe simplemente construir un conjunto de coeficientes de pertenencia tal que estén comprendidos entre 0 y 1, y representen el grado de pertenencia entre los datos y los centros de clasificación.

Los grados de pertenencia, para un correcto funcionamiento, también deben cumplir las siguientes condiciones:

$$u_{ij} \in [0,1] \; \forall i,j \qquad \sum_{j=1}^{C} u_{ij} = 1 \; \forall i \qquad 0 < \sum_{i=1}^{i} u_{ij} < N \; \forall i$$

7.8 Anexo VIII - Herramientas M. del algoritmo propuesto 3

En este apartado se recogerán aquellas herramientas matemáticas de carácter general que se utilizan para el algoritmo 3 – Segmentación de iris en condiciones ruidosas.

7.8.1 Proyección Integral

Sea **A** una imagen de $Width \times Height$ entonces la **integral proyectiva vertical**, denotada por $P_v(A)$, es una tabla de tamaño $Height$ queda definida como:

$$P_v(A) = \frac{1}{height} \sum_{y=0}^{y=height-1} A(x,y)$$

Y la **integral proyectiva horizontal**, denotada por $P_h(A)$, es:

$$P_h(A) = \frac{1}{width} \sum_{x=0}^{x=width-1} A(x,y)$$

7.8.2 Algoritmo *k-means*

Este algoritmo fue introducido en el curso científico por MacQueen en su estudio "Some Methods for classification and Analysis of Multivariate Observations" (45).

En el aprendizaje no supervisado se pretende construir una función **F**(**x**; D) que sea un buen estimador de vector aleatorio $X \in R^p$ a partir de un conjunto de entrenamiento D= {x_i, i=0... N-1}.

Los cuantificadores vectoriales construyen su solución (**F**) de forma local asignando un conjunto infinito de valores (R^p) a un conjunto finito de K valores discretos (proceso de cuantificación), siendo K un parámetro del cuantificador a determinar por el usuario.

Estos valores discretos se denominan vectores prototipo {$\mathbf{w}_i$, i=1...K} y son utilizados de la siguiente manera en un cuantificador vectorial euclidiano **F:**

$$F(x) = \sum_{i=1}^{K} w_i \cdot g_i(x), \qquad con\ g_i(x) = \begin{cases} 1 & x \in R_i \\ 0 & sino \end{cases} y\ R_i = \left\{ x \middle| \|x - w_i\| \leq \min_{j=1..K} \|x - w_j\| \right\}$$

Así **F** aproxima localmente **x** con el vector prototipo más cercano. Para determinar el conjunto de vectores prototipo se suele minimizar el error cuadrático medio definido como:

$$I^{u}_{emp}\left[F(x;D)\right] = \frac{1}{2N}\sum_{i=0}^{N-1}\|x_i - F(x_i)\|^2 = \frac{1}{2N}\sum_{i=0}^{N-1}\sum_{j=0}^{p-1}\left(x_{ij} - F_j(x_i)\right)^2$$

La solución ha dicho problema de minimización, es decir:

$$\left\{w_j^*, j = 1..K\right\} = arg\min_{\{w_j\}} I^{u}_{emp}\left[F(x;D)\right] = \frac{1}{2N}\sum_{i=0}^{N-1}\left\|x_i - arg\min_{\{w_j\}}\|x_i - w_s\|\right\|$$

Resulta ser los centroides calculados a partir de las muestras de entrenamiento que caen en cada R_j, es decir, los estimadores de la esperanza de pertenecer a cada región de Voronoi[20] R_j

$$w_j = \frac{\sum_{s=0}^{N-1} g_j(x_s)\cdot x_s}{\sum_{s=0}^{N-1} g_j(x_s)}$$

Así al minimizar I^{U}_{emp} (que no es sino, una forma de expresar el error cuadrático medio), los K vectores prototipo se distribuyen sobre el espacio de entrada de manera que aproximan de forma discreta la densidad de probabilidad desconocida $p_X(\mathbf{x})$. Donde **x** es denso o disperso, los code-vectores[21] tienden a ser densos o dispersos. La función densidad de probabilidad será bien estimada por ese conjunto de vectores prototipo, y por consiguiente X será bien aproximada, si K es lo suficientemente adecuado para el problema en cuestión.

En el siguiente apartado se verá la versión del algoritmo *k-means* que es el encargado de calcular el conjunto de vectores semilla que minimizan el funcional I^{U}_{emp} o lo que lo mismo, el error cuadrático medio expresado como un funcional.

Supongamos que disponemos de un conjunto de muestras $\{\mathbf{x}_i, i=0...N-1\}$ pertenecientes a un espacio de entrada extraído de un proceso computacional a modelar por el sistema de aprendizaje. El algoritmo K medias en su versión en línea realiza descenso de gradiente en línea sobre I_{Uemp}.

El algoritmo en línea se caracteriza por tomar los valores de manera unitaria, es decir, proceso de manera local cada vector sin tener en consideración el resto. Esto conlleva un aumento en el

[20] Región de Voronoi: Clasificación del espacio euclídeo, también denominado Polígono de Thiessen, extensamente explicado en (49).

[21] Code-vector: es el vector perteneciente al espacio a clasificar que define una clase, también es denominado semilla o centroide.

coste computacional del algoritmo pero provee la capacidad de adaptarse con mayor precisión al entorno a diferencia de consecutivas mejoras producidas sobre este algoritmo.

Este algoritmo consta de los siguientes parámetros y pasos:

Algoritmo *k-means* (C, K, D, α[n], cíclico, rlen)

- C es el conjunto de vectores prototipo o code-vectores
- K es el número de code-vectores
- D es el conjunto de entrenamiento.
- α [n] es la función del tamaño de paso en función del tiempo discreto n asociado a la ecuación iterativa
- cíclico indica si vale 1 que el algoritmo muestrea cíclicamente el conjunto de muestras. Si vale 0,el algoritmo muestra la base de datos de forma secuencial
- rlen es el número de veces que el algoritmo iterativo se va ejecutar

El algoritmo por lo tanto queda:

1. $n = 0$
2. Inicialización del conjunto de code-vectores $\mathrm{C} = \{\mathrm{w_j}[0], \mathrm{j} = 1 \ldots \mathrm{K}\}$
3. $x[n+1] = \begin{cases} x_n \, mod N & si\ cíclico \\ x_n & sino \end{cases}$
4. Calcular el code-vector actual más cercano a $x[n+1], W_w[n]$, utilizando la distancia euclídea:

$$w_w[n] = \underset{w_j, j=1..K}{\arg\min} \left\| x[n+1] - w_j[n] \right\|$$

5. Actualizar únicamente el code-vector ganador de la siguiente manera:

$$w_w[n+1] = w_w[n] + \alpha[n] \cdot (x[n+1] - w_w[n])$$

6. $n = n + 1$
7. Si $n < rlen \Rightarrow$ ir al paso 3 sino ir siguiente.
8. Fin

7.9 Anexo IX - OpenCV (Open Source Computer Vision)

Oficialmente lanzada en 1999, el proyecto OpenCV fue inicialmente una iniciativa del departamento de investigación de Intel para aplicaciones avanzadas y computacionalmente pesadas, parte de las serie del proyecto incluyen trazado y seguimiento en tiempo real y visualización de muros en 3D. La mayor contribución de este proyecto fue incluida por el equipo de realización de librerías Intel, al igual que un gran número de expertos en optimización de Intel con sede rusa. En los primeros tiempos OpenCV se marcó un conjunto de objetivos, estos son:

- Investigación avanzada de visión mediante soporte no solo abierto pero altamente optimizado para ofrecer una básica infraestructura de visión. Se busca no reinventar la rueda, dejar unos cimientos efectivos y de fácil acceso.
- Diseminar el conocimiento de visión mediante el soporte a las infraestructuras más comunes para facilitar la construcción posterior por desarrolladores, de tal manera que el código fuese fácilmente legible y transferible.
- Aplicaciones basadas en visión avanzadas con fines comerciales mediante la portabilidad, ejecución – código abierto y gratuito altamente optimizado – con una licencia que no requiera que aplicaciones comerciales sean a su vez gratuitas si usan esta librería.

La producción de esta librería ha ido ampliando progresivamente el marco de posibilidades que ofrecía dejando un amplio contexto de visión computacional que se invita al lector a profundizar debido a la gran calidad y optimización del producto y su carácter gratuito en (46).

Existen cuatro grandes componentes en esta biblioteca, los cuales aportan a distinto nivel operaciones y funciones de gran interés. Estas son:

- CXCORE – El núcleo de la librería (biblioteca), en ella se contemplan todas las estructuras básicas para su posterior uso. A su vez facilita en un conjunto de operaciones algebraicas sobre matrices de gran eficiencia temporal. Gracias a esto se puede determinar que este componente es una herramienta algebraica de gran potencial y utilidad.
- Librería CV – Esta biblioteca es el grueso de la aplicación aquí se recoge las principales funcionalidades que OpenCV aporta a la visión computacional. Procesamiento de imágenes, análisis estructural, seguimiento de objetos en visión, reconocimiento de patrones y calibración de cámaras para reconstrucciones en 3D.
- Maquina de aprendizaje – Este componente ofrece una herramienta de aprendizaje para la detección y clasificación de características en imágenes, agrupamiento de datos y regresión de datos. Basado en un algoritmo Boost de rápido aprendizaje pero con la necesidad de un gran número de ejemplos.

- Aplicación de alto nivel – Ofrece funciones de alto nivel y una interfaz gráfica para la percepción de las distintas funcionalidades contempladas en el resto de componentes, al igual que visualización de imágenes.

8. Referencias

Estas referencias han sido realizadas mediante la norma Internacional ISO 690 elaborada por el comité técnico de Documentación ISO/TC 46.

1. *Probing the Uniqueness and randomness of iriscodes: results from 200 billion iris pair comparisons.* **Daugman, John.** 2006, IEEE Procedings 94, págs. 1927-1935.

2. *The Human Eye Structure and Functions.* **Oyster, Clyde.** 1999, Sinauer Associates.

3. *Epigenetic randomness, complexity and singularity of human iris patterns.* **John Daugman, Cathryn Downing.** 2001, Proc. R. Soc. Lond. B 268, págs. 1737-1740.

4. *La couleur de l'iris.* **Bertillon, A.** 1885, Rev. Sci 36, págs. 65-73.

5. **Leonard Flom, Aran Safir.** *Iris Recognition System. 641349* USA, 1987. Design.

6. **Daugman, John.** *Biometric personal identification system based on iris analysis. 5291560* USA, Marzo de 1994. Software.

7. *How iris recognition works.* **Daugman, John.** 2003, IEEE TRans. Circ. Syst. Video Technology, Vol. 94, págs. 21-30.

8. *Iris recognition: An emerging biometric technology.* **Wildes, Richard P.** 1997, Proc. IEEE, Vol. 85, págs. 1348–1363.

9. *Image understandin for iris biometrics: A Survey.* **Kevin W. Bowyer, Karen Hollingsworth, Patrick J. Flynn.** 2, San Francisco : Elsevier, 12 de Octubre de 2008, Computer Vision and Image Understanding, Vol. 110, págs. 281-307. ISSN 1077-3142.

10. **Tan, Professor Tieniu.** Center of Biometrics and Security Research. [En línea] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences , Octubre de 1992. http://www.cbsr.ia.ac.cn/IrisDatabase.htm.

11. **Phillips, Jonathon.** NIST ICE Challenge Evaluation. [En línea] Departamento de Seguridad Interna de Estados Unidos, 2006.

12. *UBIRIS: A Noisy Iris Image Database .* **Alexandre, Hugo Proença y Luís A.** Heidelberg : SpringerLink, 18 de noviembre de 2005, Image Analysis and Processing – ICIAP 2005, Vol. 3617, págs. 970-977. ISSN 1611-3349.

13. *New methods in iris recognition.* **Daugman, John.** 5, s.l. : IEEE, 2007, IEEE Trans. Syst. Man Cyber., Vol. 37, págs. 1167–1175.

14. **Scotti, Ruggero Donida Labati y Fabio.** Noisy Iris Segmentation with Boundary Regularization and Reflection Removal. *Image and Vision COmputing.* s.l. : Elsevier, 2009.

15. *Interconversion between truncated cartesian and polar expansions of images.* **Chirikjian, W. Park y G.S.** s.l. : IEEE, IEEE Transactions on Image Processing, Vol. 16, págs. 1946-1955.

16. *High confidence visual recognition of persons by a test of statistical independence.* **Daugman, John.** 1993, Trans. Pattern Analysis and Machine Intelligence,, págs. 1148-1161.

17. **DCMS.** Iris Recognition Research at DCMS. [En línea] DCMS. http://www.irisep.mixdes.org/Eng/database.html.

18. **Bath, University of.** University of Bath Iris Image Database. [En línea] http://www.bath.ac.uk/elec-eng/research/sipg/irisweb/.

19. *Iris recognition, in: Biometric Systems: Technology, Design and Performance Evaluation.* **Wildes, Richard P.** s.l. : Springer-verlag, 2005, págs. 63–95.

20. *An efficient iris recognition system.* **Ya-ping Huang, Si-wei Luo, En-yi Chen.** s.l. : IEEE, 2003. International Conference of Machine Learning and Cybernetics. Vol. 1, págs. 450–454. ISBN: 0-7803-7508-4.

21. *A practical iris acquisition system and a fast edges locating algorithm in iris recognition.* **Yuanning Liu, Senmiao Yuan, Xiaodong Zhu, Qingliang Cui,.** s.l. : IEEE, 2003. Instrumentation and Measurement Technology Conference. Vol. 1, págs. 166–168. ISBN: 0-7803-7705-2.

22. *Iris recognition using collarette boundary localization.* **Hanho Sung, Jaekyung Lim, Ji-hyun Park, Yillbyung Lee.** s.l. : IEEE, 2004. Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. . Vol. 4, págs. 857- 860. ISBN: 0-7695-2128-2.

23. *Experiments with an improved iris segmentation algorithm.* **Xiaomei Liu, Kevin W. Bowyer, Patrick J. Flynn.** s.l. : IEEE, October de 2005, IEEE Workshop on Automatic Identification Technologies,, Vol. 4th, págs. 118–123. ISBN: 0-7695-2475-3.

24. **Masek, Libor.** Recognition of human iris patterns for biometric. *Master's thesis.* s.l. : University of Western Australia, 2003.

25. *The algorithm of iris image preprocessing.* **Mei, Pan Lili Xie.** s.l. : IEEE, Octubre de 2005, Fourth IEEE Workshop on Automatic Identification Advanced Technologies, págs. 134- 138. ISBN: 0-7695-2475-3.

26. **XiaoFu He, PengFei Shi.** A Novel Iris Segmentation Method for Hand-Held Capture Device. [aut. libro] ICB International Conference Proceedings. *Advances in Biometrics.* Hong Kong : Springer Berlin / Heidelberg, 2006, Vol. 3832, págs. 479-485.

27. *Iris Localization with Dual Coarse-to-fine Strategy.* **Xinhua Feng, Chi Fang, Ziaoqing Ding, Youshou Wu.** Hong Kong : s.n., 2006. 18th International Conference on Pattern Recognition. ICPR. . págs. 553-556. ISBN: 0-7695-2521-0.

28. *Fast Iris Detection for Personal Identification Using Modular Neural Networks.* **Al-Braky, H. M.** Sydney : IEEE Xplore, 2001, The IEEE International Symposium on Circuits and Systems ISCAS , Vol. 3, págs. 581-584. ISBN: 0-7803-6685-9.

29. *Fast modular neural nets for human face detection.* **H. M. Al-Bakry, M.A. Abo-Elsoud, M.S. Kamel.** Como : IEEE Xplore, 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN, Vol. 3, págs. 320-324. ISBN: 0-7695-0619-4.

30. **Gail A. Carpenter, Stephen Grossberg.** *Pattern recognition by self-organizing neural networks.* Cambridge : MIT Press, 1991. ISBN 978-0-262-03176-9.

31. **Kohonen, T.** *Self-Organization and Associative Memory.* Finland : Springer, 1988. ISBN:0-387-51387-6.

32. *An Appearance-Based Method for Iris Detection.* **Jiali Cui, Li Ma, Yunhon Wang, Tienu Tan, Zhenan Sun.** Beijing : National Laboratory of Pattern Recognition.

33. **Mäenpää, Topi.** An Iterative Algorithm for Fast Iris Detection. *LNCS.* Heidelberg : Springer Berlin, 2005, Vol. 3781, págs. 127-134.

34. *A knowledge-based approach to the iris segmentation problem.* **Almeida, Pedro de.** Madrid : Elsevier, 2009.

35. *Iris segmentation methodology for non-cooperative recognition.* **Hugo Proença, L.A. Alexandre.** 2, Covilha : IEEE Xplore, 6 de Abril de 2006, IEE Proceedings -Vision, Image and Signal Processing, Vol. 153, págs. 199- 205. ISSN: 1350-245X.

36. *Iris segmentation methodology for non-cooperative recongnition.* **H. Proença, L.A. Alexandre.** s.l. : IEEE, 2006. Vol. 153. ISSN: 1350-245X.

37. *Efficient and robust segmentation of noisy iris images for non-cooperative iris recognition.* **Tieniu Tan, Zhaofeng Hea y Zhenan Suna.** s.l. : Elseveir, 2009, Image and Vision Computing. doi:10.1016/j.imavis.2009.05.008 .

38. *Color gamut transform pairs.* **Smith, Alvy Ray.** New York : ACM, 1978. Proceedings of the 5th annual conference on Computer graphics and interactive techniques. págs. 12-19. ISSN:0097-8930 .

39. **Rumelhart, D.E. y Mcclelland, J.L.** Parallel distributed processing: explorations in the microstructure of cognition. s.l. : MIT Press,Cambridge, MA, Vol. 1.

40. **Freeman James, Skapura David.** *Neural Networks.* Redwood City : Adisson-Wesley, 1991. ISBN:0-201-51376-5.

41. **Hebb, D.O.** *The Organization of Behavior.* New York : Lawrence Erlbaum Associates, 1949. ISBN: 978-0805843002.

42. *Neural networks and physical systems with emergent collective computational abilities.* **Hopfield, John J.** s.l. : National Academy of Sciences, 1982, Vol. 9, págs. 2554-2558. ISBN:0-262-01097-6.

43. *A Tutorial on Support Vector Machines for Pattern Recognition.* **Burges, Christopher J. C.** 2, Hingham : Kluwer Academic Publishers, 1998, Data Mining and Knowledge Discovery , Vol. 2, págs. 121 - 167 . ISSN:1384-5810.

44. *FCM: The fuzzy c-means clustering algorithm.* **James C. Bezdek, Robert Ehrlich, William Full.** 2-3, s.l. : Elsevier, 1984, Computer & Geosciences, Vol. 10, págs. 191-203. ISSN: 0098-3004.

45. *Some Methods for classification and Analysis of Multivariate Observations.* **MacQueen, J. B.** s.l. : University of California Press, 1967, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability., págs. 281–297.

46. **Gary Bradski, Adrian Kaehler.** *Learning OpenCV: Computer Vision with the OpenCV Library.* Safari Books Online. s.l. : O'Reilly Media, Inc., 2008. pág. 555. 0596516134, 9780596516130.

47. *UBIRIS: A noisy iris image database.* **Hugo Proença, Luís A. Alexandre.** Caligari, Italy : 13th International Conference on Image Analysis and Processing, 2005. Lecture Notes in Computer Science - ICIAP . Vol. 1. ISBN 3-540-28869-4.

48. *The Use of Multiple Measurements in Taxonomic Problem.* **Fisher, Ronald A.** s.l. : Cambridge University Press, 1936, Annals Eugen, Vol. 7, págs. 179-188. ISBN: 0-02-844690-9.

49. *Nouveles applications des paramétres continus á la théorie de formas quadratiques.* **Voronoi, G.F.** 1908, J. Reine Angew. Math, págs. 79-178.