



Universidad
Carlos III de Madrid

Departamento de Tecnología Electrónica

PROYECTO FIN DE CARRERA

IMPLEMENTACIÓN EN C# DE UN MÉTODO DE IDENTIFICACIÓN DE IRIS MEDIANTE TÉCNICAS MPEG

Autor: Pau Pérez Audureau

Tutor: Raúl Sánchez Reillo

Director: Inmaculada Tomeo Reyes

Leganés, Marzo de 2013

Título: IMPLEMENTACIÓN EN C# DE UN MÉTODO DE IDENTIFICACIÓN
DE IRIS MEDIANTE TÉCNICAS MPEG

Autor: Pau Pérez Audureau

Tutor: Raúl Sánchez Reillo

Director: Inmaculada Tomeo Reyes

EL TRIBUNAL

Presidente: Juan Vázquez Martínez

Vocal: Fernando García Fernández

Secretario: David R. Sánchez Montero

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 5 de marzo de 2013 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Me gustaría agradecer a los miembros del GUTI su ayuda y colaboración desinteresada, sin la cual no habría sido posible realizar este trabajo. Especialmente a Inmaculada Tomeo Reyes por haberme aconsejado, apoyado y animado en los momentos más difíciles, tanto del desarrollo del proyecto como en la escritura de la memoria y a Iván Rubio Polo que es el autor de la librería de segmentación utilizada y que además me ha brindado valiosos consejos y reconfortado en la toma de decisiones durante el desarrollo de este proyecto. Quiero expresar también mi más profundo agradecimiento a Raúl Sánchez Reíllo por haberme ayudado a redactar esta memoria y hacer cuanto ha estado en su mano para permitirme presentar este proyecto en la mayor brevedad posible.

Índice general

1	: Introducción y objetivos.....	1
1.1	Introducción.....	1
1.2	Objetivos	2
1.3	Fases del desarrollo.....	2
1.4	Estructura de la memoria	3
2	: El iris como modalidad biométrica.....	4
2.1	Anatomía del ojo	4
2.1.1	Globo ocular	4
2.1.1.1	Túnica externa o fibrosa: cornea y esclerótica	5
2.1.1.2	Túnica media o vascular: iris, cuerpo ciliar y coroides	6
2.1.1.3	Túnica interna o neurosensorial: retina	7
2.1.2	Etapas de formación del ojo	9
2.1.2.1	Formación embrionaria	9
2.1.2.2	Formación en el primer mes de vida	10
2.1.2.3	Formación del iris.....	11
2.1.3	El iris.....	11
2.1.3.1	Descripción general	11
2.1.3.2	Coloración y detalles del iris.....	14
2.2	Reconocimiento de Iris.....	15
2.2.1	Introducción	15
2.2.2	Historia	17
2.2.3	Características de los sistemas de reconocimiento biométrico mediante iris ocular	18
2.2.3.1	Universalidad	19

2.2.3.2	Unicidad	19
2.2.3.3	Estabilidad.....	20
2.2.3.4	Facilidad de captura.....	20
2.2.3.5	Rendimiento	21
2.2.3.6	Aceptación por parte de los usuarios	22
2.2.3.7	Robustez frente a la burla del sistema	22
2.2.3.8	Coste.....	23
2.2.4	Etapas de los sistemas de reconocimiento basados en iris.....	24
2.2.4.1	Proceso de adquisición de la muestra	24
2.2.4.2	Preprocesado.....	28
2.2.4.2.1	Control de calidad.....	28
2.2.4.2.2	Segmentación.....	29
2.2.4.2.3	Normalización	30
2.2.4.2.4	Detección y enmascaramiento de oclusiones.....	32
2.2.4.3	Extracción de características	32
2.2.4.4	Comparación de códigos y proceso de decisión.....	33
3	: Lenguaje C#.....	34
3.1.1	Introducción	34
3.1.2	Origen.....	35
3.1.3	Conceptos generales.....	35
3.1.3.1	Common Language Runtime (CLR).....	35
3.1.3.2	Microsoft Intermediate Language (MSIL)	36
3.1.3.3	Librería de Clases Base (BCL).....	37
4	: Método implementado.....	39
4.1	Descripción del algoritmo utilizado.....	39
4.2	Detalles de la Implementación	42
4.2.1	Clases auxiliares implementadas	42
4.2.1.1	Clase Iris	44
4.2.1.2	Clase Resultado.....	46
4.2.1.3	Clase ReticulaMatchEventArgs.....	47
4.2.2	Implementación del algoritmo y sus variantes	47
4.2.2.1	Clase Matcher	48
4.2.2.2	Clase PartialMatcher.....	52
4.2.2.3	Clase ThreadedMatcher.....	53
4.2.3	Aplicaciones auxiliares	55
4.2.3.1	Convertidor.....	55
4.2.3.2	Visor.....	56

4.2.3.3	Calculador	56
4.2.3.4	DemoMatcher.....	57
4.2.3.5	DemoThreaded	59
4.2.3.6	VisorEstadísticos.....	59
5	: Resultados	61
6	: Conclusiones y trabajos futuros	66
	Bibliografía	68
	Anexo A: Detección de párpados en imágenes de iris.	72

Índice de figuras

Figura 1: Partes fundamentales del globo ocular	5
Figura 2: Imagen histológica de la cornea.....	6
Figura 3: Partes de la túnica media o vascular.....	7
Figura 4: Partes del neuroepitelio.....	8
Figura 5: Imagen de la papila, mácula y fovea de la retina	9
Figura 6: Embrión a las ocho semanas. Desarrollo general y detalle del ojo ...	10
Figura 7: El iris como diafragma.....	12
Figura 8: Localización del iris	12
Figura 9: Zonas del iris (1).....	13
Figura 10: Zonas del iris (2).....	13
Figura 11: Rasgos anatómicos del ojo: collarettes y criptas.....	14
Figura 12: Iris de distintos colores.....	15
Figura 13: Dos irises distintos	16
Figura 14: Profesor John G. Daugman.....	17
Figura 15: Aniridia o ausencia de iris	19
Figura 16: Reconocimiento de iris en las instalaciones militares de Fort Belvoir, VA	20
Figura 17: FAR y FRR en función del umbral escogido. La EER es el punto de cruce entre ambas curvas	22
Figura 18: Burla al sistema.....	23
Figura 19: Sistemas de adquisición patentados por Daugman y Wildes	26
Figura 20: Un prototipo del sistema “ <i>Iris on the Move</i> ”	27
Figura 21: Ojo con las fronteras límbica y de la pupila detectadas (imagen superior) e iris normalizado y con el contraste realzado (imagen inferior). La línea roja representa el punto en el que comienza la normalización (izquierda de la imagen inferior) y la flecha el sentido en el que se recorre.....	31
Figura 22: Logo de c# .net	34
Figura 23: Campos generados en la comparación de dos irises del mismo sujeto (superior) y de sujetos distintos (inferior).	41
Figura 24: Ejemplo de aumento de contraste. Imagen original, con el rango dinámico aumentado y con el histograma ecualizado. Histogramas de cada caso.....	43
Figura 25: Diagrama de la clase <i>Iris</i>	44
Figura 26: Diagrama de la clase <i>Resultado</i>	46

Figura 27: Diagrama de la clase <i>Matcher</i>	48
Figura 28: Diagrama de flujo del método <i>Match</i>	51
Figura 29: Diagrama de la clase <i>PartialMatcher</i>	52
Figura 30: Diagrama de la clase <i>ThreadedMatcher</i>	53
Figura 31: Ejemplo del convertidor en funcionamiento.....	55
Figura 32: Ejemplo del visor en ejecución.....	56
Figura 33: Ejemplo del calculador en ejecución.	57
Figura 34: Ejemplo del <i>DemoMatcher</i> en funcionamiento.....	58
Figura 35: Imagen del <i>DemoThreaded</i> indicando que dos irises pertenecen a la misma persona.....	59
Figura 36: Imagen del visor de estadísticos.....	60
Figura 37: Iris con máscara fija.	62
Figura 38: Iris con máscara automática.	62
Figura 39: FRR y FAR en función del umbral seleccionado.....	65

Índice de tablas

Tabla 1: Ventajas e inconvenientes de los sistemas de reconocimiento mediante iris.....	16
Tabla 2: Comparación de los sistemas de adquisición patentados por Daugman y Wildes.....	26
Tabla 3: EER para comparaciones sin máscara.	62
Tabla 4: EER para comparaciones con máscara fija.....	62
Tabla 5: EER para comparaciones con máscara automática.....	63

Acrónimos

BCL	Biblioteca de Clases Base Del inglés <i>Base Class Library</i>
<i>Bl</i>	Longitud de Bloque Del inglés <i>Block Length</i>
CASIA	Academia China de Ciencias, Instituto de Autoamtización Del inglés <i>Chinese Academy of Science, Institute of Automation</i>
CLR	Entorno en tiempo de ejecución de Lenguaje Común Del inglés <i>Common Language Runtime</i>
cm	Centímetro(s)
CPU	Unidad Central de Procesamiento Del inglés <i>Central Processing Unit</i>
DLL	Biblioteca de Enlace Dinámico Del inglés <i>dynamic-link library</i>
DNI	Documento Nacional de Identidad
EER	Tasa de Error Igual Del inglés <i>Equal Error Rate</i>
FAR	Tasa de Falsa Aceptación Del inglés <i>False Acceptance Rate</i>
FRR	Tasa de Falso Rechazo Del inglés <i>False Rejection Rate</i>

FTE	Tasa de Fallo en el Reclutamiento Del inglés <i>Failure To Enroll</i>
GPU	Unidad de Procesamiento Gráfico Del inglés <i>Graphics Processing Unit</i>
GUTI	Grupo Universitario de Tecnologías de la Identificación
IEC	Comisión Electrotécnica Internacional Del inglés <i>International Electrotechnical Commission</i>
IR	Infrarrojo
ISO	Organización Internacional para la estandarización Del inglés <i>International Organization for Standardization</i>
JFK	<i>John Fitzgerald Kennedy</i> (aeropuerto)
JIT	“En el momento” Del inglés <i>Just-in-Time</i>
LED	Diodo Emisor de Luz Del inglés <i>Light-Emitting Diode</i>
LUT	Tabla de búsqueda Del inglés <i>Look up Table</i>
MAD	Diferencia Absoluta Media Del inglés <i>Mean Absolute Difference</i>
mm	Milímetro
MPEG	Grupo de Expertos en Imagen en Movimiento Del inglés <i>Moving Picture Experts Group</i>
MSE	Error cuadrático mínimo Del inglés <i>Minimum Squared Error</i>
MSIL	Lenguaje Intermedio de Microsoft Del inglés <i>Microsoft Intermediate Language</i>
<i>Nh</i>	Vecindario Del inglés <i>Neighborhood</i>

RAM	Memoria de Acceso Aleatorio Del inglés <i>Random Access Memory</i>
RPM	Revoluciones Por Minuto
SAD	Suma de Diferencias Absolutas Del inglés <i>Sum of Absolute Difference</i>
SATA	Conexión Serie de tecnología Avanzada Del inglés <i>Serial Advanced Technology Attachment</i>
SDK	Kit de Desarrollo de Software Del inglés <i>Software Development Kit</i>
std	Desviación típica Del inglés <i>Standard deviation</i>
VA	Virginia (Estado de los Estados Unidos de América)
XML	Lenguaje de Marcas Extensible Del inglés <i>eXtensible Markup Language</i>
XOR	O Exclusivo Del inglés <i>eXclusive Or</i>

1: Introducción y objetivos

1.1 Introducción

Existen numerosas situaciones en la sociedad actual en las que las personas tienen que identificarse fehacientemente. Éstas van desde el acceso a terminales informáticos o información sensible, hasta la verificación de la identidad de una persona por parte de las autoridades, ya sea en un control rutinario o en el acceso a un territorio a través de una frontera.

En el caso de acceso a terminales, los usuarios suelen usar nombres de usuario y contraseñas, mientras que en otros casos se utilizan tarjetas de identificación (como puede ser el DNI). Existen situaciones en las que se recurre a una combinación de ambas técnicas, especialmente si las tarjetas se ven dotadas de medios electrónicos, como ocurre por ejemplo con las tarjetas inteligentes.

Estos métodos, que pueden ser más o menos seguros según su implementación, sufren sin embargo de un problema común: se basan en la posesión por parte del usuario de entes externos a él (contraseñas, tarjetas) que pueden ser extraviados, usurpados o copiados por terceras personas.

Esto hace que exista un creciente interés por parte de diversos sectores de la sociedad, tanto privados como públicos, en crear sistemas de reconocimiento basados en la información biométrica de los usuarios. La gran ventaja de estos sistemas es que, al basarse en características propias y a menudo inalienables de los individuos, no se pueden perder ni robar y dificultan, por tanto, los posibles robos de identidad. No obstante, esta idea no es nueva; no hay que olvidar que la mayoría de los documentos que acreditan la identidad vienen acompañados de una foto, que no es más que un soporte que contiene

información biométrica de su propietario, la cual suele ser cotejada por un ser humano, con los requerimientos de tiempo y posibles errores que esto implica.

No fue hasta el auge y abaratamiento de los sistemas informáticos cuando se empezó a plantear el tratamiento y cotejamiento de información biométrica de forma automatizada. Se exploró en primer lugar el reconocimiento automatizado de individuos mediante el uso de huella dactilar, por razones históricas (ya se empleaban estas técnicas, aunque no de forma automatizada) y por la disponibilidad de enormes archivos de huellas por parte de los gobiernos. Desde hace unas décadas, se han abierto numerosas vías de investigación, basadas por ejemplo, en la estructura facial, de la mano, o, como el caso que se expone en este proyecto, del iris.

1.2 Objetivos

El objetivo principal de este proyecto es implementar un software de reconocimiento de iris basado en el algoritmo desarrollado por D. M. Monro y S. Rakshit en [1], usando el lenguaje de programación de Microsoft, C#.

Para ello, se ha desarrollado una serie de librerías que permiten la representación y comparación de irises y la adquisición de resultados. Además, se ha implementado un conjunto de aplicaciones que permiten el tratamiento automatizado de una gran cantidad de irises y la presentación de los resultados.

Se trata pues, de un sistema de reconocimiento de iris completo (a excepción del proceso de captura) que contiene un conjunto de clases y aplicaciones escritas en C# que permiten la representación y almacenamiento de irises normalizados, su comparación con propósitos de reconocimiento y la explotación e interpretación de los resultados obtenidos durante dicho proceso.

1.3 Fases del desarrollo

El desarrollo de este proyecto se ha realizado atravesando distintas fases, cubriendo cada una de ellas distintos aspectos de su desarrollo:

- Fase 0: Investigación y documentación sobre reconocimiento de iris y tratamiento de imágenes.
- Fase 1: Diseño e implementación de un modelo de datos y aplicaciones auxiliares para el pre-procesado y almacenamiento de datos biométricos (iris).
- Fase 2: Implementación en C# del algoritmo de reconocimiento de iris basado en los trabajos de D. M. Monro and S. Rakshit [1]

- Fase 3: Prueba y obtención de tasas de error y requisitos de tiempo de dicho algoritmo.
- Fase 4: Refinamiento del algoritmo de reconocimiento de iris.
- Fase 5: Implementación a modo de trabajo dirigido de un algoritmo de detección de oclusiones al iris.
- Fase 6: Prueba y obtención de tasas de error y requisitos de tiempo de dicho algoritmo, con las modificaciones aportadas en la fase 4 y la exclusión de los párpados detectados mediante el algoritmo de la fase 5.

1.4 Estructura de la memoria

Los sistemas de reconocimiento de iris presentan cuatro etapas principales, al igual que la mayoría de sistemas de identificación biométrica: captura de los datos biométricos, pre-procesado de los datos capturados, extracción de características propias del usuario, y finalmente, comparación de las características extraídas con un patrón previamente almacenado.

Este proyecto, cuyo objetivo principal es la implementación del algoritmo de extracción de características y comparación, se centrará pues en las últimas etapas, que en este caso, son indisociables.

En este documento, se comienza por hacer un repaso de diversas tecnologías existentes en el campo del reconocimiento de iris, así como de las principales características de la anatomía del ojo (Apartado 2.1). También se presentan brevemente las principales características del lenguaje de programación usado (Capítulo 3).

Además de describir e implementar el mecanismo desarrollado (véase el Capítulo 4), se expondrán los resultados obtenidos tanto en lo referente a tasas de error, como en lo referente a los requerimientos de tiempo (Capítulo 5). Para completar el trabajo, en el Capítulo 6, se indicarán las conclusiones principales obtenidas, así como las líneas futuras a considerar.

2: El iris como modalidad biométrica

En el presente capítulo, se describen y analizan distintas tecnologías de identificación mediante iris. Se hace también una breve descripción de la anatomía del ojo y de las características deseables en un sistema de reconocimiento biométrico.

2.1 Anatomía del ojo

En este apartado se presentan los fundamentos específicos del funcionamiento del iris como órgano en particular, además de conocer su disposición y misión dentro del órgano más complejo en el que se encuentra situado: el globo ocular [2] [3]

2.1.1 Globo ocular

El globo ocular, a través de sus estructuras, recibe los estímulos luminosos externos, los codifica y transmite a través del nervio óptico al cerebro. El ojo descansa sobre una cavidad ósea, en la mitad anterior de la órbita, rodeado de músculos extraoculares, grasa y tejido conectivo. Sólo está expuesta su parte más anterior, y está protegida por el reborde orbitario óseo.

Es un órgano esferoidal, de unos 7,5 g de peso. Su diámetro anteroposterior, transversal y vertical en el adulto normal es de unos 24 mm.; en los hipermétropes es menor, y mayor en los miopes.

El proceso de formación del globo ocular influye significativamente en la formación del iris y sus correspondientes características, que permiten el uso de éste como rasgo biométrico.

Las diferentes capas que componen el globo ocular se encuentran agrupadas en tres túnica: túnica externa o fibrosa, túnica media o vascular y la túnica interna o neurosensorial.

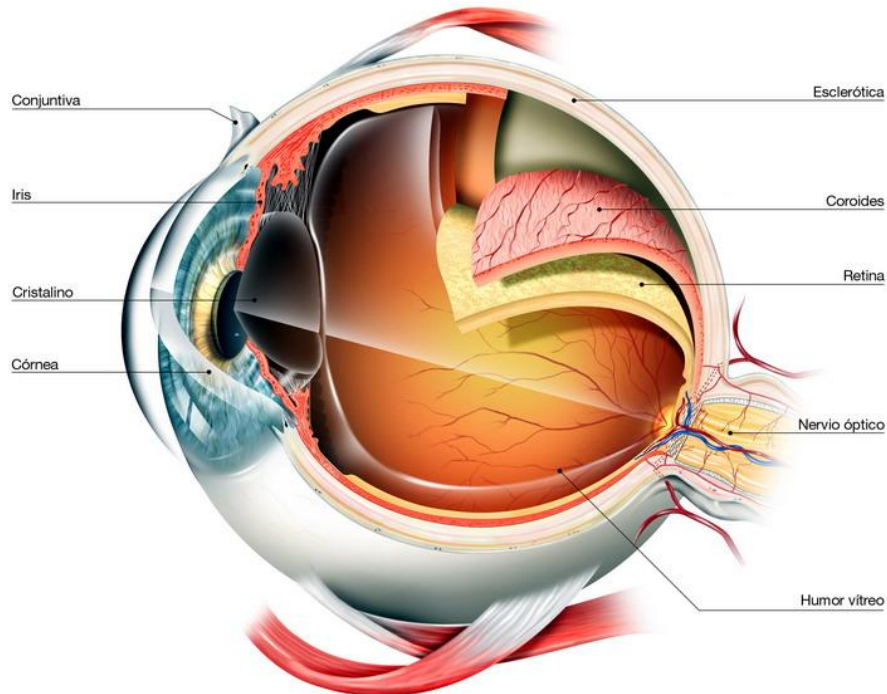


Figura 1: Partes fundamentales del globo ocular

2.1.1.1 Túnica externa o fibrosa: cornea y esclerótica

Esta primera capa es la de mayor consistencia, dándole al globo ocular su forma constante y contribuyendo al mantenimiento de la presión intraocular. Está constituida por la córnea y la esclerótica.

- **Córnea:** La córnea es un tejido transparente, casi invisible, que recubre la capa más exterior del ojo en su zona anterior, siendo la continuación de la esclera, cubriendo a su vez el iris del ojo. Sus funciones son transmitir y refractar la luz que entra en el ojo. Separa el aire con un índice de refracción de 1 y el humor acuoso con un índice de 1.33, constituyendo la principal estructura refractante del ojo, permitiendo enfocar la luz en la retina, comportándose como una lente convergente con un poder refractivo de aproximadamente 42 dioptrías, constituyendo un total de dos tercios del poder refractivo necesario. También actúa como estructura protectora de tejidos y humores intraoculares. Presenta una cara anterior convexa, recubierta constantemente por la película lagrimal, y una cara posterior bañada por el humor acuoso, que constituye la pared anterior de la cámara anterior del ojo. Es ligeramente ovalada con un diámetro medio de 11 a 12 mm, y un

Capítulo 2

grosor de unas 500 μm en su parte central y 670 μm , aproximadamente, en la periferia.

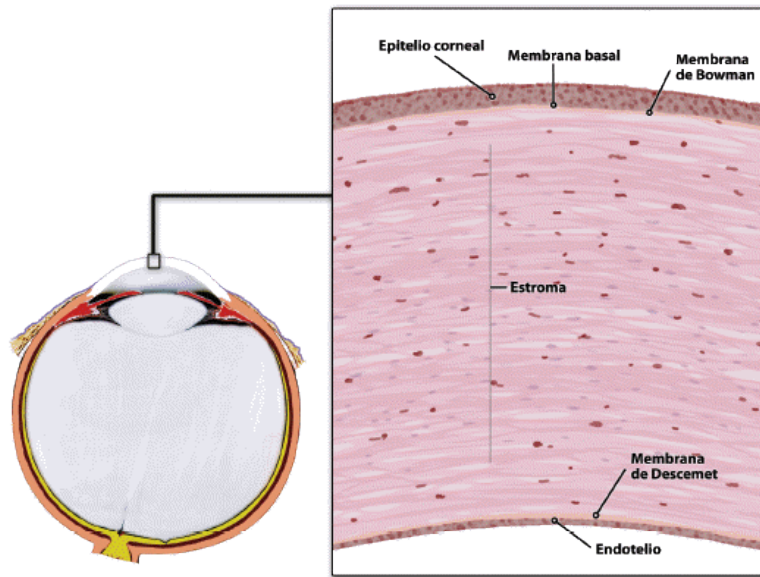


Figura 1. Imagen histológica de córnea normal (200x): epitelio corneal estratificado y membrana de Bowman (A), estroma (B), membrana de Descemet (C) y endotelio (D).

Figura 2: Imagen histológica de la cornea

- **Esclerótica:** La esclera o esclerótica constituye las tres cuartas partes de la superficie del globo ocular. Es de color blanquecino y está formada por haces conjuntivos con fibras elásticas muy resistentes, lo que proporciona consistencia al ojo. Es una membrana fibrosa muy resistente que protege los tejidos intraoculares. Soporta la tensión de los músculos intraoculares y contribuye a mantener la forma y tono muscular.
- **Limbo esclero-corneal:** Se trata de una zona especial de transición entre la esclera y la córnea, y forma la pared externa del ángulo que forman la córnea, el iris y la esclera en la periferia de la cámara anterior, denominado ángulo iridocorneal. A este nivel existen estructuras de drenaje de humor acuoso, además de ser zona de abordaje quirúrgico para determinadas técnicas.

2.1.1.2 Túnica media o vascular: iris, cuerpo ciliar y coroides

Denominada úvea, consta de tres porciones bien diferenciadas: iris, cuerpo ciliar y coroides. Las dos primeras integran la úvea anterior, y la coroides constituye la úvea posterior.

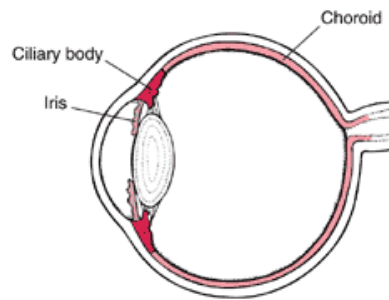


Figura 3: Partes de la túnica media o vascular

- **Iris:** Membrana coloreada y circular del ojo que determina el tamaño de la pupila, regulando la cantidad de luz que alcanza la retina. Una descripción más detallada del iris puede encontrarse en el apartado 2.1.4.4.
- **Cuerpo ciliar:** Es una estructura muscular y fibrosa, situada alrededor del cristalino en forma de anillo, que limita por detrás con la coroides y por delante con el iris. Forma parte de la úvea; está constituido por el músculo ciliar, el músculo de Brücke y el músculo de Müller (fibras circulares). En su estructura se identifican dos porciones: la pars plicata o procesos ciliares (donde se produce el humor acuoso) y la pars plana. Sus funciones son: contraer y dilatar sus fibras musculares, controlando así el espesor del cristalino (músculo ciliar), y segregar el humor acuoso, contribuyendo a su drenaje.
- **Coroides:** La úvea posterior o coroides, está situada entre la retina y la esclerótica, y llega por delante hasta el cuerpo ciliar. Es una membrana conjuntiva que reviste el globo ocular por dentro, y cuya riqueza en células pigmentarias de color negro y brillante le confiere un papel de pantalla a la luz. Su naturaleza vascular la hace fuente primordial de nutrición del ojo y permite una función reguladora de la temperatura ocular. Su función principal es el aporte sanguíneo a las capas más externas de la retina. De la esclerótica la separa un espacio virtual denominado espacio supracoroideo, que termina por delante en la inserción del músculo ciliar en el espolón escleral y por detrás a 4 o 5 mm de la pupila. Por su cara interna, la coroides se adhiere al epitelio pigmentario retiniano a través de la membrana de Bruch, que constituye una capa interna y transparente de la membrana coroides que está en contacto con el epitelio pigmentario de la retina.

2.1.1.3 Túnica interna o neurosensorial: retina

La retina es la capa más interna del globo ocular, y en la que realmente se realiza el proceso de la visión, ya que es tejido neuroepitelial, parte del sistema nervioso central, y está conectada con el cerebro por el nervio óptico.

La retina es una fina cutícula transparente, con un alto consumo de oxígeno, formada por tejido nervioso, que tapiza la cara interna de la membrana coroides y limita su superficie interna con el vítreo. Por delante termina

Capítulo 2

integrada en el cuerpo ciliar a través de la ora serrata. En su parte central y anterior se distinguen mácula y papila del nervio óptico.

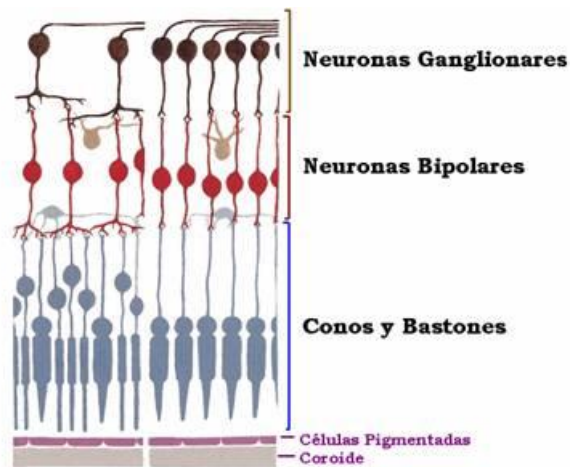


Figura 4: Partes del neuroepitelio

- **Fóvea:** La fóvea es una depresión poco profunda de la retina, situada en el polo posterior del ojo en el centro de la mancha amarilla. Las diversas capas de la retina van adelgazando progresivamente hacia el centro de la fóvea, punto en el que existen solamente las cuatro primeras; es el asiento de la visión central. Es el área de la retina que proporciona la visión de más alta resolución y precisión.
- **Papila:** La papila, o disco óptico, determina la mancha ciega en el campo visual. Es el rasgo más característico al observar el fondo de ojo. Es de color blanco amarillento con una forma redondeada bien definida. Contiene la arteria y vena centrales de la retina. Corresponde al punto de entrada del nervio óptico en la retina y también el punto por el cual entran en el ojo las arterias retinianas y salen del ojo las venas retinianas. Está formada exclusivamente por la capa de fibras nerviosas, los vasos sanguíneos, y carece de sensibilidad visual.
- **Mácula:** La mácula es la parte central de la retina. Se sitúa en el eje visual, donde se encuentra la mayor concentración de conos y la mejor discriminación de la forma y del color.

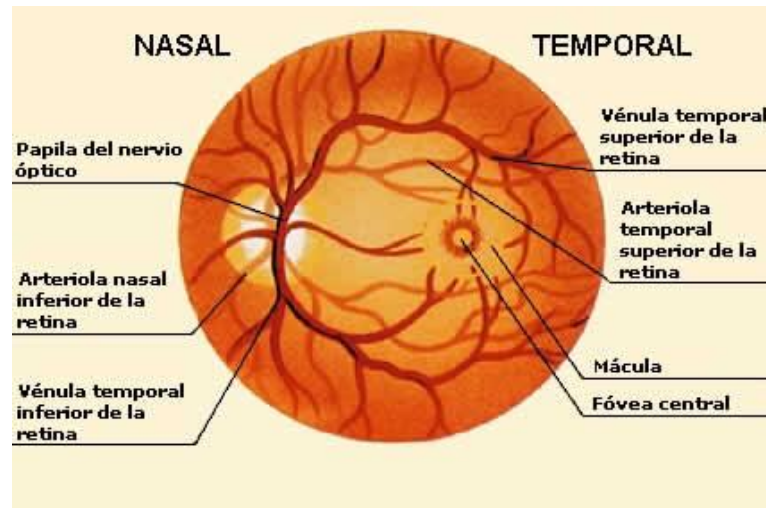


Figura 5: Imagen de la papila, mácula y fóvea de la retina

2.1.2 Etapas de formación del ojo

A continuación se detallan las etapas de formación del ojo, ya que tienen una especial importancia en el proceso de formación de los patrones del iris.

2.1.2.1 Formación embrionaria

Las etapas de formación embrionaria más significativas se detallan a continuación:

- 4 semanas (embrión de 6 mm. y 0,5 gr aproximadamente): Destacan ya las vesículas que darán lugar a los ojos.
- 5 semanas (embrión de 14 mm. aproximadamente): Los ojos comienzan a perfilarse.
- 2º mes (feto de 4 cm. y 5 gramos aproximadamente): Los ojos ya empiezan a destacarse, son unos pequeños huecos en el cráneo (véase Figura 6). Comienza la formación de los párpados, sobre el cristalino de los ojos, que a lo largo de los siguientes días, tomarán su forma ovalada. Los párpados superior e inferior están unidos y cerrados, no los abrirá hasta el séptimo mes.



Figura 6: Embrión a las ocho semanas. Desarrollo general y detalle del ojo

- 5º mes (feto de 25 cm. y 250 gr aproximadamente): Aparecen las cejas y las pestañas, que empiezan a cubrirse de un incipiente lanugo o vello.
- 7º mes (feto de 35 cm. y 1.200 gr aproximadamente): Abre y cierra los ojos ya que los globos oculares están perfectamente formados y en disposición de cumplir sus funciones. Percibe la luz como un débil resplandor rojizo.
- 9º mes (feto de 45/50 cm. y 2.500/3.200 gramos aproximadamente): El proceso de formación anatómica ha terminado, y el dispositivo visual está completamente desarrollado, a excepción de la parte más importante de la retina, que acaba su desarrollo a finales del primer año de vida del niño.

2.1.2.2 Formación en el primer mes de vida

Las distintas fases de formación del ojo que atraviesa un recién nacido son las siguientes:

- En la primera semana:
 - Presenta el reflejo fotomotor: la pupila se contrae con la luz.
 - Parpadea ante los estímulos luminosos con rechazo de la cabeza hacia atrás. El cierre por acercamiento de un objeto tarda unos meses en presentarse.
 - Cierra los párpados al intentar abrírseles.
 - El cierre de los párpados es desigual y no simultáneo hasta alcanzar el primer mes de vida.
 - Son raros los movimientos de parpadeo, que van aumentando hasta alcanzar en la adolescencia 20 por minuto.
- En la segunda semana:
 - Empieza a inmovilizar los ojos, como si ya fijase la visión.

- Desaparece el rechazo de la cabeza hacia los estímulos luminosos.
- En la tercera semana:
 - Se produce un aumento en los movimientos oculares.
 - Aparece el reflejo al miedo o amenaza.
- Al cumplir el mes:
 - Empieza a coordinar los movimientos de cabeza y ojos.
 - Se produce la fijación de la mirada y hacia finales del primer trimestre ya es capaz de seguir con seguridad un objeto en movimiento.

2.1.2.3 Formación del iris

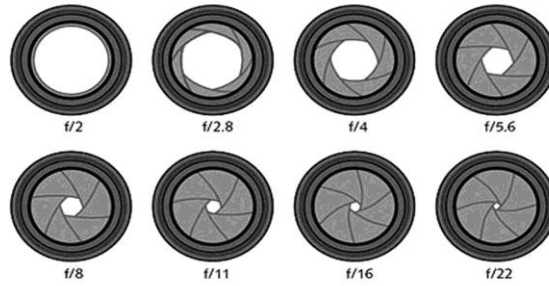
El iris se forma a partir del borde de la copa de fibras ópticas primitiva, que brota durante el tercer mes de vida del embrión. El iris se estabiliza durante el octavo mes de gestación, y se mantiene estable durante la vida completa de una persona, con la salvedad de cambios producidos en la pigmentación, que implican un cambio de color pero en ningún caso estructural. Por esto, tras nacer, es característico el color del iris, que suele ser de un gris azulado, hasta alcanzar su coloración definitiva, durante el primer año, cuando las estructuras que aportan la unicidad característica del iris ya se han terminado de formar, y no cambiarán durante el resto del ciclo vital de la persona. Las estructuras formadas, son aleatorias y no responden a ningún patrón genético, ya que el único factor influido por la genética es la pigmentación del iris, que no influye en el desarrollo del sistema biométrico basado en las características propias del iris.

2.1.3 El iris

En este apartado se detallan las peculiaridades del iris, que permiten, entre otras cosas, su uso como medio de identificación biométrica.

2.1.3.1 Descripción general

El iris, altamente pigmentado cumple el papel de diafragma, variando su apertura en función de la cantidad de luz, para permitir mayor o menor entrada de luz a la retina (véase Figura 7).



Diafragma de una cámara fotográfica según su apertura



Pupila del ojo humano según la luz incide en ella

Figura 7: El iris como diafragma

El iris se localiza en el centro del ojo, limitado por la pupila y el limbo, justo dentro de la concentración circular de las arterias que confluyen en el ojo. Su forma de cono plano truncado se debe a su posición en el cristalino, que tiene forma convexa. No es una estructura simétrica, dado que la pupila se localiza ligeramente hacia la zona nasal respecto al centro de la cornea.

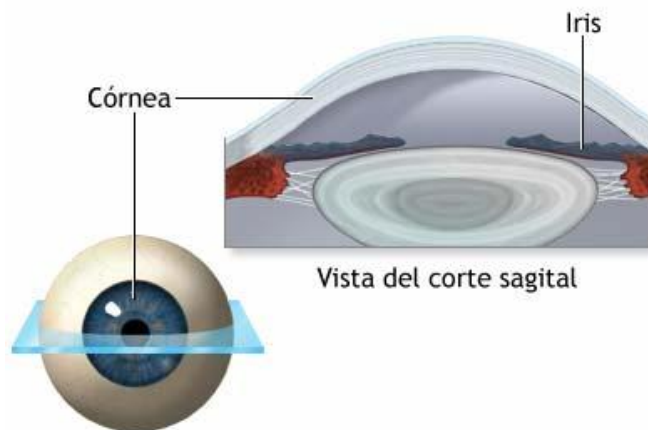


Figura 8: Localización del iris

La pupila constituye un orificio de tamaño variable, cuyo diámetro normal se considera comprendido entre 2 y 4 mm., pudiendo alcanzar 8 mm. en dilatación (midriasis) máxima y 0.5 mm. en contracción (miosis) máxima, controlando así la cantidad de luz que penetra del ojo, actuando a modo de diafragma. La base o raíz del iris se inserta sobre el cuerpo ciliar, siendo aquí mínimo su espesor, lo que explica la especial debilidad ante traumatismos que pueden provocar su desinserción (iridodialisis).

El iris contiene células pigmentadas y musculares y consta de cuatro zonas (véanse Figuras 9 y 10):

- **Membrana limitante anterior:** formada por fibroblastos y melanocitos con forma de estrella.
- **Estroma del iris:** capa de tejido fibroso constituido por colágeno en su mayor parte donde se encuentran fibroblastos ahusados (células del estroma), capilares sanguíneos, nervios y macrófagos pigmentados. Alrededor de la pupila, el estroma termina en el músculo esfínter de la pupila. Los capilares del iris suelen estar radialmente dispuestos y disponen de numerosos canales anastomósicos.

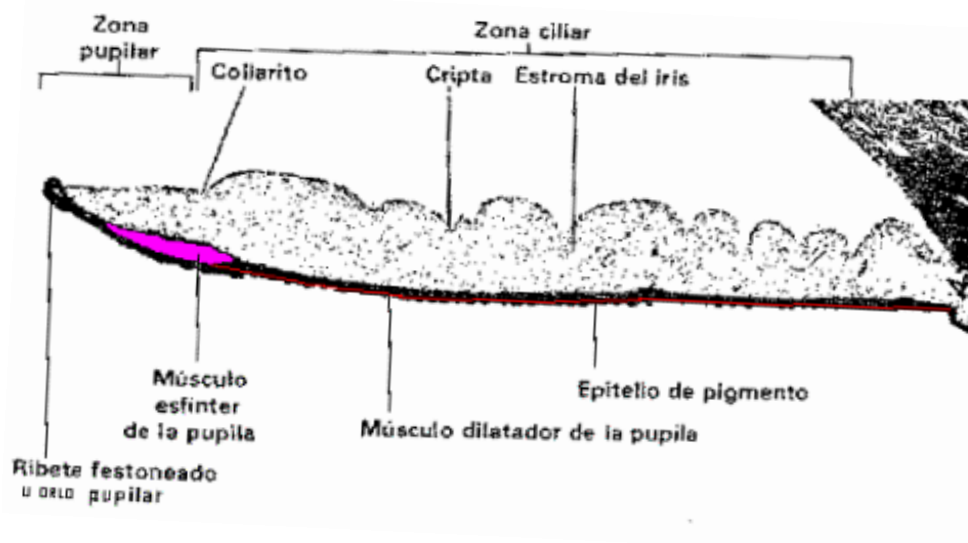


Figura 9: Zonas del iris (1)

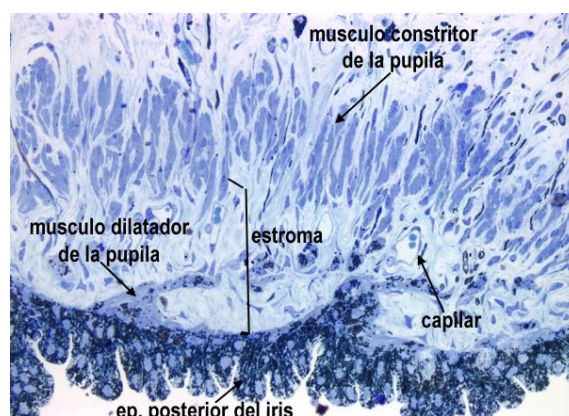


Figura 10: Zonas del iris (2)

- **Músculo dilatador del iris (Membrana de Bruch):** se extiende desde la base del iris hasta el esfínter de la pupila. Es responsable junto con el esfínter del movimiento del iris.

- **Epitelio posterior pigmentado:** compuesto de dos capas de células pigmentadas con melanina. La melanina tiene por objeto absorber luz y reducir la distorsión óptica.

La función principal del iris es la de regular la cantidad de luz que penetra en el ojo, ajustando el tamaño de la pupila. Cuando está completamente contraído, tiene aproximadamente un 13%, de su tamaño normal en relajación.

A pesar de que los detalles del patrón del iris varían de persona a persona, algunos rasgos anatómicos son comunes (véase Figura 11). Por ejemplo, en la zona media del iris aparece una región en zigzag, larga y circular, denominada collarete, un reborde circular que marca el sitio del círculo vascular menor del iris. Dispuesto alrededor del radio circular del iris y combinándose con el collarete, se encuentran a menudo dispuestas diversas crestas ligeramente elevadas. También presenta algunas áreas oscuras de estructura oval, y de relativamente profundo relieve, que se denominan criptas de Fuchs, y que se forman en la red de colágeno del estroma, que sostiene la estructura global del iris.

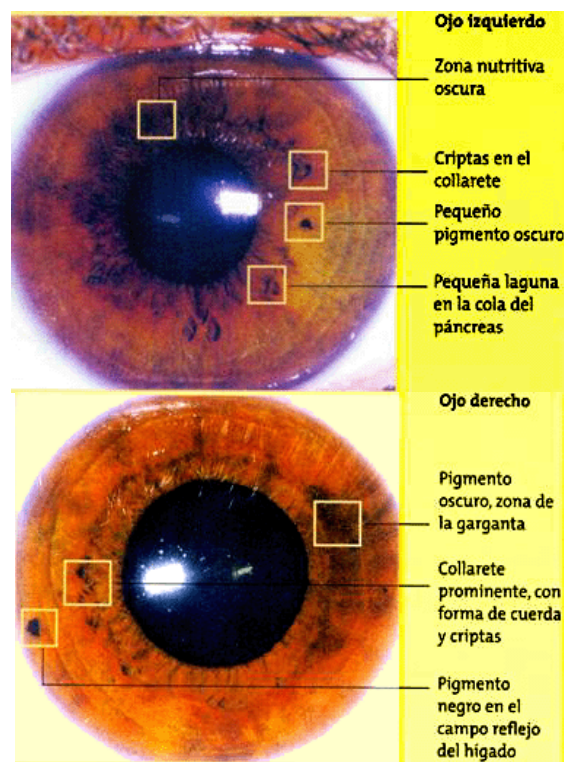


Figura 11: Rasgos anatómicos del ojo: collaretes y criptas

2.1.3.2 Coloración y detalles del iris

La coloración del iris depende de la cantidad de melanina presente. Si es escasa, la reflexión del pigmento del epitelio pigmentario produce un color azulado. En iris de color azulado se pueden observar contracciones

concéntricas en su superficie que adopta una forma irregular y con criptas. Por otro lado, si la cantidad de melanina es moderada, el iris es de color avellana, mientras que si la cantidad de melanina es grande el iris tendrá un color marrón o incluso negro. La pigmentación disminuye con la edad. En los casos de albinismo, los fibroblastos pigmentados carecen de melanina y el iris aparece de color rojo debido a los capilares sanguíneos.



Figura 12: Iris de distintos colores

Mientras que la anatomía general, fisiología y color del iris se determinan de manera genética, los detalles más finos se desarrollan de manera totalmente aleatoria durante el periodo de gestación. Por ejemplo, las manchas oscuras que aparecen en algunos irises de color verde, no aparecen duplicadas en el otro ojo. A pesar de que en algunos casos el grado elevado de pigmentación de los ojos marrones no permite distinguir a simple vista una estructura muy detallada, mediante el uso de iluminación infrarroja es posible desvelar estructuras ricas en detalles.

2.2 Reconocimiento de Iris

2.2.1 Introducción

El reconocimiento basado en iris ocular ha despertado un creciente interés en los últimos años, debido a su alta precisión y al interés despertado en ciertos sectores que requieren de una técnica de identificación fiable, como por ejemplo la banca.

Los patrones existentes en el iris son considerados una de las técnicas de identificación biométrica más fiables, pues el iris es único para cada persona y no cambia durante toda la vida. A diferencia de las huellas dactilares, que se

Capítulo 2

encuentran expuestas a agresiones externas como cortes o quemaduras, o las estructuras óseas (pómulos, nariz, arco ciliar, etc.), que se usan para el reconocimiento facial y que se encuentran expuestas a deformaciones por golpes o cirugía, el iris se encuentra muy protegido.



Figura 13: Dos irises distintos

Daugman sugiere en [4] que la probabilidad de encontrar dos irises iguales es de 1 entre 7000 millones, aunque algunos estudios afirman que esta probabilidad es tan baja como 1 entre 10^{78} . Según diversos estudios ([5], [6]), en el patrón del iris hay gran cantidad de información que permite identificar unívocamente a una persona. Aunque la pigmentación y la estructura del iris están genéticamente ligadas, los detalles de los patrones no lo están ya que se desarrollan de forma aleatoria a la par que el feto.

El iris se desarrolla durante el crecimiento prenatal. Antes del nacimiento, se produce la degeneración, que da como resultado la apertura de la pupila y los patrones únicos del iris [7]. Prueba de ello es que, aunque genéticamente idénticos, los irises de un individuo (izquierdo y derecho) son únicos y estructuralmente distintos, lo que permite utilizar el iris para propósitos de reconocimiento. Dada la alta unicidad del iris, resulta posible obtener tasas de error cuasi nulas. Otra característica que refuerza el interés de estas técnicas es que la información necesaria se obtiene de forma no invasiva.

De forma esquemática, los sistemas de reconocimiento biométrico basados en iris presentan las siguientes ventajas e inconvenientes:

Tabla 1: Ventajas e inconvenientes de los sistemas de reconocimiento mediante iris

VENTAJAS	INCONVENIENTES
Sistema no invasivo. El usuario debe situarse frente al sistema de captura pero no es necesario contacto alguno.	Requiere de la colaboración del usuario en la fase de captura.
El sistema es invariable ante el uso de gafas o lentes de contacto (siempre que no contengan inscripciones o impresiones que ocluyan el iris), pues estos elementos no modifican la estructura de ojo.	Se requieren determinadas condiciones de iluminación en la fase de captura de la imagen del iris para conseguir una muestra válida.
Presentan elevadas tasas de reconocimiento (unicidad).	Existen enfermedades que pueden afectar negativamente al reconocimiento de las características del iris.

<p>Tasas de error espectacularmente bajas, fundamentalmente Tasa de Falsa Aceptación (o FAR, del inglés False Acceptance Rate) tan baja como se desee.</p>	<p>Elevado coste (fundamentalmente de los medios de captura, aunque se están abaratando drásticamente).</p>
<p>Gran estabilidad debida a la protección que ofrece la córnea.</p>	<p>Susceptible a la calidad de la toma en la fase de captura y a la compresión de las fotografías.</p>
<p>Robustez frente a la burla del sistema. Principalmente gracias a las técnicas de reconocimiento de sujeto vivo y a la dificultad para modificar la estructura del iris.</p>	<p>Coste computacional de moderado a alto en función de la técnica empleada.</p>

2.2.2 Historia

La idea de utilizar el iris para reconocer a una persona fue originalmente propuesta por el oftalmólogo Frank Burch en 1936. En 1985, los oftalmólogos Leonard Flom y Aran Safir pusieron de manifiesto la unicidad de los patrones del iris, aunque hasta 1987 no se les concedió la patente [8] para el concepto de reconocimiento mediante iris [14]

El profesor John G. Daugman se encargó de desarrollar los algoritmos necesarios para realizar el reconocimiento automático de individuos a partir del patrón del iris. Los trabajos realizados por Daugman, que se recogen parcialmente en [6] supusieron el nacimiento definitivo de la identificación basada en patrones de iris. Dichos algoritmos fueron patentados en 1994 [8], sentando las bases de los actuales sistemas y productos comerciales de reconocimiento de iris.



Figura 14: Profesor John G. Daugman

Posteriormente, Aran Safir y Leonard Flom, fundaron conjuntamente con John Daugman la compañía *Iris Corporation*, propietaria de la patente y encargada de distribuir las licencias a las diferentes compañías desarrolladoras de sistemas de reconocimiento. Una de estas empresas fue *Sensar Corporation*, que creó una cámara especial, capaz de obtener imágenes de iris en los cajeros automáticos. Estas dos compañías (*Iriscan Corp.* y *Sensar Corp.*) se unieron creando *Iridian Technologies*.

Agencias relacionadas con los cuerpos de seguridad y justicia de los Estados Unidos comenzaron a utilizar este tipo de dispositivos en 1994, siendo la prisión del estado de Lancaster (Pennsylvania), la pionera en el uso de esta tecnología para la identificación de sus reclusos. También se puede encontrar esta tecnología en entornos más enfocados al público general, como por ejemplo en el aeropuerto internacional Charlotte/Douglas en Carolina del Norte o en el aeropuerto alemán de Frankfurt, donde se permite a los viajeros habituales registrar su patrón de iris con para reducir el tiempo necesario en el proceso de embarque. También en el campo de la seguridad aeroportuaria, tenemos los ejemplos de las instalaciones de los aeropuertos Schiphol en Ámsterdam, JFK en Nueva York o Heathrow en Londres, donde estos sistemas son empleados tanto para realizar el embarque de viajeros, como en los controles de pasaportes e inmigración. Últimamente se puede encontrar también esta tecnología en los controles de aduanas de algunos aeropuertos de Israel o Emiratos Árabes Unidos.

Una de las aplicaciones relacionadas con los patrones de iris que está experimentando un desarrollo más exhaustivo, es la utilización de dispositivos de reconocimiento automático mediante iris en los cajeros automáticos, o más generalmente en aplicaciones relacionadas con la banca. El auge de los *smartphones*, que combinan potencia de cálculo con cámaras de alta definición, está generando un gran interés en el control de acceso a los sistemas de banca a distancia a través del iris. Un ejemplo de ello es la colaboración de la empresa española *Mobeel* con *Bankinter*, publicado en [10]. También se considera muy interesante la utilización de estos sistemas de reconocimiento para el control de acceso y seguridad en entornos domésticos. Entre las empresas que desarrollan este tipo de dispositivos y sistemas pueden destacarse Panasonic, LG, Oki, British Telecom, Unisys, Siemens o IBM.

2.2.3 Características de los sistemas de reconocimiento biométrico mediante iris ocular

A continuación se realiza un repaso de las características deseables de un sistema de reconocimiento biométrico, relacionándolas con las prestaciones de los sistemas basados en iris.

2.2.3.1 Universalidad

Este factor mide la porción de población que es susceptible de ser reconocida mediante un sistema concreto de reconocimiento biométrico.

La población que tiene la posibilidad de hacer uso de estos sistemas es muy amplia, dado que muy pocas personas carecen de ambos ojos a la vez, y que aquellas personas que padecen aniridia o ausencia de iris (véase Figura 15), la suelen padecer en un único ojo. Basta con tener uno de los dos iris en buen estado para poder llevar a cabo tanto el proceso de reclutamiento como el de reconocimiento.



Figura 15: Aniridia o ausencia de iris

Las personas con problemas de ceguera pueden en muchos casos hacer uso de estos sistemas, aunque en otros casos la oclusión total del iris hace imposible el reclutamiento. Por otro lado, las personas mayores desarrollan en ocasiones patologías oculares que ocluyen parcialmente el iris, algo que debe ser tomado en cuenta.

Se puede considerar por tanto, que la universalidad de los sistemas de reconocimiento biométrico basados en patrones de iris es alta, aunque no absoluta.

2.2.3.2 Unicidad

Esta es una de las principales características que debe cumplir un sistema de reconocimiento biométrico. La unicidad evalúa la probabilidad de que dos individuos posean las mismas características, en este caso concreto, que dos personas tengan un mismo patrón de iris. Tal y como se ha visto anteriormente en la sección 2.2.1, esta probabilidad es muy baja, lo que hace del iris un rasgo muy apropiado para llevar a cabo procesos de reconocimiento biométrico.

Otro aspecto relacionado con la unicidad de los patrones de iris es la medida del grado de correlación entre los iris de ojos genéticamente idénticos. Una fuente de iris genéticamente idénticos serían los ojos derecho e izquierdo de una misma persona. Estos dos iris tendrán la misma relación genética, como la tendrán los cuatro patrones de iris que se pueden obtener de gemelos

monocigóticos. En el estudio de John Daugman y Cathryn Downing detallado en [11] se indica que los detalles de los patrones de iris obtenidos de ojos genéticamente idénticos son incorrelados. Esto significa que el grado de independencia de los ojos genéticamente idénticos es similar al presente entre los ojos de individuos sin ningún tipo de relación genética. Este factor de alto grado de unicidad, deberá ser considerado a la hora de desarrollar y utilizar el sistema de reconocimiento, ya que la captura de la imagen del iris deberá realizarse para uno de los dos ojos de un usuario de manera específica, al no ser los patrones de los ojos intercambiables.

La textura del iris es considerada la característica fenotípica visible en el rostro de una persona que presenta una mayor unicidad. Como resultado del elevado nivel de unicidad que presentan las características extraídas de una imagen de iris, este tipo de sistemas presentan un grado de fiabilidad muy alto.

2.2.3.3 Estabilidad

La estabilidad mide la posibilidad de seguir reconociendo a un individuo a lo largo del tiempo. El desarrollo y aparición de las estructuras que configurarán el patrón del iris, que queda determinado durante la última etapa del desarrollo embrionario, constituye una estructura estable a lo largo de la vida, si no tenemos en cuenta posibles accidentes u operaciones quirúrgicas. Otra característica que proporciona una gran estabilidad al iris, es la protección y el aislamiento frente a agentes externos, gracias a su situación tras la córnea y el humor acuoso.

2.2.3.4 Facilidad de captura

La captura del patrón de iris es un proceso relativamente sencillo, en cuanto a las acciones requeridas por parte del usuario. Esta captura se realiza de forma no invasiva utilizando para ello una cámara de alta resolución, con una iluminación cercana al infrarrojo, aunque también se puede llevar a cabo con una cámara fotográfica convencional, una cámara fotográfica digital, o incluso mediante capturas de una cámara de vídeo. El gran avance que se está experimentado en la captura de imágenes digitales ha permitido abaratar en gran medida los sistemas de captura.



Figura 16: Reconocimiento de iris en las instalaciones militares de Fort Belvoir, VA

2.2.3.5 Rendimiento

El rendimiento de un sistema de reconocimiento biométrico viene definido por las tasas de error. A la hora de evaluar este aspecto de un sistema, se tienen en cuenta la FAR (*False Acceptance rate*), FRR (*False Reject Rate*), FTE (*Failure To Enroll rate*) y EER (*Equal Error Rate*). La tasa de falsa aceptación o FAR mide la tasa de individuos que consiguen identificarse bajo una identidad que no es la suya. La tasa de falso rechazo o FRR es la tasa de usuarios que no consiguen identificarse bajo su propia identidad. La tasa de error de reclutamiento o FTE mide la proporción de individuos que no consiguen ser dados de alta en el sistema. Esto puede deberse a un problema de universalidad o bien a un fallo del operario que maneja el sistema de reclutamiento. Finalmente, la tasa de igual error o EER es el punto en el que la FAR y la FRR son iguales.

Dado que los sistemas de reconocimiento biométricos se basan en la comparación de datos obtenidos en condiciones y momentos distintos (reclutamiento y posteriormente reconocimiento), no se debe esperar que éstos sean exactamente iguales. Por ello, en la mayoría de los casos, se recurre a una medida de similitud de los datos. Habrá que definir, por tanto, un umbral de similitud por encima del cual se considere que los datos son lo suficientemente parecidos como para pertenecer a la misma persona. Este umbral es el denominado umbral de decisión.

El valor de las tasas de error variará fundamentalmente en base al umbral de decisión definido en el módulo de comparación de muestras, que puede ser modificado de forma que se optimice el rendimiento del sistema. Hay que tener en cuenta que la FRR y FAR son complementarias, ya que para un sistema dado, si se disminuye la FRR para volver el sistema “más permisivo”, se produce un aumento en la FAR. Lo mismo ocurriría a la inversa. Por otro lado, este hecho asegura que exista una EER, ya que si la FAR es creciente en función del parámetro que se compara con un umbral, la FRR será decreciente (véase Figura 17).

Cabe destacar que la utilización de cada una de estas características de manera independiente para evaluar el desarrollo del sistema resultaría inútil. El objetivo buscado es encontrar unos criterios de decisión que permitan maximizar el rendimiento del sistema, reduciendo al mínimo e incluso buscando unas probabilidades de error nulas.

Los sistemas de reconocimiento de iris presentan niveles de rendimiento muy elevados, pudiendo obtenerse tasas de falsa aceptación muy bajas, o incluso nulas, para valores de falso rechazo reducidos.

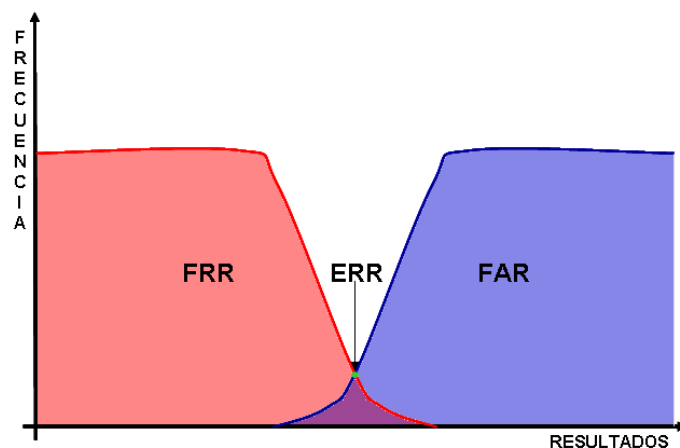


Figura 17: FAR y FRR en función del umbral escogido. La EER es el punto de cruce entre ambas curvas

2.2.3.6 Aceptación por parte de los usuarios

Este factor es un factor clave para poder implantar con éxito un sistema de reconocimiento biométrico. En la mayoría de los casos, resulta necesaria la colaboración del usuario a la hora de utilizar el sistema, o que al menos éste no tome medidas para ocultar las características que se pretenden medir.

En un principio, los usuarios de sistemas basados en iris mostraban algunas dudas en lo referente a su utilización, debido a las reticencias que suelen presentar las personas ante aquellas circunstancias relacionadas con sus ojos. Sin embargo, el desarrollo de nuevos dispositivos de captura, así como la forma no intrusiva en la que ésta se realiza (existe cierta distancia entre el sujeto y las cámaras utilizadas para tomar la imagen del iris), han producido una creciente aceptación a la hora de utilizar estos sistemas de reconocimiento.

En cualquier caso, la mejor manera de aumentar la aceptación consiste en explicar a los usuarios que estos sistemas no son en ningún caso nocivos para ellos.

2.2.3.7 Robustez frente a la burla del sistema

La robustez que presentan los sistemas de reconocimiento de iris frente a posibles ataques en los que se intenta falsear la muestra de entrada es muy elevada.

Los primeros sistemas comerciales de reconocimiento de iris podían ser engañados fácilmente mediante el uso de fotografías de alta calidad de un iris impresas en papel.



Figura 18: Burla al sistema

A medida que la tecnología ha ido madurando, se han desarrollado mecanismos que permiten diferenciar un iris genuino de otro falso. Un ejemplo de ello sería el estudio de la variación del diámetro de la pupila experimentado frente a diferentes condiciones de iluminación en imágenes tomadas consecutivamente. Este estudio puede realizarse fácilmente mediante el uso de cámaras que permitan variar dinámicamente el nivel de luz de la imagen. Existe también un movimiento ocular llamado *hippus* que consiste en leves contracciones de la pupila, incluso ante niveles de iluminación constantes, y que permite la detección de sujeto vivo sin que sea necesaria la modificación de las condiciones de captura de la imagen. Las variaciones en el tamaño de la pupila debido al *hippus* varían desde 0.3 mm hasta 1.5mm.

Por otro lado, la modificación de la estructura del iris mediante cirugía, resulta muy arriesgada, pudiendo provocar daños permanentes en la visión del sujeto sometido a la operación, por lo que resulta inverosímil que se use ésta técnica para falsificar iris.

2.2.3.8 Coste

Este es, sin lugar a duda, uno de los principales inconvenientes de este tipo de sistemas. Sin embargo, la reducción en el precio de las cámaras fotográficas y de video digitales en la actualidad, está permitiendo que los costes necesarios para el desarrollo e implementación de dispositivos y sistemas para el reconocimiento de iris sean cada vez más competitivos.

2.2.4 Etapas de los sistemas de reconocimiento basados en iris

Los sistemas de reconocimiento de iris presentan cuatro etapas principales, al igual que la mayoría de sistemas de identificación biométrica: captura de los datos biométricos, preprocesado de los datos capturados, extracción de características propias del usuario, y finalmente, comparación de las características extraídas con un patrón previamente almacenado [12],[13]

La primera etapa consiste en la adquisición de la imagen del iris. Posteriormente se lleva a cabo el preprocesado de la imagen, en el que se localiza y aísla la estructura del iris, detectando sus límites exterior e interior (frontera con la esclerótica y la pupila respectivamente) y posibles oclusiones debidas a la presencia de párpados o pestañas. También en esta etapa se procede a la normalización del tamaño de la imagen, ya que diferentes distancias entre el ojo y la cámara en las distintas capturas de la muestra de entrada, pueden resultar en diferentes tamaños de imagen para el ojo de un mismo individuo.

El paso siguiente consiste en realizar la extracción de características de la imagen del iris ya preprocesada, con el objetivo de obtener un vector de características o *IrisCode* [13] , que posteriormente será comparado con el patrón almacenado en la fase de reclutamiento.

El proceso de reconocimiento finalizará con la etapa de comparación, en la que se pueden aplicar diferentes algoritmos, como por ejemplo los basados en determinar la mínima distancia entre el patrón almacenado y los *IrisCode* obtenidos en las capturas realizadas en cada uno de los accesos al sistema.

2.2.4.1 Proceso de adquisición de la muestra

El proceso de reconocimiento mediante iris ocular comienza con la captura de una imagen del ojo del sujeto a reconocer.

La captura de la imagen del iris del individuo no es un proceso sencillo, debido al reducido tamaño de éste. Durante la captura se deben tener en cuenta algunos aspectos importantes, como la distancia entre el sujeto y el dispositivo de captura, o las condiciones de iluminación existentes. Una mala iluminación o una distancia incorrecta entre la persona a identificar y la cámara, pueden derivar en la captura de una imagen de iris con una calidad insuficiente para realizar el resto de pasos del proceso de reconocimiento.

Al contrario de lo que se puede pensar, no es necesario proceder a un escaneo del ojo por medio de láseres. Las primeras aplicaciones usaban cámaras que tomaban fotografías en color en el espectro visible. Esto presentaba el inconveniente del control de la iluminación. Si la cámara se encuentra en una habitación oscura, será necesario recurrir al uso de flashes, por lo que se pueden producir reflejos en el iris, ocultando la información que se encuentre

tras ellos. Por otro lado, los destellos pueden resultar molestos para el usuario, volviendo el sistema más intrusivo. Por estas razones en la actualidad se emplean casi exclusivamente cámaras en el espectro infrarrojo o cuasi-infrarrojo. Presentan varias ventajas: se puede controlar la potencia de la iluminación sin deslumbrar al usuario, la melanina (principal pigmento del iris) refleja mejor las longitudes de onda mayores (las que componen el espectro infrarrojo) por lo que se revelan con más claridad los detalles biométricos y la frontera entre el iris y la pupila se desvela de forma más pronunciada.

Normalmente se usa iluminación cercana al infrarrojo. Esto presenta varias ventajas ya que por un lado permite controlar la iluminación (particularmente en potencia) sin por ello resultar intrusivo para el sujeto. Por otro lado la iluminación cercana al infrarrojo permite resaltar la estructura de iris altamente pigmentados (oscuros). Hay que saber que la melanina (pigmento presente en los iris) absorbe en gran medida la luz del espectro visible reflejando en mayor medida las mayores longitudes de onda. Se puede usar también, iluminación natural o artificial en el espectro visible, aunque esto puede presentar inconvenientes, como ya se ha señalado.

La imagen del iris se captura desde una distancia de entre 10 y 50 centímetros, dependiendo del sistema empleado, mediante una cámara de alta resolución. Esto presenta la ventaja de no resultar invasivo para el usuario y además evita el contacto directo con el sensor, lo que resulta positivo en cuanto a higiene se refiere. Normalmente se requiere que el usuario mire directamente a la cámara, aunque también existen métodos para corregir imágenes que fueron tomadas “fuera de eje”.

Tras un proceso preliminar de enfoque, que se puede realizar actuando sobre el objetivo de la cámara o dando instrucciones al sujeto para que se acerque o aleje de la misma, se procede a un descarte de las posibles reflexiones debidas al uso de gafas, y se obtiene una imagen digital del iris.

Las posibles variaciones en el tamaño de la pupila, debidas a la potencia de la iluminación empleada, no interfieren en ningún caso en la unicidad ni en la posibilidad de identificación de los patrones de iris. Este fenómeno de contracción-dilatación puede ser aprovechado a la hora de confirmar que el iris es real y no una falsificación, en cuyo caso la pupila no respondería a las variaciones de iluminación.

La imagen capturada en un primer momento contiene partes útiles que serán utilizadas en el proceso de reconocimiento y otros componentes, irrelevantes para el mismo (por ejemplo, la pupila, los párpados, etc.). El hecho de que el iris esté ocluido en parte por los párpados, las pestañas o por gafas o lentes de contacto, puede aumentar la complejidad del algoritmo de preprocesado.

A continuación se presentan dos sistemas de adquisición patentados por Daugman [8] en 1994 y Wildes *et al.*[14] en 1996.

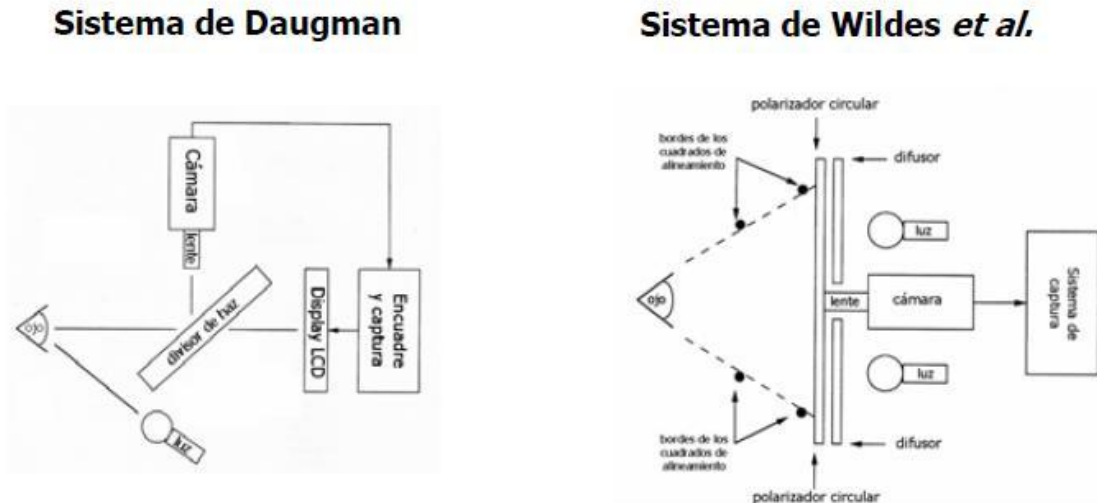


Figura 19: Sistemas de adquisición patentados por Daugman y Wildes

Tabla 2: Comparación de los sistemas de adquisición patentados por Daugman y Wildes

	Sistema Daugman	Sistema Wildes et al.
Resolución y enfoque	<ul style="list-style-type: none"> - 100~200 píxeles en el diámetro del iris (ISO [15] requiere al menos 200). - A 15~45 cm. del usuario. - Objetivo de 330 mm. 	<ul style="list-style-type: none"> - 256 píxeles en el diámetro del iris - A 20 cm. del usuario - Objetivo de 80 mm
Iluminación	<ul style="list-style-type: none"> - Simple y compacto. - Iluminación puntual mediante LED. - Cámara de vídeo estándar. - Evita reflexión en gafas, pero no el efecto "ojos rojos" (reflexión especular de la córnea). - La iluminación puntual obliga a desechar una zona del ojo por la reflexión. 	<ul style="list-style-type: none"> - Algo más sofisticado. - Iluminación difuminada (reflejo menos molesto). - Filtro polarizador y cámara de alta sensibilidad (bajos niveles de iluminación). - Los filtros polarizadores circulares evitan la reflexión especular.
Posicionamiento y centrado del iris	<ul style="list-style-type: none"> - Pantalla de cristal líquido con la secuencia de vídeo correspondiente a la zona, de forma que el usuario pueda ajustar la posición del ojo (como si estuviera frente a un espejo). - Se extrae una imagen de forma automática a partir de la secuencia adquirida según criterio de máximo contraste entre iris y pupila. 	<ul style="list-style-type: none"> - Presenta una retícula de ajuste basada en la superposición relativa de dos cuadrados. - Cuando el iris está en la posición adecuada, los cuadrados coinciden, y el usuario mismo selecciona la imagen mediante un botón.

Algunas consideraciones comunes a los dos sistemas son las siguientes:

- Ambos sistemas iluminan con luz visible, lo que implica niveles relativamente bajos para no mermar el confort del usuario. Esto obliga al uso de diafragmas relativamente grandes para dejar pasar la mayor cantidad de luz posible, lo que produce una profundidad de campo (profundidad de la zona enfocada, dada una longitud focal) pequeña. En estos sistemas es del orden de 1 cm. Desechan la información de color, codificando en escala de grises a 8 bits. La iluminación infrarroja es, por lo tanto, viable.
- Ambos sistemas usan las ventajas que ofrece la toma de secuencias de imágenes (video) para el posicionamiento y selección de imágenes. Sin embargo esto no evita el hecho de que sea fundamental la colaboración del usuario, que es además bastante restrictiva en cuanto a posicionamiento, lo que puede llegar a resultar tedioso. Se puede decir que esto afecta en cierta medida a la universalidad de estos sistemas ya que requieren de una estrecha colaboración del usuario para funcionar correctamente.

En la actualidad existen sistemas más sofisticados, con especial mención al sistema “*Iris on the Move*” [16] operativo desde 2010, que permite la captura de muestras del iris de una persona que camina a través de un portal similar a los arcos de seguridad de un aeropuerto.

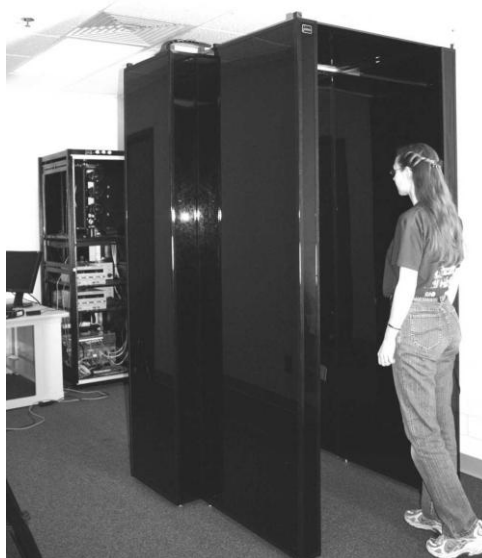


Figura 20: Un prototipo del sistema “*Iris on the Move*”

El sistema “*Iris on the Move*” es capaz de tomar imágenes de iris con suficiente calidad para el reconocimiento, de una persona que camina a velocidad normal a través de un arco similar a los que se encuentran en un aeropuerto. El arco proporciona iluminación infrarroja y estroboscópica que se sincroniza con las cámaras de video situadas en el rack que se observa al fondo de la imagen en la Figura 20 (a unos 3 metros). En este caso se consideran tres cámaras superpuestas, lo que permite que el volumen cubierto por el sistema (en el que

se deben situar los ojos del usuario) tenga un tamaño de aproximadamente 60 cm. de alto por 20 cm. de ancho por 10 cm. de profundidad. Este volumen es modulable en función del número de cámaras que se utilicen, teniendo en cuenta que cada una de ellas cubre un volumen de 20x20x10 centímetros.

Este tipo de sistema de captura permite mayor libertad al usuario, o al menos le evita tener que posicionarse de forma precisa y estática frente al sensor. No elimina sin embargo la necesidad de una cierta cooperación por parte del mismo, ya que si el usuario toma medidas para no ser identificado, el sistema será incapaz de reconocerlo. Estas medidas podrían ser el uso de gafas muy oscuras, entrecerrar los párpados o simplemente no mirar al frente, que es donde se encuentran las cámaras.

2.2.4.2 Preprocesado

La fase de preprocesado, la segunda del proceso de reconocimiento, consta a su vez de distintos procesos: control de calidad, segmentación, normalización y detección y enmascaramiento de oclusiones.

2.2.4.2.1 *Control de calidad*

La primera parte de la fase de preprocesado consiste en evaluar la calidad de la imagen para averiguar si satisface los criterios mínimos para usarse en las siguientes fases. Aunque algunos autores consideran esta fase como parte integrante del proceso de adquisición (lo que tiene sentido ya que se puede considerar que éste termina cuando se ha obtenido una imagen de calidad suficiente), aquí se considera parte del preprocesado por dos razones: primero, porque las técnicas empleadas para obtener una métrica de calidad coinciden o son similares a las empleadas en otras fases del preprocesado, y segundo, porque se considera que la fase de captura es una fase fundamentalmente referente a hardware, mientras que el control de calidad se realiza mediante software, al igual que el resto del preprocesado.

Hay que tener en cuenta que no existen criterios unificados para medir la calidad de una imagen de iris, ya que dependiendo de las técnicas de localización y comparación a utilizar, se da distinta importancia a los diversos factores que intervienen en la calidad. No obstante, según el estándar ISO/IEC 29794-6 (*Biometric sample quality – Part 6: Iris image*) [17] surgido en 2009 y actualmente en desarrollo, existen ciertas métricas (20 en total según el estándar), que siempre deberían tenerse en cuenta a la hora de determinar la calidad de una imagen de iris. Entre ellas, cabe destacar la importancia del enfoque, el movimiento de la imagen y la desviación de mirada [18], además de la cantidad de iris que queda ocluido por párpados y pestañas.

En lo referente al enfoque, métrica de calidad por excelencia, se pueden considerar varias técnicas a la hora de medirlo. La mayoría se basan en el uso de transformadas, ya sea la transformada de Fourier en 2D o la transformada wavelet, que permite un tratamiento más localizado. En cualquier caso, lo que

se pretende es medir la energía en las bandas superiores y asegurar que ésta sea superior a cierto umbral, lo que significa que existe un gran nivel de detalle y por lo tanto que la imagen está suficientemente enfocada.

Un área de trabajo muy activa en lo referente a calidad consiste en el entrenamiento de redes neuronales capaces de clasificar imágenes como buenas o malas. Existen diversos trabajos, cómo el publicado por Proença y Alexandre [19], que desecha imágenes que contengan ruido (oclusiones y reflejos), o el publicado por Krichen *et al.* [20], que tiene en cuenta además el enfoque.

En el caso de que la imagen se identifique como deficiente, existen técnicas para intentar mejorar su calidad. Una oclusión demasiado importante no se podrá resolver a menos de que se tome otra imagen, pero un enfoque ligeramente deficiente e incluso algunas reflexiones pueden ser tratadas con el fin de mejorar la calidad de la imagen de forma artificial y así superar el umbral mínimo.

2.2.4.2.2 Segmentación

Una vez se ha determinado que la imagen tiene una calidad suficiente, se procede a la segmentación. Ésta consiste en separar el iris del resto de la imagen. El iris se encuentra situado entre la pupila y la esclerótica (zona blanca del ojo). Se trata pues de determinar la localización de las dos fronteras que le separan de dichas zonas.

Algunos autores modelan dichas fronteras cómo circunferencias concéntricas [21]. Sin embargo, se ha demostrado que no son circulares y mucho menos concéntricas. En realidad, se trata de fronteras cuasi circulares, aunque pueden presentar una tendencia elipsoidal. Si la frontera con la esclerótica presenta una excentricidad considerable con respecto a la frontera con la pupila (lo que sucede a menudo), se incurre en una situación en la que o bien se pierde una porción del iris, o bien se incluye una porción de la esclerótica en la zona reservada al iris. En la mayoría de los casos, el considerar estas fronteras como circulares no produce problemas graves de segmentación, sin embargo, el considerarlas concéntricas sí puede acarrear errores más serios.

Las técnicas empleadas para realizar la segmentación son variadas y difieren según los autores, pero se pueden clasificar en tres familias.

La primera familia es la basada en operadores integro-diferenciales, inicialmente propuesta por Daugman [13]. La solución al problema consiste en encontrar los parámetros r , x_0 e y_0 (radio y centro de la circunferencia) que maximicen el operador que se muestra a continuación:

$$\max(r, x_0, y_0) \left\{ G(r) * \frac{\partial}{\partial r} \oint_{r, x_0, y_0} \frac{I(x, y)}{2\pi r} ds \right\} \quad (1)$$

Analizando (1) por partes, se observa que define una función de tres parámetros como el resultado de la integral de la imagen ($I(x,y)$) sobre una circunferencia de radio r y centro (x_0,y_0) . La integral se normaliza dividiendo entre $2\pi r$ para que la circunferencia no afecte al resultado. Posteriormente, se halla el máximo de la derivada de dicha función en la dirección radial para todo (x_0,y_0) y se selecciona el máximo r para el (x_0,y_0) que la haga máxima. La función $G(r)$ es una función de suavizado. En definitiva, con este método se busca la frontera circular que presente la máxima diferencia de luminosidad a ambos lados, lo cual tiene sentido teniendo en cuenta que la pupila es más oscura que el iris, que es a su vez más oscuro que la esclerótica. Efectivamente, la derivada presentará máximos en las zonas de transición entre las distintas partes que estamos buscando, pues son zonas en las que la variación de luminosidad es máxima.

La segunda gran familia de técnicas consiste en el uso del método de Wildes *et al.* [22] que consiste en el uso de mecanismos de detección de bordes, como por ejemplo el propuesto por Canny [23], basado también en un operador diferencial. Tras la detección de bordes se aplica la transformada de Hough [24] para detectar circunferencias. La principal desventaja de este método consiste en el coste computacional que supone.

Para paliar este efecto, se ha desarrollado una tercera familia de técnicas que consiste en utilizar métodos híbridos, que hacen uso de los anteriores pero introduciendo una primera fase que permite obtener cierta orientación sobre la ubicación y tamaño de la pupila y de la frontera límbica (la que separa el iris de la esclerótica). Una aproximación buena consiste en usar la información lumínica de la imagen para detectar estas zonas. En efecto, la pupila será en general una de las zonas más oscuras de la imagen, y la esclerótica la más clara. Lili y Mei [25], por ejemplo, proponen explotar el hecho de que el histograma de la imagen de un ojo presenta tres picos, correspondientes a la zona de la pupila, el iris y la esclerótica. Otros, como Bonney *et al.* [26] utilizan el plano de bits menos significativo, en el que zonas con mucho detalle aparecen como ruido, para detectar zonas de poco detalle como por ejemplo la pupila. Después usan operaciones de erosión y dilatación para homogeneizar dicha zona (para eliminar posibles reflexiones) y obtener una máscara que localice a la pupila.

2.2.4.2.3 **Normalización**

El paso siguiente a la segmentación consiste en la obtención de una imagen normalizada del iris, resultante de la representación del iris en forma rectangular en vez de en forma de corona. La idea es sencilla, cada punto o pixel del iris se puede localizar mediante coordenadas pseudo-polares. El “cero” en la dirección radial se encuentra en la frontera pupila-iris, y el máximo, que se puede tomar arbitrariamente como “uno”, se encontrará en la frontera iris-esclerótica. Esto permite tener en cuenta las deformaciones que sufre el iris con las contracciones de la pupila, suponiendo que fueran lineales. Aunque en la práctica esto no es exactamente así, puede considerarse una buena aproximación. Por otro lado, el “cero” en la dirección angular se toma de forma

arbitraria, ya que no existe forma de determinar la horizontal (algunos autores emplean fotos en las que aparecen los dos ojos para determinar dicha orientación [27]).

Seguidamente se mapea cada pixel de la imagen original a una nueva imagen rectangular en coordenadas cartesianas en la que la coordenada radial se mapea a la coordenada vertical y la angular a la horizontal. Intuitivamente, este paso se puede interpretar como el hecho de coger la corona que contiene el iris, aplicarle un corte y deformarla para convertirla en un rectángulo. Hay que tener en cuenta que con este método se pierde definición a medida que nos alejamos de la pupila, hecho que no tiene mayor importancia, puesto que de todas formas la cantidad de información es menor en las zonas externas del iris. La mayor importancia de este paso reside en el hecho de que el resultado es una imagen de tamaño fijo (típicamente 512x64 píxeles) independientemente del tamaño del iris en la imagen original, que se puede ver afectado, entre otros, por la lejanía del sujeto a la cámara o por las dilataciones de la pupila.

Una vez normalizada la imagen, para facilitar la fase de extracción de características, se pueden aplicar técnicas que mejoren el contraste de la imagen, como por ejemplo la ecualización del histograma, con el fin de resaltar la textura del iris.

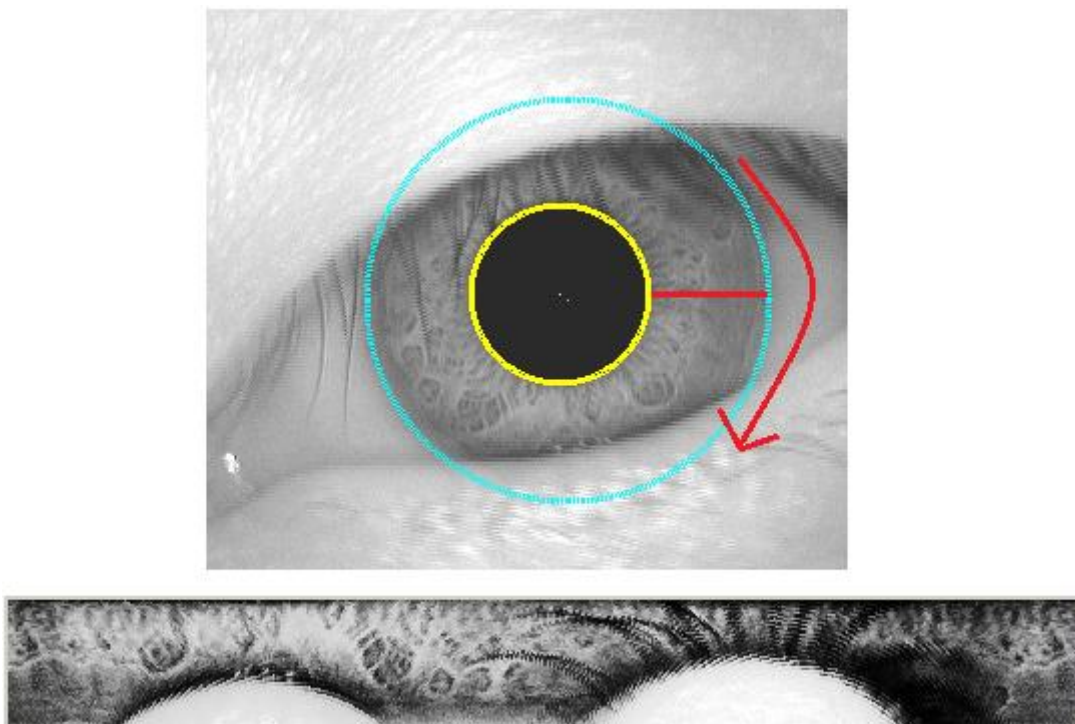


Figura 21: Ojo con las fronteras límbica y de la pupila detectadas (imagen superior) e iris normalizado y con el contraste realzado (imagen inferior). La línea roja representa el punto en el que comienza la normalización (izquierda de la imagen inferior) y la flecha el sentido en el que se recorre.

2.2.4.2.4 **Detección y enmascaramiento de oclusiones**

En la fase de preprocesado se debe además localizar y enmascarar las posibles oclusiones del iris, como las provocadas por párpados o pestañas. Dependiendo del método empleado para la segmentación, las oclusiones pueden detectarse al mismo tiempo que se localiza el iris, o bien de forma posterior, una vez se ha obtenido una imagen normalizada.

2.2.4.3 **Extracción de características**

La fase de extracción de características es la fase fundamental del proceso de reconocimiento. En ella se pretende extraer un conjunto de características que representen el iris de un usuario con el fin de poder compararlas con las almacenadas en el momento del reclutamiento.

Es la fase que más se diferencia con el método implementado en este proyecto, ya que en general se usan *IrisCodes* que son la salida de algún tipo de filtrado, mientras que en este proyecto no se procede a la extracción de características propiamente dicha sino que se compraran las imágenes directamente sin pasar por una representación basada en *IrisCodes*.

Los métodos basados en *IrisCodes*, como el de Daugman, generalmente se basan en convolucionar la imagen con algún tipo de filtro, de Gabor en el caso de este autor, y codificar la salida de dicha operación. Daugman por ejemplo utiliza dos bits por cada zona filtrada para codificar la fase de la salida del filtro (sólo se codifica en base al cuadrante del plano complejo en el que se encuentra dicha fase). Otros autores como Sun et al. [28] usan el campo de vectores gradiente de la imagen convolucionado con una función gaussiana 2-D y codifican el ángulo obtenido para cada vector como pertenecientes a una de seis categorías posibles.

También se usa la transformada wavelet, o de paquetes wavelet, para proceder a la extracción de características cómo por ejemplo Rydgren et al. en [29]. Si bien se han probado numerosos tipos de wavelet para la representación de la textura del iris, Thornton et al. [30] pusieron de manifiesto que la que mejor resultados daba era la wavelet de Gabor.

En cualquier caso y cómo ya se ha dicho, el proceso de extracción de características siempre se resume a aplicar algún tipo de transformación a la imagen del iris reclutado para almacenar características de dicha transformación cómo un código identificativo de dicho iris, para poder compararlo con el código obtenido durante el proceso de identificación. Esto se diferencia fundamentalmente del método aquí propuesto, pues el algoritmo utilizado en este proyecto no pasa por dicho código intermedio sino que procede a la comparación directa de las dos imágenes del iris (la obtenida en el reclutamiento y la obtenida durante el proceso de identificación).

2.2.4.4 Comparación de códigos y proceso de decisión

Una vez que se tiene un iris reclutado y codificado en base a sus características, habrá que compararlo con el que se toma en el momento de la identificación. Para ello se le aplica las mismas técnicas que las aplicadas en el momento del reclutamiento para generar un *IrisCode* equivalente.

Una vez obtenido, hay que comparar dichos códigos. El método que propone Daugman, y la mayoría de autores que emplean técnicas que también generan códigos binarios, es calcular la distancia de Hamming. Esta distancia mide el número de bits distintos que tienen los códigos. Tiene la gran ventaja de poder calcularse de forma muy rápida ya que bastará hacer una operación de OR exclusivo (XOR) entre los dos códigos y contar el número de “unos” que aparezcan en el resultado. De una forma más general, se pueden definir otro tipo de distancias que dependerán del método empleado en la extracción de características y comparación. En cualquier caso estas distancias siempre medirán el grado de similitud entre dos irises. En el caso de este proyecto, por ejemplo, se emplea la desviación típica de la componente horizontal del campo de vectores generados durante el proceso de comparación.

Resulta obvio que si la distancia es cero los dos códigos serán iguales y por lo tanto se podrá asegurar que los irises pertenecen a la misma persona. En la práctica sin embargo esto nunca sucede ya que las imágenes, aun siendo del mismo iris nunca son iguales. Puede haber diferencias de iluminación, de tamaño de la pupila u otros factores que produzcan códigos ligeramente distintos.

Habrà pues que definir un umbral para la distancia, que permita determinar si los códigos son lo suficientemente similares. Esto se ha hecho empíricamente por Daugman y Downing en [11] y llegaron a la conclusión que en el caso de irises distintos, la distancia Hamming normalizada (al número de bits) seguía una distribución binomial de media 0,499 y desviación típica de 0,032.

3: Lenguaje C#

3.1.1 Introducción

Microsoft.NET es un conjunto de nuevas tecnologías, desarrolladas por Microsoft con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Ésta es la llamada plataforma .NET.

Para crear aplicaciones para la plataforma .NET, Microsoft ha publicado el denominado kit de desarrollo de software conocido como .NET Framework SDK, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución y Visual Studio.NET, entorno de desarrollo integrado, que permite hacer todo lo anterior desde una interfaz visual basada en ventanas.

Ambas herramientas pueden descargarse gratuitamente desde <http://www.msdn.microsoft.com/net>. En este proyecto sólo se hace uso del lenguaje C# propiamente dicho y del entorno de desarrollo mencionado. No se ahondará por lo tanto en las características de integración con otros lenguajes o servicios de la plataforma.



Figura 22: Logo de c# .net

3.1.2 Origen

C# es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET.

Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella. Carece, por lo tanto, de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET

La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención de Microsoft con C# es facilitar la migración de códigos escritos en este lenguaje y facilitar su aprendizaje a los desarrolladores habituados a él. Sin embargo, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic o Java.

Este último lenguaje habría resultado ideal para la plataforma .NET, pero debido a diferencias con la empresa creadora del mismo, Sun Microsystems, que fue posteriormente adquirida por Oracle, Microsoft ha tenido que desarrollar un nuevo lenguaje que añadiese, a las ya probadas virtudes de Java, las modificaciones que tenía pensado añadirle para mejorarlo aún más y hacerlo un lenguaje orientado al desarrollo de componentes.

En resumen, C# es un lenguaje de programación que toma las mejores características de lenguajes anteriores como Visual Basic, Java o C++ y las combina en uno solo. El hecho de ser relativamente reciente no implica que sea inmaduro, pues Microsoft ha escrito la mayor parte de la BCL (Librería de Clases Base) usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el *.NET Framework SDK*.

3.1.3 Conceptos generales

3.1.3.1 Common Language Runtime (CLR)

El Common Language Runtime (CLR) es el núcleo de la plataforma .NET. Es el motor encargado de gestionar la ejecución de las aplicaciones desarrolladas y a las que ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad. Es el equivalente a la máquina virtual de Java. Las principales características y servicios que ofrece el CLR son:

- **Modelo de programación consistente:** modelo de programación orientado a objetos. Todos los servicios son modelados como objetos.

- **Modelo de programación sencillo:** se eliminan registros de Windows, GUIDs, HRESULTS, IUnknown, etc... Estos conceptos no existen en la plataforma .NET
- **Eliminación del “infierno de las DLLs”:** Este problema consiste en que al sustituirse versiones viejas de DLLs compartidas puede que aplicaciones que fueron diseñadas para ser ejecutadas con estas librerías dejen de funcionar. En la plataforma .NET las versiones nuevas de las DLLs pueden coexistir con las viejas. Se simplifica mucho la instalación y desinstalación de software.
- **Ejecución multiplataforma:** El CLR actúa como una máquina virtual. Y se han desarrollado versiones del CLR para la mayoría de las versiones de Windows: Windows 95, Windows 98, Windows ME, Windows NT 4.0, Windows 2000, Windows XP y Windows CE (que puede ser usado en CPUs que no sean de la familia x86). Por otro lado se ha firmado un acuerdo con Corel para portar el CLR a Linux y también hay terceros que están desarrollando de manera independiente versiones de libre distribución del CLR para el sistema operativo de código abierto. Así mismo, dado que la arquitectura del CLR está totalmente abierta, es posible que en el futuro se diseñen versiones del mismo para otros sistemas operativos.
- **Gestión de memoria:** El CLR incluye un recolector de basura que evita que el programador tenga que tener en cuenta cuándo ha de destruir los objetos que dejen de serle útiles. Gracias a este recolector se evitan errores de programación muy comunes como intentos de borrado de objetos ya borrados, agotamiento de memoria por olvido de eliminación de objetos inútiles o solicitud de acceso a miembros de objetos ya destruidos.

3.1.3.2 Microsoft Intermediate Language (MSIL)

Todos los compiladores que generan código para la plataforma .NET generan código escrito en el lenguaje intermedio conocido como Microsoft Intermediate Language (MSIL). MSIL es el código máquina de la máquina virtual de C# (el CLR).

MSIL ha sido creado por Microsoft tras consultar a numerosos especialistas en la escritura de compiladores y lenguajes tanto del mundo académico como empresarial. Es un lenguaje de un nivel de abstracción mucho más alto que el de la mayoría de los códigos máquina de las CPUs existentes, e incluye instrucciones que permiten trabajar directamente con objetos (crearlos, destruirlos, inicializarlos, llamar a métodos virtuales, etc.), tablas y excepciones (lanzarlas, capturarlas y tratarlas).

La principal ventaja del MSIL es que facilita la ejecución multiplataforma y la integración entre lenguajes al ser independiente de la CPU y proporcionar un formato común para el código máquina generado por todos los compiladores

que generen código para .NET. Sin embargo, dado que las CPUs no pueden ejecutar directamente MSIL, antes de ejecutarlo habrá que convertirlo al código nativo de la máquina sobre la que se vaya a ejecutar. De esto se encarga un componente del CLR conocido como compilador JIT (Just-In-Time) que va convirtiendo dinámicamente el código MSIL a ejecutar, en código nativo según sea necesario. El compilador JIT se distribuye en tres versiones:

- **Normal:** Es el que se suele usar por defecto, y sólo compila el código MSIL a código nativo a medida que va siendo necesario, pues así se ahorra tiempo y memoria al evitar tener que compilar innecesariamente código que nunca se ejecute (no siempre se ejecutan todas las líneas de código de un software). Cada método es compilado individualmente y una vez hecho esto se inserta el código compilado en cada llamada a dicho método. Esto lo vuelve más eficiente que otros lenguajes interpretados como Java.
- **Económico:** Funciona de forma similar al compilador JIT normal solo que no realiza ninguna optimización de código al compilar, sino que traduce cada instrucción MSIL por su equivalente en el código máquina sobre la que se ejecute. Está especialmente pensado para ser usado en dispositivos que dispongan de poca potencia de CPU y poca memoria, pues aunque genere código más ineficiente es menor el tiempo y memoria que necesita para compilar. Es el compilador JIT usado por defecto en Windows CE.
- **Precompilador:** Se distribuye como una aplicación de línea de comandos llamada *ngen.exe* mediante la que es posible compilar completamente cualquier ejecutable o librería que contenga código gestionado y convertirlo a código nativo, de modo que posteriores ejecuciones del mismo se harán usando esta versión ya compilada y no se perderá tiempo en hacer la compilación dinámica.

3.1.3.3 Librería de Clases Base (BCL)

La Librería de Clase Base (BCL) es una librería incluida en el *.NET Framework* formada por cientos de tipos de datos que permiten acceder a los servicios ofrecidos por el CLR y a las funcionalidades más frecuentemente usadas a la hora de escribir programas.

Esta librería está escrita en MSIL, por lo que puede usarse desde cualquier lenguaje cuyo compilador genere MSIL. Es tal la riqueza de servicios que ofrece que pueden crearse lenguajes que carezcan de librería de clases propia y sólo usen la BCL como por ejemplo C#.

Dado la extensión de la BCL, ha sido necesario organizar las clases en espacios de nombres que agrupen clases con funcionalidades similares. Los espacios de nombres más usados son:

- **System:** Tipos muy frecuentemente usados, como los tipos básicos, tablas, excepciones, fechas, números aleatorios, recolector de basura, entrada/salida en consola, etc.
- **System.Collections:** Colecciones de datos de uso común como pilas, colas, listas, diccionarios, etc.
- **System.Data:** Manipulación de bases de datos. Forman la denominada arquitectura ADO.NET.
- **System.IO:** Manipulación de ficheros y otros flujos de datos.
- **System.Net:** Realización de comunicaciones en red.
- **System.Reflection:** Acceso a los metadatos que acompañan a los módulos de código.
- **System.Runtime.Remoting:** Acceso a objetos remotos.
- **System.Security:** Acceso a la política de seguridad en que se basa el CLR.
- **System.Threading:** Manipulación de hilos.
- **System.Web.UI.WebControls:** Creación de interfaces de usuario basadas en ventanas para aplicaciones Web.
- **System.Windows.Forms:** Creación de interfaces de usuario basadas en ventanas para aplicaciones estándar.
- **System.XML:** Acceso a datos en formato XML.

4: Método implementado

4.1 Descripción del algoritmo utilizado

El algoritmo implementado en este proyecto, difiere fundamentalmente de los propuestos hasta la fecha en el sentido de que no pasa por un *IrisCode* intermedio, sino que compara directamente las imágenes almacenadas durante el proceso de reclutamiento y en el momento de la identificación. A diferencia de las demás técnicas de identificación mediante patrones del iris que se basan en transformaciones de la imagen para extraer sus características, el algoritmo aquí implementado se basa en técnicas de estimación de movimiento similares a las empleadas en la codificación y compresión de video, como por ejemplo los algoritmos MPEG.

Estas técnicas, provienen del hecho que en una secuencia de vídeo, no suele haber grandes variaciones en fotogramas contiguos, sino que las diferencias se suelen limitar a un pequeño número de objetos que se mueven de un fotograma a otro. No resulta pues extraño, que para ahorrar espacio de almacenamiento, o ancho de banda en el caso de la transmisión de vídeo, se codifique el flujo de datos de la siguiente manera. Cada cierto tiempo se codifica un fotograma completo, y para los siguientes sólo se expresan las diferencias con respecto al original. Básicamente se codifican los cambios en el espacio que han sufrido los distintos elementos de la toma.

Para llevar a cabo esta tarea se suele emplear una técnica denominada *Block Matching* o emparejamiento de bloques. Esta técnica consiste en dividir la imagen nueva en macro bloques y buscar el que más se ajuste en la imagen

original. Para ello se compara cada macro bloque con versiones desplazadas de la imagen original con el fin de determinar el que produce un error menor.

Existen cuatro aspectos fundamentales a tener en cuenta y estos son: tamaño del área de búsqueda, el tamaño de los macro bloques, la estrategia de búsqueda y la métrica del error. El primer aspecto es fundamental, ya que sería demasiado costoso computacionalmente buscar en toda la imagen, por lo que se tratará que el área sea lo más pequeña posible, pero esto puede producir errores si los cambios (en el espacio) de una imagen a otra son lo suficientemente grandes. El segundo aspecto, el tamaño del macro bloque es igualmente determinante ya que cuanto más pequeño sea, mejor será la detección de detalles finos a cambio de un tiempo de cálculo mayor. La estrategia de búsqueda, si bien no interfiere en la tasa de errores cometidos si que puede influir en gran medida sobre el tiempo que lleve encontrar cada macro bloque. Por último la métrica de error puede intervenir tanto en el coste computacional, cómo en la bondad del ajuste que se alcance. Normalmente se suele usar la SAD (*Sum of Absolute Differences*) o suma de diferencias absolutas, MAD (*Mean of Absolute Differences*) o diferencia absoluta media y el MSE (*Minimum Squared Error*) o error cuadrático mínimo.

En el caso del reconocimiento de iris se utiliza una técnica similar a la expuesta. El método, propuesto por D.M. Monro y S. Rakshit en [1] consiste en dividir la imagen tomada en el momento del reclutamiento (patrón) en macrobloques con un solapamiento del 50%, y buscar dichos bloques en la imagen tomada en el momento de la identificación (muestra).

La métrica del error es la MAD, el tamaño de los bloques de doce por doce píxeles y el tamaño de la zona de búsqueda es también de doce píxeles en la vecindad del bloque escogido. Estos valores han sido determinados de forma experimental y son los que proporcionan una mejor EER. Cabe destacar que para asegurar que los bloques de la periferia de la imagen tengan una zona de búsqueda del mismo tamaño que la del resto de los bloques, se extiende la imagen de muestra. Esta operación consiste en copiar las últimas doce columnas de la imagen normalizada al inicio de ésta y, análogamente copiar las doce primeras columnas al final. Esto se puede hacer ya que, dado el proceso de normalización, existe continuidad entre el principio (parte izquierda) y el final de la imagen normalizada, dado que proviene de la transformación de la corona del iris

Para cada macrobloque del patrón, se busca, dentro de la zona delimitada el bloque que mejor corresponda en la imagen de muestra. Como ya se ha expuesto, se usa la MAD para averiguar cuál de los bloques de la imagen de muestra corresponde con mayor probabilidad con el tomado del patrón. Esto consiste en restar, píxel a píxel los valores de luminosidad de cada bloque, promediándolo. Se toma el valor absoluto de dichas diferencias ya que lo que interesa es medir la "distancia" (en luminosidad) que separa los píxeles de ambos bloques. La estrategia de búsqueda es una de las más sencillas posibles. Se busca de izquierda a derecha y de arriba abajo dentro de la zona delimitada. Si se encontrara una MAD de cero para algún bloque candidato el algoritmo se detendría y lo seleccionaría como bloque definitivo. Sin embargo

esto no suele ocurrir ya que es fácil que existan diferencias de iluminación entre los patrones y las muestras por lo que el caso general consiste en recorrer toda la zona de búsqueda y seleccionar el bloque con una menor MAD.

Para cada bloque del patrón se designa un bloque como la mejor coincidencia en la imagen de muestra y se computa su distancia con respecto al primero. Se obtiene pues un campo de vectores que representan el desplazamiento que habría que aplicar a cada porción de la imagen original para reconstituir la imagen de muestra.

La idea fundamental para el reconocimiento, es que, para dos imágenes del mismo iris, el campo de vectores estará fuertemente correlado, mientras que para dos irises distintos, presentará direcciones aleatorias. No se trata de minimizar la norma de dichos vectores ya que dos irises iguales pueden presentar una norma media elevada. Hay que tener en cuenta que la componente horizontal de dichos vectores representa una rotación del iris, mientras que la vertical representa las posibles deformaciones debidas a un cambio en el tamaño de la pupila.

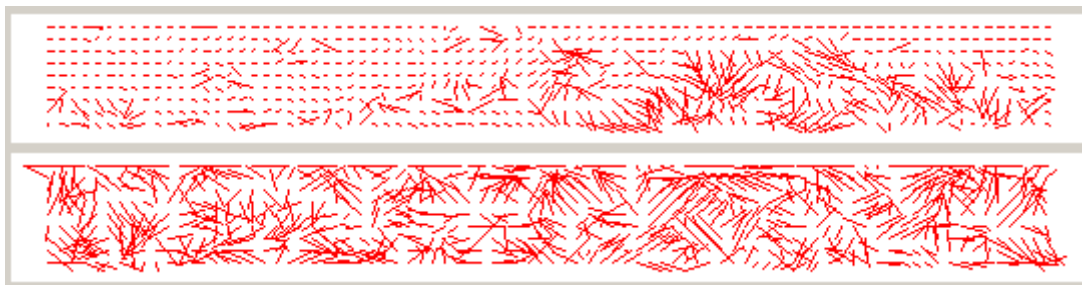


Figura 23: Campos generados en la comparación de dos irises del mismo sujeto (superior) y de sujetos distintos (inferior).

Como se puede observar en la figura 25 el campo generado en la comparación de las fotografías del iris de un mismo sujeto presenta una estructura relativamente ordenada. Hay que notar que la parte derecha presenta una zona en la que los vectores siguen direcciones aleatorias. Esta zona corresponde a la ocupada por el párpado, ya que en este caso, no se ha aplicado ninguna máscara. Para propósitos de reconocimiento se enmascararán dichas zonas para no perturbar el algoritmo.

La imagen inferior, correspondiente a la comparación de dos irises distintos, presenta sin embargo direcciones aleatorias, no sólo en la zona del párpado sino también en toda la superficie del iris. Habrá que escoger pues una métrica que permita caracterizar dicha aleatoriedad.

Se ha escogido la desviación típica de la componente horizontal de los vectores de translación como medida de similitud entre dos irises. La componente horizontal presenta la ventaja de estar más correlada (en irises iguales) que la vertical ya que ésta se ve afectada sólo por la rotación y no por

deformaciones complejas de modelar. La desviación típica no depende de la media y por lo tanto no se ve afectada por rotaciones debidas a que la imagen del iris no fuese tomada de forma totalmente horizontal.

Por último cabe destacar que la finalidad de este proyecto consiste en implementar el algoritmo de comparación propiamente dicho. Las fases previas a la comparación como la captura de los datos biométricos o el preprocesado no se han realizado. Se ha trabajado directamente con la colección de ojos CASIA versión 1 y el preprocesado se ha hecho con la librería *IrisBiometricsCS* proporcionada por el GUTI (Grupo Universitario de Tecnologías de Identificación) dirigido por el director de este proyecto, Dr. Raúl Sánchez Reíllo.

4.2 Detalles de la Implementación

4.2.1 Clases auxiliares implementadas

Antes de comenzar a trabajar con el algoritmo propiamente dicho se han implementado una serie de clases auxiliares con el fin de facilitar y normalizar todo el proceso. También se han aportado una serie de modificaciones a la librería *IrisBiometricsCS*. A continuación expondremos dichas modificaciones.

En primer lugar se modificó el método que realiza el cambio de coordenadas pseudo-polares a coordenadas cartesianas. El método original no mapeaba las partes superiores e inferiores del iris ya que estas zonas son las que suelen estar enmascaradas por los párpados. Sin embargo, el algoritmo propuesto por Monro tiene en cuenta estas partes por lo que se modificó el método de transformación para que mapeara el iris completo. En segundo lugar se añadió una fase de ecualización del histograma con el fin de aumentar el contraste de la imagen normalizada. Lo que se pretende con esta operación es distribuir los niveles de gris de una forma uniforme con el fin de obtener un histograma lo más plano posible.

Un histograma es un diagrama de barras que representa (en ordenadas) la cantidad de píxeles de una imagen que tiene un determinado nivel de luminosidad (abscisas) que en el caso de una imagen en escala de grises es el nivel de gris. Estos van desde el negro (a la izquierda) hasta el blanco (a la derecha). Se puede interpretar como la f.d.p. (función de densidad de probabilidad) de los niveles de gris en la imagen.

Concretamente a cada píxel con un determinado nivel de gris, se le asigna un nuevo nivel de gris que depende de la cantidad de píxeles en la imagen original que tengan un nivel de gris igual o inferior a ese. Para ello se usa la siguiente fórmula:

$$s_k = \frac{I_{max}}{N} \sum_{i=0}^k n_i \quad (2)$$

Dónde s_k representa el nivel de gris que tendrán los píxeles que originalmente tenían un nivel k , I_{max} es el nivel de intensidad máximo que se desea en la

imagen nueva, N es el número total de píxeles y n_i es el número de píxeles en la imagen original con un nivel de i . En la práctica, con el fin de acelerar el proceso y no tener que realizar el cálculo para cada píxel, se construye una LUT (*Look Up Table* o tabla de búsqueda) que contiene los nuevos niveles para cada nivel de gris en la imagen original, que se irá consultando para realizar los cambios.

Se puede observar que se consiguen dos efectos. Por un lado, se expande el rango dinámico de la imagen (en el caso en el que no fuera ya máximo de por sí). Es decir que se consigue una imagen en la que hay píxeles con niveles que van desde el 0 (negro), hasta I_{max} (blanco). Por otro lado, dado el procedimiento que se emplea, y como se puede ver en la figura 26, se comprime el contraste para aquellos niveles que están poco presentes en la imagen, para poder expandir el contraste en los niveles que tienen mayor frecuencia de aparición. Como estos son los píxeles más abundantes, se puede decir que globalmente se ha aumentado el contraste, aunque localmente (para los niveles menos probables) se haya contraído.

Por último, cabe destacar que esta transformación es biyectiva, es decir, que existe una correspondencia uno a uno entre los niveles de gris de la imagen original y los de la imagen transformada, conservando por lo tanto la estructura de la imagen y su entropía (cantidad de información)

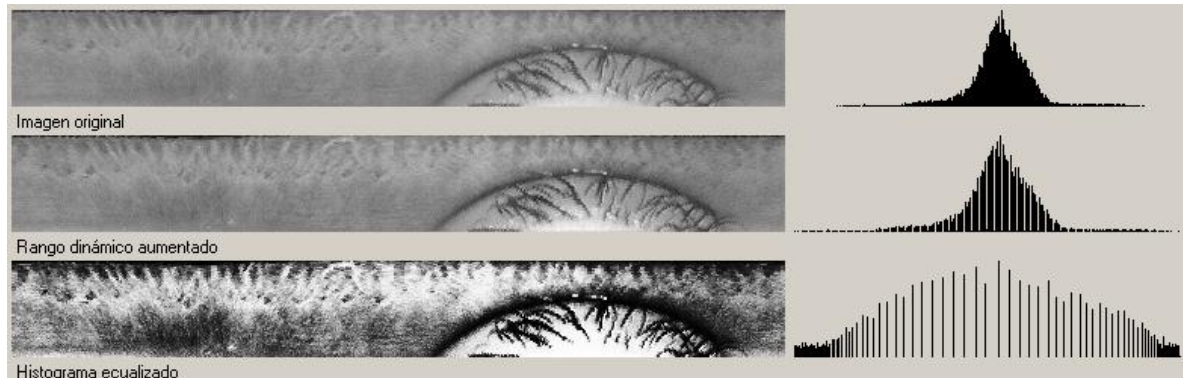


Figura 24: Ejemplo de aumento de contraste. Imagen original, con el rango dinámico aumentado y con el histograma ecualizado. Histogramas de cada caso.

Una vez explicadas las modificaciones aportadas a la librería proporcionada por el GUTI, se va a proceder a detallar los aspectos fundamentales de la implementación de la librería desarrollada para este proyecto. Se trata de un conjunto de clases, agrupadas en un paquete llamado *MPEGIris*. Contiene, además de las clases necesarias para realizar la comparación de dos irises, un subpaquete llamado *MPEGIris.Filtros*, que no se detallará aquí pues corresponde al trabajo realizado en el contexto de un trabajo dirigido enfocado al cálculo de máscaras que permiten excluir las oclusiones del iris del proceso de identificación.

El paquete MPEGIris contiene tres clases auxiliares propiamente dichas, que se encargan de la representación de los irises, de los resultados obtenidos al compararlos y de los eventos que se generan para poder representar paso a paso el proceso de comparación. Estas clases son la clase *Iris*, la clase *Resultado*, y la clase *ReticulaMatchEventArgs*

4.2.1.1 Clase Iris

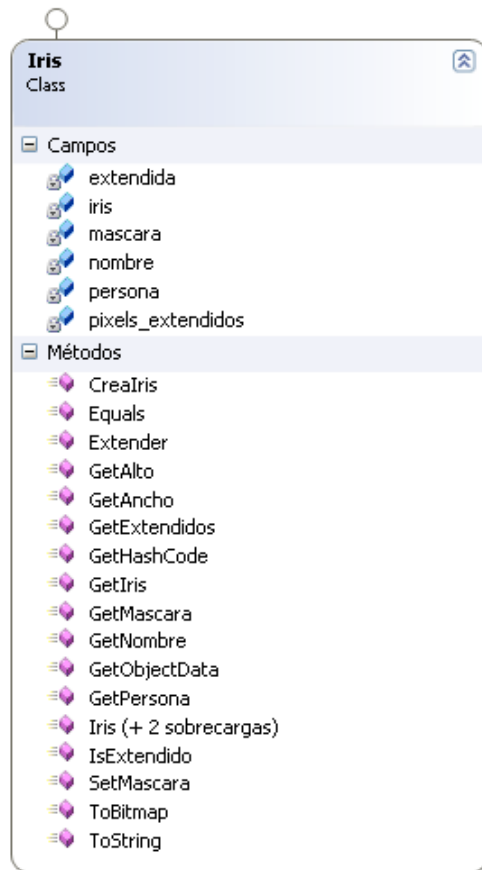


Figura 25: Diagrama de la clase *Iris*.

La clase *Iris* es la encargada de proporcionar una representación unificada, dentro del sistema, de un iris. Permite abstraer el formato original en el que se tomaron las fotografías. Además, al implementar la interfaz *ISerializable* permite escribir dicha representación en disco, e incluso transmitirla por una red de datos.

Permite pues el almacenamiento, no solo de la información del iris en sí, sino de la máscara asociada al mismo, el nombre del sujeto al que pertenece y algunos parámetros que pueden ser de interés para el proceso de identificación. Implementa también una serie de métodos que permiten entre otras cosas crear una instancia de *Iris* a partir de una fotografía escrita en disco o extenderla. El uso de esta representación compacta de los diversos parámetros presenta la ventaja de que las fases de preprocesado y

enmascaramiento se pueden realizar con anterioridad al proceso de comparación con lo que se acelera este último.

A continuación se presentan los diversos campos que la componen:

- **iris:** matriz del tipo byte que contiene la información del iris en escala de grises de 8bits.
- **mascara:** matriz del tipo bool que contiene la máscara que se le aplicará al iris en el momento de la comparación, con el fin de excluir las posibles oclusiones, de dicho proceso. Se marca a true las posiciones de los píxeles que corresponden a oclusiones.
- **nombre:** nombre que identifica, en forma de string, unívocamente al iris. En el caso de este proyecto sigue el formato XXX_Y_Z, dónde XXX es un código de tres dígitos que identifica a la persona a la que pertenece el iris, Y es un dígito que representa el momento de la toma (puede ser 1 o 2 ya que la colección CASIA v1 dispone de archivos tomados en dos momentos distintos) y Z es otro dígito que diferencia varias tomas hechas en el mismo momento para un mismo usuario.
- **persona:** string que representa a la persona a la que pertenece el iris. Corresponde al código XXX del nombre.
- **extendida:** bool que indica si la imagen del iris ha sido extendida o no.
- **pixels_extendidos:** int que indica el número de píxeles que se han extendido.

Cabe destacar que todos estos campos se han declarado como privados con el fin de poder controlar quién, cuándo y cómo se accede a ellos.

De los métodos que componen esta clase, se detallan los más relevantes:

- **Crealris:** método estático que acepta un string, como parámetro que representa una ruta de archivo en disco. Se encarga de leer el archivo (de imagen) del disco, segmentar y preprocesar dicho archivo mediante la clase *IrisBiometricsCS*, para devolver una instancia de tipo *Iris*.
- **Equals:** método que sobrescribe el método homónimo de la clase *Object*. Se diferencia de este último en el sentido de que devuelve true si el *Iris* pertenece a la misma persona independientemente de si se trata del mismo objeto (puede tratarse de tomas distintas).
- **Extender:** método que acepta un int (num_pixels) y extiende el iris en un número de píxeles igual a num_pixels. Cómo ya se ha comentado la

imagen normalizada del iris, que tiene forma rectangular, procede de la deformación de la corona del iris. Por lo tanto, existe continuidad entre la parte izquierda y derecha de la imagen normalizada.

La extensión, consiste en pegar una porción (de ancho `num_pixels`) de la parte derecha de la imagen en la parte izquierda y viceversa. Esto asegura que la zona de búsqueda tendrá igual tamaño en los extremos de la imagen y en el centro.

- **GetObjectData:** método necesario para la implementación de la interfaz *ISerializable*. Se encarga de definir los campos que deberán ser serializados.
- **ToBitmap:** método que devuelve un `Bitmap` para poder visualizar la información contenida en el campo *iris*. Es una imagen del iris normalizado.
- **ToString:** método que devuelve un `string` que indica el *nombre*, si el iris está extendido o no y su tamaño en píxeles (ancho x alto).
- **Iris:** Es el constructor de la clase. Está sobrecargado dos veces. Permite la creación de un iris proporcionando el nombre y la matriz *iris*, o bien proporcionando estos datos y además la *mascara*. Por último hay una versión específica para la deserialización.

Existen otra serie de métodos, fundamentalmente *Getters*, que no revisten un gran interés y cuyo nombre es autoexplicativo.

4.2.1.2 Clase Resultado

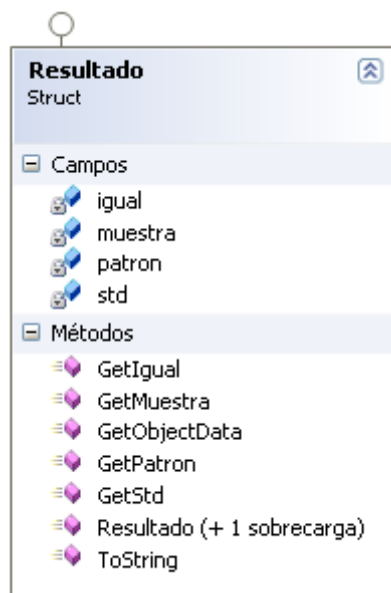


Figura 26: Diagrama de la clase *Resultado*.

La clase *Resultado* contiene los siguientes campos:

- **igual:** es un bool que está a true cuando el resultado proviene de la comparación de dos irises de la misma persona
- **muestra:** es un string que representa el nombre de la muestra.
- **patron:** es un string que representa el nombre del patrón.
- **std:** es un float que contiene el resultado de la comparación, es decir, la desviación típica de la componente horizontal del campo de vectores generado durante la comparación.

En esta clase, los campos son igualmente privados, por lo que contiene métodos *get* para acceder a ellos. Contiene también los métodos necesarios para poder serializar las instancias de esta clase. Por último, tiene definido un método **ToString** que devuelve un string con el nombre del patrón, de la muestra, si provienen de la misma persona, y la desviación típica obtenida como resultado de la comparación.

4.2.1.3 Clase **ReticulaMatchEventArgs**

Esta clase se usa para lanzar un evento cada vez que se encuentra, en la imagen de muestra, el bloque correspondiente al de la imagen patrón. Contiene el valor de la MAD calculado y las retículas (representación del bloque) del patrón y de la muestra.

4.2.2 Implementación del algoritmo y sus variantes

El algoritmo está implementado de dos maneras. Una de ellas, es una versión monohilo que fue usada fundamentalmente para una primera exploración y las pruebas preliminares. Ahora se usa con propósitos de demostración. Es la clase que hace uso de los eventos *ReticulaMatchEventArgs*. La segunda, es una versión multihilo, derivada de la primera, que se ha implementado con el fin de explotar toda la potencia de los equipos actuales multinúcleo.

Para el desarrollo de este algoritmo, se han utilizado tres clases: la clase *Matcher*, correspondiente a la versión monohilo, y las clases *PartialMatcher* y *ThreadedMatcher*, que implementan la versión multihilo.

4.2.2.1 Clase Matcher

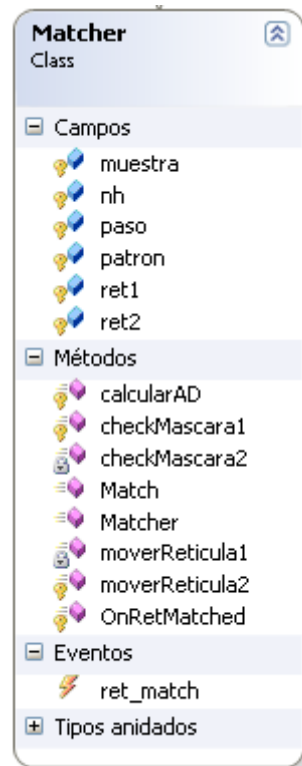


Figura 27: Diagrama de la clase *Matcher*

Esta clase contiene una implementación completa del algoritmo, con la particularidad de que lanza eventos cada vez que se encuentra una correspondencia de bloques. Además, se define aquí la estructura *Retícula*, que permite almacenar la posición de las retículas (áreas de los bloques de comparación).

Los campos contenidos en esta clase son:

- **patron:** *Iris* correspondiente al patrón.
- **muestra:** *Iris* correspondiente a la muestra.
- **ret1:** *Reticula* correspondiente al patrón.
- **ret2:** *Reticula* correspondiente a la muestra.
- **nh:** int que representa el tamaño del área de búsqueda en la imagen extendida.
- **paso:** int que representa la distancia que se debe de mover la *Reticula ret1* para recorrer el patrón. En este caso, como queremos un solapamiento del 50%, tendrá un valor de la mitad del tamaño de las retículas.

Los métodos presentes en esta clase son:

- **calcularAD:** método que calcula la MAD entre los píxeles contenidos en las imágenes en las posiciones indicadas por las retículas. Para ello, utiliza las posiciones de las retículas y va restando uno a uno los píxeles del patrón y de la muestra que se encuentran en las zonas delimitadas por éstas.
- **checkMascara1, checkMascara2:** método que devuelve un bool, que indica si la *Reticula ret1* o *ret2* han entrado en una zona enmascarada.
- **moverReticula1:** método que se encarga de ir moviendo la *Reticula ret1* sobre el patrón. Para ello mueve, cada vez que es invocado, la retícula una cantidad *paso* a la derecha, hasta que alcanza el límite derecho del patrón. Una vez alcanzado dicho límite, devuelve la retícula a la izquierda, moviéndola una cantidad *paso* hacia abajo. En cada movimiento, comprueba mediante el método **checkMascara1** que la retícula no ha entrado en una zona enmascarada. De ser así, se llama a sí mismo.

El método devuelve true, siempre que el movimiento sea posible, mientras que devuelve false cuando no es posible. Esto sucede cuando la retícula ha alcanzado el borde inferior derecho del patrón. Dado que la retícula se inicializa en la parte superior izquierda del patrón, es obvio que invocando sucesivamente este método, se recorre todo el patrón.

- **moverReticula2:** método que se encarga de ir moviendo la *Reticula ret2* sobre la muestra. Para ello mueve, cada vez que es invocado, la retícula un píxel hacia la derecha, hasta que alcanza el límite derecho de la zona de búsqueda. Una vez alcanzado dicho límite, devuelve la retícula al límite izquierdo de la zona de búsqueda, moviéndola un píxel hacia abajo. En cada movimiento, comprueba mediante el método **checkMascara2** que la retícula no ha entrado en una zona enmascarada. De ser así, se llama a sí mismo.

El método devuelve false cuando la retícula ha alcanzado la zona inferior derecha de la zona de búsqueda, y true en caso contrario. Dado que la retícula se inicializa en la parte superior izquierda de la zona de búsqueda, es obvio que invocando sucesivamente este método, se recorre toda la zona de búsqueda.

Se podría haber implementado un método de representación y movimiento de la zona de búsqueda. Sin embargo, como ésta depende de la posición de la *ret1*, no ha sido necesario

- **Match:** este es el método principal, donde se calcula la desviación típica de la componente horizontal de los vectores del campo generado durante la comparación.

La desviación típica marca la medida de similitud entre el patrón y la muestra, y su cálculo es, por lo tanto, la parte fundamental del algoritmo de reconocimiento.

- El método comienza inicializando las retículas en sus posiciones de origen.
- Tras esto, utiliza iterativamente el método **calcularAD**.
- En cada iteración:
 1. Comprueba si la MAD es menor que la MAD mínima calculada hasta el momento para una posición dada de la *ret1*.
 - Si es menor, retiene la *ret2* como retícula candidata, computa su distancia en el plano horizontal a la *ret1* y almacena dicha distancia.
 2. Mueve la *ret2* y vuelve a realizar el paso 1.
- Esto sucede hasta que el método **moverReticula2** devuelve false, significando que se ha alcanzado el límite de la zona de búsqueda, o bien si se obtuviera una MAD de 0, lo que significaría que se ha encontrado la mejor coincidencia.
- En este momento, están almacenadas la *ret2* y la distancia en el plano horizontal a *ret1* de la mejor coincidencia para dicha *ret1*.
- Se almacena la distancia en un vector permanente, se lanza el evento que indica que se ha encontrado dicha coincidencia mediante el método **OnRetMatched** y se mueve la *ret1*.
- A continuación, se repite el proceso explicado para la nueva *ret1*. Este proceso se repite hasta que el método **moverReticula1** devuelve false, indicando que se ha recorrido todo el patrón.
- En este momento tenemos un vector que contiene todas las distancias en el plano horizontal de las mejores coincidencias de la *ret2* para sus respectivas *ret1*.
- Después, sólo queda calcular la media y la desviación típica de las distancias contenidas en el vector permanente y devolver dicho resultado.

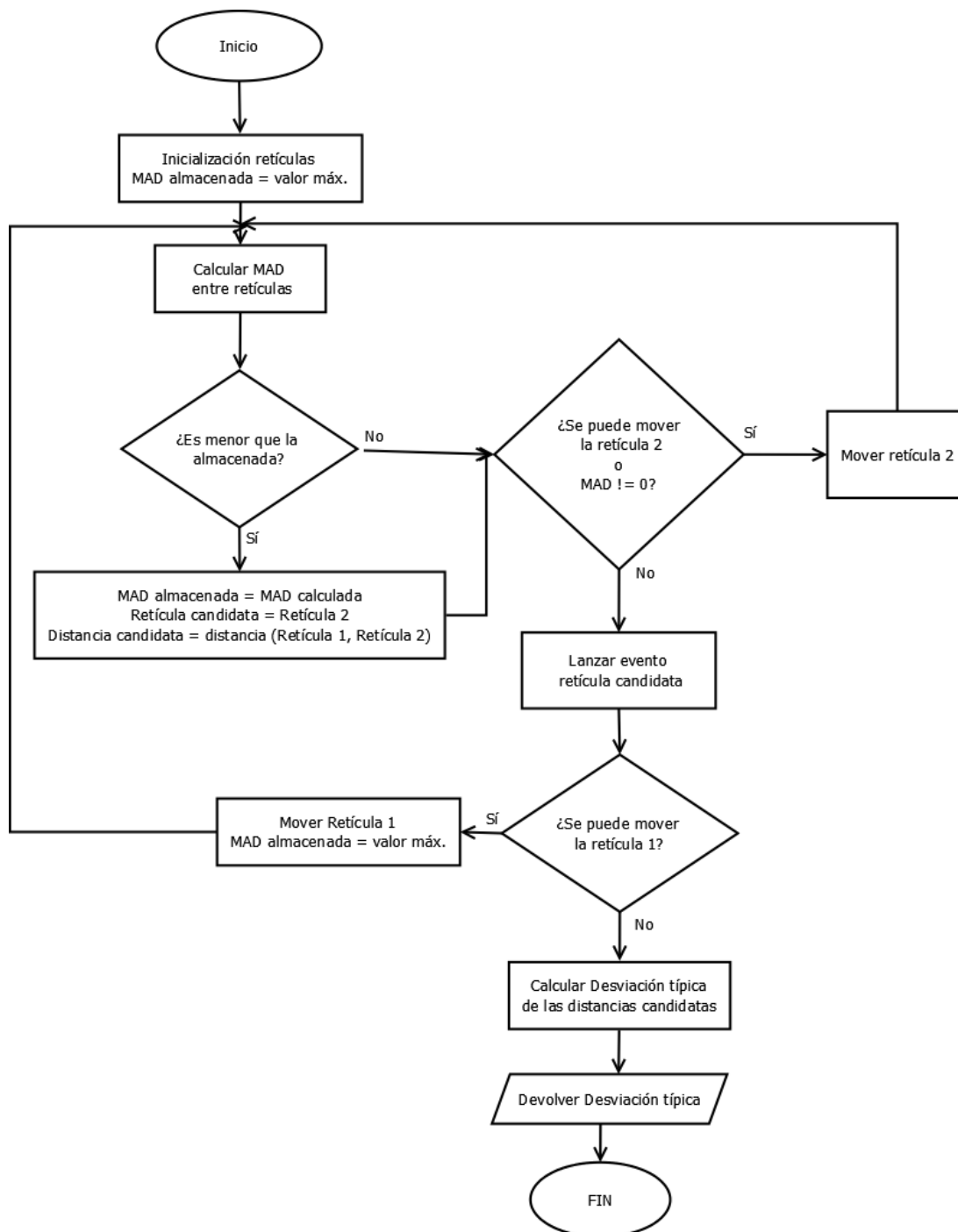


Figura 28: Diagrama de flujo del método **Match**.

- **OnRetMatched**: método que se encarga de lanzar los eventos cuando se encuentra la mejor coincidencia de una retícula a otra.
- **Matcher**: este método es el constructor de la clase. Se le pasan como parámetros los *Iris patron* y *muestra*. Si la muestra no está extendida, la extiende. Además, crea las retículas, calcula el tamaño de la zona de muestra en función del tamaño de la extensión de la muestra y el *paso* en función del tamaño de la *ret1*.

4.2.2.2 Clase PartialMatcher

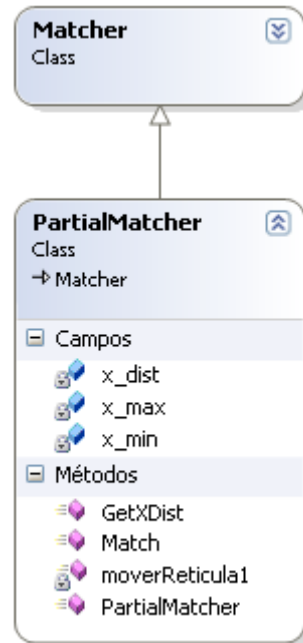


Figura 29: Diagrama de la clase *PartialMatcher*

La clase *PartialMatcher* hereda de la clase *Matcher*. Se diferencia de su clase padre en el sentido de que no recorre todo el patrón sino sólo una parte de él. Se le debe, por lo tanto, especificar en el constructor los puntos en los que debe comenzar y terminar su análisis.

Se han sobrescrito los siguientes métodos:

- **moverReticula1**: el método funciona básicamente como su método padre, con la excepción de que devuelve false cuando llega al límite que se le ha especificado, en vez de cuando llega al límite del patrón. A su vez, comienza en el punto especificado en vez de en el origen del patrón.
- **Match**: funciona de forma análoga a su método padre, con la excepción de que, al no recorrer el patrón completo, no es capaz de dar una medida de similitud entre el patrón y la muestra, sino que lo que devuelve es la parte del vector permanente correspondiente a la zona que se le ha especificado.

Se han implementado los siguientes métodos:

- **GetXDist:** método que permite acceder al vector permanente calculado en el método **Match**.
- **PartialMatcher:** método constructor de la clase. Se le pasan los *Iris patrón* y *muestra* y las coordenadas x del patrón por las que debe empezar y acabar.

4.2.2.3 Clase ThreadedMatcher

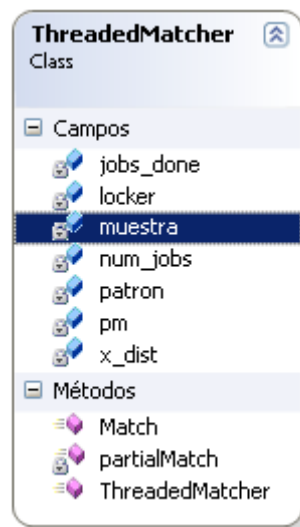


Figura 30: Diagrama de la clase *ThreadedMatcher*

La clase `ThreadedMatcher` se encarga de dividir el trabajo, ejecutarlo de forma paralela y reconstruir los resultados.

Para ello, divide el patrón en tantas porciones verticales como núcleos tenga el equipo en el que se ejecuta. Para averiguar el número de núcleos se utiliza la variable `Environment.ProcessorCount`. Si el equipo se basa en procesadores Intel con la tecnología `HyperThreading` esta variable devuelve un número que es el doble de la cantidad de núcleos presentes realmente. Se debe a que la tecnología `HyperThreading` simula dos núcleos lógicos en cada uno físico. Por ello, en el caso de los procesadores equipados con dicha tecnología se obtienen el doble de porciones que de núcleos físicos.

El método de comparación de iris presentado en este proyecto presenta la ventaja de que se puede tratar de forma independiente cada porción, puesto que la información de una no interviene en el procesamiento de las otras. Aprovechando esta ventaja, se le pasa cada porción a un `PartialMatcher` distinto (se crean tantos como porciones haya) y se ejecuta cada `PartialMatcher` en un hilo creado a tal efecto.

El hilo principal espera a que se hayan ejecutado todos los *PartialMatchers* de forma independiente y una vez que se han completado, recoge la información de distancias mediante sus respectivos métodos **GetXDist** para reconstruir el vector permanente completo. Una vez realizado esto, calcula la media y desviación típica de dicho vector, obteniendo la medida de similitud entre los dos irises.

A continuación, se describen los campos del *ThreadedMatcher*.

- **patron, muestra:** *Iris* que representan el patrón y la muestra
- **num_jobs:** int que contiene el número de porciones y, por lo tanto, el número de hilos y de trabajos que se deben realizar.
- **jobs_done:** int que contiene el número de hilos (*PartialMatchers*) que han terminado su trabajo. Se usa para saber cuando está lista la información para el cálculo de la medida de similitud.
- **pm:** array de *PartialMatchers* que contiene los *PartialMatchers* para poder manejarlos.
- **x_dist:** ArrayList (array dinámico) en el que se irán insertando los vectores parciales y que terminará conteniendo la información del vector permanente.
- **locker:** object con el atributo readonly (sólo lectura) que sirve para el control de acceso concurrente a los datos.

Los métodos contenidos en esta clase son:

- **Match:** método principal de la clase, que se encarga de crear un array de Threads, y asignar a cada uno un punto de ejecución contenido en el método **partialMatch**. Arranca cada Thread (hilo), y espera a que **jobs_done** sea igual a **num_jobs**. Una vez hecho esto, calcula la medida de similitud (desviación típica) y la devuelve.
- **partialMatch:** este método se encarga de invocar el método **Match** sobre cada uno de los *PartialMatchers* contenidos en el vector **pm**. Utiliza el cerrojo **locker**, para controlar el acceso al vector dinámico **x_dist** e ir añadiendo los vectores permanentes parciales a éste último. Cada vez que se hace esto, aumenta en una unidad la variable **jobs_done**, indicando que se ha completado uno de los trabajos pendientes.
- **ThreadedMatcher:** es el constructor de la clase. Se le pasan los *Iris patron y muestra*. Averigua el número de núcleos lógicos de los que dispone el sistema que, como se ha visto, pueden ser el doble del número de núcleos físicos, y divide el patrón en porciones. Al calcular el tamaño de las porciones, el método se asegura de que este contenga un número entero de retículas, con el fin de asegurar que se buscará en

toda la porción. Esto puede provocar que al final tengamos una porción más que el número de núcleos lógicos. Ésta contendrá la zona restante que no cabía en las demás porciones debido al tamaño de las retículas. Después de realizar esto, genera e inicializa el array **pm** pasándole los correspondientes *PartialMatchers*, con sus porciones asociadas.

4.2.3 Aplicaciones auxiliares

En este apartado se detallan una serie de aplicaciones que se han desarrollado, que permiten realizar las distintas fases del sistema de reconocimiento de iris.

4.2.3.1 Convertidor

Es una aplicación basada en Windows forms, es decir, una aplicación gráfica que permite seleccionar un directorio que contenga una serie de imágenes de iris en formato .bmp.

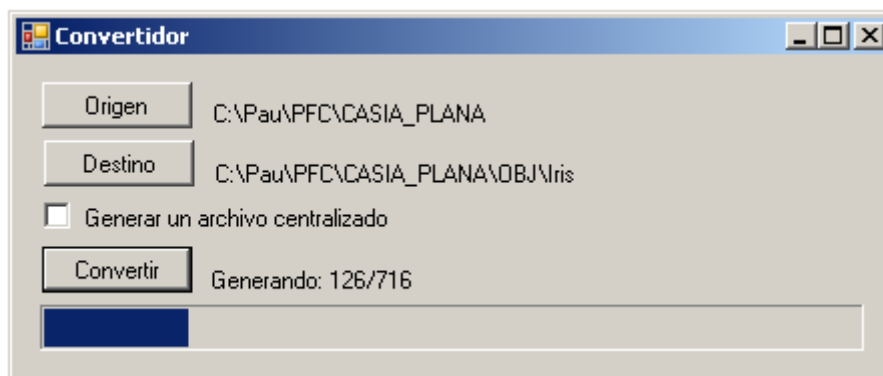


Figura 31: Ejemplo del convertidor en funcionamiento.

Seleccionando un directorio destino, permite escribir en dicho directorio los archivos de *Iris* (formato .irs) provenientes de las imágenes del directorio origen. También se encarga de seleccionar qué archivos se tomarán como patrones y cuáles como muestra. El proceso es sencillo ya que, de los irises encontrados en el directorio origen, seleccionará la primera ocurrencia para una persona como patrón, y las siguientes como muestra.

Para diferenciar las muestras de los patrones, crea dos subdirectorios (PATRONES y MUESTRAS) dentro del directorio destino, y guarda en cada uno de ellos los archivos correspondientes.

Marcando la opción “Generar un archivo centralizado”, permite escribir dos archivos de *Iris* centralizados (formato .irss), en el directorio destino seleccionado. Uno de ellos, el *PatronesCentralizado.irss*, contendrá los patrones en un solo archivo, mientras que el otro, el *MuestrasCentralizado.irss*,

Capítulo 4

contendrá las muestras. Estos archivos presentan la ventaja de que se pueden almacenar todas las muestras o patrones en un único archivo, facilitando su almacenamiento, lectura y procesado.

Cabe destacar, que en realidad lo que se escribe en dicho archivo es un array de *Iris*, lo que se tendrá en cuenta en el momento de su lectura.

4.2.3.2 Visor

Esta es otra aplicación basada en Windows forms, que permite la visualización de *Iris* almacenados en disco.

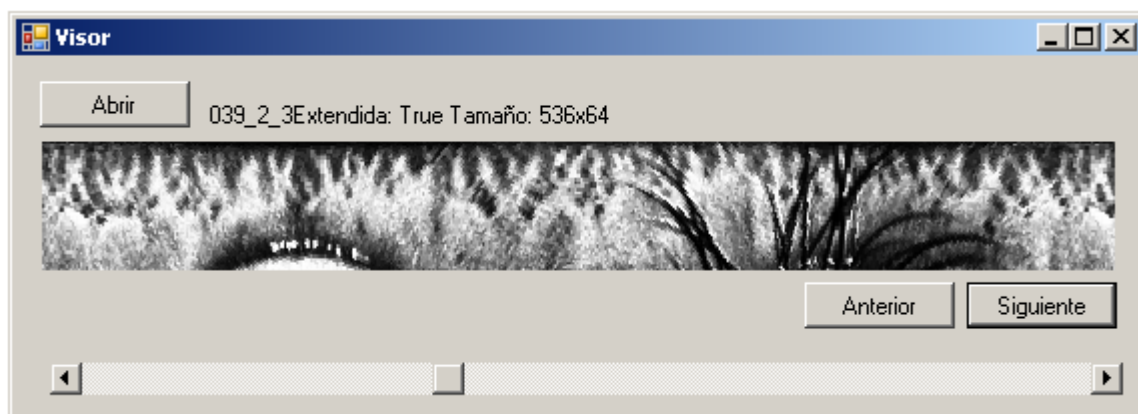


Figura 32: Ejemplo del visor en ejecución.

Mediante el botón “Abrir” se podrán seleccionar o bien un archivo individual de *Iris* (*.irs), o un archivo centralizado de *Iris* (*.irss). Con los botones “Anterior” y “Siguiete”, se podrá navegar a través del directorio que contiene los archivos de *Iris* individuales, o a través del archivo de *Iris* centralizado.

A través de una barra de *scroll*, se indica al usuario la posición que ocupa el Iris que se está viendo dentro del archivo centralizado o, en su caso, del directorio consultado. También permite seleccionar el iris que se desea ver.

4.2.3.3 Calculador

Es una aplicación Windows forms que permite calcular la medida de similitud entre los patrones y las muestras, generando un archivo de resultados centralizado.

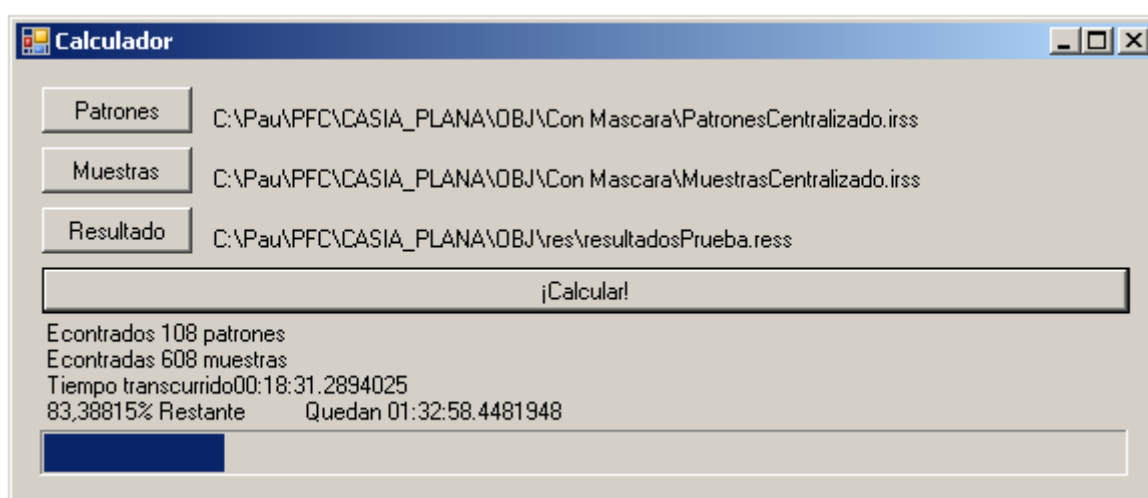


Figura 33: Ejemplo del calculador en ejecución.

Con el botón “Patrones”, se seleccionará el archivo de patrones centralizado, mientras que con el botón “Muestras”, el archivo de muestras centralizado. Con el botón “Resultado”, se seleccionará el archivo en el que se desea escribir los resultados de cada una de las comparaciones. De no existir este archivo, que es lo más probable, se generará a tal efecto.

Una vez seleccionados los tres archivos, bastará con presionar el botón “Calcular”, para que el programa comience a comparar de forma automática cada patrón con todas las muestras, escribiendo a medida los resultados en el archivo seleccionado.

Dado que, en general, existirán una gran cantidad de patrones y una cantidad aún mayor de muestras, el programa informa al usuario de las comparaciones que ha hecho hasta el momento, el tiempo empleado para hacerlas, el tiempo estimado en realizar el resto de comparaciones (en base al tiempo empleado para hacer las comparaciones hechas hasta el momento), y presentará de un modo gráfico, una barra de progreso que resume estos parámetros.

4.2.3.4 DemoMatcher

Esta es una aplicación basada en Windows forms que realiza el proceso de comparación presentándolo, en tiempo real, de una forma gráfica al usuario. Con el botón “Patrón”, se selecciona el patrón que se desea comparar. Con el botón “Muestra”, se selecciona la muestra con la que se quiere comparar. El proceso de comparación se inicia automáticamente al seleccionar la muestra.

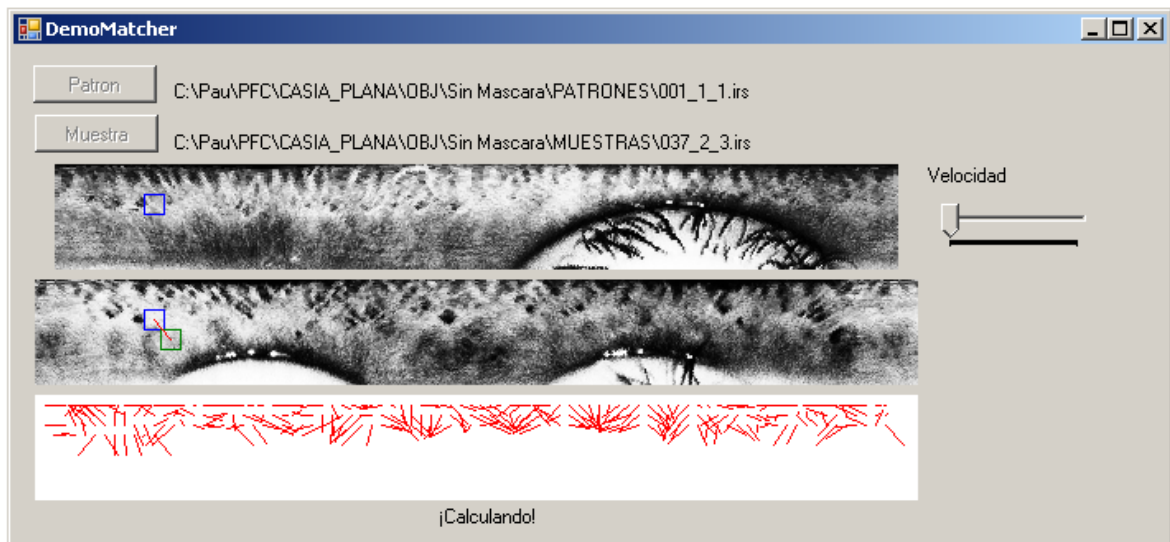


Figura 34: Ejemplo del DemoMatcher en funcionamiento.

En este punto, aparecen tres imágenes:

En la primera imagen se representa el patrón. Se puede observar sobreimpreso un cuadrado azul que materializa la posición de la retícula correspondiente al patrón. Ésta se irá moviendo sobre todo el patrón a medida que se vaya realizando la comparación.

En la segunda imagen se representa la muestra. Sobre ella aparecen dos cuadrados: un cuadrado azul que representa la posición relativa de la retícula correspondiente al patrón (ésta sólo está presente para observar el desfase entre la retícula del patrón y la propia de la muestra), y un cuadrado verde que es la retícula correspondiente a la muestra. Para cada posición de la retícula azul, se verá una posición, generalmente distinta, de la retícula correspondiente a la muestra (la verde), que corresponde a la mejor coincidencia de la retícula azul sobre la muestra. Por último, se representa una línea roja que une el centro de la retícula azul con el centro de la retícula verde. Esta línea sirve para materializar la magnitud del desplazamiento entre una y otra retícula.

Finalmente, existe una tercera imagen que se irá generando a medida que se realiza la comparación y que representa un registro gráfico de las líneas rojas que se pueden observar en la segunda imagen, o dicho de otro modo, representa el campo de vectores generado durante la comparación.

Mediante un control deslizante, se podrá seleccionar la velocidad a la que se mueve la retícula azul. Esto es útil para poder ralentizar la comparación y así poder observar mejor el proceso.

Una vez finalizada la comparación, aparecerá en la parte de abajo de la ventana, la desviación típica de la componente horizontal del campo de vectores generado que, como ya se ha dicho, es la medida de similitud entre el patrón y la muestra.

Cabe destacar que esta aplicación se podría usar para comparar dos irises. Sin embargo, dado que su propósito es ilustrar de manera gráfica el proceso de comparación, resulta lenta.

4.2.3.5 DemoThreaded

Esta aplicación, desarrollada también en Windows forms, permite comparar una muestra con un patrón. Para ello se pulsará en el botón “Patrón” para seleccionar el patrón, y en el botón “Muestra” para seleccionar la muestra. Pulsando el botón “Calcular”, aparece la medida de similitud del patrón con la muestra, y una indicación de si se considera que el iris pertenece a la misma persona.

Esta aplicación sí que está más orientada a su uso en un entorno “real”, puesto que emplea la clase *ThreadedMatcher* y, por lo tanto, realiza la comparación mucho más rápido que en el caso anterior.

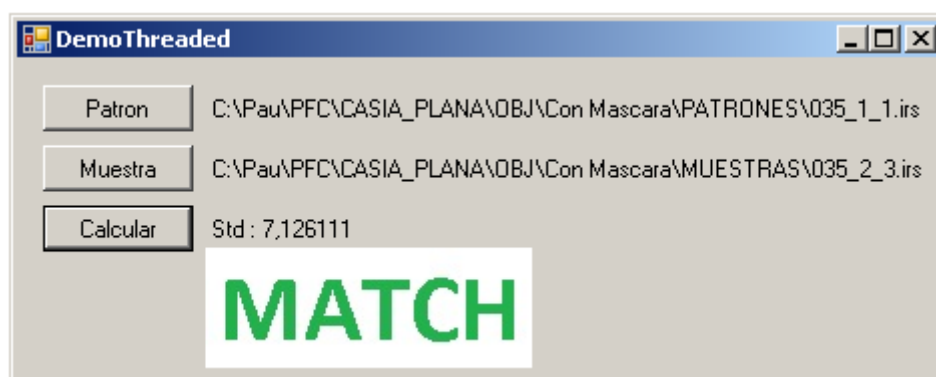


Figura 35: Imagen del DemoThreaded indicando que dos irises pertenecen a la misma persona.

4.2.3.6 VisorEstadísticos

Esta aplicación permite la visualización de los resultados obtenidos al usar la aplicación “Calculador”.

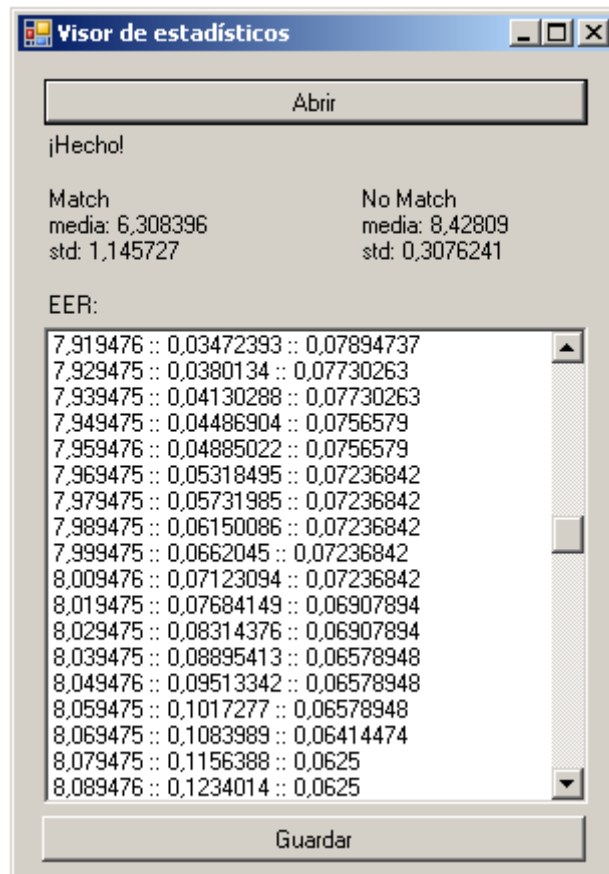


Figura 36: Imagen del visor de estadísticos.

A través del botón “Abrir”, se selecciona el archivo de resultados (*.ress) deseado. A continuación, la aplicación presenta en dos columnas la media y la desviación típica de las desviaciones típicas (medidas de similitud) obtenidas en cada comparación. En la primera columna (Match), aparecen los datos correspondientes a las comparaciones de los irises de una misma persona. En la segunda (No Match), aparecen los datos correspondientes a la comparación de un patrón y una muestra pertenecientes a personas distintas.

En el panel inferior se reflejan tres columnas. La primera corresponde a un umbral hipotético por debajo del cual una medida de similitud se consideraría correspondiente a la comparación de dos irises de la misma persona. La segunda, representa la FAR (False Acceptance Rate) para el umbral de la primera columna. La tercera, representa la FRR (*False Reject Rate*), también para el primer umbral.

Finalmente, a través del botón “Guardar” se permite guardar dichos resultados en un archivo de texto separado por comas e interpretable por Microsoft Excel, con el fin de poder utilizar este software para el procesamiento de los datos y la generación de gráficas.

5 : Resultados

Para evaluar el algoritmo implementado se ha empleado la base de datos CASIA, en su versión 1.0. De ella se han extraído 716 imágenes de 108 ojos distintos. De entre las 4 imágenes disponibles para cada ojo se tomó una imagen como patrón y entre una y tres como muestras. Esto se debe a que algunas imágenes no presentaban la calidad suficiente para hacer una segmentación correcta con la librería IrisBiometricsCS.

El sistema empleado en la evaluación es un Dell XPS 8300. El sistema está equipado con un procesador Intel core i7-2600 con cuatro núcleos a 3.4GHz y tecnología HyperThreading. Además cuenta con 8 GB de RAM a 1333MHz y un disco duro SATA de 7200 rpm. Cuenta también con un tarjeta gráfica ATI Radeon HD 5700, aunque esta no interviene en la comparación de irises. El sistema operativo instalado es Windows 7, en su versión de 64 bits.

Se han realizado comparaciones variando los tamaños de los macrobloques usados en la comparación (B_l), y del tamaño de la zona de búsqueda (N_h). En las siguientes tablas se muestran las EER (*Equal Error Rates*) logradas para cada valor de los mencionados parámetros, tomando como métrica de similitud la desviación típica (std) de la componentes horizontal (X) y vertical de los vectores de desplazamiento calculados. La tabla 3 corresponde a la comparación de imágenes que incluyen porciones de párpado, la tabla 4 corresponde a los resultados obtenidos usando una máscara fija que cubre un octavo de la altura de la imagen en la zona del párpado inferior y la mitad de la altura en la zona del superior. La tabla 5 presenta los resultados obtenidos usando máscaras calculadas de forma automática, usando para ello un algoritmo desarrollado en el contexto de un trabajo dirigido y presentado en el anexo A. A continuación se muestran ejemplos de enmascaramiento fijo y enmascaramiento automático.



Figura 37: Iris con máscara fija.



Figura 38: Iris con máscara automática.

Tabla 3: EER para comparaciones sin máscara.

<i>Bl</i> (píxeles)	<i>Nh</i> (píxeles)	EER		Tiempo Medio (s)
		X	Y	
8	8	0.1	0.1	0.09
	12	0.08	0.12	0.15
	16	0.08	0.13	0.29
12	8	0.08	0.09	0.09
	12	0.07	0.12	0.14
	16	0.07	0.12	0.24
16	8	0.07	0.1	0.06
	12	0.06	0.12	0.12
	16	0.06	0.13	0.18

Tabla 4: EER para comparaciones con máscara fija.

<i>Bl</i>	<i>Nh</i>	EER		Tiempo Medio (s)
		X	Y	
8	8	0.1	0.09	0.06
	12	0.08	0.11	0.12
	16	0.08	0.12	0.22
12	8	0.08	0.08	0.05
	12	0.06	0.1	0.10
	16	0.06	0.12	0.17
16	8	0.06	0.09	0.04
	12	0.05	0.11	0.08
	16	0.05	0.13	0.15

Tabla 5: EER para comparaciones con máscara automática.

<i>Bl</i>	<i>Nh</i>	EER		Tiempo Medio (s)
		X	Y	
8	8	0.1	0.08	0.06
	12	0.08	0.11	0.09
	16	0.08	0.13	0.22
12	8	0.08	0.08	0.05
	12	0.07	0.1	0.08
	16	0.07	0.12	0.13
16	8	0.07	0.09	0.04
	12	0.06	0.10	0.08
	16	0.06	0.13	0.15

En estas tablas, aparece también el tiempo medio empleado en cada comparación entre dos irises. Se analizará más detenidamente en las conclusiones.

Se puede ver que en general los resultados tomando la std de la componente X son más fiables que los que se obtienen tomando la componente Y. Esto se debe a que la componente X se ve fundamentalmente afectada por rotaciones del iris, a las que es inmune este algoritmo (siempre que se mantengan en la zona delimitada por *Nh*), mientras que la componente Y se ve afectada por deformaciones del iris debidas a la contracción de la pupila. Al ser estas deformaciones difíciles de modelar, el algoritmo no es capaz de corregirlas y por lo tanto se ve afectado por ellas.

Por otro lado se puede observar que cuanto más grande es el tamaño del bloque que se usa para las búsquedas (*Bl*) menor es la EER. Al ser más grande el tamaño del bloque, se utilizan más píxeles en cada comparación y por lo tanto más información. Tiene otra ventaja asociada, que es que, como siempre se usan solapamientos del 50%, al ser más grandes los bloques se puede cubrir la imagen con menos desplazamientos, acelerando el algoritmo. Sin embargo el tamaño del bloque tiene un límite, ya que cuantos menos bloques se usen, menos vectores desplazamiento se podrán calcular y por lo tanto se degradará la calidad de la std calculada.

En cuanto al tiempo que emplea el algoritmo para cada comparación, hay varios factores que afectan. Como ya se ha dicho el tamaño de los bloques (*Bl*) que se usan para la comparación de las imágenes tiene una importancia fundamental. Intervienen también otros factores, como el tamaño del vecindario (*Nh*) puesto que este determina el número de bloques que hay que comparar

con el de la imagen original. Otro factor que hace variar el tiempo consumido en cada comparación es el tamaño de la máscara que se puede emplear en cada una de las imágenes (patrón y muestra). Dado que las partes enmascaradas no se deben comparar se calcula la suma de las máscaras de las dos imágenes y se descartan las zonas recubiertas. Esto reduce el área de búsqueda y por lo tanto el tiempo de respuesta. Finalmente, el hecho de que se haya utilizado la tecnología .NET para implementar este algoritmo, presupone que se va a emplear fundamentalmente en sistemas Windows (si bien es verdad que existen implementaciones del CLR para otras plataformas). Esto introduce un problema de predictibilidad de los tiempos de ejecución, ya que Windows no es un sistema operativo de tiempo real y es difícil controlar los accesos al procesador. Nunca se sabe si se va a ejecutar un análisis por parte del antivirus, descargar una actualización o cualquier proceso que pueda ralentizar la ejecución del algoritmo de reconocimiento.

En cada una de las tablas se ha resaltado en verde los mejores resultados en cuanto a EER. Se puede ver que el mejor resultado absoluto (5%) se obtiene en el caso de máscara fija para valores de $Bl = 16$ y $Nh = 12$ o 16 . Se empleará preferiblemente $Nh = 12$ ya que el tiempo necesario para la comparación de los iris se reduce a la mitad. Este resultado se obtiene para un valor umbral de la desviación típica de 8.045.

Estos resultados se pueden comparar con los resultados obtenidos por Monro y Rakshit que obtuvieron una EER del 0.7%, sorprendentemente baja, sobre todo si se compara con los resultados obtenidos por el International Biometric Group, que esponsorizado por el U.S. Department of Homeland Security, comparó distintos sistemas comerciales y encontraron EERs de entre el 1.5% y el 2.5% [31].

Otro factor importante a tener en cuenta es la estabilidad de las tasas de error en torno a este valor, ya que puede resultar interesante disminuir una de las tasas (FAR o FRR) sin perjudicar demasiado a la otra. A modo de ejemplo, si lo que se desea es reducir la tasa de falsa aceptación (FAR) por debajo del 1% la tasa de falso rechazo (FRR) aumentará hasta el 7,5%. Sin embargo el reducir la FRR por debajo del 1% aumentará la FAR por encima del 80%. Este hecho se ilustra en la figura 39 que se muestra a continuación.

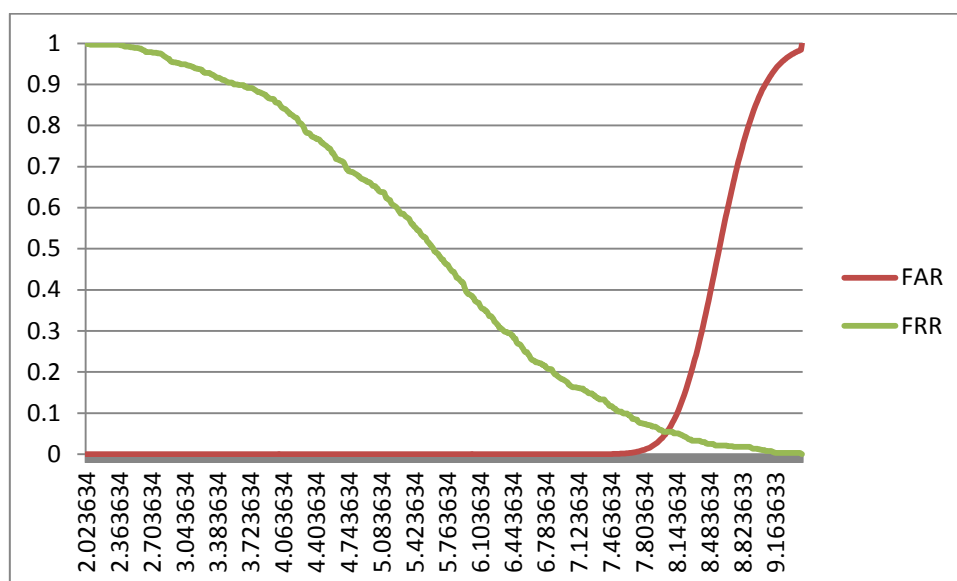


Figura 39: FRR y FAR en función del umbral seleccionado.

6: Conclusiones y trabajos futuros

El trabajo aquí realizado consiste en la implementación de un algoritmo novedoso de identificación mediante iris, propuesto por D. M. Monro and S. Rakshit. Estos investigadores han publicado resultados en [1], dónde alcanzaban una EER de tan sólo el 0,7%.

Aquí la mínima EER que se ha conseguido ha sido del 5%. Esto se puede deber a diversos factores.

En primer lugar, el preprocesado de los irises se ha hecho con técnicas presumiblemente distintas a las utilizadas para el artículo original. Cabe destacar que el control de calidad de la imagen que proporciona la librería IrisBiometricsCS ha sido desactivado. Sólo se han rechazado imágenes que debido a su pobre calidad no podían ser segmentadas correctamente y se han conservado imágenes que presentaban enfoques deficientes u otros problemas como oclusiones de gran tamaño al iris, ya que la cuasi totalidad de las imágenes presentaban algún tipo de defecto y el descartarlas habría supuesto un conjunto insuficiente de muestras.

En segundo lugar, aunque el artículo [1] presenta en detalle el algoritmo utilizado, no se proporciona una implementación del mismo, con lo que podría existir alguna diferencia entre la implementación realizada por el equipo de D. M. Monro y la realizada en este trabajo.

De la implementación aquí realizada se pueden extraer varias conclusiones. La primera, y quizás más llamativa es el tiempo empleado en la comparación de iris. Los sistemas de reconocimiento de iris se pueden emplear para la identificación o la verificación de un usuario. El primer caso consiste en comparar el iris de dicho usuario con una base de datos, con el fin de

determinar su identidad. En el segundo caso, el usuario clama una identidad y se trata de averiguar si dice la verdad.

El algoritmo aquí implementado, tarda 0.08 segundos en realizar una comparación. En el caso de la verificación, en el que el usuario aportaría su identidad mediante por ejemplo, una tarjeta electrónica, este tiempo es despreciable. Si se trata del caso de la identificación de un ciudadano español, en el peor de los casos habría que realizar algo más de cuarenta y siete millones de comparaciones, lo que implicaría un tiempo de más de cuarenta y tres días.

Esta es una de las principales desventajas de este algoritmo frente a otros basados en *IrisCodes*, ya que estos realizan la mayor parte del trabajo en el momento del reclutamiento, y a la hora de realizar la identificación, sólo deben comparar los códigos. En general esto se resume a comparar un conjunto de bits y esto es extremadamente rápido, ya sea usando procesadores de propósito general o circuitos dedicados.

Otro punto importante es la diferencia de EER obtenida en esta implementación con respecto a la original. La principal explicación reside en la diferencia de preprocesado y criterios de aceptación de las imágenes. Esto demuestra que en un sistema de reconocimiento mediante iris no sólo importa el algoritmo de comparación propiamente dicho, sino que toda la cadena, desde la toma, pasando por la segmentación y el preprocesado, la extracción de características y finalmente el proceso de decisión son cruciales para minimizar las tasas de error.

Los principales problemas encontrados en esta implementación son los tiempos empleados en el proceso de comparación de iris y las tasas de error obtenidas. El proceso de comparación se podría acelerar si se usase hardware dedicado o tecnologías como CUDA, que permite utilizar la potencia de cálculo y las posibilidades de paralelización de las GPUs de Nvidia. Ya se ha demostrado que el problema es fácilmente separable y por lo tanto computable de forma distribuida. Otra solución sería emplear un *cluster* de computación, pero su elevado coste podría ser redhibitorio. Por otro lado las tasas de error se podrían mejorar refinando el preprocesado. Se estima que los mejores resultados se obtendrían siendo más restrictivos en cuanto a la calidad de las imágenes, y mejorando la detección de oclusiones al iris.

Por último, cabe destacar que este trabajo es fundamentalmente teórico, pero se podría implantar en un sistema real, que incluyera la fase de captura, lo que permitiría controlar la calidad de las imágenes utilizadas.

Bibliografía

- [1] Monro, D. M. and Rakshit, S., "Rotation-Independent Iris Matching by Motion Estimation". *IEEE International Conference on Image Processing, ICIP 2007*, vol. 2, pp. 401-404, 2007.
- [2] Johannes Sobotta, Reinhard Putz, Reinhard Pabst, Renate Putz "Atlas de anatomía humana: Cabeza, cuello y miembro superior", Tomo 1, pages 352-378, 2006.
- [3] Jack J. Kanski, "Oftalmología Clínica", Elsevier España, 2004
- [4] J. G. Daugman, "The importance of being random: statistical principles of iris recognition," *Pattern Recognition* 36, 279-291 (2003).
- [5] Daugman, J., "Recognizing persons by their iris patterns". *Biometrics: Personal Identification in Networked Society*, Jain, A., Bolle, R. y Pankanti, S. (eds.), pp. 103–122, Springer, 1999.
- [6] Daugman, J., "High confidence visual recognition of persons by a test of statistical independence." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15(11), pp. 1148-1161, 1993.
- [7] Westmoreland, B., Lemp, M. y Snell, R., "Clinical Anatomy of the Eye", Wiley-Blackwell, 2ª Ed., 1998.
- [8] John Daugman. Biometric personal identification system based on iris analysis. U.S. Patent No.5, 291, 560, March 1994.
- [14] Flom, L. and Safir, A., US. Patent No. 4641349, U.S. Government Printing Office, Washington, DC, 1987.

- [10] García Vega, M.A., “Cuando el ‘broker’ es el iris”, El País, p 17 (Negocios), 12 de junio de 2011.
- [11] John G Daugman, Cathryn Downing, Epigenetic randomness, complexity, and singularity of human iris patterns, *Proceedings of the Royal Society, B*, 268, Biological Sciences, pp 1737 – 1740, 2001.
- [12] R. Wildes, Iris Recognition: An Emerging Biometric Technology, *Proceedings of the IEEE*, vol 85, no. 9, pp. 1348-1363, 1997.
- [13] John Daugman, How Iris Recognition works, *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, january 2004
- [14] Wildes et al. Automated, non-invasive iris recognition system and method. U.S. Patent No. 5,572,596 and 5,751,836, 1996 and 1998.
- [15] ISO/IEC Standard 19794-6. Information technology - biometric data interchange formats, part 6: Iris image data. Technical report, International Standards Organization, 2005.
- [16] J. R. Matey, O. Naroditsky, K. Hanna, R. Kolczynski, D. Lolacono, S. Mangru, M. Tinker, T. Zappia, and W. Y. Zhao. Iris on the Move™: Acquisition of images for iris recognition in less constrained environments. *Proceedings of the IEEE*, 94(11):1936–1946, 2006.
- [17] ISO/IEC 4th Working Draft 29794-6, Information Technology – Biometric sample quality – Part 6: Iris image.
- [18] N. Kalka, J. Zuo, N. Schmid and B. Cukic., “Image quality assessment for iris biometric”. *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6202, Biometric Technology for Human Identification III, 2006.
- [19] Hugo Proença and Luis A. Alexandre. A method for the identification of noisy regions in normalized iris images. In *Int. Conf. on Pattern Recognition*, pages 05–408, August 2006.
- [20] Emine Krichen, Sonia Garcia-Salicetti, and Bernadette Dorizzi. A new probabilistic iris quality measure for comprehensive noise detection. In *Biometrics: Theory, Applications, and Systems*, Sept 2007.

Bibliografía

- [21] Yuanning Liu, Senmiao Yuan, Xiaodong Zhu, and Qingliang Cui. A practical iris acquisition system and a fast edges locating algorithm in iris recognition. In IEEE Instrumentation and Measurement Technology Conference, pages 166–168, 2003.
- [22] Richard P. Wildes. Iris recognition: An emerging biometric technology. Proceedings of the IEEE, 85(9):1348–1363, September 1997.
- [23] John Canny, A computational approach to edge detection, IEEE Transactions on patterns analysis and computer intelligence, vol. PAMI-8, no. 6, november 1986
- [24] Paul V. C. Hough, Method and means for recognizing complex patterns, U.S. Patent 3,069,654
- [25] Pan Lili and Xie Mei. The algorithm of iris imageprocessing. In Fourth IEEE Workshop on Automatic Identification Technologies, pages 134–138, October 2005.
- [26] Bradford Bonney, Robert Ives, Delores Etter, and Yingzi Du. Iris pattern extraction using bit planes and standard deviations. In Thirty-Eighth Asilomar Conference on Signals, Systems, and Computers, volume 1, pages 582–586, November 2004.
- [27] Ya-ping Huang, Si-wei Luo, and En-yi Chen. An efficient iris recognition system. In Int. Conf. of Machine Learning and Cybernetics, volume 1, pages 450-454, November 2002.
- [28] Zhenan Sun, Tieniu Tan, and Yunhong Wang. Robust encoding of local ordinal measures: A general framework of iris recognition. In Proc. BioAW Workshop, pages 270–282, 2004
- [29] Erik Rydgren, Thomas Ea, Frederic Amiel, Florence Rossant, Amara Amara, and Carmen Sanchez- Ávila. Iris features extraction using wavelet packets. In Int. Conf. on Image Processing, pages II:861–864, 2004.
- [30] Jason Thornton, Marios Savvides, and B.V.K. Vijaya Kumar. An evaluation of iris pattern representations. In Biometrics: Theory, Applications, and Systems, Sept 2007.
- [31] International Biometric Group. Independent Testing of Iris Recognition Technology, Final Report. Mayo 2005. <http://www.hSDL.org/?view&did=464567>
- [32] Coomonte Belmonte, Rafael. Sistema de reconocimiento de personas mediante su patrón de iris basado en la transformada wavelet. Proyecto fin de Carrera. Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad Politécnica de Madrid. Mayo 2006.

- [33] José Antonio González Seco. El lenguaje de programación C#. <http://dis.um.es/~bmos/privado/bibliografia/LibroCsharp.pdf>

Anexo A: Detección de párpados en imágenes de iris.

1. Introducción

En los últimos años el reconocimiento basado en el patrón del iris ocular ha experimentado un gran auge debido a los excelentes resultados obtenidos y al gran interés que están mostrando algunos sectores económicos para incorporar dicha técnica a sus sistemas de identificación. La idea de utilizar el patrón del iris para reconocer a un individuo fue inicialmente propuesta por el oftalmólogo Frank Burch en 1936. En 1985, los doctores Leonard Flom y Aran Safir, oftalmólogos, pusieron de manifiesto que no hay dos iris semejantes, aunque no fue hasta 1987 que les fue concedida una patente para el concepto de reconocimiento mediante iris ocular [1]. Con el objetivo de automatizar el sistema, los doctores contactaron con el profesor John G. Daugman para que fuera éste quien desarrollara los algoritmos necesarios para realizar el reconocimiento biométrico a través del patrón del iris. Estos algoritmos fueron patentados en 1994 y son la base de todos los sistemas de reconocimiento por iris existentes.

En el contexto de la identificación biométrica, los patrones existentes en el iris son considerados una de las alternativas más fiables, pues el iris es único y diferenciador para cada persona y no cambia durante toda la vida. Según diversos estudios [2, 3], en el patrón del iris hay gran cantidad de información que permite identificar unívocamente a una persona. Aunque la coloración y la estructura del iris están genéticamente ligadas, los detalles de los patrones no lo están. El iris se desarrolla durante el crecimiento prenatal de acuerdo a un estricto proceso de formación. Antes del nacimiento, se produce la degeneración, que da como resultado la apertura de la pupila y los patrones

únicos del iris [4]. Aunque genéticamente idénticos, los iris de un individuo son únicos y estructuralmente distintos, lo que permite utilizar el iris para propósitos de reconocimiento. Dada la extremadamente alta unicidad de las técnicas de reconocimiento de iris, resulta posible obtener tasas de error nulas. Otra característica que refuerza el interés de estas técnicas es que los datos necesarios se obtienen de forma no invasiva.

De acuerdo con la estructura general de los sistemas de identificación biométrica, los sistemas de reconocimiento de iris presentan cuatro etapas principales: captura de los datos biométricos, preprocesado de los datos capturados, extracción de las características propias del usuario y, finalmente, verificación de las características extraídas con un patrón previamente almacenado.

Este trabajo dirigido, cuyo objetivo principal es detectar y eliminar las partes ocluidas del iris debidas a los párpados, se centrará en la etapa correspondiente al preprocesado, y dentro de ésta, en la eliminación mediante una máscara de la información de los párpados que queda dentro de la frontera del iris. Aparte de implementar y describir el mecanismo desarrollado (véase Sección 3), se expondrán los resultados obtenidos tanto en lo referente a la detección de párpados propiamente dicha, como a su influencia en los resultados obtenidos en términos de tasas de error, para lo que se considerará un algoritmo de identificación de iris basado en técnicas MPEG [5] (véase Sección 4). Para completar el trabajo, en la Sección 5, se indicarán las conclusiones principales obtenidas, así como las líneas futuras a considerar.

2. Etapas de los sistemas de reconocimiento de iris

Tal y como se ha comentado previamente, los sistemas de reconocimiento de iris presentan cuatro etapas o bloques principales: captura, preprocesado, extracción de características y verificación.

Inicialmente, se captura una imagen del iris mediante una cámara de iris, sensor específicamente diseñado para llevar a cabo esta tarea.

Tras la captura, se aplican a la imagen una serie de operaciones de procesamiento necesarias para poder llevar a cabo la etapa posterior, correspondiente a la extracción de características. Entre estas operaciones, destacan las siguientes:

- Segmentación o localización del iris, con la que se detectan los bordes del iris dentro de la imagen (frontera iris-esclerótica y frontera iris-pupila).
- Operaciones de mejora de la calidad de la imagen, tales como la ecualización.
- Normalización, necesaria para compensar el tamaño del iris, debido a la distancia del sujeto respecto al objetivo y a la dilatación o contracción de la pupila.
- Eliminación de información no biométrica del iris, entre la que se puede incluir tanto la información correspondiente a los párpados como a las pestañas.

La fase de extracción de características es una de las fases fundamentales del sistema de reconocimiento de iris. En ella, se somete la imagen del iris resultante de la fase de preprocesado a una serie de transformaciones matemáticas hasta obtener un vector de datos con información suficiente para los propósitos de reconocimiento. Dicho vector se denomina vector de características o *iriscode*. Básicamente, el algoritmo de extracción de características se desarrolla en dos fases. Inicialmente, se filtra la imagen (por ejemplo, mediante un filtro de Gabor bidimensional en el caso del sistema desarrollado por el profesor John G. Daugman [6]) y posteriormente, se genera el vector de características aplicando un proceso de codificación.

Finalmente, una vez extraído el vector de características de cada muestra capturada, se ha de comparar con un patrón, que es un vector de características previamente almacenado en una base de datos y obtenido tras una fase de reclutamiento. Para realizar la comparación se pueden considerar distintas medidas de distancia (por ejemplo, distancia de Hamming). Es fundamental tener en cuenta que existe una máscara que indica si el iris contiene o no información biométrica. Esta máscara ha de considerarse a la hora de realizar las comparaciones, pues los valores del vector asociados a partes del iris que no contienen información biométrica no habrán de tenerse en cuenta. Para terminar, un concepto importante cuando se habla de comparación en este tipo de sistemas, es que no se trata de una comparación binaria o de igualdad, sino que la variación de las muestras, bien por variaciones en la captura o bien por leve variación de las características del sujeto, hacen que los resultados de la comparación sean ligeramente diferentes aunque todas las muestras pertenezcan al mismo usuario. Según esto, se podría decir que el resultado de la comparación es una probabilidad de semejanza y, por tanto, para determinar el éxito o fracaso de la comparación, habrá que determinar un umbral en esa probabilidad.

3. Método de detección de párpados

Tal y como se ha comentado en la sección anterior, en un sistema de reconocimiento biométrico basado en iris, es fundamental la correcta segmentación de éste con el fin de poder procesarlo sin considerar las porciones que no contienen información.

En este caso, se usará una librería proporcionada por el Grupo Universitario de Tecnologías de la Identificación (GUTI) que se encarga de segmentar el iris. Para ello, se modela la región del iris como dos círculos no concéntricos, de los cuales el exterior representa la frontera iris-esclerótica, y el interior la frontera iris-pupila. Esta librería, sin embargo, no incluye mecanismo alguno de detección de párpados, por lo que existen partes del iris ocluidas por los párpados inferior y superior, al igual que por las pestañas. Esto provoca que los algoritmos que se emplean a continuación para la extracción de características del iris consideren los párpados como partes integrantes de éste, lo cual induce errores. Así, resulta fundamental eliminar las partes de la imagen que no contienen información, considerando además que su extensión puede variar en función de lo profundo que sea el parpadeo en el momento de la captura. Por todo lo mencionado anteriormente, en este trabajo dirigido se ha implementado una solución que permita detectar las oclusiones del iris debidas a los párpados. Aparte de los

párpados, habría que detectar también las pestañas, pero eso es algo que queda fuera del ámbito del mismo.

Para detectar los párpados se considerará un método que consta de dos fases, que serán explicadas en detalle en las siguientes Subsecciones. Inicialmente, se aplica un filtrado de Canny al iris segmentado para realzar los bordes, y en segundo lugar, se procede a detectar los bordes correspondientes al párpado. Esta tarea no es trivial ya que el iris contiene zonas de alto contraste que producen bordes que no corresponden al párpado pero que también se ven realzados por el filtrado. Para paliar los efectos de estas zonas, se aplica un filtrado de mediana previo al realce de los bordes con la intención de minimizar los bordes que aparecen dentro del iris y que no corresponden a los límites del párpado.

3.1. Filtrado de Canny

En 1986, J.F. Canny publicó un artículo [7] sentando las bases y demostrando la idoneidad de su método para la detección de bordes diferenciales en imágenes digitales en 2D. Concretamente, el método consiste en tres fases: filtrado, decisión inicial e histéresis.

La primera fase se basa en un filtrado de convolución de la derivada primera de una función gaussiana normalizada sobre la imagen realizada en dos direcciones, horizontal y vertical. Esta función consta de dos parámetros fundamentales, su media (m) y desviación típica (s). En este caso se ha elegido una media nula, ya que de otra forma el filtrado produciría un desplazamiento espacial, poco deseable a la hora de hacer coincidir la máscara obtenida con la imagen original. La desviación típica tiene un valor unitario. Este parámetro se ha elegido tras realizar varios experimentos y es el que, a priori, ofrece el mejor resultado en cuanto al realce de los bordes de los párpados sin producir artefactos indeseados. Valores menores producen muchos artefactos e incluso imposibilitan la detección de los bordes, y valores mayores no realzan suficientemente estos últimos. Esto se debe a que para desviaciones típicas grandes, la derivada primera de la función gaussiana no presenta “picos” lo suficientemente grandes como para realzar de forma efectiva los bordes. Por último, cabe destacar que al trabajar en un entorno discreto, se usa una versión discreta sobre un rango acotado entre $-5s$ y $5s$ de esta derivada, en la que los valores se toman en variaciones de un pixel. Esto produce un vector de once elementos cuyos valores se pueden observar en la Figura 1.

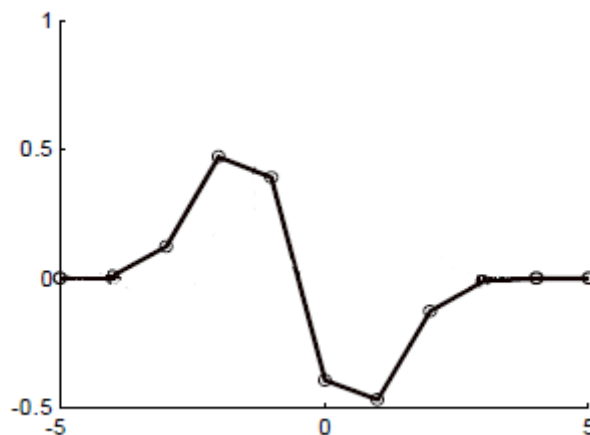


Figura 1.- Valores del vector usado en la convolución

Al aplicar la convolución en cada una de las direcciones se obtienen dos matrices, dx y dy . Cada una de ellas contiene una imagen cuyos niveles de gris representan la variación entre píxeles colindantes, ya sea en la dirección horizontal o vertical respectivamente. Una vez realizado este proceso se procede a una primera toma de decisiones en la que se identifican los píxeles candidatos a borde. En concreto se consideran aquellos que cumplen:

$$dx(i,j) > dx(i-1,j) \text{ y } dx(i,j) > dx(i+1,j) \text{ o bien } dy(i,j) > dy(i,j-1) \text{ y } dy(i,j) > dy(i,j+1).$$

Una vez hecho esto se procede a calcular la energía de cada uno de estos candidatos, definida como la raíz cuadrada de la suma de los cuadrados de los valores de los píxeles de estas matrices.

Para terminar, se aplica una función de histéresis basada en dos umbrales, superior (Th) e inferior (Tl). Se consideran parte del borde definitivo aquellos píxeles cuya energía es superior a Th y parte del fondo definitivo aquellos píxeles cuya energía es inferior a Tl . Asimismo, se consideran parte del borde definitivo aquellos píxeles que se encuentran en un vecindario de 3×3 con respecto a los bordes definitivos y cuya energía se encuentra entre los dos umbrales citados. En este caso, se ha determinado experimentalmente que los valores $Tl = 0.2$ y $Th = 0.3$ son valores que producen, en general, bordes suficientemente conexos sin definir como bordes demasiados píxeles correspondientes a la zona del iris. Al final de este proceso se obtiene una imagen binarizada en la que los píxeles correspondientes a bordes definitivos tienen un valor de 1 (blanco) y los correspondientes a fondo definitivo de 0 (negro). En la Figura 2 se puede observar un ejemplo de imagen binarizada.

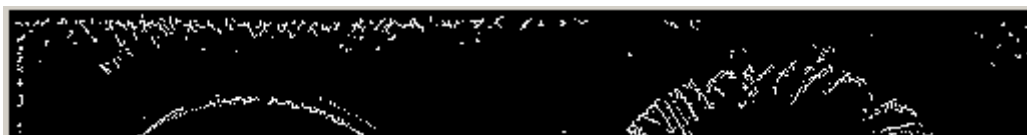


Figura 2.- Ejemplo de imagen binarizada

Cabe destacar que la zona correspondiente al párpado superior (a la derecha de la imagen) contiene un conjunto de bordes diagonales que se deben a la presencia de

pestañas. Estos bordes provocan graves problemas a la hora de detectar los párpados. En el siguiente epígrafe se verá cómo se resolvió este problema.

3.2. Detección de bordes

Una vez obtenida una imagen binarizada con los bordes resaltados, queda detectar cuáles de éstos pertenecen al párpado y cuáles no. Para ello, se emplean las características espaciales de los párpados en la imagen. Estas son fundamentalmente tres, y se basan en tres hipótesis. La primera, es que los párpados se sitúan en el borde inferior de la imagen: la segunda, es que los centros de los párpados se encuentran a un cuarto y tres cuartos aproximadamente de la imagen en sentido horizontal, y la tercera, es que los párpados tienen forma de curva que empieza y termina en el borde inferior de la imagen.

De acuerdo a esto, el algoritmo comienza buscando, desde la izquierda de la imagen, un grupo de al menos tres píxeles consecutivos que se encuentren en el borde inferior de la imagen siguiendo un patrón ascendente. Una vez encontrados, se van recorriendo y almacenando los píxeles consecutivos en una dirección (ascendente, horizontal o descendente) que se va prediciendo en función de los detectados anteriormente. Como se puede observar en la Figura 2, los bordes no siempre son conexos, lo que obliga al algoritmo a interpolar aquellos puntos que faltan. Así, se lleva una cuenta del número de píxeles consecutivos que se han interpolado para, a partir de un umbral (diez píxeles), parar el algoritmo y así evitar interpolar un número demasiado elevado de píxeles, algo que podría conducir a desviarse del borde. Se considera que se ha detectado un párpado cuando la trayectoria nos devuelve al borde inferior de la imagen. Para la detección del párpado inferior se emplea un método similar, aunque modificado para tener en cuenta la presencia de posibles pestañas que llevarían a error al algoritmo original. Estas modificaciones consisten principalmente en la inclusión de condiciones más restrictivas para poder cambiar de dirección ascendente a descendente, y en la comprobación de píxeles más exteriores (situados más arriba en la imagen) en la cercanía de un pixel ya detectado como borde.

El algoritmo devuelve una matriz de booleanos que se construye rellenando con valores lógicos igual a *true* los píxeles que se encuentran entre el borde inferior de la imagen y el borde de los párpados detectado. Esta matriz representa la zona enmascarada, correspondiente a los píxeles que están en el estado *true*.

4. Resultados

Una vez descrito en detalle el método de detección de párpados desarrollado, se procederá a exponer los resultados. Por un lado, se analizará la influencia de los parámetros de filtrado en la detección de párpados, y por otro lado, se analizará la influencia del mecanismo implementado en los resultados globales de un sistema completo de reconocimiento de iris. Para ello, se atenderá a los valores de tasa de error y se considerará un algoritmo de reconocimiento mediante iris basado en técnicas

MPEG [5]. La base de datos a utilizar en ambos casos es la base de datos CASIA v1.0, que contiene 756 imágenes de iris obtenidas a partir de 108 sujetos.

4.1. Influencia de los parámetros de filtrado en el resultado de la detección

A continuación se muestran una serie de capturas de pantalla que permitirán ilustrar la influencia de los distintos parámetros de filtrado en la imagen binarizada. Para ello, se muestra primero el resultado obtenido considerando los parámetros estándar, y a continuación, el resultado obtenido tras la optimización de dichos parámetros. Las zonas de interés de la imagen binarizada se han resaltado mediante rectángulos verdes, y los parámetros modificados durante la optimización, mediante flechas rojas.

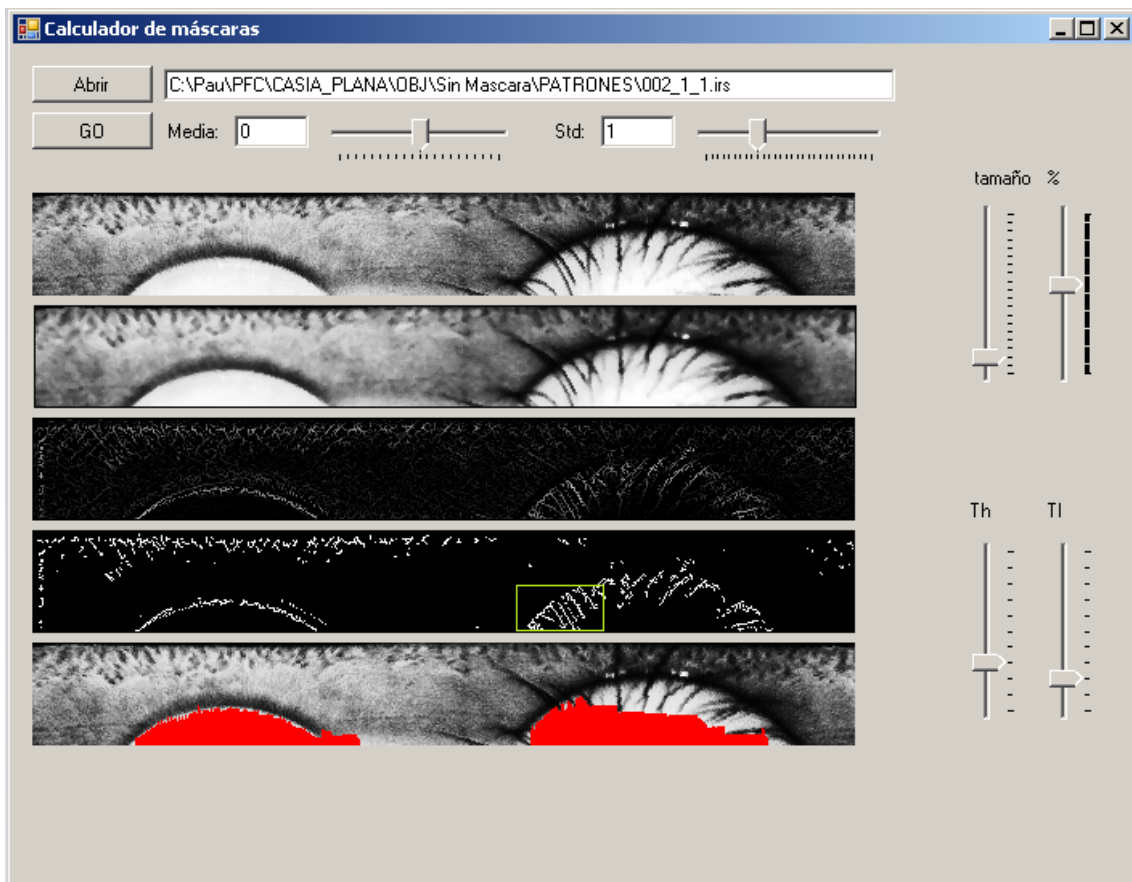


Figura 3.1.- Tratamiento estándar de la imagen de CASIA 002_1_1.

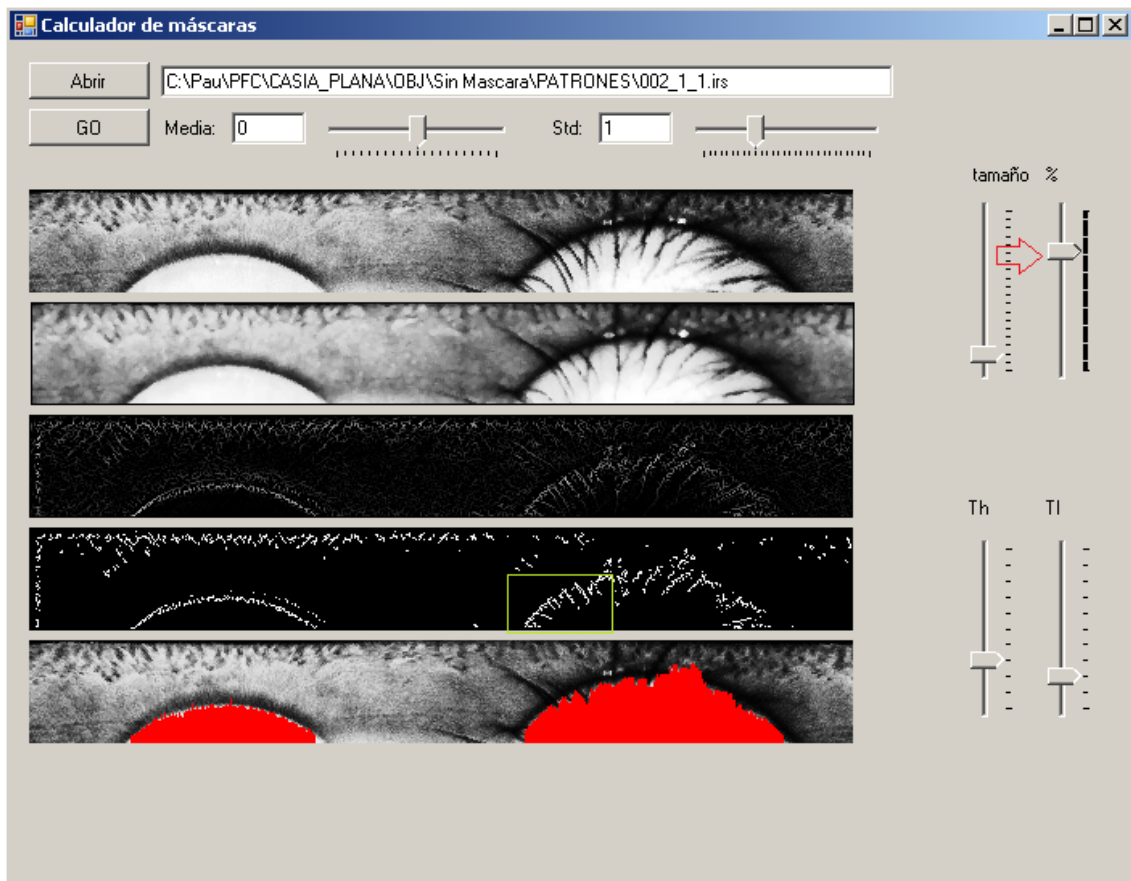


Figura 3.2.- Tratamiento optimizado de la imagen de CASIA 002_1_1.

En el caso de la primera figura, se observa que la detección del párpado inferior es correcta, mientras que el párpado superior no se ha detectado correctamente. En el segundo caso, se ha cambiado el filtro de mediana por un filtro de percentil 80. Esto aclara la imagen y, por tanto, disminuye el contraste existente entre pestañas y párpados, lo que produce un resultado mucho más deseable en el párpado superior sin por ello deteriorar la calidad del inferior. Nótese en particular la disminución de los bordes presentes en la imagen de histéresis (marcado en verde en ambas Figuras, en la cuarta imagen comenzando por arriba).

Considérese a continuación un caso distinto (véanse Figuras 4.1 y 4.2). En este otro caso, se puede observar que al usar los parámetros estándar no se obtiene un buen resultado con ninguno de los párpados. Uno de los problemas de la muestra de iris utilizada es que el párpado inferior no cumple la hipótesis de estar pegado al borde inferior de la imagen (este hecho se puede apreciar en la imagen de histéresis de la Figura 4.1, cuarta imagen comenzando por arriba). Esto se debe a un bajo contraste en las zonas inferiores del párpado. Dicho problema se ha resuelto aplicando la operación de convolución con una función gaussiana de media 2.5, lo que provoca un desplazamiento hacia abajo de toda la imagen. Por otro lado, el borde del párpado superior es poco conexo y hay muchas pestañas que molestan. Dicho problema se ha resuelto cambiando el filtro de mediana por otro de percentil 80, como en el caso anterior, pero esta vez ha sido necesario aumentar además el tamaño del filtro a 5x5 en lugar de 3x3, lo que acentúa aún más el suavizado. Por último, se han rebajado los

Anexo A

niveles tanto superior como inferior del proceso de histéresis, lo que produce que más puntos sean retenidos como borde y que aumente la densidad de éste.

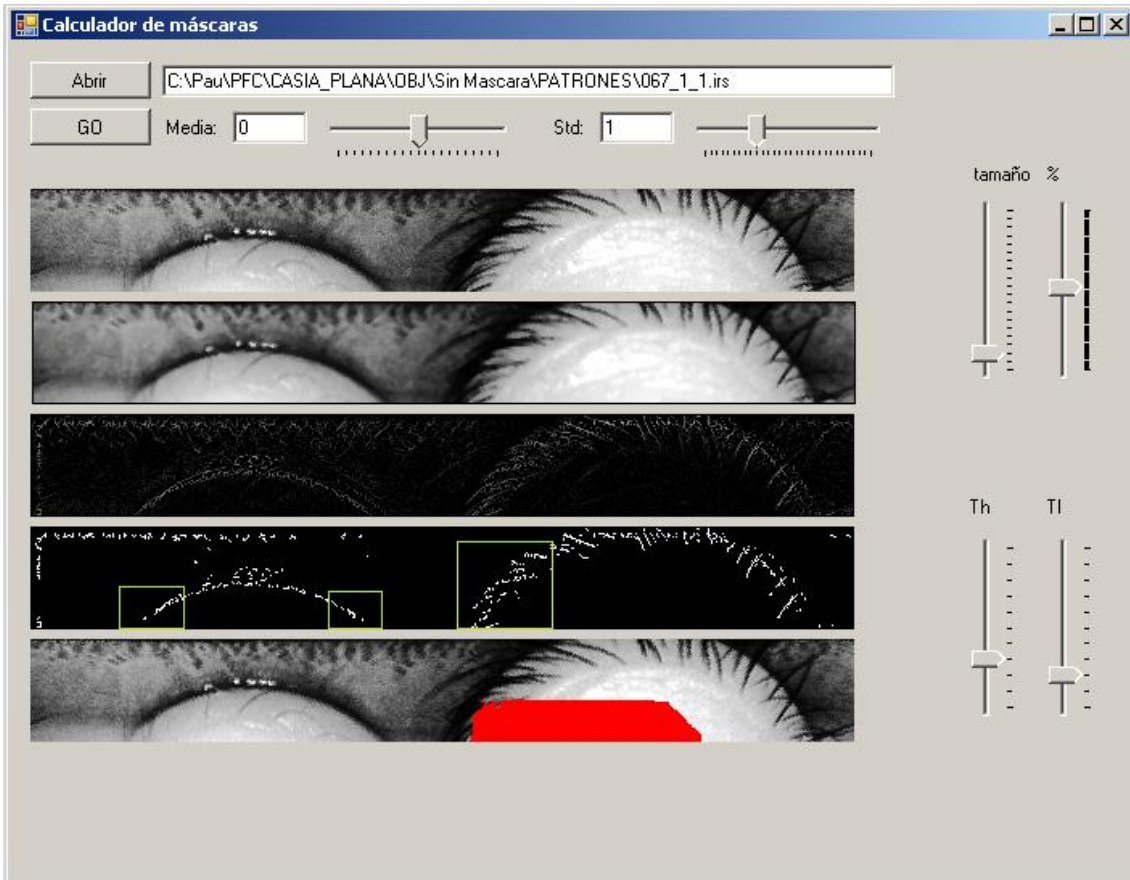


Figura 4.1.- Tratamiento estándar de la imagen de CASIA 067_1_1.

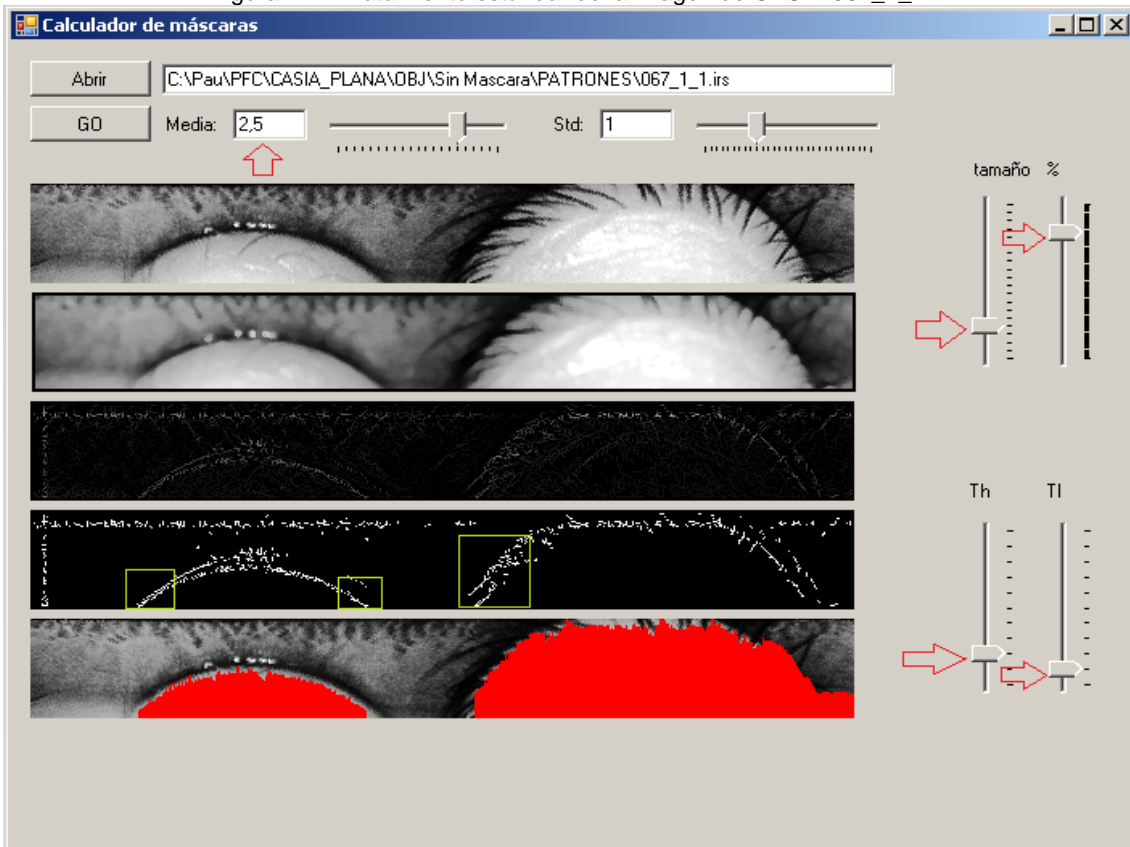


Figura 4.2.- Tratamiento optimizado de la imagen de CASIA 067_1_1.

4.2. Influencia del método de detección de párpados en las prestaciones de un sistema de reconocimiento mediante iris

En este caso, se han empleado dos procedimientos para evaluar la influencia del método de detección de párpados implementado. El primero de ellos consiste en una evaluación subjetiva considerando un subconjunto de 108 muestras de iris pertenecientes a 108 usuarios distintos y provenientes de la base de datos CASIA v1.0. El segundo consiste en una evaluación objetiva de los resultados obtenidos utilizando la totalidad de la base de datos CASIA v1.0 y un algoritmo de reconocimiento mediante iris basado en técnicas MPEG [5], y comparando los casos con irises sin enmascarar y enmascarados.

Los procedimientos anteriores se han repetido para varios conjuntos de parámetros con el fin de establecer el mejor compromiso. A continuación se detallan los resultados para tres de estos conjuntos. El primero, llamado conjunto estándar, es el que a ojos del autor alcanza el mejor compromiso. El segundo es una leve modificación del primero, que consiste en cambiar el filtro de mediana previo por uno de percentil 80 (en concreto, se trata del caso correspondiente a la Figura 3.2). El tercero conlleva una modificación más radical y se corresponde con el conjunto de variaciones utilizado en el caso de la Figura 4.2.

La evaluación subjetiva consiste en la observación de las máscaras producidas y en la determinación subjetiva del ajuste de las mismas a los párpados. En la figura 5 se puede observar lo que se considera un buen ajuste y un mal ajuste, y en la Tabla 1 se muestran los porcentajes de acierto y fallo desglosados en función del conjunto de parámetros del filtro.

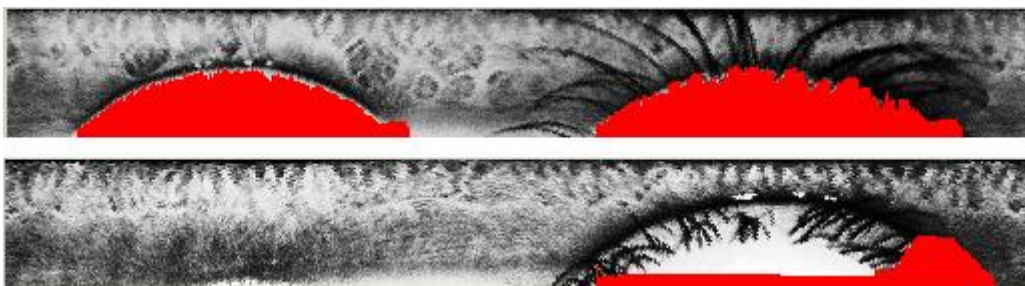


Figura 5.- Ejemplo de máscara ajustada (imagen superior) y no ajustada (imagen inferior).

Tabla 1.- Porcentaje de aciertos y de fallos desglosados en función del conjunto de parámetros del filtro. Método subjetivo.

	Párpado inferior			Párpado superior		
	Máscara correcta	Defecto de máscara	Exceso de máscara	Máscara correcta	Defecto de máscara	Exceso de máscara
Conjunto estándar	75%	20%	5%	54%	22%	24%

Anexo A

Conjunto percentil 80	65%	29%	6%	46%	16%	38%
Conjunto radical	56%	38%	6%	47%	10%	43%

Como se puede observar, en el caso del conjunto estándar, la detección del párpado inferior es bastante buena, ya que en un 75% de los casos la máscara se ajusta bastante bien. En el caso de los párpados superiores, que contienen mayor cantidad de ruido debido a la presencia de pestañas, el ajuste correcto de la máscara no se produce más que en un 54% de los casos aproximadamente. Para otros conjuntos de parámetros el porcentaje de aciertos es inferior. Otro dato a tener en cuenta es el tiempo empleado en calcular las máscaras. Se puede afirmar que éste se sitúa entre 0.05 y 0.08 segundos por máscara en función de los parámetros empleados, siendo el caso más rápido el correspondiente al conjunto estándar. El caso más lento es el tercero (conjunto radical), que produce un número más elevado de puntos en la imagen de histéresis, lo que aumenta el coste computacional.

La evaluación objetiva consiste en ejecutar cinco veces un algoritmo de identificación de iris basado en técnicas MPEG, desarrollado en el contexto del proyecto final de carrera del autor, y obtener de las tasas de error de dicho algoritmo considerando distintos enmascaramientos. La tasa de error considerada a la hora de comparar es la comúnmente denominada *equal error rate* (EER).

En el primero de los cinco casos, se usaron irises sin enmascarar y se obtuvo una EER del 7% (EER de referencia). En el segundo caso, se enmascararon todos los irises de igual forma, considerando una máscara estática que cubría un octavo de la altura en la zona del párpado inferior y la mitad de la altura en la zona del superior. La EER obtenida en este caso fue del 6%. En el tercer caso se emplearon irises enmascarados con el algoritmo presentado, usando el conjunto estándar de parámetros. La EER obtenida fue del 30%. Los dos últimos casos correspondían al conjunto percentil 80 y al conjunto radical. En ambos casos se obtuvieron valores de EER entorno al 32%, no observándose ninguna diferencia significativa entre ambos casos.

Este resultado parece a priori algo desesperanzador, pero puede deberse a que el algoritmo de reconocimiento es muy sensible a la superficie de iris disponible para sus cálculos, y una máscara un píxel más grande que otra puede suponer una pérdida de 144 píxeles útiles para el cálculo, lo que puede explicar esta tasa de error tan llamativa. Esto también explicaría el aumento de la EER en los dos últimos casos, puesto que en ellos el porcentaje de máscara por exceso es muy superior al del conjunto estándar, sobre todo en el caso del párpado superior, que suele ser el que más ocluye el iris.

5. Conclusiones y líneas futuras

Como se ha podido observar a lo largo de este trabajo, la detección de párpados no es un problema trivial. En concreto, el algoritmo propuesto es altamente sensible a los parámetros usados en el proceso de filtrado de Canny. Sin embargo, si estos se eligen cuidadosamente se pueden obtener resultados satisfactorios, teniendo en cuenta que se consideran satisfactorios aquellos casos en los que la máscara se ajusta al párpado de forma correcta, lo que no significa que deba observarse mejoría obligatoria en el valor de EER de un algoritmo de reconocimiento de iris concreto.

En la base de datos considerada, compuesta por muestras de sujetos asiáticos, el párpado ocupa una parte importante del iris segmentado, lo que implica una gran pérdida de información biométrica útil. No obstante, si bien las zonas centrales del párpado no contienen información, por tratarse de piel que suele presentar amplias zonas de luminosidad constante, las zonas de la frontera entre iris y párpado podrían aprovecharse para tratar de disminuir de manera considerable la EER. Y es que si se observa bien, existe una alta correlación en la forma y posición de los párpados de un mismo sujeto, mientras que en el caso de sujetos distintos, dicha correlación es mucho más baja. Y dado que la forma y posición de los párpados dependen de cada sujeto, ¿por qué no tenerlos en cuenta en un sistema de identificación biométrica? Este hecho es realmente significativo en el caso que nos ocupa, pues el algoritmo de extracción de características empleado tiene en cuenta la posición y la forma de los párpados, por lo que interesaría comprobar el comportamiento global del sistema cuando se considera la información correspondiente a la frontera entre los párpados y el iris.

Otro tema muy distinto es la detección de las pestañas, que presentan un comportamiento mucho más aleatorio, incluso tratándose de un mismo sujeto. El caso de las pestañas es un tipo de oclusión que convendría detectar y enmascarar previamente al proceso de extracción de características, aunque se salga del ámbito de este trabajo.

Otra posible mejora a considerar como trabajo futuro, podría ser la aceleración del proceso de detección de párpados. El código que se presenta junto a este trabajo no está optimizado, ni para el moderno hardware actual presente en los PCs, que suele estar dotado de procesadores capaces de ejecutar procesos en paralelo, ni para su uso en posibles microprocesadores empotrados o DSPs. Se ha observado, sin embargo, que los tiempos empleados por un sistema PC estándar actual no son excesivamente elevados, siendo del orden de los 50 ms. Así, una mejora que se podría aportar en cuanto a velocidad de proceso se refiere, consistiría en emplear una red neuronal celular multicapa [8], que permite el procesado paralelo y casi instantáneo de la detección de bordes. Esto aceleraría el proceso de Canny de forma considerable, tras lo que se

podrían aprovechar las características multinúcleo de los procesadores actuales para detectar los párpados de forma paralela e independiente, o incluso atacando por ambos lados a los párpados (si se dispusiera de cuatro núcleos). Esto podría suponer una reducción en el tiempo de detección de más de un 75% con respecto al tiempo empleado actualmente.

Finalmente, cabe destacar que el método empleado para la detección del párpado, una vez realzados su bordes, se ha realizado basándose en hipótesis razonables, pero, no obstante, intuitivas. Así, podría ser interesante probar otras técnicas radicalmente distintas como por ejemplo el uso de medidas como la energía (muy superior en el párpado de sujetos de piel clara que en su iris o a la inversa en el caso de sujetos de piel oscura), o incluso la entropía, que como se puede intuir sería mucho más baja en un párpado que sobre las regiones del iris.

Agradecimientos

Antes de acabar me gustaría agradecer a los miembros del GUTI su ayuda y colaboración desinteresada, sin la cual no habría sido posible realizar este trabajo. Especialmente a Inmaculada Tomeo Reyes por haberme aconsejado, apoyado y animado en los momentos más difíciles y a Iván Rubio Polo que es el autor de la librería de segmentación mencionada en la introducción y que además me ha brindado valiosos consejos y reconfortado en la toma de decisiones durante el desarrollo de este trabajo.

Bibliografía

1. Flom, L. and Safir, A., US. Patent No. 4641349, U.S. Government Printing Office, Washington, DC, 1987.
2. Daugman, J., "Recognizing persons by their iris patterns". In *Biometrics: Personal Identification in Networked Society*, Jain, A., Bolle, R. y Pankanti, S. (eds), pp. 103–122. Springer, 1999.
3. Daugman, J., "High confidence visual recognition of persons by a test of statistical independence." In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15(11), pp. 1148-1161, 1993.
4. Westmoreland, B., Lemp, M. y Snell, R., "Clinical Anatomy of the Eye", Wiley-Blackwell, 2ª Ed., 1998.
5. Monro, D. M. and Rakshit, S., "Rotation-Independent Iris Matching by Motion Estimation", In *IEEE International Conference on Image Processing, ICIP 2007*, vol. 2, pp. 401-404, 2007.
6. Daugman, J., "Recognizing persons by their iris pattern", in: Jain, Bolle, Pankanti, *Biometrics: Personal Identification in Networked Society*, Amsterdam: Kluwer, Ch. 5, 1999.
7. Canny, J. F., "A Computational Approach to Edge Detection", In *IEEE PAMI*, vol. 8(6), pp. 679-698, 1986.
8. Jaramillo, M.A., Fernández, J.A. y Martínez de Salazar, E., "Implementación del Detector de Bordes de Canny sobre Redes Neuronales Celulares", Seminario Anual de Automática, Electrónica Industrial e Instrumentación, SAAEI 2000.