



**ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA**

**INGENIERÍA INFORMÁTICA**

**Curso Académico 2009/2010**

**Proyecto de Fin de Carrera**

**ANÁLISIS DE MÉTODOS DE IDENTIFICACIÓN  
PERSONAL BASADOS EN SILUETAS**

**Autor: Jesús Herráez Soria**

**Tutores: Ángel Sánchez Calle**

**José Vélez Serrano**

## **Resumen**

El objetivo del proyecto es comparar distintas técnicas de verificación de identidad de personas basadas en la silueta facial de perfil. Las técnicas se basan en distintos modelos de representación de los perfiles y toman como referencia la definición de nueve puntos de interés sobre las siluetas.

Las tareas principales del proyecto han sido:

- Crear una base de datos de siluetas. Esta tarea incluye la captura de las imágenes, el tratamiento para convertirlas en siluetas y la definición de los puntos de interés de la silueta.
- Propuesta de los distintos modelos de representación de siluetas.
- Implementación de cada uno de los métodos en Matlab.
- Experimentos de verificación de siluetas usando los modelos propuestos.
- Análisis de resultados y conclusiones.

# Índice

<b>1. INTRODUCCIÓN</b>	<b>6</b>
1.1. Introducción al Reconocimiento Facial	6
1.2. Motivación	8
1.3. Estado del Arte	9
1.4. Objetivos	11
1.5. Estructura de la memoria	12
<b>2. CREACIÓN DE LA BD</b>	<b>13</b>
2.1. Toma de fotografías	14
2.1.1. Equipo fotográfico	14
2.1.2. Técnicas para la obtención de las fotografías	15
2.1.3. Proceso de adquisición de imágenes	18
2.2. Procesado de las imágenes	19
2.2.1. Umbralización de imágenes	20
2.2.2. Extraer figura principal de la imagen	21
2.2.3. Efecto espejo	21
2.3. Asignación de puntos	22
<b>3. PROCESO GENERAL DE IMÁGENES Y CLASIFICACIÓN</b>	<b>25</b>
3.1. Etapas del Sistema de Visión Artificial	25
3.2. Descripción del clasificador	25
3.3. Descripción de las pruebas	27
3.4. Presentación de resultados	28
<b>4. MÉTODO GEOMÉTRICO</b>	<b>30</b>
4.1. Descripción del método	30
4.2. Descripción del experimento	34
4.2.1. Escenario 1: Perfiles sin giro horizontal	34
4.2.2. Escenario 2: Perfil sin giro horizontal contra todos los perfiles	35
4.2.3. Escenario 3: Todos los perfiles con todos los perfiles	36
4.3. Análisis de resultados	37
<b>5. MÉTODO DE CURVATURA LOCAL</b>	<b>39</b>
5.1. Descripción del método	39
5.2. Descripción del experimento	40

5.2.1.	Escenario 1: Perfiles sin giro horizontal	40
5.2.2.	Escenario 2: Perfil sin giro horizontal contra todos los perfiles	42
5.2.3.	Escenario 3: Todos los perfiles con todos los perfiles	43
<b>5.3.</b>	<b>Análisis de resultados</b>	<b>43</b>
<b>6.</b>	<b>MÉTODO DE CURVATURA LOCAL ACUMULADA</b>	<b>45</b>
<b>6.1.</b>	<b>Descripción del método</b>	<b>45</b>
<b>6.2.</b>	<b>Descripción del experimento</b>	<b>46</b>
6.2.1.	Escenario 1: Perfiles sin giro horizontal	46
6.2.2.	Escenario 2: Perfil sin giro horizontal contra todos los perfiles	47
6.2.3.	Escenario 3: Todos los perfiles con todos los perfiles	48
<b>6.3.</b>	<b>Análisis de resultados</b>	<b>49</b>
<b>7.</b>	<b>MÉTODO DE HISTOGRAMA LOCAL</b>	<b>51</b>
<b>7.1.</b>	<b>Descripción del método</b>	<b>51</b>
<b>7.2.</b>	<b>Descripción del experimento</b>	<b>52</b>
7.2.1.	Escenario 1: Perfiles sin giro horizontal	53
7.2.2.	Escenario 2: Perfil sin giro horizontal contra todos los perfiles	54
7.2.3.	Escenario 3: Todos los perfiles con todos los perfiles	55
<b>7.3.</b>	<b>Análisis de resultados</b>	<b>56</b>
<b>8.</b>	<b>MÉTODO DE SNAKES</b>	<b>57</b>
<b>8.1.</b>	<b>Descripción del método</b>	<b>57</b>
8.1.1.	Construcción de imágenes y <i>snakes</i>	59
8.1.2.	Configuración de la aplicación y resultados	63
8.1.3.	Ejecución en batch de la aplicación	66
<b>8.2.</b>	<b>Descripción del experimento</b>	<b>66</b>
8.2.1.	Escenario 1: Perfiles sin giro horizontal	67
8.2.2.	Escenario 2: Perfil sin giro horizontal contra todos los perfiles	68
8.2.3.	Escenario 3: Todos los perfiles con todos los perfiles	69
<b>8.3.</b>	<b>Análisis de resultados</b>	<b>70</b>
<b>9.</b>	<b>CONCLUSIONES</b>	<b>71</b>
<b>9.1.</b>	<b>Objetivos cumplidos</b>	<b>71</b>
9.1.1.	Escenario 1: Perfiles sin giro horizontal	71
9.1.2.	Escenario 2: Perfil sin giro horizontal contra todos los perfiles	72
9.1.3.	Resultados de la comparación entre todos los perfiles	73
<b>9.2.</b>	<b>Conclusiones sobre los métodos</b>	<b>74</b>
<b>9.3.</b>	<b>Futuros Trabajos</b>	<b>74</b>
9.3.1.	Determinación automática de los puntos de interés	74
9.3.2.	Unión de método global y local	74
9.3.3.	Mejora del uso de snakes.	74

<b>10.</b>	<b>BIBLIOGRAFÍA</b>	<b>76</b>
<b>A.</b>	<b>APÉNDICES</b>	<b>77</b>
<b>A.1.</b>	<b>Detalle de scripts Matlab creados</b>	<b>77</b>

# 1. Introducción

En este capítulo se realiza una introducción al proyecto fin de carrera. El reconocimiento facial es un tema que suscita gran interés entre investigadores de todo el mundo y conviene presentar los distintos tipos de sistemas que podemos encontrar, así como el papel que juega dentro de la biometría. El capítulo presenta las motivaciones que suscitan el proyecto y el estado del arte referente al reconocimiento facial de perfil.

La definición de los objetivos del proyecto y la estructura de la memoria se describen en el capítulo para que guíen la lectura de dicha memoria.

## 1.1. *Introducción al Reconocimiento Facial*

Durante las últimas décadas el reconocimiento facial ha suscitado un gran interés en investigadores de todo el mundo [7]. El tema interesa a distintas comunidades de investigación relacionadas con: la biometría, el reconocimiento de patrones y la visión por computador. Este interés común de tantos investigadores de distintas áreas, está fomentado por la necesidad del ser humano de reconocer a otros en la vida cotidiana, la aparición de numerosos sistemas dedicados a la automatización del día a día, hacen necesaria la adopción de esta capacidad por dichos sistemas.

La verificación de la identidad de una persona se puede hacer básicamente mediante tres métodos:

- **Conocimiento:** La persona conoce la respuesta adecuada a una determinada pregunta (por ejemplo: PIN o clave de acceso).
- **Posesión:** La persona posee un objeto que lo diferencia del resto (por ejemplo: una tarjeta, el DNI, una llave, etc.).
- **Característica:** La persona tiene una característica única que puede ser empleada para su verificación (por ejemplo: iris del ojo).

En los dos primeros métodos pueden aparecer errores de verificación. En función del método los errores pueden deberse a:

- Conocimiento: la persona olvida la clave, otra persona conoce la clave, se crean métodos para descubrir claves.
- Posesión: la persona pierde la tarjeta (no se puede identificar), una persona distinta encuentra la tarjeta (se identifica por otra persona).

Este tipo de errores no se dan en el método basado en la característica. Las características que se usan para la verificación pertenecen únicamente a la persona y no se pueden ni perder ni suplantar. El estudio de estas características se conoce como biometría, que más formalmente definida sería: la biometría es el estudio de métodos automáticos para el reconocimiento y/o verificación de personas basados en uno o más rasgos físicos.

La biometría engloba multitud de características que se pueden clasificar en función de los aspectos tratados (y la tecnología empleada). En la Figura 1 se pueden ver los principales aspectos así como el porcentaje de uso dentro del mercado. Como se

muestra en la figura, el reconocimiento facial tiene un gran peso dentro del mercado. La importancia del reconocimiento facial dentro de la biometría se deba a la combinación equilibrada de las variables: intrusismo, precisión, coste y esfuerzo. La Figura 2 muestra la combinación de las variables para los distintos métodos biométricos, en ella se aprecia como la huella dactilar y el reconocimiento facial son los más equilibrados.

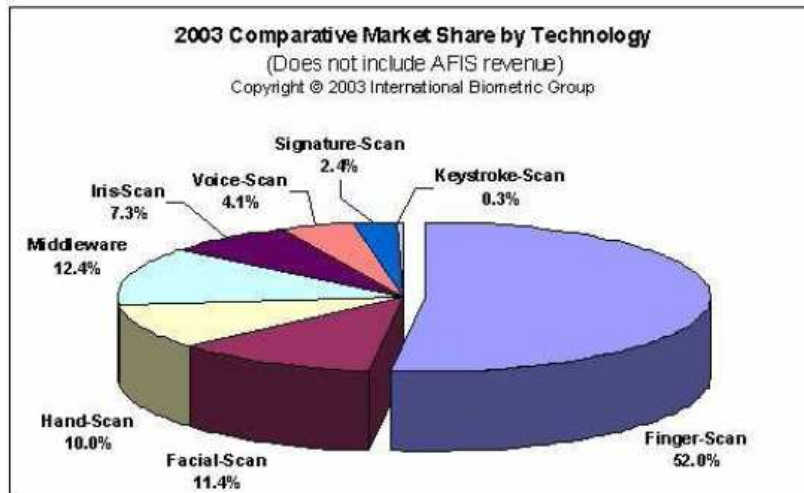


Figura 1.- Comparativa de participación en el mercado por tecnología

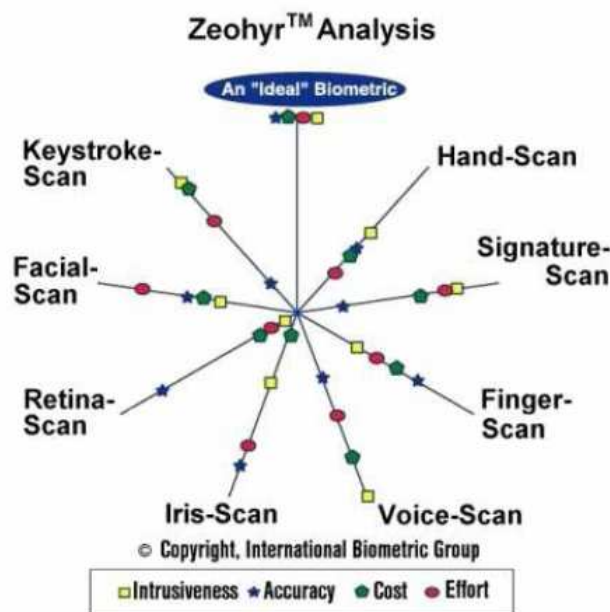


Figura 2.- Comparación de características de distintos métodos biométricos

Los posibles escenarios para el reconocimiento facial pueden ser clasificados en:

- **Verificación:** Se centra en comprobar si una persona es quién dice ser. Se resuelve mediante la comparación uno a uno. Para medir el rendimiento de un sistema de verificación, se pinta un gráfico con el ratio de personas verificadas (usuarios que son quién dicen ser) frente al ratio de impostores aceptados (personas que no son quién dicen ser y el sistema las acepta), este gráfico se

conoce como curva ROC. Un buen sistema de verificación debe equilibrar estos dos ratios en función de las necesidades del propio sistema.

- **Identificación:** Debe buscar a una persona entre otras, se resuelve con una comparación uno a muchos. El resultado de la comparación debe dar una lista ordenada según el parecido a los individuos

En función de los datos de entrada, se pueden distinguir los siguientes métodos de reconocimiento facial [9]:

- Frontal: imágenes de frente con fondo simple o complejo.
- Video: extracción de imágenes de video.
- 3D: imágenes en 3D.
- Perfil: imágenes de perfil.
- Híbridos: mezcla de los anteriores.

Los métodos basados en imágenes frontales y video han sido los más estudiados por los investigadores, dejando en segundo lugar a las imágenes de perfil [9]. En los últimos años este método está cobrando fuerza debido a la importante información contenida en el perfil. La estructura completa de la cara está presente en el perfil y no en la imagen frontal.

Para el proyecto nos vamos a centrar en la verificación (nos interesa determinar si el individuo es quién dice ser) y en los perfiles.

## **1.2. Motivación**

La identificación de personas es una acción básica en nuestra sociedad, hay numerosas ocasiones en las que se pide el DNI, o una tarjeta para verificar la identidad, esto es implica que quién pide la acreditación tiene capacidad de verificar si somos la misma persona. Cuando se pretende automatizar estos procesos, se deben buscar métodos de verificación de identidad alternativos como los basados en conocimiento o posesión. Estos métodos tienen unas debilidades ya comentadas, mientras que la evolución de la biometría está posibilitando la sustitución de estos métodos por otros más fiables.

Dentro del abanico de posibilidades de la biometría, el reconocimiento facial tiene grandes posibilidades, debido a que posee unas características (intrusismo, precisión, coste y esfuerzo) bastante equilibradas. Un gran punto a favor del reconocimiento facial es el intrusismo, el usuario no tiene que hacer nada especial para ser reconocido, el sistema capta su imagen y lo verifica. Pensemos por el contrario en el reconocimiento de iris, el usuario debe acercar su ojo al sensor que capta las propiedades del iris para que este le identifique.

El rendimiento de un sistema automático de reconocimiento facial depende de las siguientes condiciones:

- Posición en 3D de la imagen (giro horizontal y vertical)
- Iluminación [5]
- Expresión del rostro
- Oclusión (por elementos externos, gafas, gorras, bufandas, etc.)



- Pelo, tanto peinado (si oculta parte del rostro, como barba)
- Envejecimiento

Tener controladas todas estas situaciones no siempre es posible, por este motivo se debe buscar un método que minimice estas dependencias.

Las imágenes de perfil tienen información de la estructura de la cara y se quiere comprobar como afectan estas dependencias. Para hacer la comprobación se van a testear varios métodos.

### 1.3. Estado del Arte

Nicolas Davidenko [1] ha realizado un estudio sociológico para determinar la cantidad de información contenida en la silueta de una cara de perfil. En el estudio se comprobaba si una persona era capaz de asociar una silueta a una fotografía y así determinar si en la silueta existía información suficiente para identificar a dicha persona. Los resultados son positivos y revelan que en muchos casos sería posible reconocer a personas mediante su silueta. Además se pretende determinar que partes contienen más información y si con la parte del perfil próxima al frontal se podrían concluir los mismos resultados. En dicho estudio, propone la construcción de un perfil reducido usando 18 puntos que se muestran en la Figura 3, al normalizar y tomar 2 puntos como ejes se convierten en 16. La definición de la silueta de la persona queda reducida a 16 puntos (32 enteros).

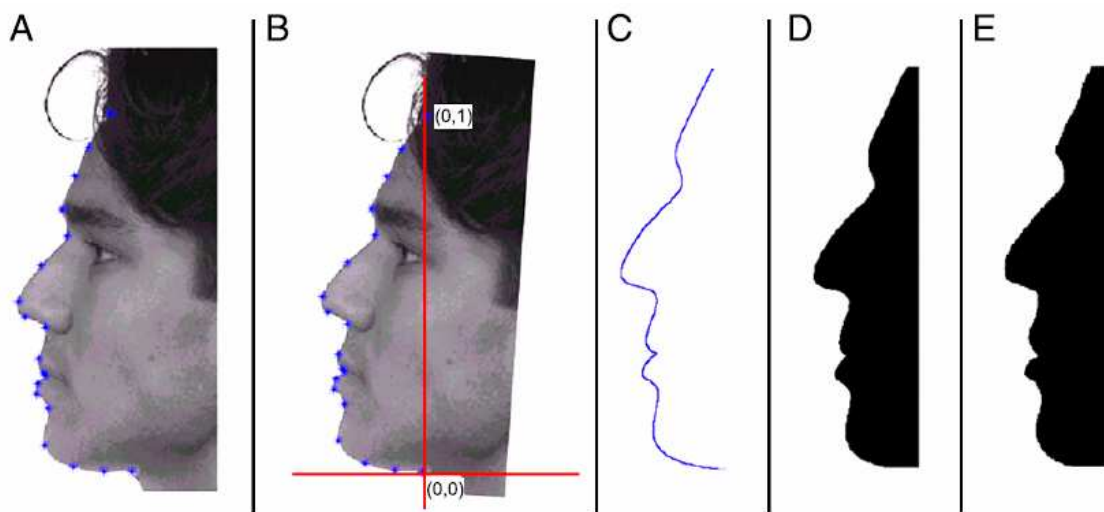


Figura 3.- Pasos en la parametrización de la silueta, A) Original con los 18 puntos, B) Normalizado a 16, C) Unión de líneas, D) Contorno relleno, E) Silueta original

Parece evidente que la parte más próxima al frontal tiene más información, la estructura del perfil tiene una baja variación en el tiempo, sin embargo con el paso de los años el perfil cambia y puede incorporar distorsiones respecto al anterior, siendo más evidentes las distorsiones cuando la edad es mayor, o cuando se producen grandes saltos de edad.

A lo largo del tiempo se han realizado trabajos para identificar personas mediante la silueta<sup>1</sup> del perfil, a continuación se describen algunos ejemplos:

- **Basados en el histograma:** El método [9] pretende evitar la definición de puntos de interés dentro de las imágenes. Como la definición de los puntos no es sencilla, compara las imágenes como si fueran histogramas. Antes de hacer la comparación normaliza las imágenes para que tengan el mismo tamaño y orientación. El método trabaja con condiciones ideales, no tiene en cuenta ni ocultación, ni variación de gesto o posición.
- **Basados en puntos de interés:** La mayoría de los métodos extraen información de las imágenes teniendo en cuenta la situación de unos puntos de interés. En [10] usan funciones de escalado espacial para determinar los puntos de interés y sobre ellos toman medidas de distancia y ángulo. El método trabaja con condiciones ideales.
- **Basados en la comparación de imágenes 3D:** El método [11] trabaja con modelos 3D, del modelo 3D extrae el contorno de distintos perfiles según el ángulo de giro y compara estos perfiles con el tomado con una cámara sencilla. De esta forma el método supera los posibles giros de la persona.
- **Basados en la aplicación de filtros morfológicos:** Se basan en la aplicación de filtros morfológicos sobre el perfil para obtener información del mismo. Esta técnica consiste en aplicar filtros morfológicos sobre el perfil de manera que simule el crecimiento/decrecimiento del pelo o barba y usar como descriptores una combinación de los perfiles obtenidos. El método se fija en el perfil completo, dando la misma importancia a las partes más invariantes (nariz, boca, frente) y al resto del contorno (que se puede ver afectado por el pelo). Los resultados obtenidos no han sido buenos debido a los siguientes motivos:
  - No se ha dado más importancia a las partes más invariantes
  - Se ha supuesto un crecimiento / decrecimiento uniforme, no teniendo en cuenta los cortes de pelo asimétricos o irregulares.
- **Basados en la transformación morfológica del la firma<sup>2</sup> [2]:** tomando como base la parte del perfil con más información que definió Davidenko [1], crea una nueva figura que mantiene proporciones y ángulos (ver Figura 4), sobre dicha figura crean distintas figuras a base de erosionar la figura base con un elemento estructurante en forma de 'Y', repiten la erosión sobre la figura base aplicando rotaciones al elemento estructurante. Para cada figura resultante toman la firma y el conjunto de las firmas construye la definición de la imagen. De esta forma representan el perfil de la persona. El método da una tasa de aciertos del 90% con perfiles sintéticos contruidos sobre los originales (con rotaciones, cambios de tamaño, etc.) y un 75% con otros perfiles reales de dicha persona. En estos perfiles no hay rotación, ni posibles cambios de posición, gafas, etc.

---

<sup>1</sup> Silueta: imagen con blanco y negro como únicos valores de color, uno se usa para identificar el fondo y el otro para identificar el objeto.

<sup>2</sup> Firma: medida de distancia de un punto de referencia a los puntos del contorno

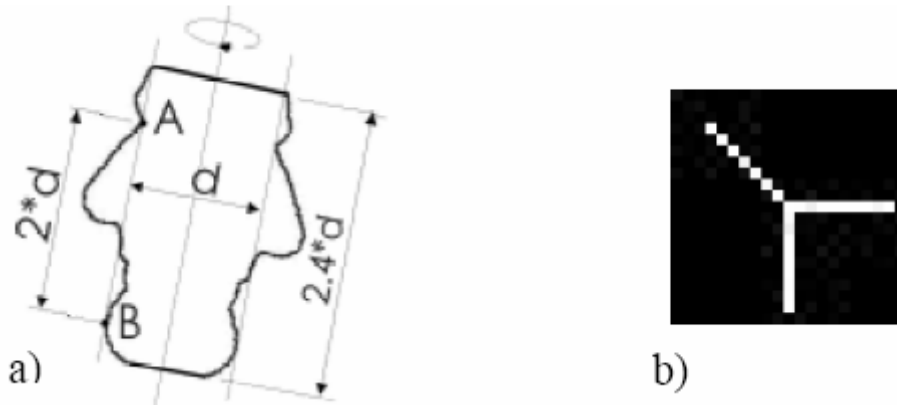


Figura 4.- Figura base para la transformación morfológica de la firma a); elemento estructurante b)

Actualmente se está trabajando en determinar las partes de la cara que son más representativas para el reconocimiento de personas [3], en una fotografía frontal en color se tiene mucha información, distancias entre ojos, distancias con la boca, la nariz, dimensiones de cada una de las partes, información de color de ojos, tonalidad de la piel, textura, etc. En la Figura 5 se pueden ver algunos de los puntos más representativos. En la silueta las opciones se reducen y hay que intentar buscar las partes más representativos e invariantes dentro de la silueta, así como las que las hacen distinguibles de las demás.

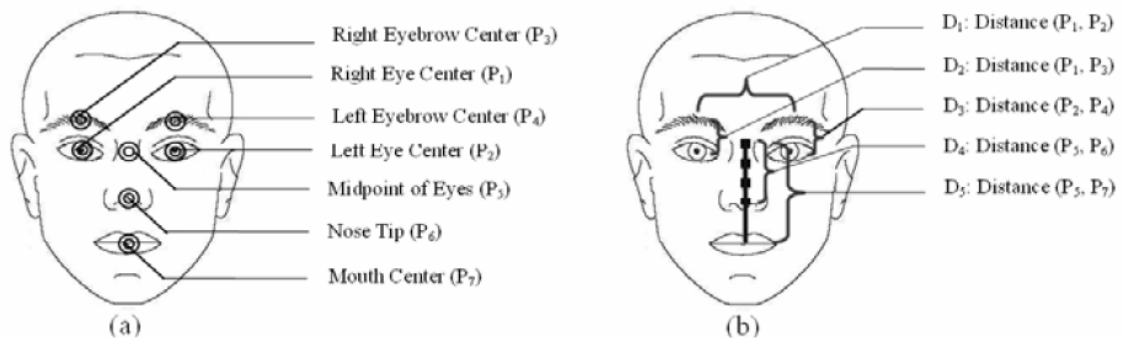


Figura 5.- Identificación de los puntos más representativos de la cara, en posición frontal

Existen distintas técnicas para la obtención automática de los puntos de la cara que permiten obtener directamente de las fotografías las coordenadas de dichos puntos [4]. Estas técnicas dan buenos resultados para casos en los que no hay elementos en el rostro que puedan ocultar parte del mismo, como pueden ser las gafas.

#### 1.4. Objetivos

El objetivo principal del proyecto es la comparación de distintos métodos de verificación de personas basado en siluetas, con el fin de determinar cuáles son las características más determinantes y qué métodos producen mejores resultados. Los resultados y conclusiones obtenidos, podrán ser utilizados para la construcción de un sistema de reconocimiento basado en siluetas.

Para llevar a cabo el objetivo principal se deberán cumplir los siguientes sub-objetivos:

- a) Creación de una Base de Datos de imágenes de perfil

- b) Definición en detalle de cada uno de los métodos e implementación de los mismos
- c) Pruebas de verificación usando cada uno de los métodos
- d) Análisis de resultados y conclusiones.

### **1.5. Estructura de la memoria**

La memoria presenta un capítulo para la creación de la base de datos, donde se describe cómo se han tomado y procesado las fotografías, un capítulo para definir las partes comunes a todos los métodos y la pruebas, un capítulo para cada método propuesto, que incluye la descripción del mismo, sus pruebas y el análisis de resultados. Finalmente se presentan las conclusiones y trabajos fuentes en el último capítulo. Se incluye también un anexo con la descripción de las partes del código desarrollado en MATLAB.

## 2. Creación de la Base de datos de imágenes

El objetivo principal del proyecto es comparar y analizar distintos métodos de verificación de siluetas, para realizar dichas comparaciones, es necesario tener una base de datos acorde a las necesidades de las pruebas. Las pruebas se deben realizar con perfiles de personas con distintas condiciones de sexo, edad, peinado (corto, largo, liso, ondulado, etc.), con/sin gafas, etc. No interesa seleccionar perfiles ideales (ejemplo: hombre de pelo corto y sin gafas) para obtener buenos resultados, es preferible obtener peores resultados y que estos sean reales. Al incluir personas con distintas características externas de aspecto, se puede decir que la base de datos es un reflejo real de la sociedad.

Queremos comprobar si los distintos métodos son capaces de distinguir a personas con cierto grado de giro horizontal y vertical. Otro requerimiento adicional es la independencia respecto al tamaño, es decir la escala de las personas fotografiadas no guarda relación entre distintas imágenes.

Por los motivos anteriormente descritos se ha creado una base de datos con 39 individuos. Por cada individuo se toman 6 fotografías, 3 de cada perfil con los siguientes ángulos de giro horizontal respecto a la cámara aproximados:

- $55^\circ \rightarrow -35^\circ$  Respecto a la posición de perfil ideal.
- $90^\circ \rightarrow$  Posición de perfil ideal.
- $125^\circ \rightarrow 35^\circ$  Respecto a la posición de perfil ideal.

A la personas que poseen gafas se les toman otras 6 fotografías en las mismas condiciones con la gafas.

En total la base de datos está compuesta por 269 fotografías de perfil de 39 individuos, de los cuáles 6 tienen gafas. La base de datos debería contener 270 imágenes, pero la captura de una de las imágenes (con gafas) no se realizo adecuadamente.

Una vez obtenidas las imágenes, se han procesado para obtener las siluetas y se han asignado los puntos de interés.

En la Figura 6 se muestra un ejemplo de las 6 fotografías tomadas, en ellas se puede ver el ángulo y la orientación de las mismas.



Figura 6.- Ejemplo de imágenes

## 2.1. Toma de fotografías

Se quiere crear una base de datos de siluetas de perfil, por ello a la hora de realizar las fotografías se ha buscado obtener el mayor contraste entre el fondo y la imagen del individuo. Al aumentar este contraste se facilita la separación entre la persona y el fondo.

A continuación se va a describir el equipo fotográfico, las condiciones en las que se ha creado la base de datos, una descripción de las técnicas usadas y el proceso de la adquisición de imágenes.

### 2.1.1. Equipo fotográfico

Lo primero es describir el equipo fotográfico usado para la obtención de las imágenes, el equipo está compuesto por una cámara reflex-digital Canon EOS 450 D (Figura 7) con un objetivo Sigma 18-200 f/3.5-6.3 DC OS (Figura 8) y un flash Canon Speedlite 580-ex II (Figura 9).



Figura 7.- Cámara Canon 450D



Figura 8.- Objetivo Sigma 18-200



Figura 9.- Flash Canon 580

Las imágenes se han tomado en configuración manual, almacenado las imágenes en RAW CR2 (Canon Raw V2) para no perder información por el procesado de las mismas. El tipo RAW almacena en la tarjeta de memoria la imagen sin comprimir y sin pérdidas.

## 2.1.2. Técnicas para la obtención de las fotografías

Se han probado distintas técnicas para obtener las siluetas de la manera más directa posible, pero no todas han sido efectivas por las condiciones ambientales. Es importante destacar cuales eran las condiciones para así entender porqué no han sido satisfactorias algunas de las técnicas. En función de las condiciones ambientales, se puede usar una u otra técnica. Además de las técnicas probadas se describen otras técnicas que darían buenos resultados pero que no se han probado por ser más costosa su preparación.

La BD se ha creado en una oficina de paredes blancas, pintadas hace algunos años (no es un blanco puro, sino que tiene tonos grises), con abundante iluminación artificial mediante fluorescentes y natural proporcionada por grandes ventanas con filtros anti-reflejos.

Las variantes probadas para la toma de imágenes han sido:

- **Frente a la ventana:** Se sitúa el objeto a fotografiar frente a la fuente de luz natural (ventana). El individuo se sitúa frente a la ventana sobre una columna de color blanco, el esquema de la escena se muestra en la Figura 10. En este caso no se usa flash y las fotografías tomadas aparentemente tienen buen aspecto. Al intentar umbralizar dichas fotografías aparecen los siguientes problemas:
  - Se solapan el nivel de intensidad del rostro y el del blanco de la pared. Este hecho aparece cuando la persona produce algo de sombra sobre la pared, o cuando la luz reflejaba en el rostro haciendo aparecer brillos.
  - Existe diferencia entre las imágenes tomadas de un perfil o de otro, la iluminación no es uniforme.
  - Las condiciones solares afectan a la cantidad de luz de la imagen, variando dichas condiciones en función de la altura del sol (por la hora) y si está o no nublado.

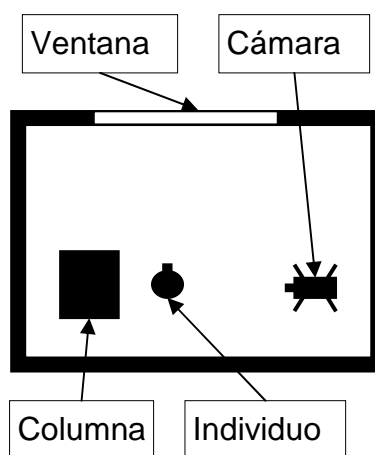


Figura 10.- Esquema de ubicación frente a la ventana

- **Contra la ventana:** Se sitúa el individuo delante de la ventana, de esta forma se pretende buscar la sombra directamente sobre-iluminando el fondo. La Figura 11 muestra el esquema de la escena. Esta técnica es muy apropiada en muchos casos, pero no es válida para la mayoría de edificios de oficinas, dichos edificios tienen gran cantidad de luz ambiental (tanto natural como artificial) y los cristales suelen ser anti-reflejos, evitando la entrada directa de la luz. Al usar esta técnica en un edificio de oficinas se ha comprobado que el rostro también aparece iluminado, dificultando la umbralización de las imágenes. La técnica es ideal en lugares en los que la principal fuente de luz es esa ventana.

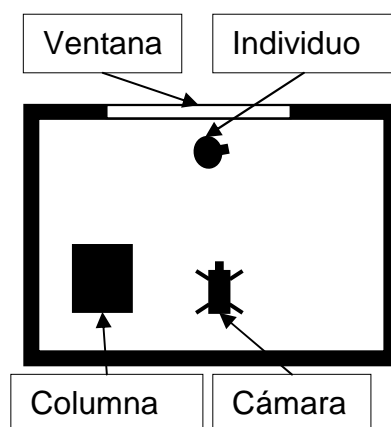


Figura 11.- Esquema de Ubicación contra la ventana

- **Iluminación mediante flash estándar:** Se sitúa el individuo junto a la pared de la ventana, evitando la luz directa de la misma, para iluminar al individuo se usa el flash integrado de la cámara. La Figura 12 muestra el esquema de la escena. Los resultados aparentes son buenos, pero al umbralizar se observa que no existe tanta diferencia entre el fondo y el rostro, esto depende del individuo a fotografiar y de la cantidad de luz que entra por la ventana. Esta técnica puede ser aceptable, aunque en algunos casos los resultados no son buenos. Siempre que se use flash hay que tener en cuenta que dicha luz produce una sombra y hay que evitar que esta sombra pueda interferir en la imagen, por eso el flash debe estar orientado de forma que la sombra quede detrás de la imagen.

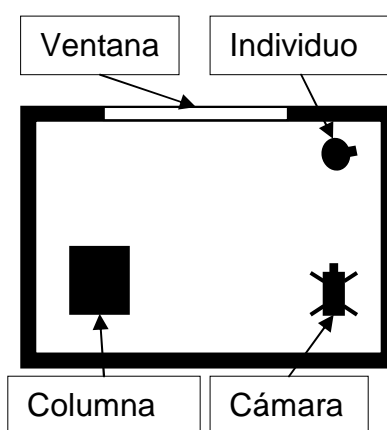


Figura 12.- Esquema de Ubicación con flash estándar



- **Iluminación mediante flash externo:** Este caso es igual que el anterior (ver Figura 13), pero en lugar de usar el flash de la cámara se usa un flash externo más potente, configurado para crear un gran foco de luz de tamaño reducido. Los resultados usando esta técnica son muy buenos, consiguiendo llevar el fondo muy cercano al blanco puro. De esta forma la umbralización es automática debido a que la potencia del flash anula el resto de fuentes de luz.

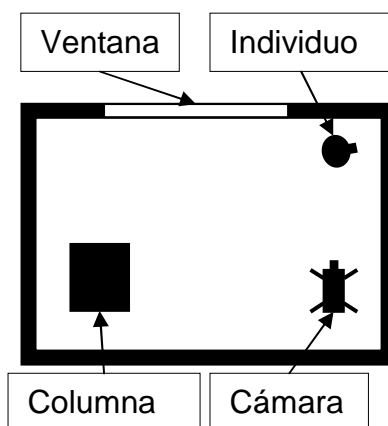


Figura 13.- Esquema de ubicación con flash externo

Otras técnicas no probadas (por haber obtenido buenos resultados con la anterior):

- **Luz artificial iluminado el fondo:** Se sitúa un flash activado por control remoto detrás del individuo apuntando directamente al fondo, la Figura 14 muestra el esquema de la escena descrita, de esta manera sólo se sobre-ilumina el fondo, consiguiendo así llevar dicho fondo a valores de blanco puro.

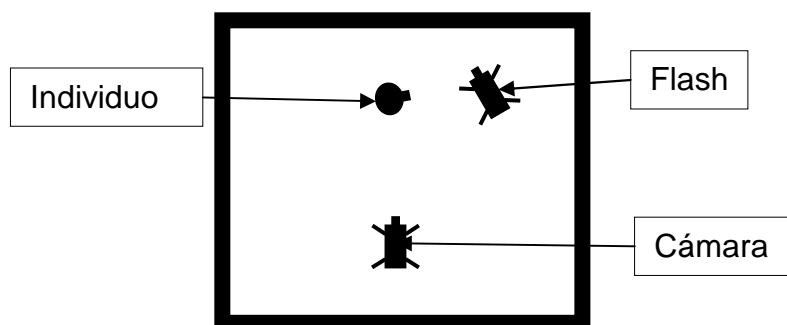


Figura 14.- Esquema Ubicación Luz Artificial iluminando el fondo

- **Luz artificial iluminando pantalla de fondo:** Se sitúa una pantalla blanca detrás del individuo y se usa un flash activado por control remoto para iluminar dicha pantalla. La escena se muestra en la Figura 15. Este caso es válido cuando el fondo está muy lejos.

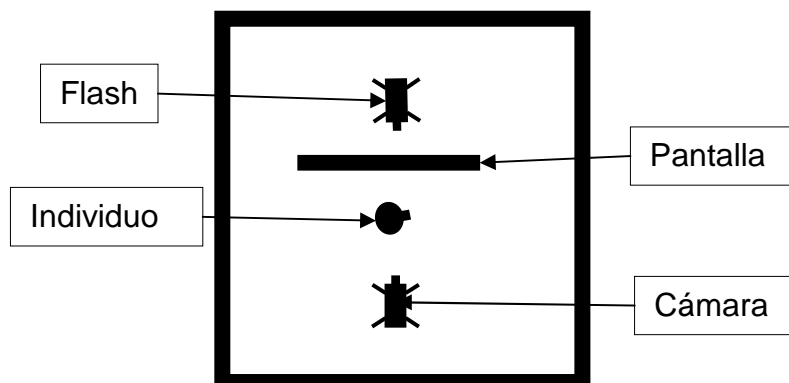


Figura 15.- Esquema de Ubicación iluminando pantalla de fondo

### 2.1.3. Proceso de adquisición de imágenes

Se necesita tener imágenes que aporten cierto realismo para comparar entre sí. Las imágenes deben retratar a las personas con una posible variación en la postura o en el gesto, de poco servirían ‘n’ imágenes iguales. Por este motivo se toman fotos de ambos perfiles, así obligamos a las personas fotografiadas a moverse y con ello conseguimos que los gestos puedan cambiar teniendo 2 fotografías en cada caso. Si por el contrario con la misma posición de la persona realizamos 2 disparos de la cámara tendremos 2 fotografías prácticamente iguales. Este principio se usa en la toma de cualquier fotografía de perfil.

Como se quiere comprobar si los sistemas son tolerantes a variaciones de ángulo horizontales se toman 3 fotos de cada perfil, con los siguientes ángulos:

- 35° → Respecto al perfil perfecto
- 0 → Perfil perfecto
- -35 → Respecto al perfil perfecto

La Figura 16 muestra un esquema de la posición de los individuos en la foto. Se ha seguido una nomenclatura en las fotos, estando compuesta esta nomenclatura por 2 números de la forma *xxxx\_yy.ext*, siendo la ‘xxxx’ el n° de individuo y el valor de ‘yy’ identifica la posición de la imagen tomada. La Figura 16 muestra las posiciones de los individuos a las que le corresponden los valores de ‘yy’ 1, 2, 3, 4, 5 y 6 respectivamente de izquierda a derecha.



Figura 16.- Variaciones de ángulo entre fotos

Para poder comprobar si el sistema permite identificar a personas con gafas o sin ellas, es necesario tener registrada a la persona con y sin gafas. Por este motivo, a las personas con gafas se la hacen 12 fotografías, 6 con gafas y 6 sin gafas. La nomenclatura es como en el caso anterior, del 1-6 sin gafas y del 7-12 con ellas.

El ángulo de giro horizontal no se ha establecido con precisión en el método, se ha marcado aproximadamente mediante unas referencias y las personas se giraban para mirar dichas referencias, el resultado es que los ángulos no son uniformes en todos los individuos. Esta situación tiene una parte positiva y otra negativa, la positiva es que se puede valorar el grado de acierto en función del ángulo y la negativa es que no se puede hacer de forma cuantitativa al no tener medidos con exactitud dichos ángulos en todos los casos. Hay que tener en cuenta que los individuos han colaborado gratuita y voluntariamente y por ello se ha pretendido que fuera fácil y rápido para ellos, limitando el nº de fotografías requeridas. Se podrían haber tomado más fotografías con distintos ángulos para tener una muestra mayor, pero no se han realizado por no abusar de los colaboradores, ya que tomar más muestras llevaría asociado una medición estricta de los ángulos.

No se ha tenido ningún cuidado en la toma de las imágenes en la distancia a la persona, ni con la distancia a la cámara, ni con la configuración del objetivo. Como el sistema debe ser tolerante a los cambios de tamaño, no tenía sentido tomar las fotografías con esta limitación.

El ángulo de giro vertical tampoco se ha tenido en cuenta, y se ha permitido a los individuos posar como han querido, para que sea el sistema quién normalice esta situación.

Se han tomado imágenes de 44 individuos. Aunque para todas ellas se pretendía usar la técnica de *‘Iluminación mediante flash externo’*, los juegos de pilas se agotaron sobre el individuo 28. Las fotografías siguientes se realizaron con la técnica *‘Iluminación mediante flash estándar’* que, por los problemas descritos en dicha técnica, las imágenes de 5 individuos no han tenido una umbralización automática y se han descartado, teniendo disponibles para las pruebas 39 individuos.

## **2.2. Procesado de las imágenes**

Las imágenes han sido tomadas en formato RAW (CR2, Canon Raw Versión 2) y para que Matlab pueda trabajar con ellas ha sido necesario convertirlas a un formato sin pérdidas, el formato elegido a sido *“tiff”* y se ha realizado usando el programa *“Adobe Photoshop Lightroom 2.4”*.

Una vez las imágenes se encuentran en un formato entendido por Matlab, se va a trabajar sobre ellas para encontrar las siluetas. Se han realizado las siguientes acciones:

- Umbralizar las imágenes
- Extraer la figura principal dentro de las imágenes
- Efectuar el “espejo” de las imágenes del segundo perfil para que tengan todos los perfiles la misma orientación.

Estas acciones se han efectuado por bloque, es decir se han ejecutado cada una de las acciones para todas las imágenes. Las imágenes de salida de cada acción han sido las imágenes de entrada de la siguiente acción. Se han creado los scripts para la ejecución en bloque, no se han creado para la ejecución lineal porque el primer paso necesita intervención humana como se explicará a continuación. La finalidad del proyecto es testear y comparar una serie de métodos, si se quisiera implementar un sistema de verificación usando algún método particular no sería muy costoso modificar las funciones para una ejecución lineal.

### 2.2.1. Umbralización de imágenes

En este paso se pasa de la imagen en color a una imagen binaria (con 2 niveles de intensidad: blanco y negro). El proceso es más complejo de lo que debería porque no todas las imágenes han sido tomadas usando las mismas técnicas (como se explicó en el apartado anterior).

El proceso busca el valor del umbral que estima oportuno y realiza la umbralización con diversos valores próximos a dicho umbral, con el fin de elegir la imagen umbralizada que sea más correcta. Conviene distinguir dos casos, en función de la técnica de obtención de fotografías usada:

- Técnica de **iluminación con flash externo**: Se puede fijar un umbral fijo de 0.91 (o cualquiera hasta el 0.98). En la Figura 17 se muestra un ejemplo de las imágenes umbralizadas para valores de 0.91 a 0.99.



Figura 17.- Ejemplo de imágenes umbralizadas para flash externo

- Técnica de **iluminación con flash estándar**: Es necesario obtener el valor estimado, y con un pequeño ajuste se podría tener el valor adecuado para la mayoría de las imágenes. El proceso genera imágenes umbralizadas con distintos valores (próximos al calculado) para dar la opción de elegir cuál de ellos es más adecuado. Si no fuera por algunas imágenes que no ha sido posible umbralizar de forma correcta, este paso podría ser automático y el valor calculado unido a la corrección podría ser suficiente. En la Figura 18 se muestra un ejemplo de las 15 imágenes umbralizadas para valores cercanos al estimado por el sistema. Se puede ver que cualquiera de ellas daría un resultado correcto.



**Figura 18.- Ejemplo de imágenes para flash estándar**

### 2.2.2. Extraer figura principal de la imagen

Las imágenes están tomadas de manera que queda espacio blanco rellenando al tamaño de la foto. Para no tener imágenes más grandes de lo necesario sin que aporte información, se recorta la imagen a la figura principal, eliminando el espacio blanco de relleno. Se invierten los colores de la imagen, de manera que la imagen quede en color blanco y el fondo en negro. Un ejemplo del proceso se puede ver en la Figura 19.



**Figura 19.- Ejemplo de extracción de la figura principal**

### 2.2.3. Efecto espejo

Se han tomado fotografías de ambos perfiles de cada persona, pero únicamente queremos trabajar con un perfil, el segundo se tomó para tener dos imágenes de cada perfil. Por este motivo es necesario transformar el segundo perfil al primero, eso se consigue realizando el espejo de las imágenes. De las 6 imágenes de cada persona, del 1-3 corresponden al perfil izquierdo y del 4-6 al perfil derecho, se ha decidido trabajar con el izquierdo, por lo que es necesario efectuar el espejo de todas las imágenes del 4-6 de cada individuo. En la Figura 20 se puede ver el resultado de la aplicación del efecto espejo.

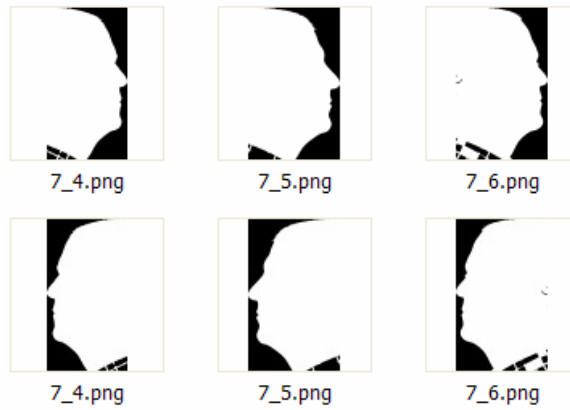


Figura 20.- Ejemplo de efecto espejo para imágenes del 4-6

### 2.3. Asignación de puntos

Una vez obtenidas las siluetas hay que definir los puntos de interés dentro de dichas siluetas. Los puntos seleccionados responden a la estructura del perfil y se pueden ver en la Figura 21.

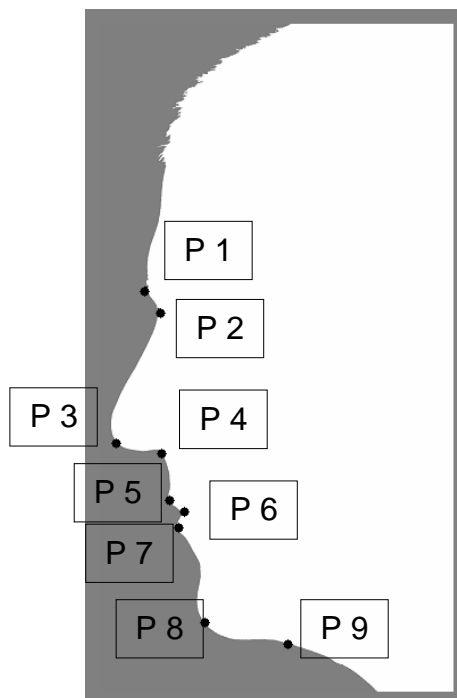


Figura 21.- Puntos de interés en la silueta

Los puntos están numerados del 1-9, empezando por el punto situado en la ceja y continuando según van apareciendo en el perfil.

El sistema debe ser tolerante a cambios de tamaño en imágenes y a giros verticales de la silueta. Por estos motivos hay que girar y redimensionar las imágenes para que estén todas normalizadas y así poder trabajar con ellas directamente. Para poder cumplir correctamente con estos requisitos se han usado distintas técnicas, que por los resultados obtenidos, han sido necesario modificar. Las técnicas consisten en la determinación de los puntos que van a servir para normalizar las imágenes, tanto en tamaño como en

ángulo. No se han tomado los puntos propuestos por Davidenko [1] por la ocultación del punto 1 y la difícil identificación del punto 9 en algunas imágenes. A continuación se van a describir dichas técnicas y los problemas encontrados:

- **Puntos 2 y 3:** En la primera aproximación se pensó usar dichos puntos para normalizar las imágenes. Los puntos 2 y 3 pertenecen a la parte inmóvil de la silueta, además la distancia que describen los puntos tiene un tamaño suficiente como para ser tolerante a fallos de precisión. Estos puntos de referencia no dieron buenos resultados porque aún siendo parte inmóvil de la silueta en algunas ocasiones el punto 2 está oculto (en la Figura 22 se puede ver un ejemplo). Para determinar la ubicación del punto 2 cuando está oculto se aproxima a un punto lo más cercano posible, en estos casos el error producido por dicha aproximación es elevado en algunas ocasiones (pensemos en individuos con gafas, o con un giro elevado), variando las proporciones y ángulo de toda la imagen.



Figura 22.- Ejemplo de ocultación punto 2

- **Puntos 2,3 y 8:** Como la proporción entre los puntos 2 y 3 podía producir errores al normalizar en tamaño, se pensó en amortiguar dicho error, haciendo la media entre la proporción 2-3 y 3-8. No se quiso dar mucho más peso al punto 8 por ser una parte móvil dentro de la silueta. Esto produjo mejores resultados, pero aún no eran los deseados.
- **Puntos 3 y 8:** Por los problemas de ocultación del punto 2 se decide eliminar dicho punto para la normalización, usando el punto 3 y el 8. Como no existen problemas de ocultación para dichos puntos y la distancia entre ellos es elevada, se podría amortiguar algún error de precisión.

La obtención de puntos se podría haber intentado hacer de forma totalmente automática [4] pero estas técnicas no tienen en cuenta la ocultación, o la existencia de elementos adicionales (gafas). Hay que recordar que el objetivo del proyecto no es definir un sistema completamente automático, sino testear y analizar una serie de métodos para permitirnos extraer conclusiones y aprender sobre las técnicas más adecuadas para la construcción de dicho sistema. Si se quiere automatizar esta parte del sistema, habría que trabajar en la asignación inteligente de los puntos teniendo en cuenta además de la curvatura, la posible posición de dichos puntos.

Para la selección de los puntos en la imagen se permiten varias opciones implementadas, en todas ellas el usuario únicamente debe pinchar a la altura que quiere indicar el punto (el valor 'X' lo calcula directamente sobre la imagen). Las opciones son:

- **Seleccionar la altura directamente en la imagen:** Esta opción tiene el inconveniente de la falta de precisión, ya que al mostrarse la imagen reducida resulta muy difícil seleccionar el punto deseado.
- **Uso de zoom en la imagen:** Se selecciona un punto aproximado y el sistema muestra esa zona ampliada, de esta forma se puede seleccionar el punto deseado con precisión. Esta opción tiene el inconveniente que la selección de los puntos es muy pesada, ya que hay que pinchar 2 veces por cada punto (una selección para zoom y otra la propia selección).
- **Selección de la altura aproximada y cálculo automático:** Se selecciona un punto aproximado y el sistema calcula el punto en el que la curvatura es máxima ó mínima en función del punto a introducir. Es un algoritmo parecido al propuesto en [4], pero no comprobando la curvatura en cada punto, sino cada 5 puntos.

Hay que indicar que la tercera opción es la más adecuada, porque al tener gran parte automatizada, la selección es más estándar y la asignación de los puntos es más rápida. Se ha implementado una función para modificar manualmente algún punto, esta función recibe el número de punto que se quiere modificar y muestra la zona del punto con zoom para una asignación manual.

La base de datos final sobre la que se han hecho las pruebas que se describen en esta memoria se ha realizado usando los puntos de referencia 3 y 8 con la asignación semiautomática en función de la curvatura y modificando los puntos que no se habían seleccionado correctamente.

Con el fin de optimizar el proceso de la creación de la BD, se han creado funciones para introducir primero los puntos necesarios para la normalización (puntos 3 y 8), crear las imágenes normalizadas y posteriormente introducir los puntos finales sobre las imágenes normalizadas. Esta estructura del proceso, unido a funciones que leen las imágenes del directorio y las van mostrando para seleccionar los puntos, ha hecho que se reduzca a un 30% el tiempo en la normalización y asignación de puntos.

Hay que resaltar que la asignación del punto 9 se hace automáticamente, porque era muy difícil establecer el criterio para poder seleccionar correctamente dicho punto. La forma de obtener dicho punto, es encontrar el punto de contacto del perfil, a una distancia determinada del punto 8. La distancia está definida por la recta de 45° pintada en la Figura 23.

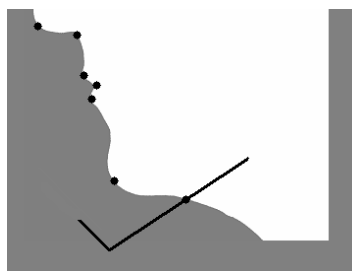


Figura 23.- Determinación del Punto 9



### 3. Proceso general de imágenes y clasificación

Este capítulo enumera las diferentes etapas de un Sistema de visión artificial y en que han consistido para el proyecto. Una vez están claras las etapas del sistema, conviene identificar las que son comunes a todos los métodos y las que no. De esta forma se puede explicar toda la parte común a los distintos métodos en el presente capítulo. La clasificación tiene una gran parte común que será explicada en el capítulo, dejando para cada uno de los métodos la parte específica de la clasificación y de la representación.

Se han creado tres escenarios de pruebas para todos los métodos y serán descritos en este capítulo.

#### 3.1. *Etapas del Sistema de Visión Artificial*

A continuación se describe brevemente en que consiste para el proyecto cada una de las etapas de los sistemas de visión por computador.

- **Adquisición de la imagen:** La adquisición de la imagen ha consistido en la toma de las fotografías en las condiciones indicadas en el apartado de la creación de la BD.
- **Preproceso:** No ha sido necesario hacer ningún tipo de filtrado ni realizado con las imágenes en este paso.
- **Segmentación:** Para extraer la información necesaria de la imagen (el perfil) se ha usado la umbralización de las mismas.
- **Representación y descripción (extracción de características):** Esta acción depende de cada método y será explicada en el capítulo correspondiente a dicho método.
- **Reconocimiento e Interpretación:** En esta etapa se tiene que construir un clasificador que determine si 2 imágenes pertenecen a la misma persona. Para todos los métodos se ha usado una red de neuronas, más concretamente un perceptrón multicapa. En función del método la configuración del perceptrón multicapa ha sido distinta.

En el capítulo de creación de la base de datos se ha hablado de las tres etapas primeras.

#### 3.2. *Descripción del clasificador*

Una vez obtenidas las características principales que identifican la silueta hay que clasificar dicha silueta. Al ser un problema de verificación las posibles clases del clasificador se reducen a dos:

- Es igual a la silueta de referencia.
- Es distinta a la silueta de referencia.

En este caso no se quiere hacer una clasificación general que indique un grupo absoluto, sino que se quieren comparar dos siluetas.

Para hacer la clasificación en el proyecto se ha optado por el uso de Redes Neuronales, más concretamente el perceptrón multicapa (FeedForward Neuronal Network).

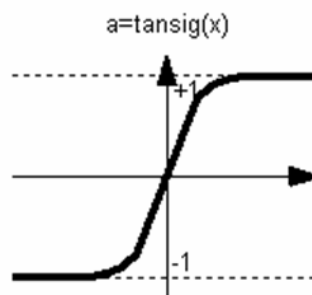
En [8] se puede encontrar una definición formal de las redes neuronales. Para simplificar se puede decir que las redes neuronales pretenden simular el comportamiento de las neuronas biológicas. Se caracterizan por la interconexión entre ellas de forma que la salida de una neurona puede ser la entrada de otra. A los elementos de unión entre neuronas se le asigna un peso a la activación de cada entrada. El proceso de aprendizaje de las neuronas consiste en la asignación de los pesos adecuados para la obtención de los resultados esperados.

Para entrenar las redes se necesita un conjunto de datos con las entradas y otro con las salidas esperadas. Al entrenar la red se asignarán los pesos a las entradas de cada una de las neuronas para que al recibir los estímulos de entrada, la salida sea la deseada. El conjunto de datos de entrenamiento se divide en dos (conjunto de entrenamiento CE y conjunto de test de entrenamiento CTE), pero por simplificar se verá como un único conjunto CE.

Otro aspecto importante para los perceptrones multicapa es el número de capas ocultas de la red y el número de neuronas por capa. Esta configuración dependerá de cada método.

Para crear el clasificador se siguen los mismos pasos en todos los métodos:

- Se define el número de neuronas de entrada y el rango de datos de cada entrada. El número de neuronas de entrada y el rango depende del vector de características del método.
- Se define el número de capas ocultas y el número de neuronas por capa.
- Se define la función de activación entre capas. Para todos los casos es la función “*tansig*” (tangente sigmoial). Esta función toma valores entre -1 y 1. La Figura 24 se puede ver la función tangente sigmoial.
- Se crea la red de neuronas con las definiciones anteriores.
- Se entrena la red de neuronas
- Se comprueba el funcionamiento con los datos de las pruebas.



**Figura 24.- Función tangente sigmoial**

Al estar en un problema de verificación, la entrada de las neuronas debe relacionar las dos imágenes que se quieren comprobar. En los casos en los que el método extraiga características de una persona, será necesario realizar la diferencia en valor absoluto de los vectores de características de ambos perfiles. El único método que compara ambos perfiles directamente es el de los snakes. El resto obtiene características de un perfil.

Todos los métodos comparten el conjunto de datos de entrenamiento. Para entrenar las redes se usan 10 personas (4 de ellas con gafas) y para cada persona se compara el perfil de posición 2 (giro de 0°) con:

- Todas las fotos de la persona (En total 83 comparaciones:  $10 \times 6 + 4 \times 6$ ).
- Todas las fotos con giro 0° de todas las personas (En total 900 comparaciones:  $10 \times (39 \times 2 + 6 \times 2)$ ).

Cuando se van a preparar los conjuntos de datos para pruebas se eliminan siempre los que se han usado para entrenar. De esta manera se asegura que no se pruebe con los datos que se ha entrenado.

### **3.3. Descripción de las pruebas**

Para testear todos los métodos se van a definir tres tipos o escenarios de prueba, los escenarios dependen de los perfiles implicados en las pruebas. Las pruebas consisten en comparar las siluetas de dos personas y determinar si pertenecen o no a la misma persona.

Los escenarios definidos para las pruebas son los siguientes:

- **Escenario 1: Perfiles sin giro horizontal.** Este es el escenario ideal, supone que se tiene almacenado en el sistema un perfil sin ningún giro horizontal y se compara con otro perfil sin ningún giro horizontal.
- **Escenario 2: Perfil sin giro horizontal contra todos los perfiles.** Este es un escenario real, se tiene almacenado un perfil sin ningún giro y se compara con otro perfil que puede tener cualquier tipo de giro. Un ejemplo del sistema sería el siguiente: Para registrar a una persona se toma una fotografía sin ningún giro y cuando la persona se presenta al sistema se le vuelve a tomar otra fotografía, esta fotografía puede tener cualquier giro.
- **Escenario 3: Todos los perfiles con todos los perfiles.** Este escenario supone que ambos perfiles pueden tener cualquier giro horizontal.

Una vez definidas las condiciones de los escenarios de las pruebas hay que determinar las imágenes que están involucradas en dichas pruebas. A la hora de definir los conjuntos de datos de las pruebas, hay que tener en cuenta que las comparaciones usadas para entrenar la red de neuronas no se deben usar para las pruebas. Para resolver este problema se han seguido los siguientes pasos:

- Se define el caso general sin tener en cuenta el entrenamiento.
- Se elimina del caso general los datos usados para el entrenamiento.

De esta forma se asegura que no se prueba con los datos de entrenamiento.

Antes de definir las imágenes de cada escenario, conviene recordar que las imágenes sin giro horizontal llevan los sufijos: 2, 5, 8 ó 11. Los sufijos 8 y 11 pertenecen a personas con gafas. En las personas que no tienen gafas estos sufijos no aplican. La base de datos tiene 39 personas y 6 de ellas tienen gafas.

En función del escenario, las imágenes seleccionadas han sido:

- **Escenario 1:** Se toma la imagen 2 de una persona como referencia y se compara con las imágenes 2, 5, 8 y 11 de todas las personas. Esta acción se repite con todas las personas de la BD. En total se hacen 2610 comparaciones,  $39 \times (39 \times 2 + 6 \times 2) - 900$ . De las 2610 comparaciones 62 son entre perfiles de la misma persona y 2548 de diferentes personas.
- **Escenario 2:** Se toma la imagen 2 de una persona y se compara con todas las imágenes de todas las personas. La acción se repite con todas las personas de la BD. En total se hacen 9535 comparaciones,  $39 \times (39 \times 6 + 6 \times 6 - 1)$  – “las que aparecen en el conjunto de entrenamiento”. De las 9535 comparaciones 185 son entre perfiles de la misma persona y 9350 de diferentes personas.
- **Escenario 3:** Se comparan todas las imágenes de una persona con todas las imágenes de todas las personas. La acción se repite con todas las personas de la BD. En total se hacen 35424 comparaciones,  $269^2 -$  “las repetidas que aparecen en el conjunto de entrenamiento”. De las 35424 comparaciones 816 son entre perfiles de la misma persona y 34608 de diferentes personas

### 3.4. *Presentación de resultados*

La prueba consiste en preguntar al clasificador si dos perfiles pertenecen a la misma persona. El clasificador responderá con un valor entre 0 y 1 indicando la semejanza de dichos perfiles. Un valor de 0 indicará que no se parecen nada y un valor de 1 que son iguales. El clasificador deberá responder a cada comparación definida por el escenario correspondiente. Como resultado de la prueba el clasificador devolverá un valor entre 0 y 1 para cada comparación de perfiles realizada.

Para mostrar los datos se trocea el intervalo [0 1] en saltos de 0,05. Este valor de salto indicará el umbral para determinar si los perfiles pertenecen a la misma persona. Si el clasificador devuelve un valor superior al umbral los perfiles pertenecen a la misma persona y si es inferior no pertenecen a la misma persona. En este punto es necesario determinar los casos para los que el clasificador acierta / falla con un valor de umbral determinado. Para comprobar dichos casos, se compara la salida estimada (valor 0 no pertenece a la misma persona, valor 1 pertenece a la misma persona) con la salida real y se obtienen los casos de fallo. La salida real se obtiene mientras se crea el escenario de pruebas. Cuando se define la comparación entre dos perfiles se determina si pertenecen o no a la misma persona. Los dos tipos de fallo posibles son:

- **Falsos Positivos:** Son los casos de fallo en los que el sistema identifica que los dos perfiles pertenecen a la misma persona y la salida real determina que no pertenecen a las mismas personas.
- **Falsos Negativos:** Este fallo se da cuando dos perfiles pertenecen a la misma persona y el sistema identifica que pertenecen a personas distintas.

La comprobación se realiza mediante la siguiente resta: valor obtenido menos valor real. El resultado de la resta puede ser:

- Valor 0: La salida deseada y obtenida son iguales. El sistema ha acertado.
- Valor 1: Se produce un FP, la salida obtenida es un '1' mientras que la deseada es '0'.

- Valor -1: Se produce un FN, la salida obtenida es '0' mientras que la deseada es '1'.

Con los datos obtenidos de las pruebas, después de realizar las comparaciones de resultados se rellenará una tabla (la Tabla 1 muestra un ejemplo) que tiene las columnas indicadas a continuación:

- **Umbral:** Valor de corte sobre el grado de semejanza que indica que 2 perfiles pertenecen a la misma persona.
- **Fallos:** Número de fallos para el umbral definido. Suma de FP y FN
- **Aciertos:** Número de aciertos para el umbral definido.
- **Falsos Positivos (FP)**
- **Falsos Negativos (FN)**
- **% Falsos Positivos:** Tanto por ciento de casos en los que el sistema determina que dos perfiles de personas distintas pertenecen a la misma persona.
- **% Falsos Negativos:** Tanto por ciento de casos en los que el sistema determina que dos perfiles de la misma persona pertenecen a distintas personas.

**Tabla 1.- Ejemplo de resultado de prueba**

Umbral	Fallos	Aciertos	FP	FN	% FP	% FN
0,00						
0,05						
...						
1,00						

Además de la tabla, se mostrará una gráfica ROC. En la grafica ROC se puede ver como evolucionan los valores de Falsos Positivos y Falsos Negativos en función del umbral.

Para cada método, se mostrará en el resumen una tabla con el valor del umbral óptimo (EER equal error rate) para cada escenario y los valores de falsos positivos y falsos negativos para dicho umbral. La Tabla 2 muestra un ejemplo. El valor del umbral óptimo es en el que se cortan las curvas de FP y FN. Como los datos se obtienen muestreando cada 0,05 se tomará el valor de muestra que minimice la diferencia entre FP y FN.

**Tabla 2.- Ejemplo de tabla resumen de método**

	ESCENARIO 1	ESCENARIO 2	ESCENARIO 3
<b>Falsos Positivos</b>			
<b>Falsos Negativos</b>			
<b>Valor de corte</b>			

## 4. Método Geométrico

Este método se basa en la extracción de ciertos puntos característicos de la silueta, para así obtener las medidas de distancias y ángulos entre ellos. Con las medidas tomadas quedará definida la silueta y su representación será reducida a los valores obtenidos. Estos valores guardan relación de proporción entre partes por distancias y ángulos.

Para probar el método se ha creado y entrenado un perceptrón multicapa con 2 capas ocultas.

En la Tabla 3 se muestra un resumen de los resultados obtenidos para cada uno de los escenarios definidos, en ella se puede ver el % de Falsos Positivos y Falsos Negativos para un umbral (óptimo).

**Tabla 3.- Resumen del resultado del método trigonométrico**

	<b>ESCENARIO 1</b>	<b>ESCENARIO 2</b>	<b>ESCENARIO 3</b>
<b>Falsos Positivos</b>	5,85%	21,45%	21,11%
<b>Falsos Negativos</b>	6,45%	10,27%	17,28%
<b>Valor de corte</b>	0,3	0,05	0,05

Los resultados determinan que el método es muy sensible a la ubicación de los puntos de interés.

### 4.1. Descripción del método

Según el estudio de Davidenko [1] en las siluetas hay suficiente información para que una persona sea capaz de diferenciar a otra. Se necesita extraer información de las siluetas de manera que se pueda diferenciar unas siluetas de otras. Una de las cosas en las que nos fijamos las personas es en las proporciones de las partes de la cara. El método geométrico trata de extraer la información del perfil referente a estas proporciones y relaciones entre los puntos significativos de la cara.

Es importante a la hora de definir estas propiedades, tener en cuenta que algunos puntos son invariantes ó inmóviles y otros en cambio pueden depender de la expresión o gesto de la persona. Puntos inmóviles se consideran aquellos en los que el cambio de posición es muy pequeño. Según los puntos mostrados en la Figura 21, podemos definir los puntos 1, 2, 3, 4 como inmóviles, 5, 6, 7 como móviles (por la expresión de la boca) y los puntos 8 y 9 aún siendo dependientes de la expresión de la boca, tienen una relación de 'inmóviles' con respecto al punto 7. En la Figura 25 se puede ver como cambian los puntos 5,6 y 7 por el gesto de la boca.

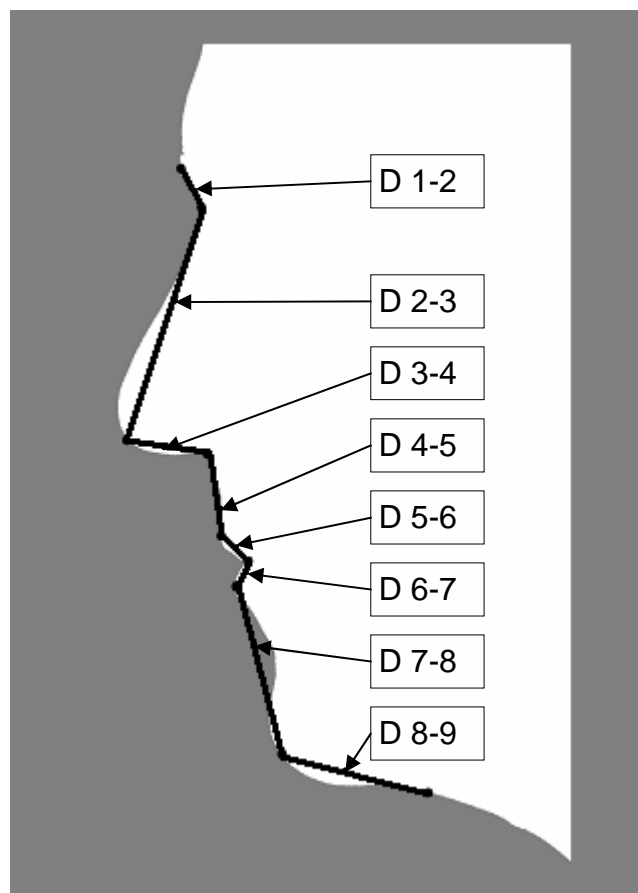


**Figura 25.- Cambio en los puntos móviles**

Teniendo en cuenta las relaciones entre los puntos móviles e inmóviles, se van a definir unas relaciones de tamaño/distancia y ubicación/ángulo entre puntos de forma que queden registradas todos los rasgos importantes del rostro. La relaciones definidas para un silueta se pueden comparar con otras por tener normalizadas las imágenes en tamaño y ángulo. Se han tomado 15 medidas de cada silueta repartidas de la siguiente manera:

- 8 medidas de tamaño/distancia.
- 4 medidas de ángulos relativos entre puntos.
- 3 medidas de ángulos respecto a un punto calculado.

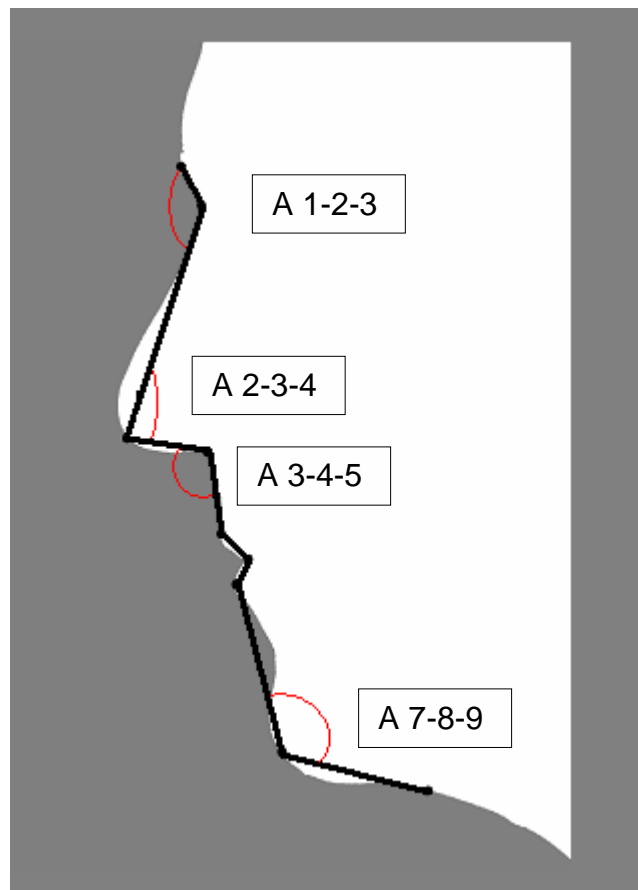
En la Figura 26 se pueden ver las medidas de distancia tomadas sobre la imagen, estas medidas corresponden a las distancias consecutivas entre puntos. La medida 1-2 corresponde con la distancia entre los puntos 1 y 2, de esta forma se toman las medidas de distancia de todos los puntos contiguos y se obtienen las proporciones entre las distintas partes de la cara.



**Figura 26.- Medidas de distancia**

Estas medidas de proporción guardan gran información de las personas condicionando las características del individuo. Por ejemplo si la 'D 3-4' es muy grande, la persona a la que pertenece la cara tendrá una gran nariz, o una nariz que sobresale mucho del rostro. Al unir las medidas 'D 3-4' y 'D 2-3' se obtiene información más concreta de la nariz y podrá indicar si la nariz tiene gran tamaño, o si es muy puntiaguda. La distancia 'D 7-8' proporciona información del tamaño de la mandíbula, 'D 5-6' y 'D 6-7' dan información de la forma de la boca. Cada medida de distancia aporta información sobre el individuo.

Con las medidas de distancia se obtiene información sobre las proporciones de las partes de la cara. Si se une esta información a las medidas de determinados ángulos se obtiene una información más completa. En la Figura 27 se pueden ver las medidas de ángulos entre puntos relativos. Estas medidas reflejan la relación de ubicación entre puntos (inmóviles). Así se obtiene el ángulo de la cavidad ocular, el de la nariz, el de la base de la nariz y el de la barbilla. Se han omitido los ángulos entre puntos móviles, porque podrían variar considerablemente y anularía dichas medidas.



**Figura 27.- Medidas de ángulos entre puntos**

Hasta ahora se han obtenido los ángulos que forman los puntos entre sí. También interesa tener información de giro o rotación de algunos segmentos de la figura y así completar la información del rostro. Para poder tener información de giro de algunos segmentos independientemente de los puntos contiguos, es necesario definir un tercer punto de referencia. Este punto de referencia se calcula en base al punto 3 y se desplaza la coordenada 'X' una cantidad elevada de píxeles (aprox. 2/3 de la distancia entre los



puntos 3 y 8). Al desplazar la coordenada 'X' el punto queda alejado del "borde" del perfil. Una vez obtenido el punto de referencia, se calculan los ángulos entre los segmentos 1-2, 2-3 y 4-5. En la Figura 28 se puede ver la ubicación del punto de referencia (PR) y los ángulos tomados.

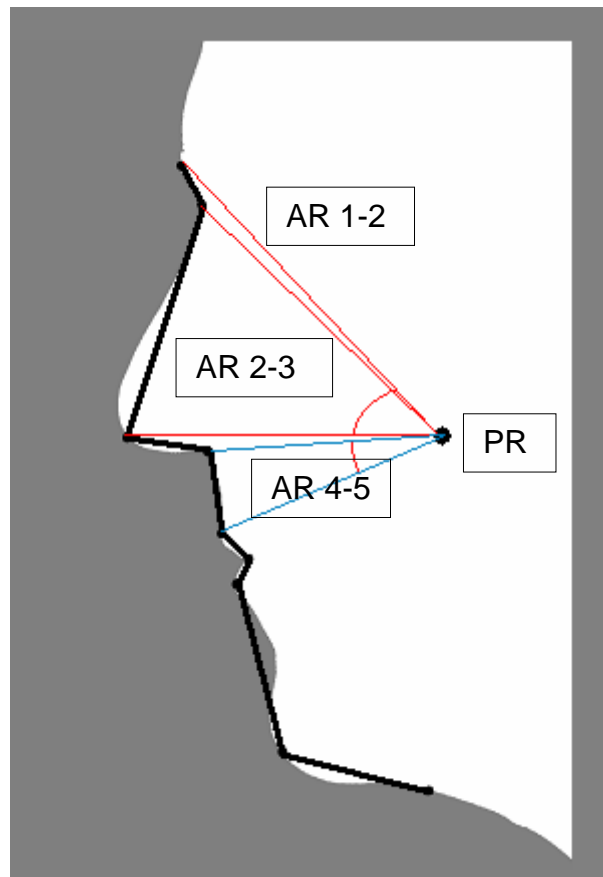


Figura 28.- Ángulos respecto al punto de referencia

Con estas 15 medidas se tiene una definición completa de la estructura del rostro de la persona. Hay que tener en cuenta que al tomar estas 15 medidas se está dejando pasar información, el motivo de dejar pasar ésta información es que no responde a la estructura o esqueleto de la persona y que puede cambiar con el paso del tiempo, o con un cambio de peso de la persona. En la Figura 26 se ven las partes del perfil que se desprecian al no estar la información del borde dentro de las líneas marcadas.

Una vez obtenidas las características, es necesario definir un clasificador que nos diga si dos perfiles pertenecen o no a la misma persona. Para el clasificador se ha elegido un perceptrón multicapa con 2 capas ocultas. La primera capa oculta tiene 14 neuronas y la segunda 3. Se han probado distintas configuraciones y es esta la que ha dado mejores resultados. En la Figura 29 se puede ver un esquema de la red. Se ha elegido esta configuración, porque después de realizar diversas pruebas ha sido la configuración que ha dado mejores resultados.

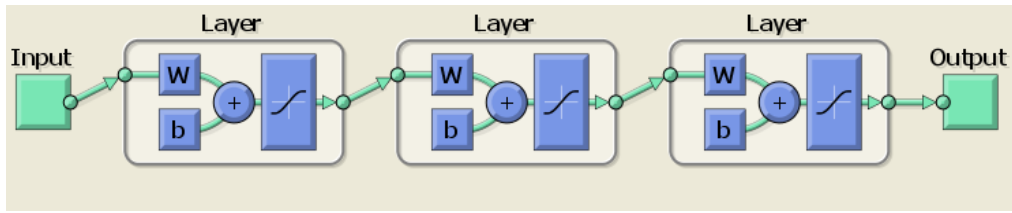


Figura 29.- Perceptrón multicapa para el método geométrico

## 4.2. Descripción del experimento

El experimento consiste en probar el grado en el que el método es capaz de determinar si dos siluetas pertenecen a la misma persona. Se ha probado el método para cada uno de los escenarios descritos en el capítulo 3.

### 4.2.1. Escenario 1: Perfiles sin giro horizontal

Los resultados de las pruebas para este escenario se muestran en la Tabla 4. El Gráfico 1 muestra la variación de los FP y FN con el umbral.

Tabla 4.- Resultado para método geométrico escenario 1

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	2548	62	2548	0	100,00%	0,00%
0,05	557	2053	556	1	21,97%	0,00%
0,10	388	2222	387	1	15,39%	0,00%
0,15	289	2321	288	1	11,41%	0,00%
0,20	228	2382	225	3	8,94%	3,45%
0,25	181	2429	177	4	7,03%	5,17%
0,30	153	2457	149	4	5,89%	5,17%
0,35	130	2480	122	8	4,87%	10,34%
0,40	110	2500	102	8	4,10%	10,34%
0,45	100	2510	92	8	3,70%	10,34%
0,50	89	2521	78	11	3,17%	15,52%
0,55	69	2541	57	12	2,32%	17,24%
0,60	64	2546	49	15	1,99%	22,41%
0,65	62	2548	43	19	1,75%	29,31%
0,70	55	2555	35	20	1,42%	31,03%
0,75	54	2556	30	24	1,22%	37,93%
0,80	48	2562	22	26	0,89%	41,38%
0,85	42	2568	11	31	0,45%	48,28%
0,90	41	2569	9	32	0,37%	50,00%
0,95	65	2545	3	62	0,12%	100,00%
1,00	62	2548	0	62	0,00%	100,00%

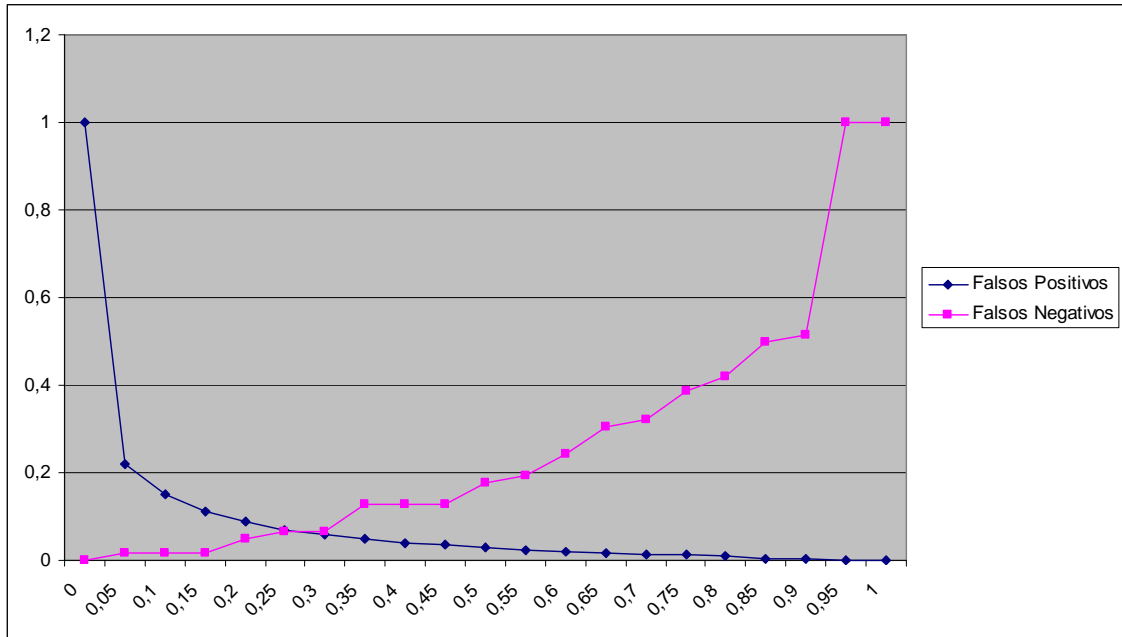


Gráfico 1.- Resultado método geométrico escenario 1

#### 4.2.2. Escenario 2: Perfil sin giro horizontal contra todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 5. El Gráfico 2 muestra la variación de los FP y FN con el umbral.

Tabla 5.- Resultado para método geométrico escenario 2

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	9350	185	9350	0	100,00%	0,00%
0,05	2025	7510	2006	19	21,97%	0,00%
0,10	1384	8151	1347	37	15,39%	0,00%
0,15	1071	8464	1024	47	11,41%	0,00%
0,20	882	8653	823	59	8,94%	3,45%
0,25	739	8796	675	64	7,03%	5,17%
0,30	634	8901	565	69	5,89%	5,17%
0,35	561	8974	482	79	4,87%	10,34%
0,40	488	9047	404	84	4,10%	10,34%
0,45	434	9101	349	85	3,70%	10,34%
0,50	384	9151	294	90	3,17%	15,52%
0,55	333	9202	237	96	2,32%	17,24%
0,60	305	9230	201	104	1,99%	22,41%
0,65	276	9259	163	113	1,75%	29,31%
0,70	248	9287	130	118	1,42%	31,03%
0,75	230	9305	99	131	1,22%	37,93%
0,80	211	9324	71	140	0,89%	41,38%
0,85	194	9341	44	150	0,45%	48,28%
0,90	184	9351	30	154	0,37%	50,00%
0,95	195	9340	10	185	0,12%	100,00%
1,00	185	9350	0	185	0,00%	100,00%

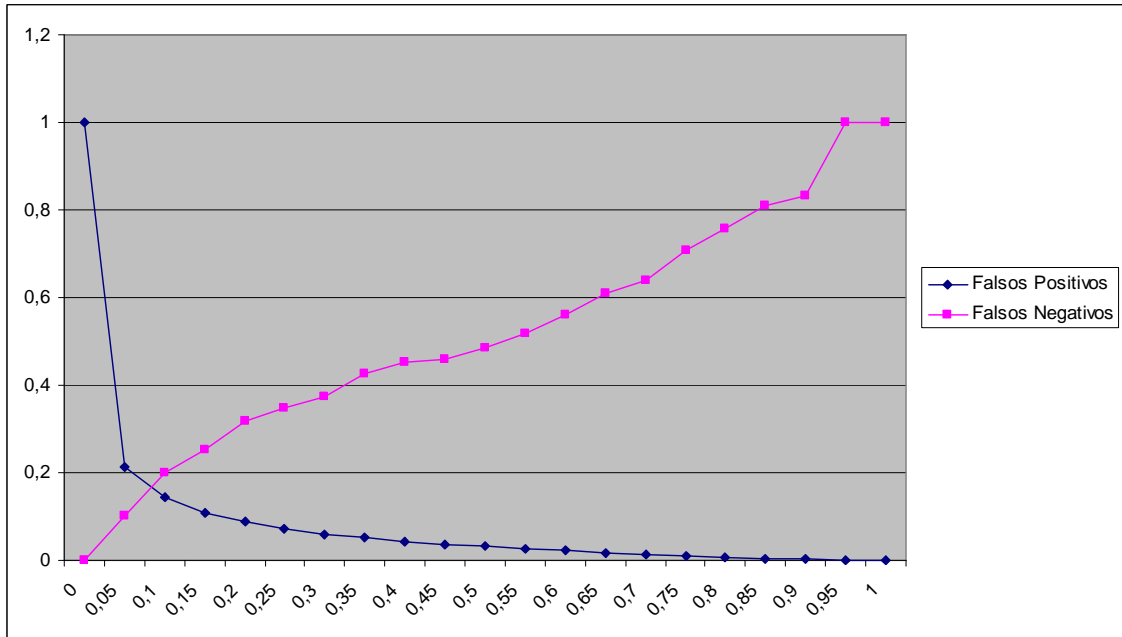


Gráfico 2.- Resultado para método geométrico escenario 2

### 4.2.3. Escenario 3: Todos los perfiles con todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 6. El Gráfico 3 muestra la variación de los FP y FN con el umbral.

Tabla 6.- Resultado para método geométrico escenario 3

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	34608	816	34608	0	100,00%	0,00%
0,05	7446	27978	7305	141	21,97%	0,00%
0,10	5033	30391	4814	219	15,39%	0,00%
0,15	3943	31481	3683	260	11,41%	0,00%
0,20	3313	32111	3007	306	8,94%	3,45%
0,25	2835	32589	2483	352	7,03%	5,17%
0,30	2503	32921	2121	382	5,89%	5,17%
0,35	2239	33185	1827	412	4,87%	10,34%
0,40	1984	33440	1544	440	4,10%	10,34%
0,45	1791	33633	1325	466	3,70%	10,34%
0,50	1629	33795	1133	496	3,17%	15,52%
0,55	1458	33966	932	526	2,32%	17,24%
0,60	1351	34073	784	567	1,99%	22,41%
0,65	1267	34157	670	597	1,75%	29,31%
0,70	1171	34253	541	630	1,42%	31,03%
0,75	1082	34342	411	671	1,22%	37,93%
0,80	1010	34414	302	708	0,89%	41,38%
0,85	945	34479	199	746	0,45%	48,28%
0,90	915	34509	123	792	0,37%	50,00%
0,95	859	34565	46	813	0,12%	100,00%
1,00	816	34608	0	816	0,00%	100,00%

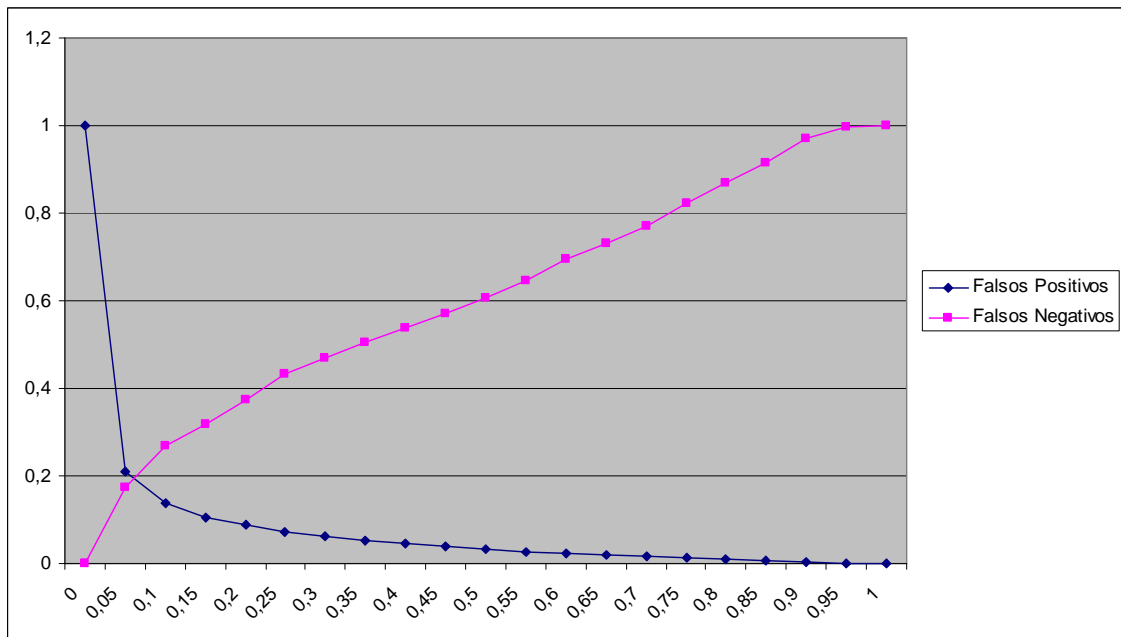


Gráfico 3.- Resultado para método geométrico escenario 3

### 4.3. Análisis de resultados

Una vez obtenidos los datos de las pruebas en los distintos escenarios se debería determinar el umbral que mejor se adapta a las necesidades del sistema. Como se pretende evaluar el método en cada uno de los escenarios vamos a hablar del umbral que en cada caso hace mínimo las dos funciones FP y FN.

Se van a analizar cada uno de los escenarios:

- Escenario 1: Seleccionando un umbral de 0,3 se tiene un 5,85% de FP y un 6,45% de FN. Se obtiene muy buena respuesta cuando se comparan perfiles que no tienen ningún giro. Se debe tener en cuenta que se están comparando imágenes con gafas, cambio de expresión, etc.
- Escenario 2: El umbral está entre el 0,05 y el 0,1 que hace oscilar los valores entre un 21% - 14% para FP y entre un 10% - 20% para FN. Para este caso se puede indicar que cuando se producen giros en las siluetas la precisión de sistema baja.
- Escenario 3: Para un umbral de 0,05 se obtienen unos valores del 21% para FP y un 17% para FN. En este caso se puede decir que si no se controla el giro en ninguna imagen (ni la de referencia ni la comparada) el rendimiento baja a un 20% aproximadamente.

Con los datos obtenidos parece que el método no se comporta bien frente a giros, pero hay que explicar el trabajo adicional realizado en el escenario 1. Con los puntos marcados inicialmente en la BD se ejecutó la prueba y los resultados mostraban que el sistema fallaba alrededor del 12% (en FP y FN aprox.). No se analizaron los casos en los que el sistema fallaba con FP porque el umbral seleccionado era muy bajo (0,1 aproximadamente). Como el problema eran los FN se analizaron con detalle los casos en los que fallaba. Algunos de ellos eran perfiles muy similares en los que no tenía

justificación aparente dicho error. Comprobando la definición/ubicación de los puntos de interés se detectaron puntos que no estaban bien definidos y esto hacia tener unos valores de semejanza muy bajos. Redefiniendo dichos puntos, los valores de semejanza aumentaban considerablemente y el valor del umbral óptimo se podía subir. Esta acción se tomo en algunos casos, tanto en casos en los que los perfiles eran muy parecidos, como en los que tenían alguna diferencia o incluso alguna parte oculta. Los resultados mejoraron para ambos casos. No se ha modificado la definición de puntos de todos los casos en los que el sistema no ha acertado porque no se pretendía obtener una tasa de fallos determinada, sino comprobar si el método puede aportar buenos resultados y cuáles son sus posibles causas de error.

Si se repitiera el proceso seguido en el escenario 1 con los escenarios 2 y 3 el valor del umbral óptimo se podría subir y obtener mejores resultados. Por este motivo hay que destacar que el método funciona bien tanto en los casos con giros cómo sin giros (se producen muchos aciertos en perfiles girados), pero es muy sensible a la definición de los puntos y la mala ubicación de un punto nos puede hacer bajar mucho el grado de semejanza.

Durante el análisis de resultados, comprobando los puntos se observo que la mala ubicación del punto 9 producía muchos fallos. La dificultad de ubicación uniforme del punto 9 es elevada. Se ha probado dicho punto de las medidas y el rendimiento ha bajado, lo que indica que cuando estaba bien situado nos aportaba gran información. Por este motivo se ha decidido asignar el punto de forma automática.

## 5. Método de Curvatura Local

La curvatura local, se define como la variación del ángulo de la tangente respecto a un punto. Para la obtención de la variación del ángulo se muestrearán en 'n' puntos separados por 'm' píxeles. De esta forma se definirá la silueta como la variación del ángulo ( $2 * n$  muestras) en los puntos significativos de la silueta.

Con las características obtenidas se ha creado y entrenado un perceptrón multicapa con 2 capas ocultas.

Los resultados de las pruebas de cada escenario se pueden ver en la Tabla 7, donde se muestra el % de FP y FN para un umbral determinado (óptimo).

Tabla 7.- Resumen del resultado del método de curvatura local

	ESCENARIO 1	ESCENARIO 2	ESCENARIO 3
<b>Falsos Positivos</b>	10,40%	17,03%	23,40%
<b>Falsos Negativos</b>	30,65%	53,51%	50,86%
<b>Valor de corte</b>	0,05	0,05	0,05

### 5.1. Descripción del método

Cuando al reconocer a una persona nos centramos en alguna característica específica de dicho perfil, nos estamos fijando en la forma que tiene esa parte concreta. La forma de esa parte del rostro se define por la secuencia de puntos que pertenecen al contorno. Se podría usar un método para la definición del contorno basado en dicha secuencia de puntos. Al ser curvas todas las partes del rostro que se quieren medir se ha usado un método que mide la variación del ángulo de los puntos del contorno, de esta manera se puede obtener la información de la forma de la silueta.

En la Figura 30 se puede ver el ejemplo de la definición de puntos para dos partes del rostro (la nariz y la barbilla). En cada figura se puede ver un punto de mayor tamaño y una serie de puntos alrededor de éste. El punto de mayor tamaño corresponde con un punto de interés y será usado como referencia. Los puntos más pequeños van definiendo el contorno y el método se basa en tomar la curvatura de cada uno de estos puntos respecto al punto de interés.

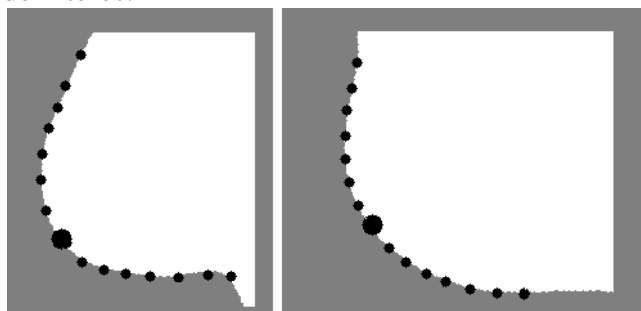


Figura 30.- Ejemplo de puntos para la curvatura local

Por definición de curvatura tenemos la variación del ángulo de la tangente en dichos puntos. En [12] se puede ver una definición de la curvatura. La Figura 31 muestra un ejemplo de curvatura.

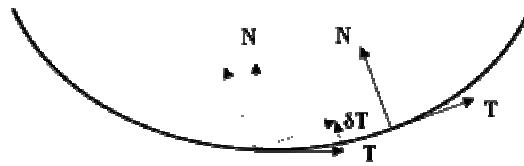


Figura 31.- Ejemplo de curvatura

Para cada punto de interés se ha medido la curvatura en 40 puntos, 20 anteriores y 20 posteriores. Así queda registrada la variación de la curvatura en la zona próxima a cada punto de interés. Como para cada punto se toman 40 medidas y tenemos 9 puntos, la silueta quedaría definida por este método con 360 características.

Es necesario crear el clasificador que nos determine si dos siluetas pertenecen o no a la misma persona y como en el caso anterior se ha creado un perceptrón multicapa con 2 capas ocultas. La primera capa tiene 12 neuronas y la segunda 3. La Figura 32 muestra el esquema del perceptrón utilizado.

Al crear la red para las características definidas por el método (360 características) y realizar pruebas, se comprobó que los tiempos de respuesta de la red eran muy elevados (tanto para el entrenamiento como en test). Al estar conectadas las neuronas de la primera capa con todas las entradas (360) el número de conexiones era muy elevado. Por este motivo se decidió reducir el n° de características con la que trabajaría la red y se tomaron 10 medidas por cada punto de interés (5 en cada sentido), así el n° total de características quedó reducido a 90. Los resultados usando un número de características u otro eran similares y se ganaba en tiempo de respuesta. En la primera aproximación con 360 características se medía la curvatura cada 5 píxeles y ahora se mide cada 20 píxeles.

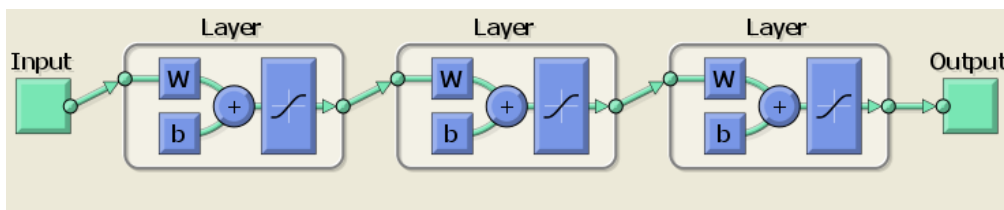


Figura 32.- Perceptrón multicapa para el método de curvatura local

## 5.2. Descripción del experimento

El experimento consiste en probar el grado en el que el método es capaz de determinar si dos siluetas pertenecen a la misma persona. Se ha probado el método para cada uno de los escenarios descritos en el capítulo 3.

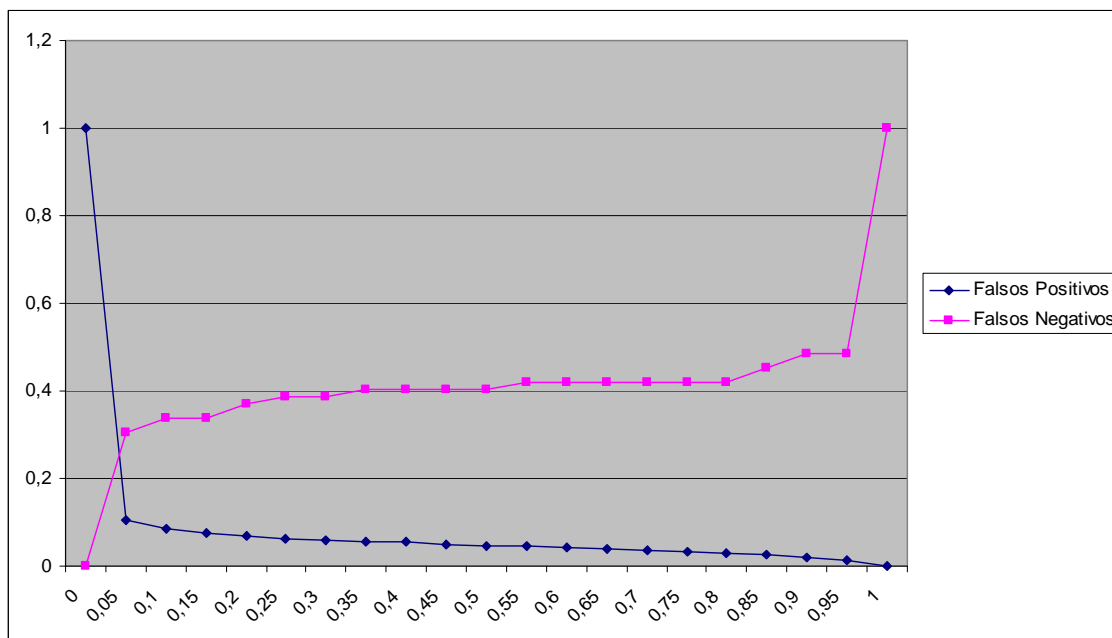
### 5.2.1. Escenario 1: Perfiles sin giro horizontal

Los resultados de las pruebas para este escenario se muestran en la Tabla 8. El Gráfico 4 muestra la variación de los FP y FN con el umbral.



**Tabla 8.- Resultado para método curvatura local escenario 1**

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	2548	62	2548	0	100,00%	0,00%
0,05	284	2326	265	19	10,40%	30,65%
0,10	237	2373	216	21	8,48%	33,87%
0,15	217	2393	196	21	7,69%	33,87%
0,20	200	2410	177	23	6,95%	37,10%
0,25	183	2427	159	24	6,24%	38,71%
0,30	176	2434	152	24	5,97%	38,71%
0,35	170	2440	145	25	5,69%	40,32%
0,40	165	2445	140	25	5,49%	40,32%
0,45	150	2460	125	25	4,91%	40,32%
0,50	142	2468	117	25	4,59%	40,32%
0,55	139	2471	113	26	4,43%	41,94%
0,60	133	2477	107	26	4,20%	41,94%
0,65	127	2483	101	26	3,96%	41,94%
0,70	121	2489	95	26	3,73%	41,94%
0,75	111	2499	85	26	3,34%	41,94%
0,80	101	2509	75	26	2,94%	41,94%
0,85	93	2517	65	28	2,55%	45,16%
0,90	81	2529	51	30	2,00%	48,39%
0,95	65	2545	35	30	1,37%	48,39%
1,00	62	2548	0	62	0,00%	100,00%



**Gráfico 4.- Resultado método curvatura local escenario 1**

## 5.2.2. Escenario 2: Perfil sin giro horizontal contra todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 9. El Gráfico 5 muestra la variación de los FP y FN con el umbral.

Tabla 9.- Resultado para método curvatura local escenario 2

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	9350	185	9350	0	100,00%	0,00%
0,05	1691	7844	1592	99	17,03%	53,51%
0,10	1445	8090	1337	108	14,30%	58,38%
0,15	1351	8184	1239	112	13,25%	60,54%
0,20	1264	8271	1148	116	12,28%	62,70%
0,25	1189	8346	1071	118	11,45%	63,78%
0,30	1148	8387	1028	120	10,99%	64,86%
0,35	1102	8433	981	121	10,49%	65,41%
0,40	1058	8477	937	121	10,02%	65,41%
0,45	1021	8514	898	123	9,60%	66,49%
0,50	980	8555	856	124	9,16%	67,03%
0,55	940	8595	814	126	8,71%	68,11%
0,60	909	8626	783	126	8,37%	68,11%
0,65	883	8652	757	126	8,10%	68,11%
0,70	849	8686	723	126	7,73%	68,11%
0,75	814	8721	688	126	7,36%	68,11%
0,80	781	8754	653	128	6,98%	69,19%
0,85	740	8795	609	131	6,51%	70,81%
0,90	682	8853	547	135	5,85%	72,97%
0,95	598	8937	463	135	4,95%	72,97%
1,00	185	9350	0	185	0,00%	100,00%

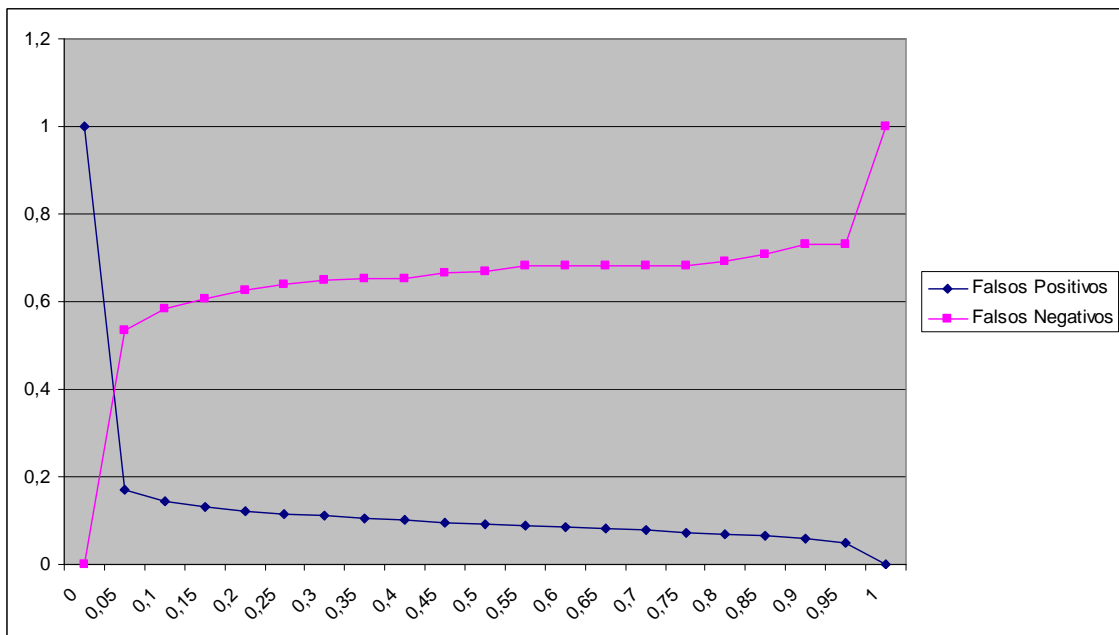


Gráfico 5.- Resultado para método curvatura local escenario 2

### 5.2.3. Escenario 3: Todos los perfiles con todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 10. El Gráfico 6 muestra la variación de los FP y FN con el umbral.

Tabla 10.- Resultado para método curvatura local escenario 3

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	34608	816	34608	0	100,00%	0,00%
0,05	8514	26910	8099	415	23,40%	50,86%
0,10	7456	27968	7003	453	20,24%	55,51%
0,15	6947	28477	6482	465	18,73%	56,99%
0,20	6549	28875	6067	482	17,53%	59,07%
0,25	6272	29152	5778	494	16,70%	60,54%
0,30	6062	29362	5555	507	16,05%	62,13%
0,35	5861	29563	5349	512	15,46%	62,75%
0,40	5674	29750	5155	519	14,90%	63,60%
0,45	5519	29905	4992	527	14,42%	64,58%
0,50	5334	30090	4801	533	13,87%	65,32%
0,55	5188	30236	4649	539	13,43%	66,05%
0,60	5057	30367	4510	547	13,03%	67,03%
0,65	4905	30519	4350	555	12,57%	68,01%
0,70	4758	30666	4196	562	12,12%	68,87%
0,75	4592	30832	4023	569	11,62%	69,73%
0,80	4415	31009	3836	579	11,08%	70,96%
0,85	4188	31236	3591	597	10,38%	73,16%
0,90	3883	31541	3265	618	9,43%	75,74%
0,95	3462	31962	2823	639	8,16%	78,31%
1,00	816	34608	0	816	0,00%	100,00%

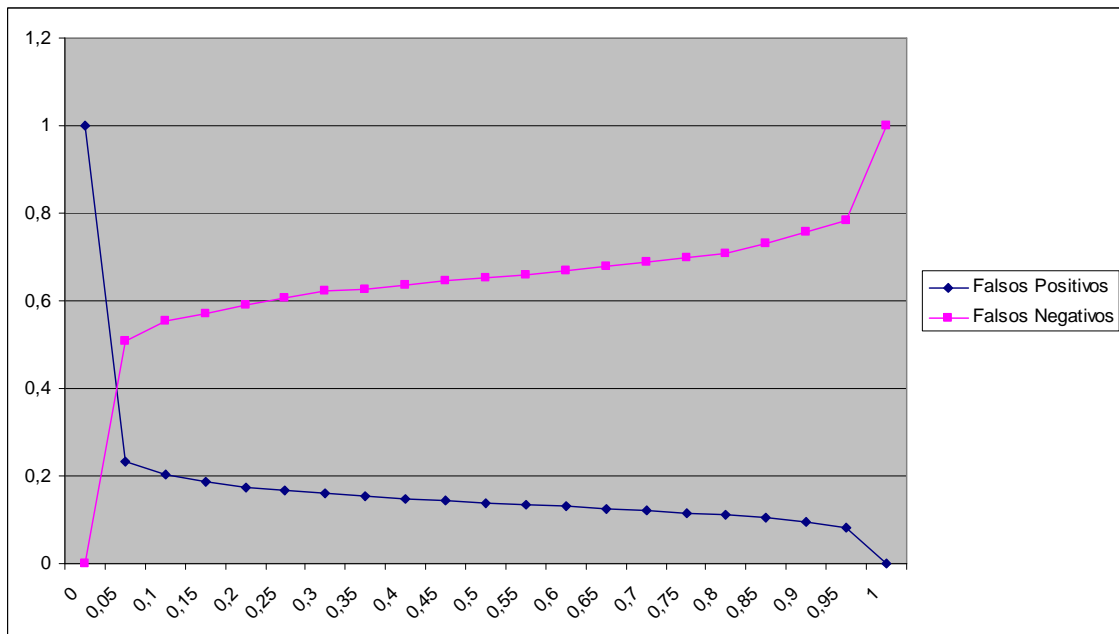


Gráfico 6.- Resultado para método curvatura local escenario 3

### 5.3. Análisis de resultados

Observando los resultados, destaca que el valor de Falsos Negativos es muy elevado incluso para el valor mínimo de umbral (para el que tenemos datos). La tasa elevada de FN se da tanto para el caso de perfiles perfectos (un 27%) como para los otros casos (50% aprox.). Este comportamiento denota una inadecuada capacidad para caracterizar las siluetas, que está en parte justificada por la definición de dicho método. Se están comparando las variaciones de curvatura de 'n' puntos respecto a un punto de interés. La ubicación de dicho punto de interés y de los puntos con los que se compara no tiene porque tener una relación de exactitud entre distintas siluetas. De esta forma se estarían comparando puntos próximos cuya unión pueden hacer variar la curvatura en dichos puntos. No parece muy adecuado para comparar la curvatura hacerlo con los valores en puntos concretos. Debería ser algo más general porque la determinación de los puntos no es exacta entre siluetas.

Aunque para el caso de perfiles perfectos, la tasa de aciertos es más alta, sigue siendo muy baja (73%) y al no tener margen para definir el umbral que más convenga a la aplicación, hacen el método poco eficiente.

La curvatura de las partes del rostro parece que es determinante para la identificación que realizamos los humanos. En resumen los resultados del método propuesto no son buenos.

## 6. Método de Curvatura Local Acumulada

La curvatura local acumulada toma como características los valores medios de curvatura respecto a los puntos de interés. El motivo por el que se toma la media es para evitar la dependencia respecto a los puntos seleccionados.

Por cada punto de interés se toman 2 medidas, la media de los puntos anteriores al de interés y la media de los puntos posteriores. En total se toman 18 características (2 por punto) y con estas se crea y entrena un perceptrón multicapa con 2 capas ocultas que actúa como clasificador.

Los resultados de las pruebas para cada uno de los escenarios se muestran en la Tabla 11.

Tabla 11.- Resumen del resultado de curvatura local acumulada

	ESCENARIO 1	ESCENARIO 2	ESCENARIO 3
<b>Falsos Positivos</b>	19,66%	21,73%	21,10%
<b>Falsos Negativos</b>	12,13%	30,27%	32,84%
<b>Valor de corte</b>	0,1	0,1	0,10

### 6.1. Descripción del método

El método tiene por objetivo extraer información de “forma” de determinadas partes del rostro, la curvatura es una buena manera de obtener dicha información cuando las formas de las que hablamos son ‘curvas’. Se pretende tener una medida que nos diga el valor de la curvatura por encima y por debajo del punto.

El caso anterior de curvaturas locales no ha aportado buenos resultados. La forma de determinadas partes del rostro tiene gran importancia para la identificación de personas. Se ha decidido tomar el valor acumulado de estas curvaturas locales, así el sistema será más independiente de la determinación del punto.

Este método se apoya en las medidas realizadas por el método original de curvatura local (40 puntos de referencia por punto de interés) y calcula la media de dichos valores para caracterizar la silueta con 18 medidas.

Para determinar si 2 siluetas corresponden a la misma persona, se ha creado y entrenado un perceptrón multicapa con dos capas ocultas. La primera capa oculta tiene 7 neuronas y la segunda 2. Esta configuración del perceptrón ha sido la que ha dado mejores resultados. En la Figura 33 se puede ver el esquema del perceptrón usado.

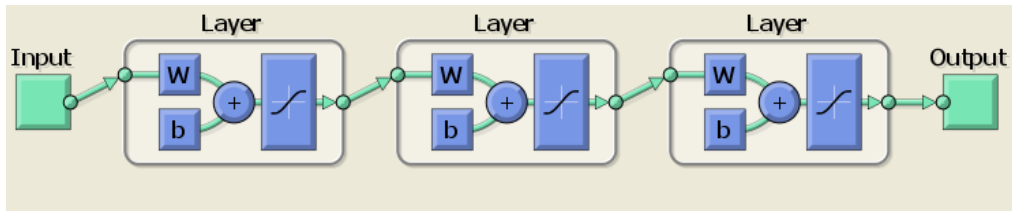


Figura 33.- Perceptrón multicapa para el método de curvatura local acumulada

## 6.2. Descripción del experimento

El experimento consiste en probar el grado en el que el método es capaz de determinar si dos siluetas pertenecen a la misma persona. Se ha probado el método para cada uno de los escenarios descritos en el capítulo 3.

### 6.2.1. Escenario 1: Perfiles sin giro horizontal

Los resultados de las pruebas para este escenario se muestran en la Tabla 12. El Gráfico 7 muestra la variación de los FP y FN con el umbral.

Tabla 12.- Resultado para método curvatura local acumulada escenario 1

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	2548	62	2548	0	100,00%	0,00%
0,05	875	1735	867	8	34,03%	12,90%
0,10	511	2099	501	10	19,66%	16,13%
0,15	368	2242	354	14	13,89%	22,58%
0,20	260	2350	245	15	9,62%	24,19%
0,25	188	2422	171	17	6,71%	27,42%
0,30	149	2461	132	17	5,18%	27,42%
0,35	122	2488	104	18	4,08%	29,03%
0,40	88	2522	70	18	2,75%	29,03%
0,45	66	2544	47	19	1,84%	30,65%
0,50	60	2550	40	20	1,57%	32,26%
0,55	54	2556	34	20	1,33%	32,26%
0,60	52	2558	29	23	1,14%	37,10%
0,65	43	2567	18	25	0,71%	40,32%
0,70	39	2571	13	26	0,51%	41,94%
0,75	33	2577	6	27	0,24%	43,55%
0,80	31	2579	4	27	0,16%	43,55%
0,85	29	2581	1	28	0,04%	45,16%
0,90	31	2579	0	31	0,00%	50,00%
0,95	33	2577	0	33	0,00%	53,23%
1,00	62	2548	0	62	0,00%	100,00%

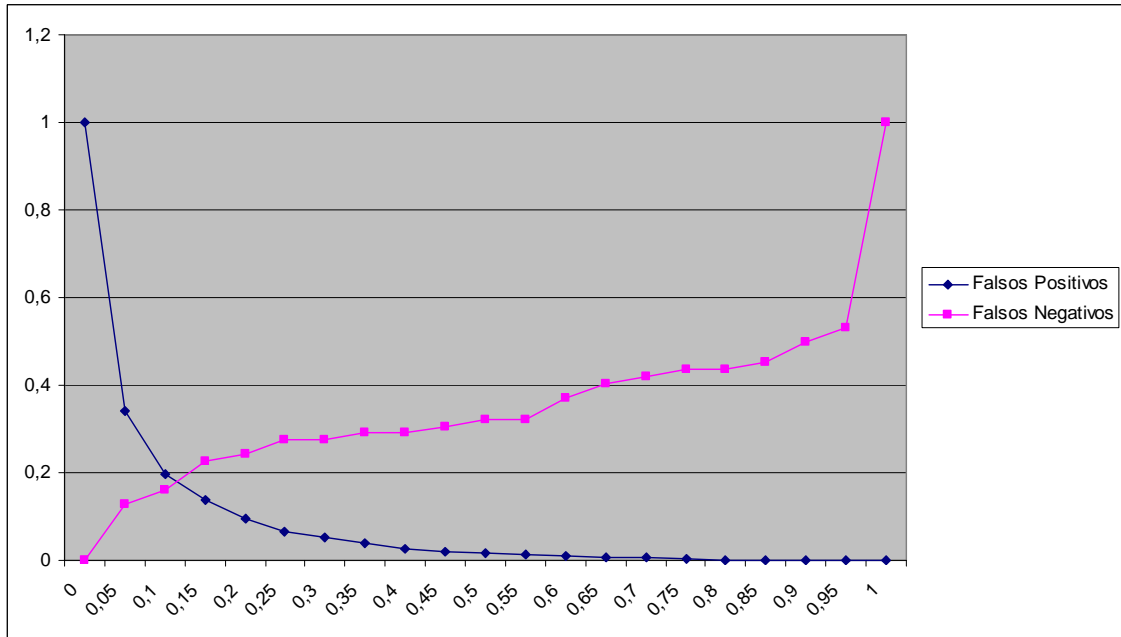


Gráfico 7.- Resultado para método curvatura local acumulada escenario 1

### 6.2.2. Escenario 2: Perfil sin giro horizontal contra todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 13. El Gráfico 8 muestra la variación de los FP y FN con el umbral.

Tabla 13.- Resultado para método curvatura local acumulada escenario 2

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	9350	185	9350	0	100,00%	0,00%
0,05	3411	6124	3372	39	36,06%	21,08%
0,10	2088	7447	2032	56	21,73%	30,27%
0,15	1486	8049	1415	71	15,13%	38,38%
0,20	1099	8436	1015	84	10,86%	45,41%
0,25	838	8697	745	93	7,97%	50,27%
0,30	684	8851	584	100	6,25%	54,05%
0,35	561	8974	457	104	4,89%	56,22%
0,40	449	9086	341	108	3,65%	58,38%
0,45	369	9166	258	111	2,76%	60,00%
0,50	312	9223	197	115	2,11%	62,16%
0,55	271	9264	148	123	1,58%	66,49%
0,60	248	9287	117	131	1,25%	70,81%
0,65	216	9319	82	134	0,88%	72,43%
0,70	197	9338	59	138	0,63%	74,59%
0,75	185	9350	40	145	0,43%	78,38%
0,80	175	9360	26	149	0,28%	80,54%
0,85	162	9373	12	150	0,13%	81,08%
0,90	157	9378	4	153	0,04%	82,70%
0,95	156	9379	0	156	0,00%	84,32%
1,00	185	9350	0	185	0,00%	100,00%

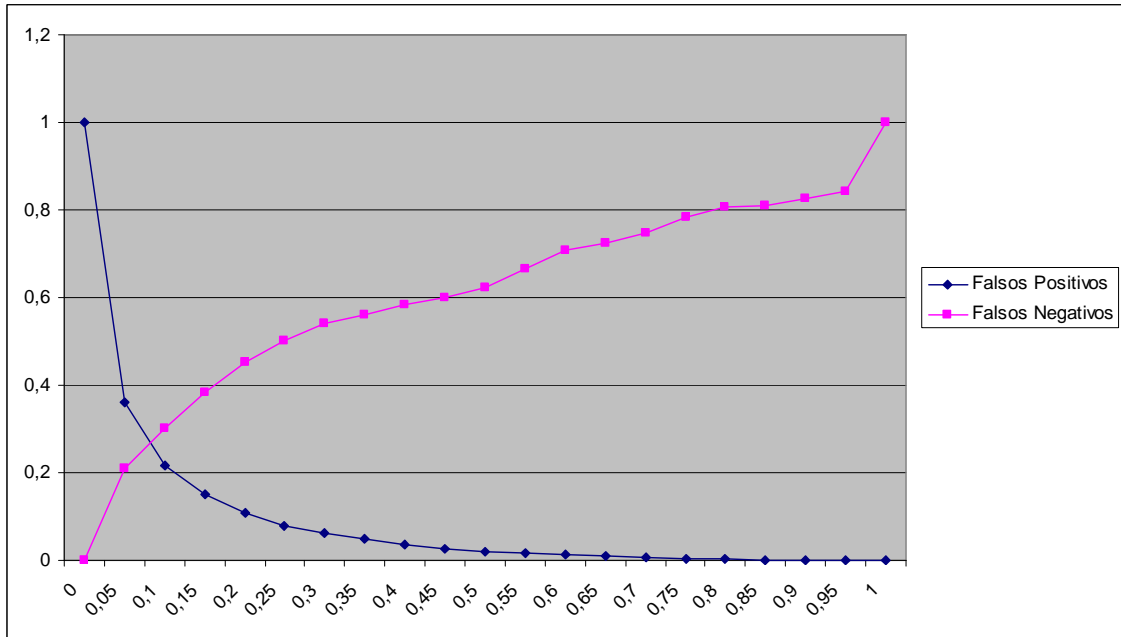


Gráfico 8.- Resultado para método curvatura local acumulada escenario 2

### 6.2.3. Escenario 3: Todos los perfiles con todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 14. El Gráfico 9 muestra la variación de los FP y FN con el umbral.

Tabla 14.- Resultado para método curvatura local acumulada escenario 3

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	34608	816	34608	0	100,00%	0,00%
0,05	12089	23335	11908	181	34,41%	22,18%
0,10	7569	27855	7301	268	21,10%	32,84%
0,15	5410	30014	5079	331	14,68%	40,56%
0,20	4068	31356	3684	384	10,64%	47,06%
0,25	3187	32237	2762	425	7,98%	52,08%
0,30	2610	32814	2135	475	6,17%	58,21%
0,35	2209	33215	1702	507	4,92%	62,13%
0,40	1877	33547	1337	540	3,86%	66,18%
0,45	1604	33820	1035	569	2,99%	69,73%
0,50	1393	34031	801	592	2,31%	72,55%
0,55	1226	34198	608	618	1,76%	75,74%
0,60	1109	34315	462	647	1,33%	79,29%
0,65	1039	34385	358	681	1,03%	83,46%
0,70	957	34467	249	708	0,72%	86,76%
0,75	887	34537	157	730	0,45%	89,46%
0,80	842	34582	91	751	0,26%	92,03%
0,85	829	34595	58	771	0,17%	94,49%
0,90	817	34607	31	786	0,09%	96,32%
0,95	811	34613	4	807	0,01%	98,90%
1,00	816	34608	0	816	0,00%	100,00%



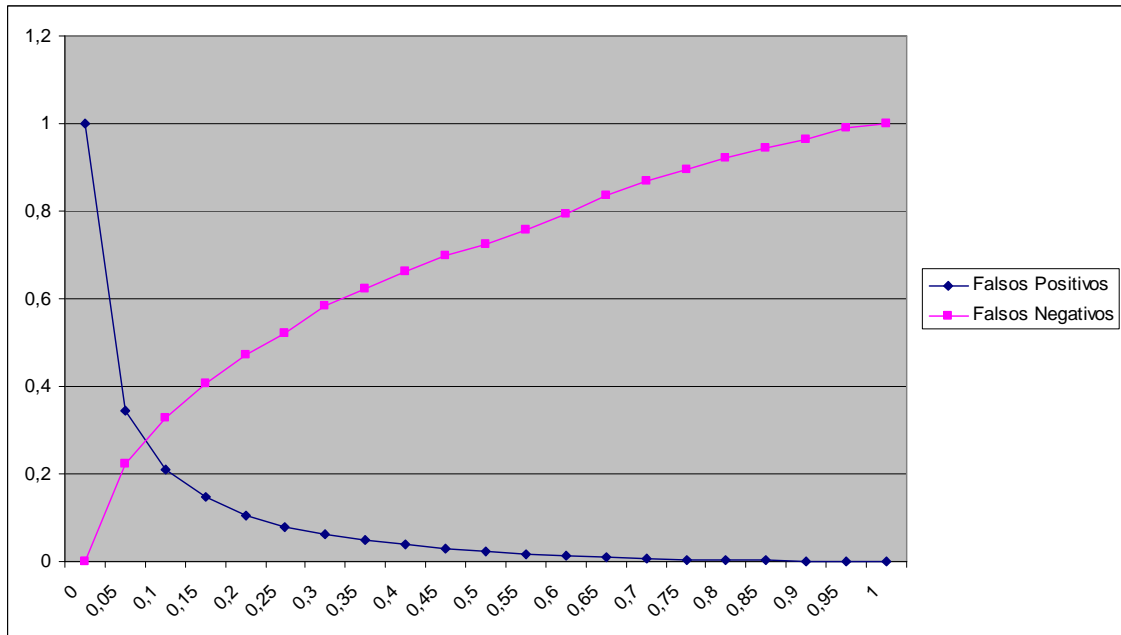


Gráfico 9.- Resultado para método curvatura local acumulada escenario 3

### 6.3. Análisis de resultados

Los resultados de las pruebas, determinan que el método mejora bastante respecto al método de curvatura local, pero no aporta la precisión suficiente. Para el caso de perfiles perfectos los resultados son medianamente aceptables, tenemos un valor de falsos negativos relativamente bajo. Si nos fijamos, no hay mucha diferencia en los resultados de las pruebas para los escenarios 2 y 3. Las curvas de los errores son similares y hay que tener en cuenta que en dichas pruebas hay gran cantidad de perfiles de la misma persona con diferencias considerables. El método se mantiene estable, lo que significa que hay una serie de perfiles con las que falla.

Analizando los casos en los que falla vemos que son aquéllos en los que tienen parte de oclusión, la expresión de la boca ha cambiado, el pelo cambia, o el individuo tiene barba, etc. Lo ideal sería que el sistema fuera capaz de superar estas diferencias pero realmente es muy difícil. El fundamento principal para el reconocimiento es la curvatura local de los puntos. Cuando se producen estos incidentes, la curvatura cambia en varios puntos con valores elevados, haciendo al sistema tomar valores de semejanza muy bajos. Vamos a describir los puntos que afecta a cada uno de los incidentes detectados:

- El peinado (flequillo) tapa los puntos 1 y 2.
- La expresión de la boca produce el cambio en los puntos 5,6 y 7 (y parte del 4).
- La barba puede incorporar variaciones en todos los puntos en los que interviene (4, 5, 7, 8 y 9).

Por estos motivos resulta muy difícil creer que un método basado únicamente en la curvatura local pueda aportar buenos resultados en estas condiciones.

Parece que fijarnos en la forma de determinadas partes del rostro no es suficiente para reconocer a una persona. Quizás sea porque un humano reconoce de forma positiva (si tiene un determinado rasgo es esta persona, suponiendo que el resto es parecido) y el

sistema lo hace de forma negativa (todos los rasgos son parecidos). Por este motivo podría ser relevante usar dicho método como complemento a otro.

En lo que se refiere a la curvatura local, hemos pasado de un extremo (comparar en cada punto), al otro (comparar todo el acumulado). Es decir de tener toda la información local y dependiente de los puntos seleccionados, a tener la información del conjunto. Parece lógico pensar que un término medio aportaría mejores resultados, de manera que se tuviera información (media) de variación de la curvatura.

Una buena aproximación podría ser la siguiente, partiendo de las 40 medidas de curvatura (20 superiores y 20 inferiores), hacer el acumulado de la media cada 5 medidas. Con cada 5 medidas tomadas, calculamos la media y proporcionamos una nueva medida. Así tendremos 8 medidas de cada punto y podremos representar más fielmente la variación de la curvatura sin caer en ninguno de los errores anteriores.

## 7. Método de Histograma Local

El histograma local mide los valores normalizados de la coordenada 'x' en las proximidades de cada punto de interés. Se fija el eje de la coordenada 'x' respecto a un punto y se toman las medidas del punto de referencia al valor de la coordenada 'x' de cada punto. Así se tienen registrados los valores de distancia respecto al punto de referencia en determinados puntos. Lo que se representa es la variación de la coordenada 'x' y la proporción entre los distintos puntos. La proporción se consigue gracias a la normalización. Al comparar entre siluetas, se compara la forma de determinadas partes del rostro.

El método toma 41 características por cada punto de interés, en total 369. Los 41 puntos corresponden a las medidas de comparación del punto de interés y otros 40 puntos próximos a él (20 en cada sentido). Con estas características se crea y entrena un perceptrón multicapa de una capa oculta que actúa como clasificador.

Los resultados de las pruebas para cada uno de los escenarios se muestran en la Tabla 15.

**Tabla 15.- Resumen del resultado de histograma local**

	<b>ESCENARIO 1</b>	<b>ESCENARIO 2</b>	<b>ESCENARIO 3</b>
<b>Falsos Positivos</b>	28,38%	29,83%	29,46%
<b>Falsos Negativos</b>	25,81%	43,78%	42,16%
<b>Valor de corte</b>	0,05	0,05	0,05

### 7.1. Descripción del método

El histograma local pretende analizar la forma de la silueta centrándose en determinadas partes del rostro. Busca la facilidad para comparar una silueta con otra valorando únicamente las proximidades de los puntos de interés. En el Gráfico 10 se puede ver un ejemplo de la representación del histograma local de 2 imágenes. En dicho gráfico se ve como varía la coordenada 'x' en las proximidades de los puntos de interés, el resultado es una figura compactada en la que se suprimen las partes alejadas de los puntos de interés. El eje de coordenadas es el índice del punto representado, el valor del eje de abscisas es el valor de la coordenada 'x' del borde de la imagen en el punto determinado.

Para normalizar los valores se establece el origen de coordenadas relativo al punto 3 (añadiendo aproximadamente 4/3 de la distancia entre los puntos 3 y 8). Así todos los valores de la coordenada 'x' se miden desde el valor 'x' del punto 3 más 800.

La silueta quedará representada por 369 características (40 +1 x PI) que determinan el valor de la coordenada 'x' en determinados puntos. Con estas características se crea y entrena un clasificador implementado mediante un perceptrón multicapa de una sola capa oculta con 6 neuronas. Por el número elevado de características se ha tenido que

reducir el a una sola capa oculta. Incrementando el número de capas ocultas el tiempo de proceso se dispara y los resultados no mejoran. En la Figura 34 se puede ver el esquema de la red de neuronas creada.

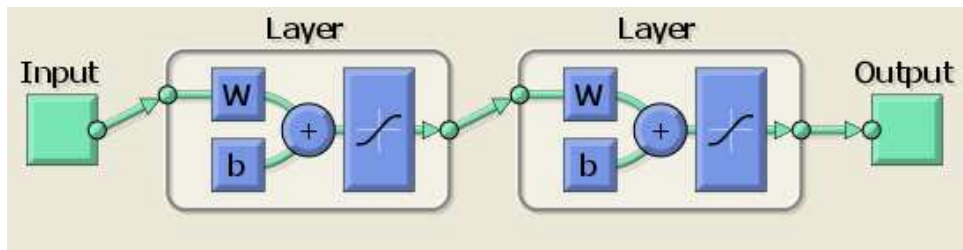


Figura 34.- Esquema de la red de neuronas para el histograma local

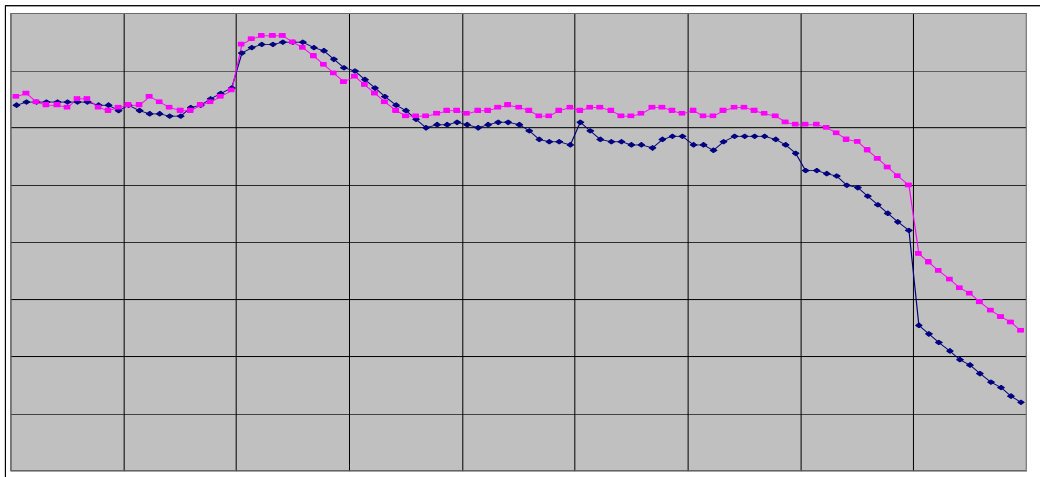


Gráfico 10 Ejemplo de 2 histogramas locales

## 7.2. Descripción del experimento

El experimento consiste en probar el grado en el que el método es capaz de determinar si dos siluetas pertenecen a la misma persona. Se ha probado el método para cada uno de los escenarios descritos en el capítulo 3.

### 7.2.1. Escenario 1: Perfiles sin giro horizontal

Los resultados de las pruebas para este escenario se muestran en la Tabla 16. El Gráfico 11 muestra la variación de los FP y FN con el umbral.

Tabla 16.- Resultado para método histograma local escenario 1

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	2548	62	2548	0	100,00%	0,00%
0,05	739	1871	723	16	28,38%	25,81%
0,10	574	2036	557	17	21,86%	27,42%
0,15	517	2093	499	18	19,58%	29,03%
0,20	474	2136	455	19	17,86%	30,65%
0,25	445	2165	425	20	16,68%	32,26%
0,30	419	2191	399	20	15,66%	32,26%
0,35	397	2213	377	20	14,80%	32,26%
0,40	379	2231	358	21	14,05%	33,87%
0,45	354	2256	332	22	13,03%	35,48%
0,50	346	2264	322	24	12,64%	38,71%
0,55	325	2285	300	25	11,77%	40,32%
0,60	317	2293	292	25	11,46%	40,32%
0,65	307	2303	281	26	11,03%	41,94%
0,70	293	2317	267	26	10,48%	41,94%
0,75	270	2340	244	26	9,58%	41,94%
0,80	255	2355	229	26	8,99%	41,94%
0,85	230	2380	202	28	7,93%	45,16%
0,90	199	2411	169	30	6,63%	48,39%
0,95	152	2458	92	60	3,61%	96,77%
1,00	62	2548	0	62	0,00%	100,00%

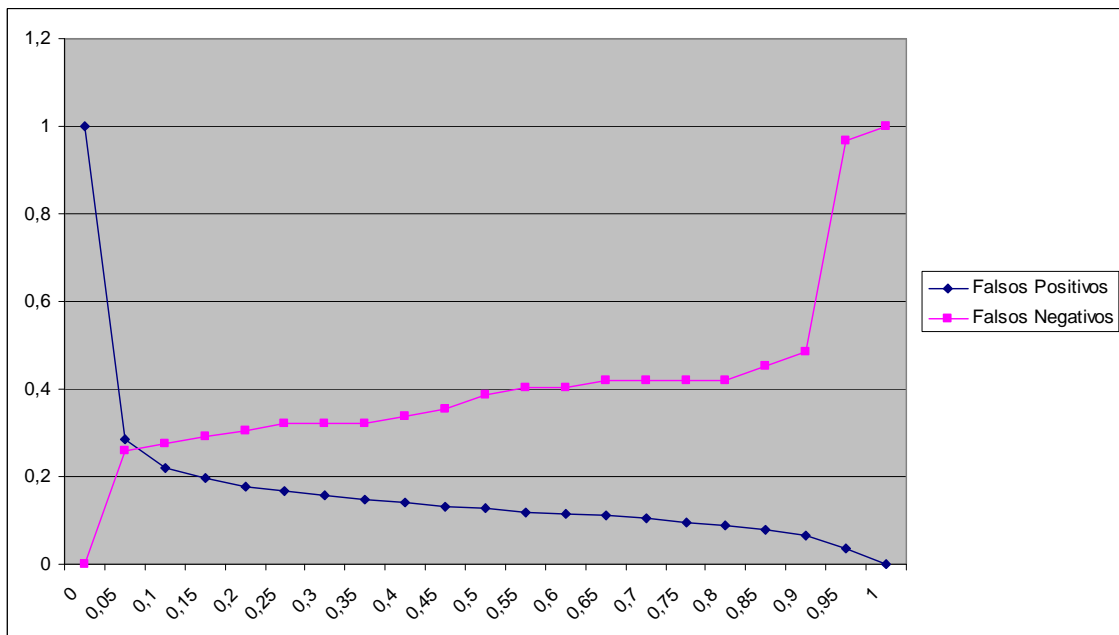


Gráfico 11.- Resultado para método histograma local escenario 1

## 7.2.2. Escenario 2: Perfil sin giro horizontal contra todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 17. El Gráfico 12 muestra la variación de los FP y FN con el umbral.

Tabla 17.- Resultado para método histograma local escenario 2

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	9350	185	9350	0	100,00%	0,00%
0,05	2870	6665	2789	81	29,83%	43,78%
0,10	2320	7215	2227	93	23,82%	50,27%
0,15	2104	7431	2010	94	21,50%	50,81%
0,20	1969	7566	1872	97	20,02%	52,43%
0,25	1854	7681	1753	101	18,75%	54,59%
0,30	1772	7763	1668	104	17,84%	56,22%
0,35	1691	7844	1586	105	16,96%	56,76%
0,40	1629	7906	1522	107	16,28%	57,84%
0,45	1573	7962	1465	108	15,67%	58,38%
0,50	1527	8008	1415	112	15,13%	60,54%
0,55	1452	8083	1339	113	14,32%	61,08%
0,60	1399	8136	1286	113	13,75%	61,08%
0,65	1341	8194	1226	115	13,11%	62,16%
0,70	1285	8250	1167	118	12,48%	63,78%
0,75	1218	8317	1099	119	11,75%	64,32%
0,80	1150	8385	1027	123	10,98%	66,49%
0,85	1065	8470	940	125	10,05%	67,57%
0,90	927	8608	797	130	8,52%	70,27%
0,95	665	8870	500	165	5,35%	89,19%
1,00	185	9350	0	185	0,00%	100,00%

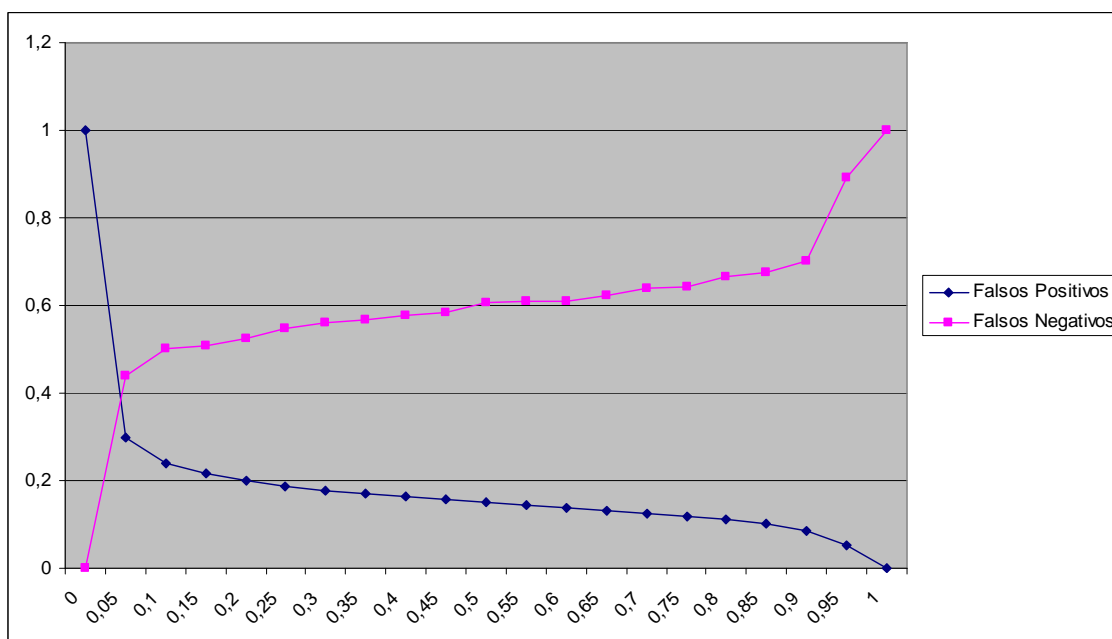


Gráfico 12.- Resultado para método histograma local escenario 2

### 7.2.3. Escenario 3: Todos los perfiles con todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 18. El Gráfico 13 muestra la variación de los FP y FN con el umbral.

Tabla 18.- Resultado para método histograma local escenario 3

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	34608	816	34608	0	100,00%	0,00%
0,05	10539	24885	10195	344	29,46%	42,16%
0,10	8824	26600	8423	401	24,34%	49,14%
0,15	8055	27369	7624	431	22,03%	52,82%
0,20	7610	27814	7154	456	20,67%	55,88%
0,25	7236	28188	6762	474	19,54%	58,09%
0,30	6944	28480	6458	486	18,66%	59,56%
0,35	6648	28776	6151	497	17,77%	60,91%
0,40	6443	28981	5937	506	17,15%	62,01%
0,45	6245	29179	5726	519	16,55%	63,60%
0,50	6065	29359	5531	534	15,98%	65,44%
0,55	5830	29594	5292	538	15,29%	65,93%
0,60	5641	29783	5094	547	14,72%	67,03%
0,65	5405	30019	4848	557	14,01%	68,26%
0,70	5159	30265	4581	578	13,24%	70,83%
0,75	4919	30505	4331	588	12,51%	72,06%
0,80	4651	30773	4043	608	11,68%	74,51%
0,85	4365	31059	3740	625	10,81%	76,59%
0,90	3869	31555	3220	649	9,30%	79,53%
0,95	2804	32620	2105	699	6,08%	85,66%
1,00	816	34608	0	816	0,00%	100,00%

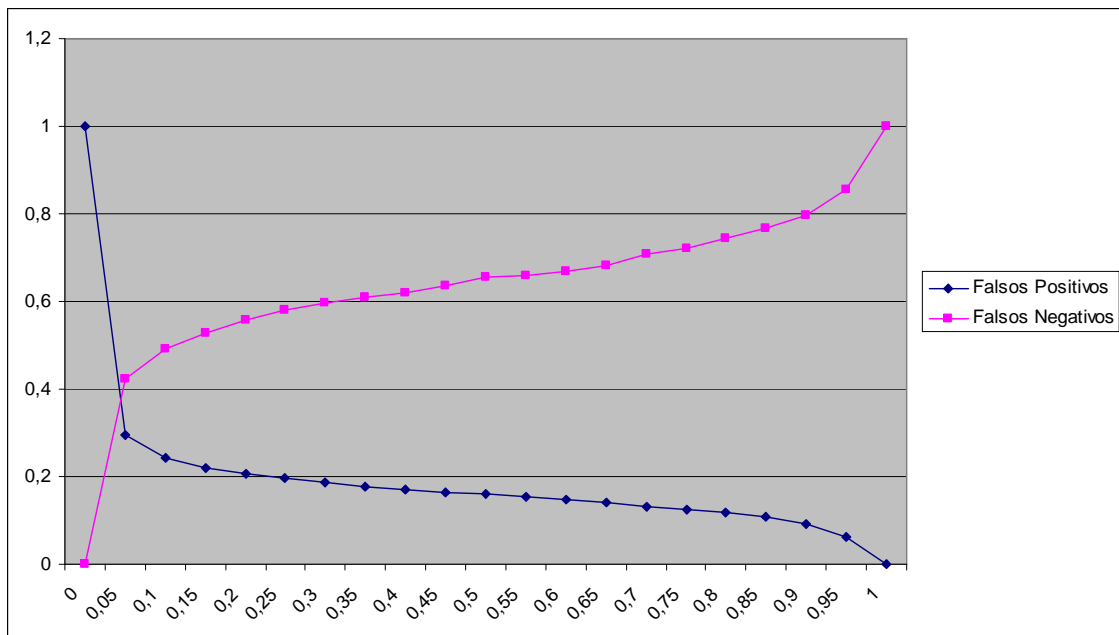


Gráfico 13.- Resultado para método histograma local escenario 3

### **7.3. *Análisis de resultados***

Los resultados obtenidos son parecidos a los del método de curvatura local. Esto debido a que se basa en el mismo principio, la forma de partes del rostro. Se implementa de forma parecida mediante la variación respecto a unos puntos concretos.

El método tiene los mismos inconvenientes que los basados en la forma de los puntos de interés, dependencia respecto a los puntos seleccionados y la dificultad para determinar la semejanza cuando se oculta alguna parte, se cambia la expresión del rostro, aparece barba, etc.



## 8. Método de Snakes

Se basa en la comparación de la silueta con un 'snake' [13]. El resultado de la comparación permite determinar el grado de semejanza entre la silueta y el *snake*. El método permite al *snake* deformarse y adaptarse en cierto grado a la imagen, aportando así una tolerancia a posibles cambios de gesto u ocultación. El *snake* se genera desde la silueta de otra imagen, de esta forma se compara una silueta con la definición reducida de otra.

Del resultado de la comparación se obtienen dos valores de semejanza. Con estos valores se crea y entrena un perceptrón multicapa que actúa como clasificador.

Los resultados de las pruebas para cada uno de los escenarios se muestran en la Tabla 15.

Tabla 19.- Resumen del resultado de histograma local

	ESCENARIO 1	ESCENARIO 2	ESCENARIO 3
<b>Falsos Positivos</b>	12,01%	21,68%	15,33%
<b>Falsos Negativos</b>	14,52%	25,95%	35,42%
<b>Valor de corte</b>	0,20	0,10	0,15

### 8.1. Descripción del método

El método se basa en el reconocimiento global que pueden hacer los humanos, consiste en comparar la imagen de silueta con la definición o recuerdo que se tiene almacenada.

Un *snake* [13] es un tipo de contorno activo [14], que consiste básicamente en un contorno que se mueve sobre una imagen a la que trata de adaptarse. El contorno tiene una cierta flexibilidad que le permite variar la propia definición del *snake* para asemejarse a dicha imagen. Para mover el *snake* sobre la imagen se usa un algoritmo iterativo que minimiza la energía consumida. La energía consumida depende de la variación necesaria del *snake* tanto en posición como en forma. Las iteraciones del bucle pueden terminar al encontrar una solución o por un límite en dichas iteraciones. Una vez terminadas las iteraciones el algoritmo devuelve los valores de semejanza entre *snake* e imagen.

El método consiste en superponer el *snake* sobre una imagen y permitir a dicho *snake* adaptarse ligeramente a la imagen. Una vez el algoritmo iterativo ha terminado se obtienen los valores de semejanza. La posibilidad de adaptarse hace que se puedan superar algunas diferencias entre la imagen y el *snake*. El grado de adaptación depende de unos parámetros de configuración, estos parámetros pueden hacer el *snake* muy rígido o por el contrario permitir que se adapte a cualquier forma. Por este motivo es necesario definir con claridad la configuración que se quiere usar porque de esta configuración van a depender las medidas tomadas.

Los valores de semejanza devueltos son dos valores coincidencia y distancia. La definición de estos valores se hará en el apartado de configuración de la aplicación y resultados. Con los valores de coincidencia y distancia se crea y entrena un perceptrón multicapa con dos capas ocultas. El número de neuronas de la primera capa es 4 y el de la segunda es 2. La Figura 35 muestra el esquema de la red de neuronas creada.

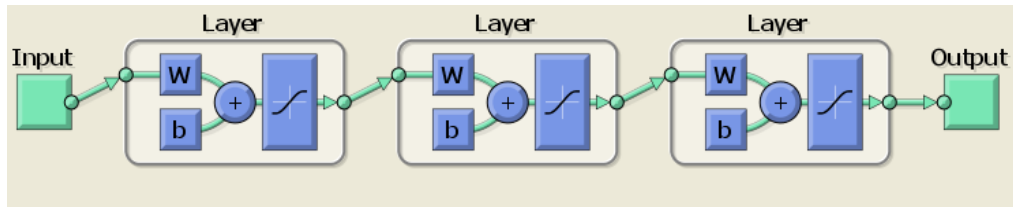


Figura 35.- Esquema de la red de neuronas para snakes

La aplicación usada es *PSnakes.exe* que es una implementación de *snakes*. La aplicación permite seleccionar una imagen y un *snake*, configurar los parámetros y calcular la semejanza entre ambos. En la Figura 36 se puede ver un ejemplo de la aplicación donde se distinguen las 3 partes principales:

- **Control:** Contiene la parte de configuración de la aplicación y el botón Comenzar.
- **Histórico:** Muestra los resultados de las comparaciones.
- **Marco de la imagen:** Muestra la imagen y el *snake*. Si se selecciona en el control se puede ver en cada iteración como se va adaptando el *snake*.

La aplicación *PSnake*, recibe como entrada una imagen y un *snake*. Se tiene una base de datos de imágenes. El formato de las imágenes no es el adecuado para la aplicación y es necesario crear las imágenes y los *snakes* acordes a la aplicación. Una vez creadas las imágenes y los *snakes* se configura la aplicación para obtener el valor de semejanza. A continuación se explicarán los pasos para crear las imágenes y los *snakes*. La aplicación permite la ejecución en modo *batch* para la comparación de imágenes, se explicará más en detalle esta parte.

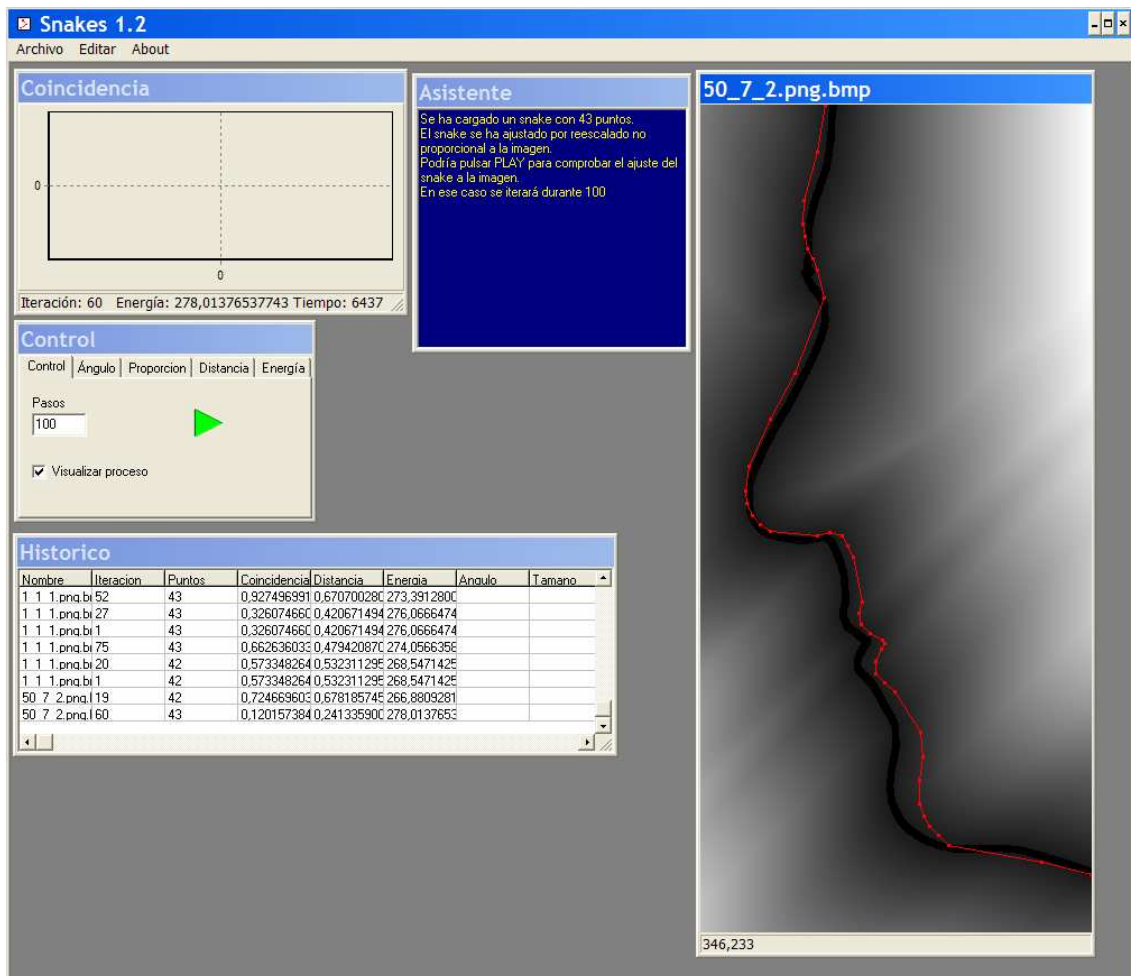


Figura 36.- Ejemplo de aplicación PSnake

### 8.1.1. Construcción de imágenes y snakes

La aplicación *PSnake* tiene como entrada una imagen en la que únicamente interesa el contorno (o píxeles que están en el borde de la imagen). La BD tiene las imágenes completas, por lo que es necesario pasar dichas imágenes al contorno. En la Figura 37 se puede ver la imagen almacenada en la BD, el contorno de dicha imagen y el contorno para la entrada de la aplicación.

Al seleccionar el contorno, se recorta a la parte más representativa de la silueta. Para recortar la silueta se toman 300 píxeles por encima del punto P1 y 200 píxeles por debajo del punto P8. Como el contorno tiene únicamente un píxel, y la aplicación intenta introducir el *snake* dentro de la imagen, se ha ensanchado el contorno de manera que sea posible dicha acción.

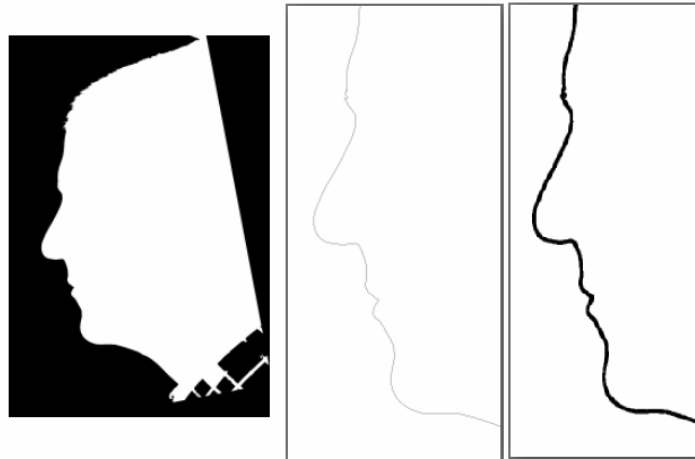


Figura 37.- Imagen de BD, contorno y contorno para psnake

La definición de un *snake* para la aplicación es un fichero xml con la lista de puntos que lo forman. La selección de los puntos que forman el *snake* se hace en función del grado de detalle que se quiere aportar a dicho *snake*. Se han definido puntos a lo largo de todo el contorno (la parte más representativa, usada también en la imagen), aumentando el número de puntos en las zonas en las que la curvatura es mayor. Este aumento de puntos nos permite tener mayor definición en dichas zonas. En la Figura 38 se pueden ver los 43 puntos que forman la definición del *snake*.

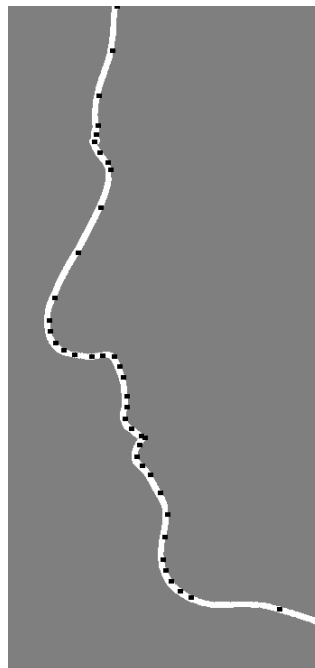


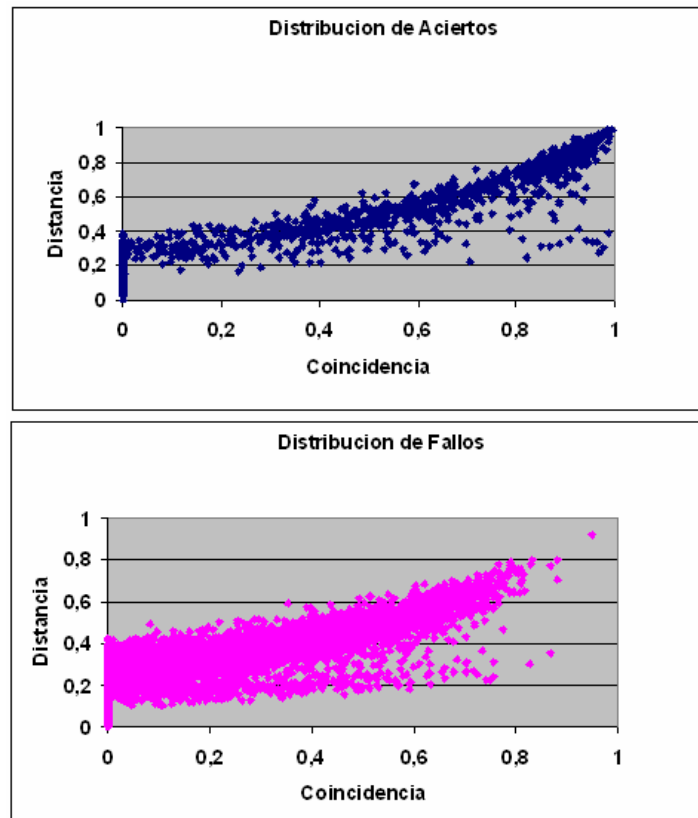
Figura 38.- Puntos en la definición del snake

La definición de estas imágenes y de estos *snakes* se han usado por la aplicación para establecer el grado de semejanza entre imagen y *snake*.

El tamaño de las imágenes en la BD es elevado y este método compara toda la imagen (el contorno) contra un *snake*, el tiempo necesario para comparar una imagen y *snake* depende del tamaño. Por este motivo se han realizado pruebas reduciendo el tamaño y comprobando si los datos obtenidos eran correctos. Se ha probado reduciendo el tamaño a la mitad y a una cuarta parte. Los datos se han tomado con la misma configuración

entre ambas pruebas. La configuración corresponde con la primera imagen de la Figura 43, donde se define una configuración de ángulo para un *snake* rígido.

La Figura 39 muestra los resultados para la reducción del 50%, en dicha figura se puede ver la distribución de los aciertos y los fallos en función de la coincidencia y de la distancia. Coincidencia y distancia son los valores de semejanza que devuelve la aplicación.



**Figura 39.- Distribución de Aciertos y Fallos para tamaño  $\frac{1}{2}$**

En la Figura 40 se puede ver la distribución de aciertos y fallos para una reducción del tamaño de un 25%. Como se puede apreciar en las gráficas la reducción de tamaño no significa una reducción de precisión sino que localiza los aciertos y fallos en conjuntos más separados.

En la Figura 41 se puede ver la localización de aciertos y fallos conjuntamente para cada uno de los tamaños. Como para las imágenes reducidas a un 25% la distribución parece más localizada, y el tiempo de proceso se ha reducido de forma lineal. Se ha tomado este tamaño para continuar con el procedimiento.

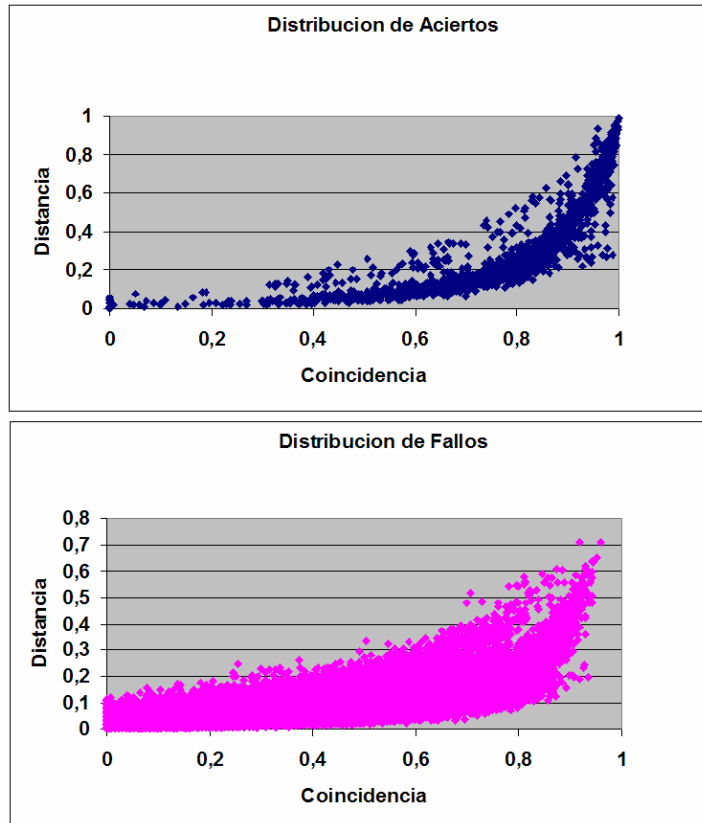


Figura 40.- Distribución de Aciertos y Fallos para tamaño  $\frac{1}{4}$

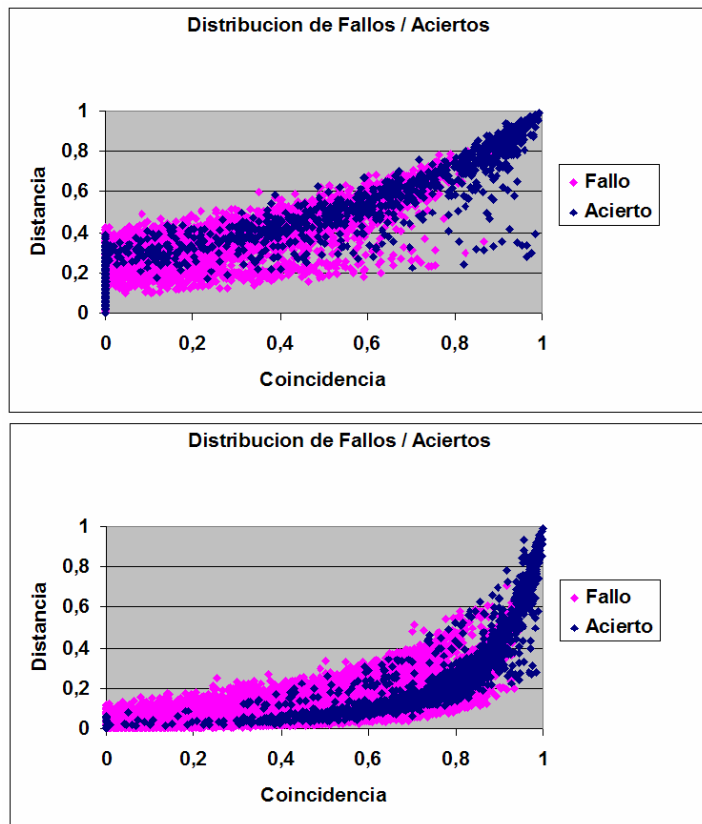


Figura 41.- Distribución de fallos y aciertos, tamaño 50% arriba, tamaño 25% abajo.

## 8.1.2. Configuración de la aplicación y resultados

La aplicación tiene una parte de configuración que modifica el comportamiento del *snake* a la hora de ajustarse a la imagen. Dicha configuración está contenida dentro de la aplicación por la ventana llamada control que se muestra en la Figura 42. En ella se pueden ver las siguientes pestañas:

- **Control:** Nos permite indicar el nº de iteraciones máximas para adaptarse a la imagen.
- **Ángulo:** Nos permite definir mediante lógica borrosa las variaciones de ángulo permitidas entre puntos.
- **Proporción:** Nos permite definir mediante lógica borrosa las variaciones de proporción permitidas entre segmentos.
- **Distancia:** Nos permite definir mediante lógica borrosa lo que considera la aplicación como una distancia elevada o pequeña
- **Energía:** Nos permite definir mediante lógica borrosa lo que considera la aplicación como energía grande o pequeña.



Figura 42.- Detalle de la ventana Control

Para el caso de los perfiles, después de hacer pruebas con la aplicación, se ha observado que la parte que más interesa es la configuración del ángulo. El ángulo está condicionando el ajuste los *snakes* usados. Se necesita un sistema medianamente rígido. Al tener todos los perfiles una estructura similar, si se pone un ángulo grande, el *snake* se deformará hasta ajustarse sin conservar su forma. Hay que tener en cuenta que la aplicación es de uso general y no únicamente para siluetas. Una vez decidido que se quiere un sistema rígido, hay que determinar con que valores proporciona mejores resultados, se ha probado con las configuraciones mostradas en la Figura 43.



Figura 43.- Configuraciones de ángulo

- Las pruebas aportan mejores resultados para la configuración menos rígida, ya que permite al *snake* adaptarse en mayor grado. La configuración quedaría definida como en la segunda imagen de la Figura 43 donde los números indican los ángulos en grados.

En la Figura 44 se puede ver la gráfica que representa el número de Falsos Positivos y Falsos Negativos en función del umbral seleccionado. La gráfica corresponde a la configuración de ángulo más rígida (definida por la configuración de ángulo “1 5”). En la Figura 45 se muestra la configuración de ángulo que hacen al *snake* menos rígida (“3 8”). Como se puede apreciar en dichas gráficas, la segunda configuración proporciona mejores resultados. Esta mejora en los resultados es ocasionada por la mayor concentración de aciertos con valor elevado de coincidencia y distancia. En la Figura 46 se puede comprobar que la configuración de ángulo menos rígida (“3 8”) tiene mayor número de aciertos en las proximidades del 1.

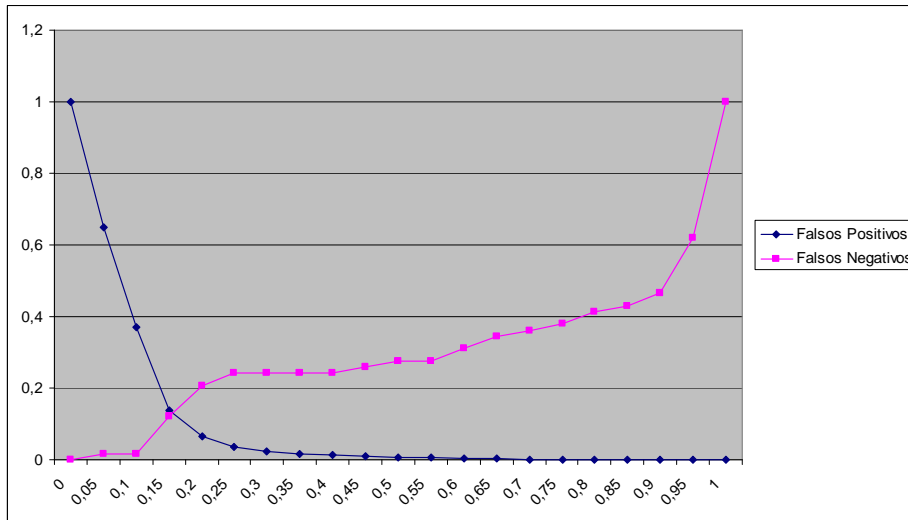


Figura 44.- FP y FN para configuración de ángulo 1 5

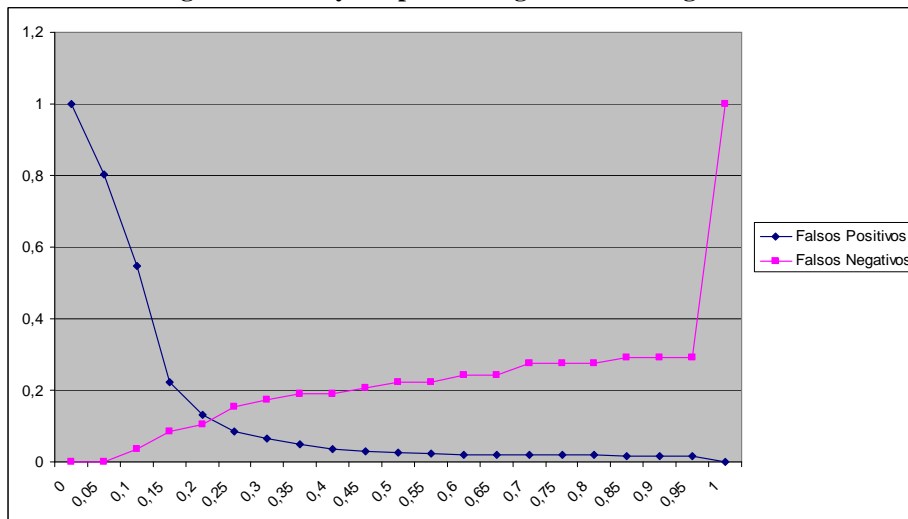


Figura 45.- FP y FN para configuración de ángulo 3 8



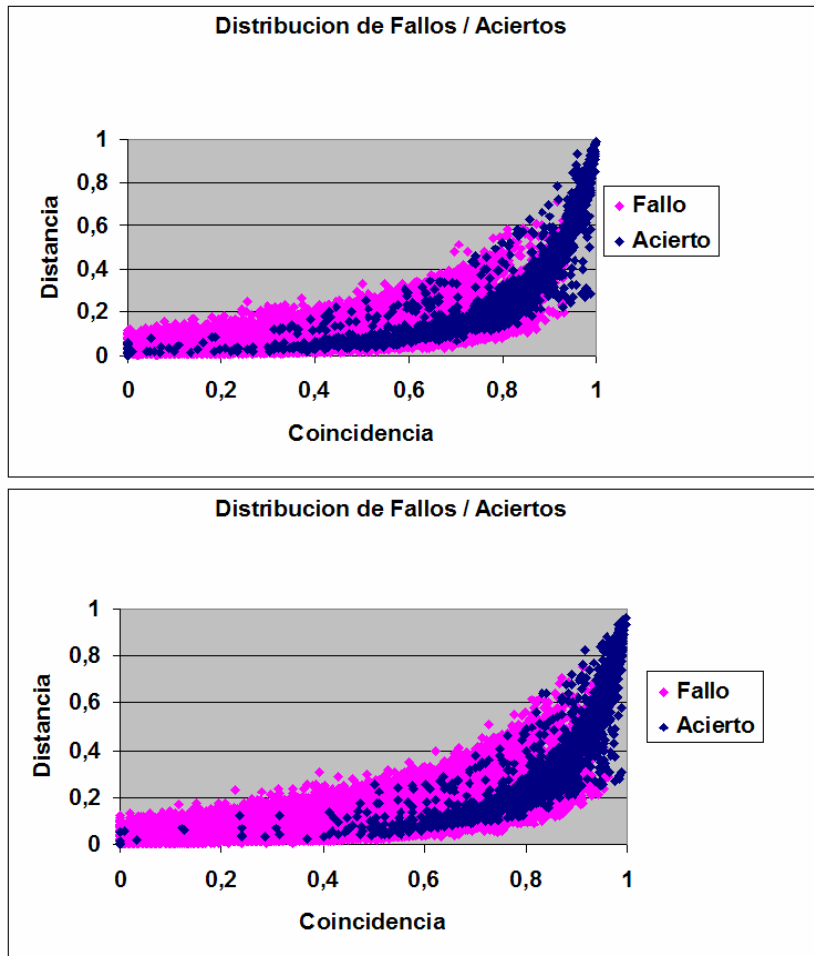


Figura 46.- Distribución de Fallos y Aciertos para configuración de ángulo 1 5 (arriba) y 3 8 (abajo)

Para las pruebas se ha limitado el nº de iteraciones a 50 porque en la mayoría de los casos en este límite la aplicación ya ha convergido a una solución válida.

Otra posibilidad de configuración de la aplicación es en el menú Editar, la opción de ajustar puntos por reescalado, que ajusta el tamaño del *snake* al tamaño de la imagen. En la Figura 47 se puede ver como queda configurada la aplicación.

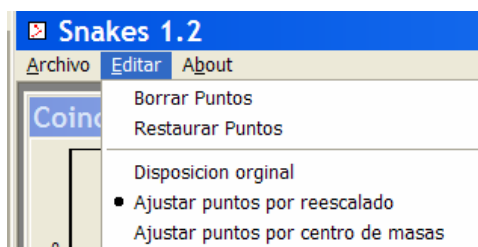


Figura 47.- Configuración de ajuste de *snake* por reescalado

Una vez configurada la aplicación, cuando se ejecuta, en la ventana de histórico nos muestra el resultado de la ejecución (y de las anteriores). En la Figura 48 se ve un ejemplo de ejecución. La información mostrada es, nombre de la imagen, iteración en la que para, número de puntos del *snake*, coincidencia, distancia y energía. Lo que determina el grado de semejanza entre la imagen y *snake*, son las medidas de coincidencia y la de distancia.

La coincidencia mide el grado de ajuste de los puntos del *snake* a la imagen y la distancia mide la distancia entre los trazos del *snake* (unión de los puntos) y la imagen.

Hay que indicar que la medida de distancia se ha modificado de la aplicación original para que asignara el valor en función de la distancia. La aplicación original computaba el nº de puntos que estaban más lejos de un umbral, y la modificación además asigna una medida en función de la distancia.

Nombre	Iteracion	Puntos	Coincidencia	Distancia	Energia	Angulo	Tamano
1 1 1.pnq.bi 52		43	0,927496991	0,670700280	273,3912800		
1 1 1.pnq.bi 27		43	0,326074660	0,420671494	276,0666474		
1 1 1.pnq.bi 1		43	0,326074660	0,420671494	276,0666474		
1 1 1.pnq.bi 75		43	0,662636033	0,479420870	274,0566356		
1 1 1.pnq.bi 20		42	0,573348264	0,532311295	268,5471425		
1 1 1.pnq.bi 1		42	0,573348264	0,532311295	268,5471425		
50 7 2.pnq.l 19		42	0,724669603	0,678185745	266,8809281		
50 7 2.pnq.l 60		43	0,120157384	0,241335900	278,0137653		

Figura 48.- Ventana de histórico

### 8.1.3. Ejecución en batch de la aplicación

La aplicación original permitía una ejecución en batch, y se ha modificado para que se adapte a la nomenclatura y ubicación de las imágenes y los *snakes*. La ejecución en batch permite la ejecución de la aplicación cruzando un directorio de imágenes con un directorio de *snakes*. El resultado se escribe en un fichero de texto en el que aparece por cada comparación una línea con el nombre del *snake*, el nombre de la imagen y los valores de coincidencia y distancia. La opción de menú está en Archivo, Falsos Positivos.

Es esta opción de modo batch la que se ha usado para la ejecución de las pruebas.

## 8.2. Descripción del experimento

El experimento consiste en probar el grado en el que el método es capaz de determinar si dos siluetas pertenecen a la misma persona. Se ha probado el método para cada uno de los escenarios descritos en el capítulo 3.

### 8.2.1. Escenario 1: Perfiles sin giro horizontal

Los resultados de las pruebas para este escenario se muestran en la Tabla 20. El Gráfico 14 muestra la variación de los FP y FN con el umbral.

Tabla 20.- Resultado para método *snakes* escenario 1

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	2548	62	2548	0	100,00%	0,00%
0,05	870	1740	865	5	33,95%	8,06%
0,10	700	1910	693	7	27,20%	11,29%
0,15	563	2047	555	8	21,78%	12,90%
0,20	315	2295	306	9	12,01%	14,52%
0,25	138	2472	124	14	4,87%	22,58%
0,30	84	2526	68	16	2,67%	25,81%
0,35	53	2557	37	16	1,45%	25,81%
0,40	40	2570	22	18	0,86%	29,03%
0,45	35	2575	16	19	0,63%	30,65%
0,50	30	2580	11	19	0,43%	30,65%
0,55	29	2581	10	19	0,39%	30,65%
0,60	29	2581	8	21	0,31%	33,87%
0,65	27	2583	6	21	0,24%	33,87%
0,70	25	2585	4	21	0,16%	33,87%
0,75	25	2585	3	22	0,12%	35,48%
0,80	25	2585	3	22	0,12%	35,48%
0,85	24	2586	2	22	0,08%	35,48%
0,90	25	2585	2	23	0,08%	37,10%
0,95	28	2582	1	27	0,04%	43,55%
1,00	62	2548	0	62	0,00%	100,00%

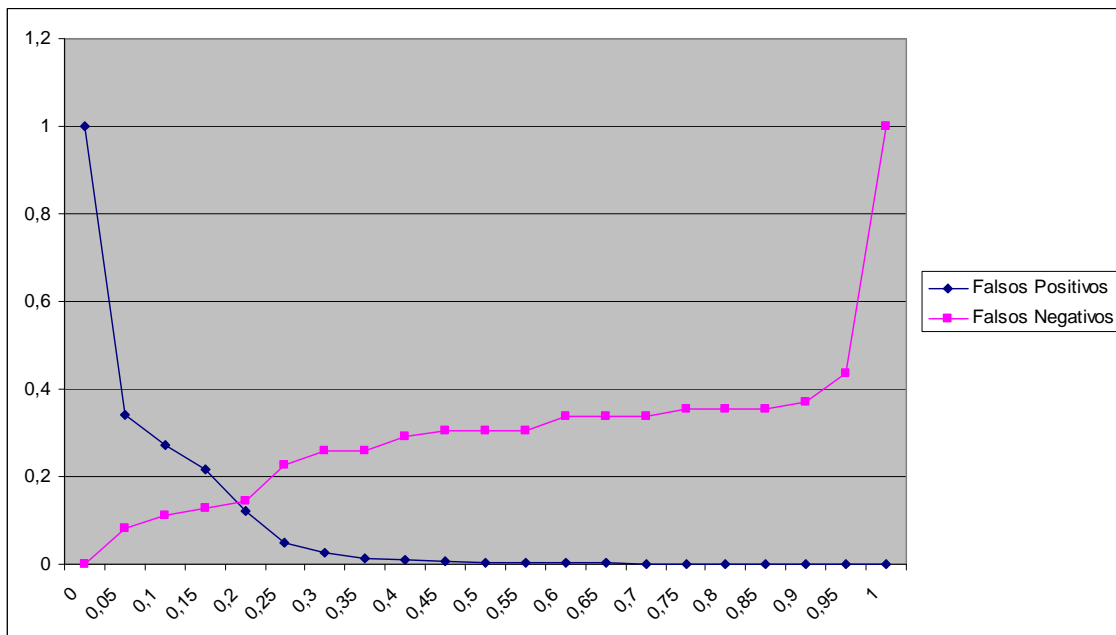


Gráfico 14.- Resultado para método *snakes* escenario 1

### 8.2.2. Escenario 2: Perfil sin giro horizontal contra todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 21. El Gráfico 15 muestra la variación de los FP y FN con el umbral.

Tabla 21.- Resultado para método *snakes* escenario 2

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	9350	185	9350	0	100,00%	0,00%
0,05	2581	6954	2544	37	27,21%	20,00%
0,10	2075	7460	2027	48	21,68%	25,95%
0,15	1660	7875	1607	53	17,19%	28,65%
0,20	917	8618	852	65	9,11%	35,14%
0,25	438	9097	353	85	3,78%	45,95%
0,30	280	9255	185	95	1,98%	51,35%
0,35	197	9338	96	101	1,03%	54,59%
0,40	171	9364	64	107	0,68%	57,84%
0,45	157	9378	45	112	0,48%	60,54%
0,50	147	9388	33	114	0,35%	61,62%
0,55	139	9396	25	114	0,27%	61,62%
0,60	139	9396	20	119	0,21%	64,32%
0,65	137	9398	17	120	0,18%	64,86%
0,70	133	9402	13	120	0,14%	64,86%
0,75	133	9402	10	123	0,11%	66,49%
0,80	130	9405	7	123	0,07%	66,49%
0,85	128	9407	5	123	0,05%	66,49%
0,90	131	9404	4	127	0,04%	68,65%
0,95	135	9400	3	132	0,03%	71,35%
1,00	185	9350	0	185	0,00%	100,00%

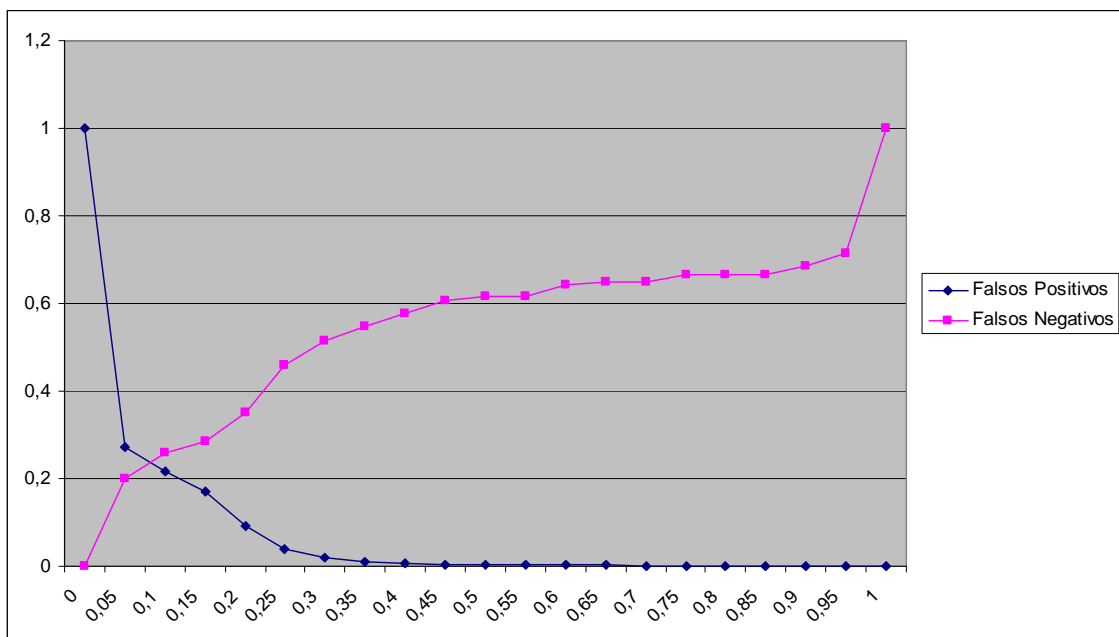


Gráfico 15.- Resultado para método *snakes* escenario 2

### 8.2.3. Escenario 3: Todos los perfiles con todos los perfiles

Los resultados de las pruebas para este escenario se muestran en la Tabla 22. El Gráfico 16 muestra la variación de los FP y FN con el umbral.

Tabla 22.- Resultado para método *snakes* escenario 3

UMBRAL	FALLOS	ACIERTOS	FALSOS POSITIVOS	FALSOS NEGATIVOS	% FP	% FN
0,00	34608	816	34608	0	100,00%	0,00%
0,05	25089	10335	25027	62	72,32%	7,60%
0,10	16120	19304	15980	140	46,17%	17,16%
0,15	5594	29830	5305	289	15,33%	35,42%
0,20	3204	32220	2818	386	8,14%	47,30%
0,25	2272	33152	1843	429	5,33%	52,57%
0,30	1738	33686	1274	464	3,68%	56,86%
0,35	1435	33989	938	497	2,71%	60,91%
0,40	1227	34197	709	518	2,05%	63,48%
0,45	1124	34300	589	535	1,70%	65,56%
0,50	1037	34387	487	550	1,41%	67,40%
0,55	981	34443	420	561	1,21%	68,75%
0,60	951	34473	383	568	1,11%	69,61%
0,65	918	34506	346	572	1,00%	70,10%
0,70	905	34519	327	578	0,94%	70,83%
0,75	894	34530	310	584	0,90%	71,57%
0,80	886	34538	298	588	0,86%	72,06%
0,85	881	34543	284	597	0,82%	73,16%
0,90	862	34562	258	604	0,75%	74,02%
0,95	850	34574	234	616	0,68%	75,49%
1,00	816	34608	0	816	0,00%	100,00%

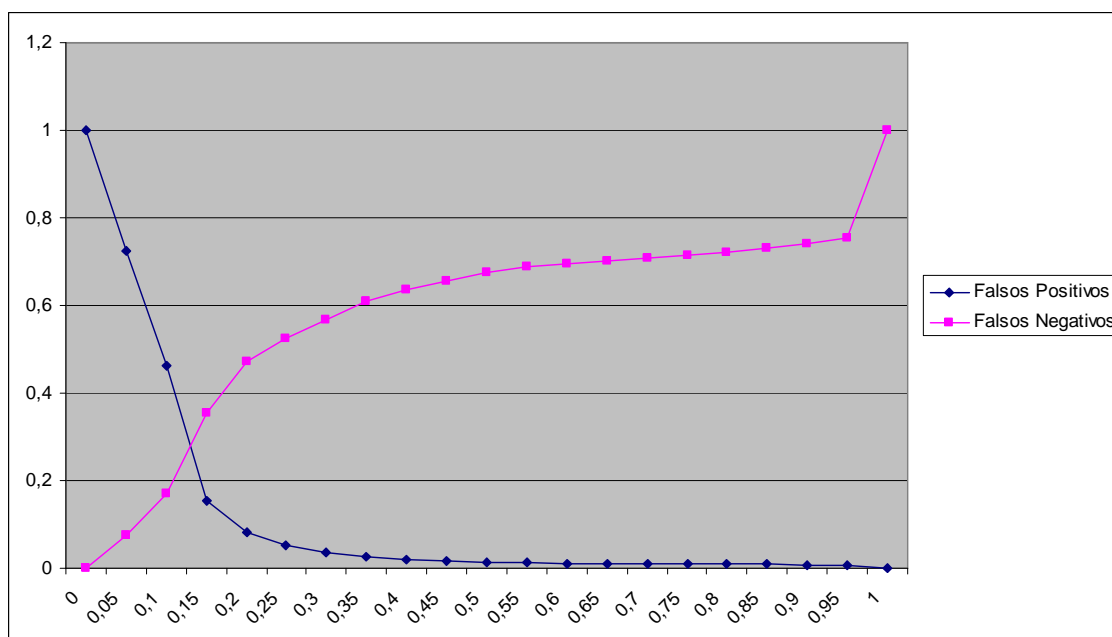


Gráfico 16.- Resultado para método *snakes* escenario 3

### 8.3. Análisis de resultados

Los resultados dependen mucho del escenario en el cual se prueba. Para el escenario 1, en el que se comparan las personas sin ningún giro, los resultados son aceptables. En torno a un 10%. Hay que tener en cuenta que, aunque sea sin giros, la expresión puede haber cambiado y existen personas con gafas. Al ser un método global, que se centra en el contorno, es muy susceptible a modificaciones del mismo. El *snake* se puede adaptar en cierto grado al contorno, pero se ha definido un *snake* rígido por lo que intenta mantener la forma original, de esta manera, las variaciones de dicho contorno van penalizando las medidas.

En el escenario 2 añadimos las imágenes con todos los giros. Este hecho hace que dichas imágenes tengan ocultación, variación de gesto y cambio de expresión, haciendo más difícil el reconocimiento. Por este motivo, el método proporciona peores resultados, porque tiene problemas para adaptar el *snake* al contorno.

El escenario 3 empeora la situación porque probamos desde todos los ángulos con todos los ángulos teniendo los mismos problemas que en el escenario 2 pero con peores resultados.

Hay que destacar algunas características del método para poder entender los resultados. Hay numerosos perfiles, que sin ser de la misma persona tienen un cierto parecido y al dotar al *snake* de cierta flexibilidad hacen que se adapte bastante bien. Por el contrario hay perfiles de la misma persona, que por ocultación, cambio de gesto, etc. no es posible dicha adaptación. El método intenta llegar a un punto medio entre un caso y otro, por este motivo cuando se permite un cierto ángulo en las imágenes los resultados empeoran.

Como muestra la Figura 46, la distribución de fallos y aciertos está solapada en gran número de casos, haciendo difícil la separación clara entre ellos. Esta situación es debido al parecido entre algunas imágenes de personas distintas y a la diferencia en imágenes de la misma persona.

## 9. Conclusiones

El principal objetivo del proyecto es la comparación entre distintos métodos de reconocimiento de siluetas. Se han realizado las pruebas y analizado los resultados para cada método. A continuación se van a comparar los resultados de todos los métodos, analizando las características principales de los métodos, sus puntos fuertes y débiles y proponiendo posibles mejoras sobre ellos.

Se han definido 3 escenarios de pruebas en función de los perfiles usados para las mismas, los resultados se van a analizar para estos 3 escenarios, observando como se comportan los métodos en cada escenario.

### 9.1. *Objetivos cumplidos*

Se han creado 5 métodos para el reconocimiento de personas, basados en distintas propiedades de las siluetas. Para realizar las pruebas, se ha creado una base de datos de siluetas de perfil con 269 imágenes de 39 personas, las imágenes se han tomado con distintos ángulos de giro horizontal y 6 personas han sido fotografiadas con y sin gafas.

Se han creado tres escenarios de prueba y se ha sometido a todos los métodos a las mismas pruebas.

Se han realizado todos los pasos previos para cumplir los objetivos del proyecto, ahora queda comparar los resultados de los métodos. Como se han definido tres escenarios, se van a comparar los resultados para cada escenario.

#### 9.1.1. Escenario 1: Perfiles sin giro horizontal

Este escenario define una situación ideal, los individuos fotografiados, no tienen ningún giro horizontal 'forzado' y pueden tener los cambios de expresión propios de la persona, cambios de gesto o peinado normales.

Como muestra el Gráfico 17, en el que se pinta el porcentaje de Falsos Positivos frente al porcentaje de Falsos Negativos de cada método. El orden de los métodos que aportan mejores resultados es:

1. **Medidas geométricas:** Se basa en medidas estructurales del perfil que difícilmente cambian. Hay que resaltar que este método es muy sensible a la correcta ubicación de los puntos de interés, ya que la situación errónea de dichos puntos produce resultados inadecuados.
2. **Snakes:** Se basa en la definición global del contorno y funciona bien en muchos casos, pero es sensible a cambios de gesto, ocultación, gafas, etc. Al basarse el método en el contorno completo, si alguna parte de dicho contorno cambia, hace bajar los valores de semejanza. Como el umbral de los valores de semejanza está muy ajustado, la bajada de dichos valores provoca que no

se reconozca al individuo. Como para este escenario los perfiles no tienen apenas variación los resultados son buenos.

3. **Suma de Curvatura:** Se basa en la media de las medidas locales de curvatura de los puntos de interés. Al no tener una visión de conjunto del perfil, el método es sensible al cambio de gesto, y un cambio de gesto de la boca puede hacer fallar al sistema (en la boca se sitúan 3 puntos de interés).
4. **Curvatura:** Se basa en distintas medidas de curvatura alrededor de los puntos de interés. Estas medidas son muy locales y muy sensibles de cualquier cambio.
5. **Histograma:** Se basa en la medida de distancia de los puntos próximos a los puntos de interés, es una media de características local. Al ser local, es muy sensible a cualquier cambio.

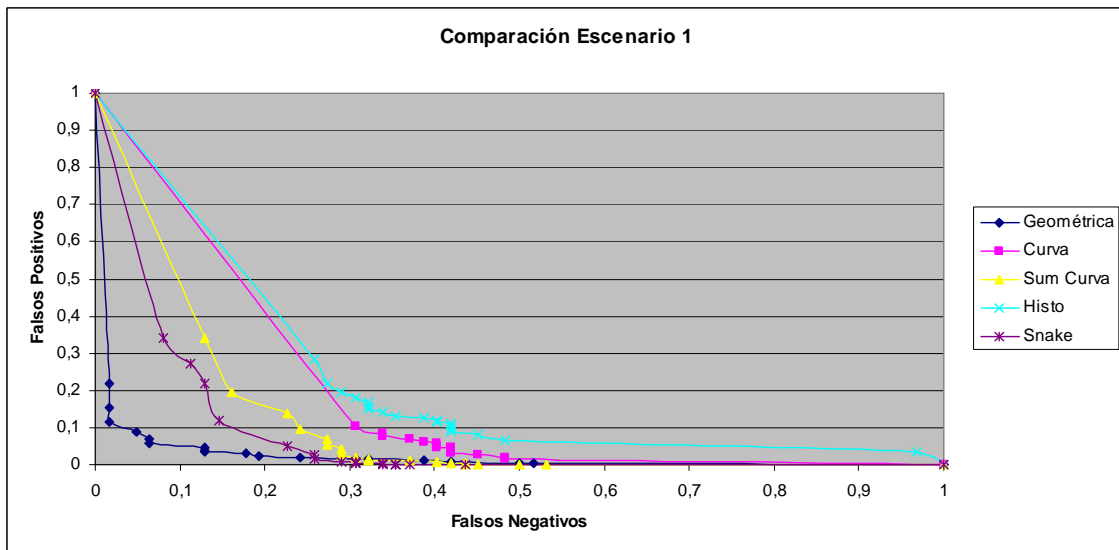


Gráfico 17.- Comparación de métodos para escenario 1

### 9.1.2. Escenario 2: Perfil sin giro horizontal contra todos los perfiles

En este escenario al incluir giros aparecen ocultaciones, cambios de perspectiva y más cambios de gesto. El Gráfico 18 muestra el comportamiento de los distintos métodos en este escenario. El orden de los métodos que aportan mejores resultados es el mismo que en el caso anterior, pero los resultados son peores. A continuación se van a indicar los principales motivos por los que se obtienen peores resultados en cada método:

1. **Medidas geométricas:** No se ha comprobado la correcta ubicación de todos los puntos en los casos de fallo, además de la ocultación de algún punto de interés.
2. **Snake:** Ocultación de partes del contorno, cambio de gesto, cambio del contorno por el ángulo de giro.



3. **Suma de Curvatura:** Ocultación, cambio de gesto, suavizado de los ángulos por el giro.
4. **Curvatura:** Ocultación, cambio de gesto, suavizado de los ángulos por el giro.
5. **Histograma:** Ocultación, cambio de gesto, suavizado de los ángulos por el giro.

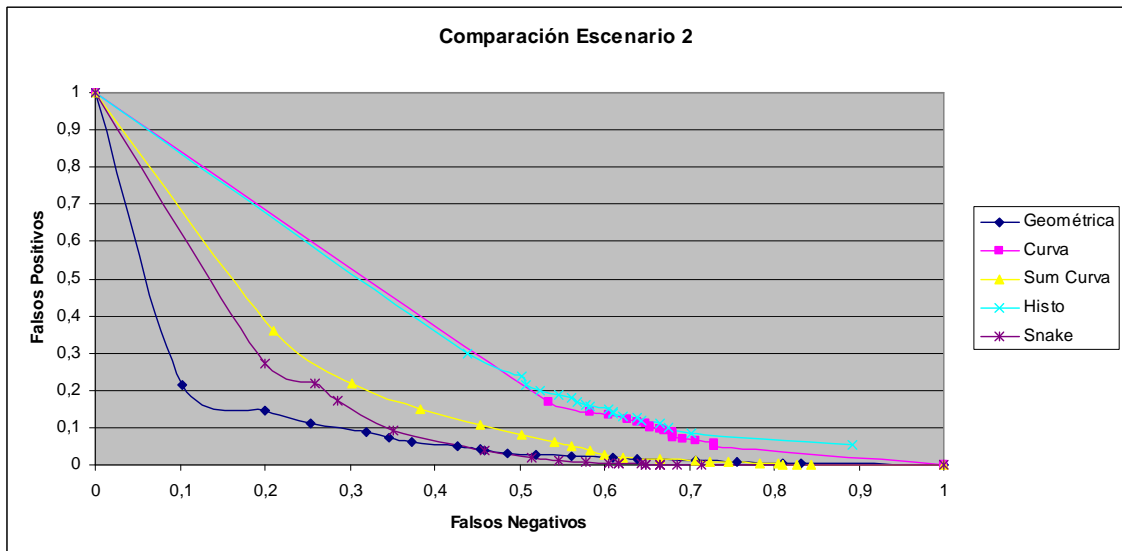


Gráfico 18.- Comparación de métodos para escenario 2

### 9.1.3. Resultados de la comparación entre todos los perfiles

Este escenario cambia el elemento con el que se compara, se comparan todos los perfiles con todos los perfiles lo que incluye más variaciones entre perfiles. Como se puede ver en el Gráfico 19 los datos son similares al caso anterior, han empeorado un poco a excepción del método de suma de curvatura que se mantiene más constante.

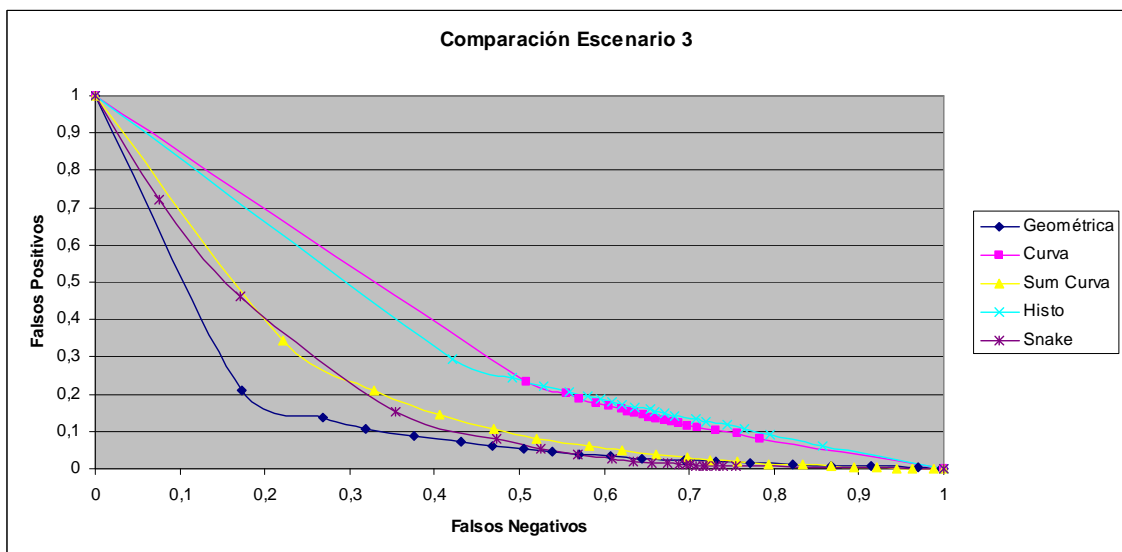


Gráfico 19.- Comparación de métodos para escenario 3

## **9.2. Conclusiones sobre los métodos**

Los métodos que aportan mejores resultados son los basados en características globales. El método geométrico además de ser un método global, se basa en la estructura facial y eso lo hace más robusto, siendo más tolerantes a posibles cambios. Los métodos locales no aportan buenos resultados, pero pueden ser usados como complemento a un método global. La suma de curvaturas tiene una tasa de aciertos similar (en el tercer escenario) al método de los snakes.

Otro aspecto a tener en cuenta a la hora de definir el sistema, es en número de características del mismo. Los sistemas que usan redes de neuronas con un número elevado de características tienen un rendimiento en tiempo peor. Los métodos de curvatura e histograma tienen un n° elevado de características y el tiempo de respuesta es mucho mayor.

## **9.3. Futuros Trabajos**

Con el objetivo de mejorar los resultados obtenidos, se van a definir algunas líneas de investigación que podrían aportar mejores resultados.

### **9.3.1. Determinación automática de los puntos de interés**

El método geométrico necesita la asignación de los puntos de interés. Esta asignación se ha realizado de forma semiautomática y se ha visto la importancia de la correcta identificación de dichos puntos.

En el proyecto, una persona selecciona un punto de referencia y el sistema calcula el punto (próximo al seleccionado) donde la curvatura es máxima. Para calcular este punto se van dando saltos de 5 píxeles para comprobar la curvatura por lo que se pierde precisión. Sería necesario aumentar la precisión y que no hiciera falta la introducción manual del punto de referencia. Un buen punto de partida podría ser el definido en [4].

### **9.3.2. Unión de método global y local**

Otra posible mejora podría ser la unión de un método global con uno local. Como los métodos que mejores resultados aportan son el geométrico y la suma de curvatura, se podrían unir sus características para tener mayor información de las siluetas y así ser capaces de ser más tolerantes a posibles fallos.

### **9.3.3. Mejora del uso de snakes.**

El problema del uso de snakes de forma global, es que no es muy tolerante a variaciones. Por este motivo se podría usar como método global (el actual) y completarlo con medidas locales (de cada parte representativa del perfil). Así se tendrían 2 medidas de semejanza de forma global y 18 de forma local (usando los 9 puntos de interés). Se definiría un *snake* muy rígido para las partes locales, de esta

forma se determinaría el grado de semejanza de cada parte y un *snake* menos rígido para la comparación global de la silueta.

## 10. Bibliografía

- [1] Davidenko, N. (2007). Silhouetted face profiles: A new methodology for face perception research. *Journal of Vision*, 7(4):6, 1–17, <http://journalofvision.org/7/4/6/>, doi:10.1167/7.4.6.
- [2] Z. Liposcak and S. Loncaric. Face Recognition from Profiles Using Morphological Signature Transform. *Proceedings of the 21st Int'l Conference Information Technology Interfaces*, pp. 93-98, Pula, Croatia, 1999.
- [3] Abu Sayeed Md. Sohail and Prabir Bhattacharya, "Detection of Facial Feature Points Using Anthropometric Face Model, in Proc. IEEE International Conference on Signal-Image Technology and Internet-Based Systems, Hammamet, Tunisia, 2006, pp. 656–665
- [4] H. H. Thodberg and H. Olafsdottir, Adding curvature to minimum description length shape models, *Proceedings of British Machine Vision Conference*, vol. 2, pp. 251-260, 2003.
- [5] Peter N. Belhumeur, "Ongoing Challenges in Face Recognition," *Frontiers of Engineering: Reports on Leading-Edge Engineering*, National Academy of Engineering, National Academy Press, pp. 5-14, 2006
- [6] <http://www.face-rec.org/>
- [7] *Image Analysis for Face Recognition*, Xiaoguang Lu
- [8] Anderson, J. A., 1995, *Introduction to Neural Networks* (Cambridge, MA: MIT Press).
- [9] Pan, G., Zheng, L., Wu, Z.H., "Robust Metric and Alignment for Profile-Based Face Recognition: An Experimental Comparison", *Proc. Seventh IEEE Workshop on Application of Computer Vision (WACV/MOTIONS '05)*, Vol. 1, pp. 117-122, 2005.
- [10] - Z. Liposcak and S. Loncaric, "A Scale-Space Approach to Face Recognition From Profiles", *Proc. 8th Intl. Conf. on Computer Analysis of Images and Patterns*, LNCS, Springer, vol. 1689, pp. 243-250, 1999.
- [11] Kakadiaris, I.A., Abdelmunim, H., Yang, W., Theoharis, T., "Profile-based face recognition", *Proc. 8th IEEE Intl. Conf. on Automatic Face & Gesture Recognition (FG '08)*, pp. 1-8, 2008.
- [12] [http://en.wikipedia.org/wiki/Curvature#Curvature\\_from\\_arc\\_and\\_chord\\_length](http://en.wikipedia.org/wiki/Curvature#Curvature_from_arc_and_chord_length)
- [13] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour models", *International Journal of Computer Vision*, pags. 321-331, 1988.
- [14] H. Zhiqiang, H. Chongzhao, "Force fiel analysis snake: an improved parametric active contour model", *Pattern Recognition Letters*, vol. 26, pags. 513-526, 2005.
- [15] Wu, C.J., and Huang, J.S., "Human Face Profile Recognition by Computer", *Pattern Recognition*, vol. 23, No. 3-4, pp. 255-259, 1990.
- [16] Campos, J.C., Linney, A.D., Moss, J.P., "The Analysis of Facial Profiles Using Scale Space Techniques", *Pattern Recognition*, vol. 26, No. 6, pp. 819-824, 1993.
- [17] Gao, Y.S., Leung, M.K.H., "Human face profile recognition using attributed string", *Pattern Recognition*, vol. 35, No. 2, pp. 353-360, 2002.
- [18] Gentile, J.E., Bowyer, K.W., Flynn, P.J., "Profile Face Detection: A Subset Multi-Biometric Approach", *Proc. 2nd Conf. on Biometrics: Theory, Applications and Systems (BTAS '08)*, pp. 1-6, 2008.

## A. Apéndices

### A.1. Detalle de scripts Matlab creados

Todo el código usado en el proyecto está desarrollado para Matlab, únicamente está fuera de Matlab el uso de la herramienta PSnake para obtener los valores de distancia y coincidencia, una vez obtenidos los valores son importados de nuevo a Matlab para realizar las pruebas.

Dentro de Matlab se han definido scripts y funciones para realizar todas las acciones necesarias, que van desde umbralizar las imágenes hasta crear las redes de neuronas y ejecutar las pruebas, y analizar los resultados de las pruebas. Por este motivo se van a enumerar y explicar los scripts creados, el orden de los scripts es el orden cronológico de uso.

**threshold\_images.m** Función usada para umbralizar las imágenes, por las condiciones de la toma de las imágenes (con flash externo e interno) tiene 2 comportamientos para identificar el umbral:

- Con flash externo, umbraliza alrededor al 0.9
- Con flash interno, busca el valor adecuado para umbralizar

En base al valor de dicho umbral realiza distintas umbralizaciones en valores próximos a dicho valor, con el objetivo de seleccionar el más adecuado.

Lee una lista de imágenes de un directorio y sobre el directorio umbral\_auto, crea una carpeta por cada individuo y dentro de dicha carpeta crea otra por cada posición (de giro) de la imagen, en dicha carpeta crea las distintas umbralizaciones de la imagen, el nombre de la imagen lleva un sufijo indicando el valor con el que ha sido umbralizada. En la Figura 49 se puede ver un ejemplo del resultado de la ejecución.

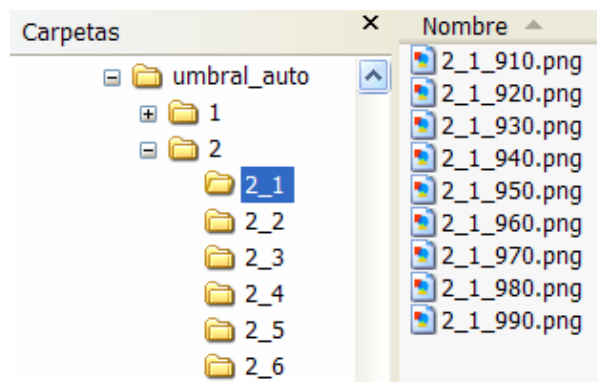


Figura 49.- Ejemplo de estructura después de umbralizar

Parámetros de entrada:

- Base\_path → Ruta del directorio donde se encuentran las imágenes.

La firma de la función es:

*function thresold\_images(base\_path)*

**getMaxImageArea.m** Función que devuelve el objeto más grande dentro de una imagen. Se usa para extraer la parte principal de la imagen e incluir en ella únicamente la parte de la silueta, eliminando el espacio vacío.

Parámetros de entrada:

- *Img* → Imagen de la que se quiere obtener el objeto principal

Parámetros de salida:

- *Img\_result* → Parte principal de la imagen.

La firma de la función es:

*function img\_result=getMaxImageArea(img)*

**extract\_images.m** Función encargada de obtener las imágenes seleccionadas después de umbralizar. Prepara estas imágenes para extraer el objeto principal y poner el fondo como negro (después de la umbralización está en blanco). Coge las imágenes situadas bajo la raíz del directorio de cada persona y las copias ya tratadas al directorio tratadas eliminando el sufijo de valor de umbral. En la Figura 50 se puede ver la estructura de directorios e imágenes de la función para la entrada.

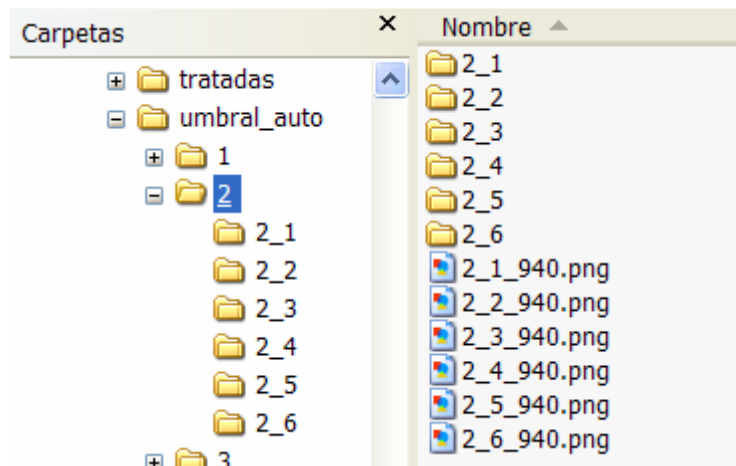


Figura 50.- Ejemplo de directorio de *extract\_images*

Parámetros de entrada:

- *Base\_path* → ruta donde se encuentran las imágenes umbralizadas.

La firma de la función es:

*function extract\_images(base\_path)*

**getImageIndex.m** Función que devuelve el índice de persona y el indicador de posición o giro de la imagen. El nombre de las imágenes llevan la siguiente nomenclatura: *xx\_yy.png* donde 'xx' corresponde con el nº del individuo e 'yy' corresponde con el indicador de posición de la imagen (del 1 al 12).

Parámetros de entrada:

- Name → Nombre de la imagen

Parámetros de salida:

- imageIndex → Vector de 2 posiciones, donde la primera posición corresponde con el individuo y la segunda con el indicador de posición.

La firma de la función es:

```
function imageIndex=getImageIndex(name)
```

**mirror\_images.m** Función encargada de reflejar la imágenes con indicador de posición 4,5,6,10,11,12 con el fin de que todos los perfiles tengan la misma orientación. En la Figura 51 se puede ver un ejemplo de cómo queda la imagen después de reflejar. La función lee un directorio y refleja todas las imágenes que deban ser giradas por la posición, guarda en la carpeta *dch* las imágenes originales.



Figura 51.- Imagen antes de y después de reflejar

Parámetros de entrada:

- Base\_path → Ruta del directorio donde se encuentran las imágenes.

La firma de la función es:

```
function mirror_images(base_path)
```

**setDataBase.m** Script encargado de establecer la ruta donde se encuentran las imágenes y donde será creada la BD. El script se ejecutará antes de empezar a trabajar para tener cargada la ruta donde está ubicada la BD, el resto de funciones trabajarán con esta ruta como referencia.

**getCurvature.m** Función que devuelve la curvatura de una lista de puntos. Calcula la curvatura del punto central de una lista con el resto de elementos de dicha lista, el punto es un vector de 2 posiciones. Si la entrada es una lista de 5 puntos '[1 2 3 4 5]' calcula la curvatura de los puntos 1, 2, 4 y 5 respecto del punto 3.

Parámetros de entrada:

- Points → Vector de  $2*n + 1$  puntos

Parámetros de salida:

- Curvatura → Vector de  $2*n$  elementos con la curvatura de cada punto respecto al punto central.

La firma de la función es:

*function curvature=getCurvature(points)*

**findRowIndex.m** Función que devuelve el índice de una lista de puntos que contiene el punto buscado.

Parámetros de entrada:

- Points → Vector de puntos
- Point → Punto buscado

Parámetros de salida:

- rowIndex → Índice del punto buscado dentro del vector.

La firma de la función es:

*function rowIndex=findRowIndex(points,point)*

**getMaxCurvaturePoint.m** Función que devuelve el punto con la máxima curvatura de un vector de puntos en las proximidades de un punto dado.

Parámetros de entrada:

- Point → Punto de referencia sobre el que se quiere calcular otro punto de máxima curvatura.
- Points → Vector de puntos próximos al punto de referencia.
- distancePoints → N° de puntos de desplazamiento para tomar la curvatura (distancia en puntos al punto de referencia)
- max\_displace → Valor de salto para calcular el punto de referencia
- convexo → Indica si se debe buscar curvatura convexa o cóncava

Parámetros de salida:

maxCurvaturePoint → Punto de máxima curvatura

La firma de la función es:

*function maxCurvaturePoint=  
getMaxCurvaturePoint(point,points,distancePoints,max\_displace, convexo)*

**existPoint.m** Función que comprueba si un punto está ya incluido en un vector de puntos.

Parámetros de entrada:

- Point → Punto a buscar
- pointList → Vector de puntos donde buscar

Parámetros de salida:

- Exist → toma el valor 1 si el punto existe en la lista, y el valor 0 en caso contrario

La firma de la función es:

*function exist=existPoint(point,pointList)*



**getNearestPoints2.m** Función que devuelve un vector de puntos vecinos a un punto de referencia. La función busca un nº de puntos cercanos y ordenados por proximidad, de forma que en el centro del vector está el punto de referencia y alrededor de él los puntos más cercanos, la función es capaz de devolver toda la lista de puntos del contorno, teniendo como centro del vector el punto de referencia. Los parámetros de entrada nos permiten configurar el número de puntos devueltos (en cada sentido) y cada cuantos puntos hay que seleccionar.

Esta es función tiene gran importancia dentro del proyecto, ya que se usa tanto en la asignación de puntos de interés, como en distintos métodos (curvatura, suma de curvatura, histograma y snakes)

Parámetros de entrada:

- Point → Punto de referencia, perteneciente al contorno de la imagen
- Img → Imagen
- Width → N° de salto de puntos contiguos para seleccionar. Por ejemplo, si tenemos los puntos 1 2 3 4 5 6 7 8 9 10 11, como punto de referencia el punto 6, si configuramos width como 2, seleccionaremos los puntos 2 4 6 8 10, para un width de 3 serían 3 6 9. Es decir cada cuantos puntos seleccionamos.
- numPoints → N° de puntos vecinos que hay que devolver en cada sentido, se supone que un punto tiene 2 direcciones de avance (pertenece a una línea), estas direcciones de avance son los sentidos.

Parámetros de salida:

- nearestPoints → Vector de puntos vecinos a un punto de referencia, el vector va ordenado de una dirección (arriba -> abajo) dejando al punto de referencia en el centro, y estando en los extremos los puntos más alejados a dicho punto.

La firma de la función es:

*function nearestPoints=getNearestPoints2(point,img,width,numPoints)*

**getProfileImage.m** Función que devuelve el contorno de una imagen dada.

Parámetros de entrada:

- Img → Imagen original de la que se quiere obtener el contorno.

Parámetros de salida:

- profileImage → contorno de la imagen

La firma de la función es:

*function profileImage=getProfileImage(img)*

**getHorizontalNearestPoint.m** Función que devuelve un punto del contorno de la imagen con la misma coordenada 'y' recibiendo un punto como entrada. La función se desplaza por el eje 'x' hasta que encuentra un punto del contorno.

Parámetros de entrada:

- Point → Punto de referencia

- `Img` → Imagen en la que buscar el punto.

Parámetros de salida:

- `nearestPoint` → Punto con la misma coordenada 'y' del punto de referencia que pertenece al contorno.

La firma de la función es:

*function nearestPoint=getHorizontalNearestPoint(point,img)*

**loadSizeImagePointsAuto.m** Función que carga una imagen para introducirla en la base de datos, únicamente almacena los puntos tomados como referencia para redimensionar las imágenes.

La función muestra la imagen y se queda esperando a que se seleccionen los puntos de la nariz y barbilla (tomados para normalizar el tamaño y ángulo) con los puntos marcados por el usuario, calcula los puntos de curvatura máxima y almacena estos puntos para poder procesar (redimensionar y girar) todas las imágenes a la vez. Almacena los puntos en el fichero con nombre '*sizeimagespoints.mat*'

Parámetros de entrada:

- `Imgpath` → Nombre de la imagen que se quiere incluir, el directorio de trabajo debe ser el directorio donde está la imagen.
- `Dbpath` → Ruta del directorio donde está almacenada la BD de imágenes
- `ImageIndex` → Posición dentro de la BD en la que situar la imagen. Si el índice es 0, la sitúa al final.

Parámetros de salida:

- `Sizeimagespoints` → Vector con una estructura de nombre y puntos, donde se almacena la imagen y los puntos introducidos.

La firma de la función es:

*function sizeimagespoints=loadSizeImagePointsAuto(imgpath,dbpath,imageIndex)*

**loadDirSizeImagePoints.m** Función encargada de llamar a `loadSizeImagePointsAuto` para todas las imágenes de una persona.

Parámetros de entrada:

- `Dbpath` → Ruta del directorio de la Base de datos
- `imageIndex` → Índice de la imagen que se quiere introducir
- `imageSubindex` → Subíndice (o indicador de posición) de la imagen desde la que se quiere empezar. El rango es de 1 a 12, cuando no se encuentra la siguiente imagen, el proceso para.

La firma de la función es:

*function loadDirSizeImagePoints(dbpath,imageIndex, imageSubIndex)*

**resizeImages.m** Función encargada de redimensionar y rotar las imágenes almacenadas en `sizeimagespoints.mat` para normalizarlas. Normaliza todas las imágenes de golpe y las almacena en el directorio `resized`.

Parámetros de entrada:

- `Dbpath` → Ruta del directorio de la base de datos.
- `startIndex` → Índice de la imagen de `sizeimagespoints` con la que comenzar el proceso, lo normal es empezar por 1.
- `endIndex` → Índice de la imagen de `sizeimagespoints` con la que terminar el proceso.

La firma de la función es:

*function resizeImages(dbpath,startIndex, endIndex)*

**showImgPoints.m** Función que muestra por pantalla, o escribe en fichero una imagen y una lista de puntos indicados. Se usa para poder visualizar los puntos de interés cargados de una imagen. En la Figura 52 se ve un ejemplo de ejecución de la función.

Muestra la imagen con el fondo en gris, y los puntos seleccionados en negro.

Parámetros de entrada:

- `img` → Imagen que se quiere mostrar
- `points` → Puntos dentro de la imagen que se quieren mostrar (resaltar)
- `wripath` → Ruta donde almacenar la imagen, si este parámetro es un string vacío, muestra la imagen por pantalla.

La firma de la imagen es:

*function showImgPoints(img,points,wripath)*

**loadImageResizedPointsAuto.m** Función encargada de mostrar una imagen (previamente normalizada en tamaño y ángulo) para seleccionar los puntos de interés y añadir la imagen a la BD. La selección de los puntos de interés es aproximada, porque se calculan los puntos próximos a los seleccionados donde la curvatura es máxima.

La BD es un vector que contiene el nombre de la imagen y los puntos seleccionados, que se almacena en el fichero 'imagespoints.mat'.

Adicionalmente a la inserción de la imagen en la BD, se escribe en el directorio 'pointsimg' la imagen con los puntos seleccionados. En la Figura 52 se puede ver un ejemplo de los puntos seleccionados, el grosor original de los puntos de salida es menor, pero con el tamaño original de los puntos, no se apreciaría en una imagen reducida, por este motivo han sido resaltados con Paint.



Figura 52.- Ejemplo de selección de puntos de una imagen

La función pide la introducción de los 9 puntos, aunque el punto 9 se calculará posteriormente, esta función está así, porque el cálculo del 9º punto se hizo cuando ya estaba la BD creada.

Parámetros de entrada:

- `imgPath` → Ruta de la imagen.
- `dbpath` → Ruta del directorio de la Base de datos.
- `imageIndex` → Índice de la imagen dentro de la BD, si se indica el valor 0 se pone al final.

Parámetros de salida:

- `imagespoints` → Vector de las imágenes introducidas (BD).

La firma de la función es:

```
function imagespoints=loadImageResizedPointsAuto(imgpath,dbpath,imageIndex)
```

**loadDirResizedImagePoints.m** Función encargada de llamar a `loadImageResizedPointsAuto` para todas las imágenes de una persona. Busca las imágenes del directorio 'resized'.

Parámetros de entrada:

- `Dbpath` → Ruta del directorio de la Base de datos
- `imageIndex` → Índice de la imagen que se quiere introducir
- `imageSubindex` → Subíndice (o indicador de posición) de la imagen desde la que se quiere empezar. El rango es de 1 a 12, cuando no se encuentra la siguiente imagen, el proceso para.

La firma de la función es:

```
function loadDirResizedImagePoints(dbpath,imageIndex, imageSubIndex)
```

**loadImagePointsAuto.m** Función encargada de introducir una imagen en BD, la función pide la selección de los puntos de la nariz y barbilla, normaliza la imagen y vuelve a mostrar la imagen normalizada para la selección de los 9 puntos de interés. La función busca los puntos de interés, usando los puntos introducidos por el usuario como referencia y selecciona los puntos de curvatura máxima.

Esta función se ha dejado de usar, por ser más rápido el proceso cuando se introducen las imágenes en bloque, usando: *loadDirSizeImagePoints*, *resizeImages* y *loadDirResizedImagePoints*.

Parámetros de entrada:

- *Imgpath* → Ruta de la imagen que se quiere introducir.
- *Dbpath* → Ruta del directorio de la BD.
- *imageIndex* → Índice dentro de la BD donde se quiere introducir, 0 para insertar al final.

Parámetros de salida:

- *imagespoints* → Vector de las imágenes introducidas (BD).

La firma de la función es:

```
function imagespoints=loadImagePointsAuto(imgpath,dbpath,imageIndex)
```

**loadDirImagePoints.m** Función encargada de llamar a *loadImagePointsAuto* para todas las imágenes de una persona.

Parámetros de entrada:

- *Dbpath* → Ruta del directorio de la Base de datos
- *imageIndex* → Índice de la imagen que se quiere introducir
- *imageSubindex* → Subíndice (o indicador de posición) de la imagen desde la que se quiere empezar. El rango es de 1 a 12, cuando no se encuentra la siguiente imagen, el proceso para.

La firma de la función es:

```
function loadDirImagePoints(dbpath,imageIndex, imageSubIndex)
```

**getZoomImagePoints.m** Función que hace zoom sobre una zona de la imagen y devuelve el punto seleccionado por el usuario. La función muestra una zona de la imagen aumentada y se queda esperando a que el usuario seleccione un punto, una vez seleccionado busca el punto perteneciente al contorno y lo devuelve.

Parámetros de entrada:

- *image* → Imagen sobre la que se quiere hacer zoom
- *centerPoint* → Punto central sobre el que se hace zoom
- *portionSize* → Tamaño del area mostrado (nº de pixeles desde el centro a cada extremo).

Parámetros de salida:

- *selectedPoint* → Punto seleccionado.

La firma de la función es:

*function selectedPoint=getZoomImagePoint(image,centerPoint,portionSize)*

**loadImagePointsManual.m** Función encargada de introducir una imagen en la BD, la diferencia de esta función respecto a la forma automática, es que en lugar de calcular el punto de máxima curvatura respecto a los puntos introducidos por el usuario, muestra por cada punto el área próximo a dicho punto de forma aumentada (hace zoom) para poder seleccionar con precisión el punto.

Esta función no se usa porque requiere mucha más interacción del usuario (tiene que pinchar el doble de puntos) y los puntos se introducen manualmente. Esta función está configurada para normalizar usando los puntos de la nariz, ya que se usó antes de cambiar la forma de normalizar.

Parámetros de entrada:

- *Imgpath* → Ruta de la imagen que se quiere introducir.
- *Dbpath* → Ruta del directorio de la BD.
- *imageIndex* → Índice donde se quiere introducir la imagen, 0 para insertar al final.

Parámetros de salida:

- *imagespoints* → Vector de las imágenes introducidas (BD).

La firma de la función es:

*function imagespoints=loadImagePointsManual(imgpath,dbpath,imageIndex)*

**updatePoints.m** Función usada para modificar un punto de interés de una imagen almacenada en la BD. La función muestra las proximidades del punto (con zoom) y se queda esperando a que el usuario seleccione la nueva ubicación del punto, una vez seleccionado el punto, calcula el punto del contorno de la imagen más próximo y lo almacena modifica la definición del punto en la BD.

La función almacena la imagen con la selección de puntos usando la función *showImgPoints*, de manera que queda actualizado con el estado real de la BD.

Parámetros de entrada:

- *imageIndex* → Índice de la imagen dentro de la BD.
- *Dbpath* → Ruta del directorio de la BD.
- *pointsNumber* → N° de punto que se quiere modificar (del 1 al 9).

Parámetros de salida:

- *imagespoints* → Vector de las imágenes introducidas (BD).

La firma de la función es:

*function imagespoints=updatePoints(imageIndex,dbpath,pointsNumber)*

**getNearestWidthPoint.m** Función que calcula el punto del contorno que está a una distancia determinada, se recorre los puntos a una distancia (en movimiento de píxeles)

determinada y para cuando encuentra un punto que pertenece al contorno de la imagen. En la Figura 23 se ve un ejemplo de los puntos chequeados hasta encontrar un punto del contorno.

Parámetros de entrada:

- `Img` → Imagen en la que buscar el punto del contorno
- `Point` → Punto de referencia, desde el que se quiere encontrar el punto del contorno.
- `Width` → Distancia del punto de referencia al punto del contorno seleccionado.

Parámetros de salida:

`resultPoint` → Punto del contorno seleccionado.

La firma de la función es:

```
function [resultPoint]=getNearestWidthPoint(img,point,width)
```

**calculate9thPoint.m** Función que calcula el punto 9 para todas las imágenes de la BD. En la Figura 23 se puede ver como se calcula dicho punto.

Cuando se creo la BD este punto lo seleccionaba el usuario, pero por la dificultad para seleccionarle correctamente, se ha creado esta función para calcularle automáticamente. Se podría meter este cálculo directamente en la función que le pide al usuario que seleccione los 9 puntos, así se ahorraría introducir dicho punto.

Parámetros de entrada:

- `Db_path` → Ruta del directorio de la BD.

La firma de la función es:

```
function calculate9thPoint(db_path)
```

**removeImagePoints.m** Función que borra una imagen de la BD. Lee la BD, borra la imagen indicada y guarda la BD.

Parámetros de entrada:

- `Dbpath` → Ruta del directorio de la BD.
- `imagenname` → Nombre de la imagen que se quiere borrar.

Parámetros de salida:

- `Imagespoints` → Vector con la imágenes de la BD.

La firma de la función es:

```
function imagespoints=removeImagePoints(dbpath,imageName)
```

**getDistance.m** Función que calcula la distancia entre 2 puntos.

Parámetros de entrada:

- Point1 → Punto 1.
- Point2 → Punto 2.

Parámetros de salida:

- Distance → Distancia entre los 2 puntos.

La firma de la función es:

*function distance=getDistance(point1, point2)*

**getAngle.m** Función que devuelve el ángulo formado por 3 puntos.

Parámetros de entrada:

Parámetros de entrada:

- Point1 → Punto 1.
- Point2 → Punto 2.
- Point3 → Punto 3.

Parámetros de salida:

- angle → Ángulo formado por los 3 puntos

La firma de la función es:

*function angle=getAngle(point1, point2, point3)*

**calculateCharacteristics.m** Función que calcula las características para los métodos: trigonometrico, curvatura, suma de curvatura e histograma.

Calcula todas las características de una imagen determinada y devuelve un vector con dichas características, para separar las características entre métodos se usa el valor 0, esta separación es únicamente para poder tener una referencias en el caso de inspección visual. Este valor 0 se pone además para separar partes dentro de las características.

El motivo por el que se hace todo en la misma función es para crear un único vector para todos los métodos y posteriormente seleccionar para cada uno la parte que interese (en la función *loadCharacteristics.m* se hace la separación).

Parámetros de entrada:

- Points → Puntos seleccionado de la imagen
- Img → Imagen sobre la que extraer las características.

Parámetros de salida:

Characteristics → Características para los 4 métodos de la imagen indicada.

La firma de la función es:

*function characteristics=calculateCharacteristics(points,img)*

**generateCharacteristics.m** Función encargada de calcular las características de todas las imágenes almacenadas en la BD. Graba las características en un fichero llamado 'characteristics.mat' en el directorio de la BD.



Llama a la función *calculateCharacteristics* por cada imagen de la BD.

Parámetros de entrada:

- *Db\_path* → Ruta del directorio donde se encuentra la BD

Parámetros de salida:

*Characteristics* → Características de todas las imágenes de la BD.

La firma de la función es:

*function characteristics=generateCharacteristics(db\_path)*

**recalculateCharacteristics.m** Función que calcula las características de una imagen que previamente las tenía ya calculadas, se usa por el caso en el que se cambie algún punto.

Carga el fichero con las características (*characteristics.mat*), calcula las características de la imagen y las vuelve a guardar en el fichero.

Parámetros de entrada:

- *Db\_path* → Ruta del directorio de la BD.
- *Index* → Índice de la imagen en la BD

La firma de la función es:

*function recalculateCharacteristics(db\_path,index)*

**generateSnakes.m** Función encargada de transformar las imágenes de la BD al formato que necesita la aplicación PSnake.exe, en la Figura 37 se puede ver un ejemplo.

Transforma todas las imágenes de la BD para usarlas en la aplicación PSnake y las almacena bajo un directorio de la BD, la función permite crear las imágenes reduciendo (o ampliando) el tamaño.

La función escribe las imágenes en formato bmp y png. En la imagen en formato 'bmp' la usa la aplicación PSnake y contiene el contorno grueso, la imagen en formato 'png' tiene el grueso original y es la entrada de la función *generateSnakesXML*.

Parámetros de entrada:

- *db\_path* → Ruta del directorio de la BD.
- *snakeSubFolder* → Directorio desde *db\_path* donde se va a situar la carpeta 'snakes' que contendrá las imágenes creadas.
- *reductionFactor* → Factor de reducción de las imágenes.

La firma de la función es:

*function generateSnakes(db\_path,snakeSubFolder,reductionFactor)*

**writeSnakeXML.m** Función que escribe en un fichero xml la información de los puntos que forman el snake. El formato del xml es el definido por la aplicación PSnake.

Parámetros de entrada:

- Points → Puntos que forman el snake
- centerPoint → Punto central del snake
- xmlPath → Ruta del fichero xml escrito.

La firma de la función es:

*function writeSnakeXML(points,centerPoint,xmlPath)*

**generateSnakesXML.m** Función encargada de crear los ficheros xml que definen los snakes de las imágenes de la BD. Escribe en la carpeta snakesXML los ficheros xml que definen los snakes. Necesita las imágenes generadas por *generateSnakes* para poder generar los ficheros xml. Además de crear los ficheros, escribe en la carpeta snakesPoints las imágenes con los puntos que definen el snake.

Parámetros de entrada:

- db\_path → Ruta del directorio de la BD.
- snakeSubFolder → Directorio desde db\_path donde se va a situar la carpeta 'snakesXML' que contendrá los xml creados.
- reductionFactor → Factor de reducción de las imágenes.

La firma de la función es:

*function generateSnakesXML(db\_path,snakeSubFolder,reductionFactor)*

**loadCharacteristics.m** Script que separa las características de la imágenes para cada método, necesita tener la variable 'characteristics' con las características de la imágenes.

Crea los siguientes vectores:

- carTrigo
- carCurva
- carSumCurva
- carHisto

**splitImageNames.m** Script que convierte el nombre de la imágenes en una matriz de 3 columnas donde el significado de las columnas es:

1. Índice de la imagen (posición dentro de la BD)
2. Identificador de la persona.
3. Identificador de la posición de la imagen (del 1 al 12).

**getSnakeTrainingData.m** Función usada para obtener los valores de entrada y salida de un conjunto de comparaciones, esta función se usa para alimentar a las redes de neuronas, para entrenar o para probar.

Los parámetros de entrada son:

- characteristics → Lista de características de los snakes, tiene n filas y 4 columnas, la primera columna es el n° del snake, la segunda el n° de la imagen comparada y la tercera y cuarta el valor de coincidencia y distancia de la comparación.
- Comparations → Lista con los elementos que se quieren comparar, tiene n filas y 2 columnas que contienen los índices del snake y de la imagen que se quiere comparar respectivamente.

- Imagespoints → BD de imágenes

Los elementos devueltos son:

- InputValues → Lista con el mismo nº de filas que comparisons y con 2 columnas que contienen distancia y coincidencia de la comparación de los índices indicados en comparisons.
- outptValues → Lista con el mismo nº de filas que comparisons con una columna que indica si la comparación se ha hecho entre un snake e imagen de la misma persona.

La firma de la función es:

```
function [inputValues outputValues]=
getSnakeTrainingData(characteristics,comparations,imagespoints)
```

**getComparisonData.m** Función encargada de generar un vector de pares de imágenes (índices de imágenes) que se quieren comparar. La función se usa para seleccionar los índices de las imágenes que se quieren comparar, se selecciona una persona (el nº de la persona) y se compara la imagen de posición 2 con:

- La otra imagen sin giro de la persona (posición 5).
- Todas las imágenes de la persona.
- Las imágenes sin giro (posiciones 2, 5, 8, 11) de todas las personas.

Parámetros de entrada:

- List\_images → matriz de 3 columnas con los nombres de las imágenes separados por la el script *splitImageNames*.
- Image\_number → N° del individuo que se quiere comparar. Se seleccionara la imagen 2 de dicho individuo.
- Equal → Índica si se quiere comparar con las imágenes del mismo individuo, 1 indica sí y 0 indica no (se compara con las imágenes de otros individuos)
- All → Índica si se quiere comparar con todas las imágenes de la misma persona. Únicamente aplica para *equal=1*.

Parámetros de salida:

- Comparations → vector con los índices de las imágenes que se quieren comparar según los parámetros introducidos.

La firma de la función es:

```
function [comparation]=getComparantionData(list_images,image_number,equal,all)
```

**generateTrainingData.m** Función encargada de generar los datos para el entrenamiento (y test) de las redes de neuronas. Devuelve 2 vectores, el primero tiene el resultado de la comparación entre imágenes y el segundo indica si la comparación pertenece a imágenes de la misma persona o no.

Parámetros de entrada:

- Characteristics → Matriz de características de las que se quieren generar los datos.

- Comparisons → Vector con los índices de las imágenes que se quieren comparar.
- Imagespoints → BD con las imágenes, se usa para identificar si las imágenes pertenecen o no a la misma persona.

Parámetros de salida:

- inputValues → Matriz con el nº de filas de ‘characteristics’ y con el nº de columnas de ‘comparations’, el valor de cada posición es el valor absoluto de la diferencia de las características.
- outputValues → Vector con el mismo nº de columnas que inputValues (y ‘comparations’), que toma el valor 0 si las imágenes pertenecen a distintas personas y el valor 1 si pertenecen a la misma.

La firma de la función es:

```
function [inputValues outputValues]=
generateTrainingData(characteristics,comparations,imagespoints)
```

**loadTestEnvironment.m** Script encargado de crear los conjuntos de datos de entrenamiento y de test de cada uno de los métodos. Funciona de 2 modos:

- Calcula los valores y los guarda en un fichero.
- Lee los datos almacenados en fichero.

La generación de los datos es pesada por el volumen de comparaciones que se realizan, por este motivo hay una variable en el script que conmuta entre cálculo y lectura.

Para las pruebas se han definido 3 escenarios, el script tiene una variable para seleccionar que escenario se quiere crear.

**getComparisonErrors.m** Función que analiza los resultados de un test y devuelve una matriz con 3 columnas:

1. Valor real
2. Valor obtenido por el test
3. Indica si ha acertado o no, y en caso de fallo, si ha sido Falso positivo, o Falso negativo. Valor 0 → Correcto, 1 → Falso positivo, -1 → Falso negativo

Parámetros de entrada:

- Real\_values → Valores reales.
- Calculated\_values → Valores devueltos en el test.
- Umbral → Umbral a partir del cual consideramos que el sistema considera que las imágenes pertenecen a la misma persona.

Parámetros de salida:

- comparationsErrors → Matriz que indica los fallos y aciertos del test para un umbral.

**createFFTTrigo.m** Script que crea y entrena la red para el método trigonométrico, y ejecuta el test. Una vez finalizado el test, tomando un umbral de 0,5 muestra información del resultado como:

- N° de elementos en el test
- N° de errores
- % de errores
- N° de Falsos Positivos
- N° de Falsos Negativos

**createFFCurva.m, createFFSumCurva.m, createFFHisto.m y createFFSnake.m**

Scripts iguales a `createFFTrigo`, cambiando únicamente que generan el proceso para el método correspondiente.

**getROC\_Data.m** Función que prepara los datos para pintar gráficas ROC, calcula en n° de Falsos positivos y Falsos negativos para cada valor de umbral definido como un incremento.

Parámetros de entrada:

- `Real_values` → Valores reales
- `Calculated_values` → Valores calculados, resultado de la ejecución del test.
- `Precisión` → Cantidad del umbral que se incrementa en cada iteración. El umbral comienza en 0 y acaba en 1.

Parámetros de salida:

- `Roc_data` → Matriz de 2 columnas que contiene el n° de Falsos positivos y Falsos negativos para cada umbral.

La firma de la función es:

*function [roc\_data]=getROC\_Data(real\_values,calculated\_values,precision)*