Centro de Investigación en Matemáticas, A.C.

Síntesis de Expresiones Faciales para el Reconocimiento de Caras en 3D

T E S I S

que para obtener el grado de

Maestro en Ciencias con Especialidad en Computación y Matemáticas Industriales

presenta

Leonel Ramírez Valdez

Director de Tesis

Dr. Rogelio Hasimoto Beltrán

Guanajuato, Gto. *Agosto de 2007*

Resumen

El reconocimiento facial automatizado tiene una gran cantidad de aplicaciones que van desde los sistemas de seguridad y control de acceso, hasta los sistemas de realidad virtual, los videojuegos interactivos y la comunicación por Internet, es por ello que ha despertado gran interés dentro de la comunidad científica en los últimos años. La información tridimensional tiene un gran potencial para superar los problemas que enfrentan los sistemas de reconocimiento facial basados en imágenes bidimensionales; estos problemas tienen que ver con cambios en iluminación, pose y expresión.

En este trabajo se presenta una alternativa para reconocimiento facial en 3D utilizando información sintetizada para atacar el problema de cambio de expresiones faciales. El proceso comienza haciendo uso del método de los elementos finitos para simular la acción de los músculos faciales en una cara genérica (o cara base), esto con el objetivo de obtener varias expresiones sintéticas como resultado de la interacción de los distintos músculos faciales. Por medio de transformaciones, se ajusta la cara base a cada una de las caras de entrada que formarán parte de la base de datos de usuarios, de manera que se puedan sintetizar expresiones para todos los usuarios. Durante esta etapa se realizan dos pasos intermedios para obtener un buen ajuste: eliminación de ruido y detección de puntos característicos en las caras de entrada.

Para la eliminación de ruido de las mallas se propone un algoritmo iterativo que combina el suavizador Laplaciano y el filtro de mediana. La idea básica del algoritmo consiste en conservar la posición original de los vértices que tengan poco cambio entre iteraciones y suavizar los vértices que sufran grandes cambios. Como resultado, el algoritmo elimina el ruido sin sobresuavizar y preservando las características finas de la malla, incluso mejor que otros algoritmos reportados en la literatura. Para la detección de puntos característicos se propone un proceso heurístico que hace uso de información de la curvatura de la cara e información estadística de la distancia entre estos puntos.

Por último, se generan imágenes de rango, tanto de la cara original como de las caras con expresiones sintetizadas, para cada uno de los usuarios almacenados en la base de datos. Este conjunto de imágenes de rango se utiliza como conjunto de entrenamiento para dos clasificadores: uno basado en PCA (*Eigenfaces*) y el otro basado en LDA (*Fisherfaces*). Los resultados demuestran que al agregar caras con expresiones sintéticas y tomar en cuenta la separación entre clases proporcionada por LDA se mejora sustancialmente la tasa de reconocimiento.

A mis padres, Mario y Gloria, y a mi abuelo, José Valdez †, quienes con su ejemplo me enseñaron a luchar para alcanzar mis metas

Agradecimientos

Quisiera comenzar agradeciendo a Dios, por estar siempre a mi lado, y por haberme puesto en este viaje lleno de retos y aventuras llamado vida. En esta etapa tan importante de mi viaje hubo muchas personas a quienes quiero expresar mi gratitud.

En primer lugar, agradezco al Dr. Rogelio Hasimoto Beltrán por su apoyo y por el tiempo dedicado para conducir a buen término este trabajo. Agradezco también al Dr. Salvador Botello Rionda y al Dr. Flavio Vigueras Gómez por su paciencia y disposición para revisar este trabajo así como por sus comentarios y sugerencias para enriquecerlo. Un agradecimiento adicional al Dr. Salvador Botello por las facilidades otorgadas para utilizar el *MEFi* dentro del trabajo realizado.

Mi más profundo agradecimiento a CIMAT por brindarme el espacio y el equipo necesario para realizar mis estudios de maestría, y más que agradecer a CIMAT, agradecer a cada una de las personas que con su trabajo dan vida y hacen de CIMAT esa gran institución, en especial a los profesores que contribuyeron a mi crecimiento académico y personal.

Un agradecimiento especial a mi familia: a mis padres, Mario y Gloria, que me han apoyado siempre y porque gracias a ellos he aprendido que todo se puede conseguir con esfuerzo y dedicación; a mi hermana Jezabel, siempre alegrándome la existencia; y a Berenice, mi mejor amiga, por estar ahí en todo momento, dispuesta a apoyarme sin condición.

A Gladiola, por brindarme su amor, apoyo y comprensión, y por ser la causa principal que motiva mis actos, sin ella esto no tendría sentido.

Agradezco a mis compañeros y amigos: Dan-El, Mauricio, Miguel, Cyntia, Oscar, Humberto, Pedro Pablo, Oscar (Dalmau), Juanita y Josué, por todo lo que aprendí de ellos, intelectual y humanamente, y también por los buenos momentos que compartimos dentro y fuera de CIMAT. A Gallo Muerto #5: Gerardo, Sergio, Ivvan, Víctor, gracias por todas las horas de música y ensayos que fueron mi terapia de relajación ante la carga de trabajo.

Finalmente, doy las gracias a CONACyT y a CIMAT que por medio de sus programas de becas me brindaron los medios económicos para realizar mis estudios de maestría.

Índice general

1. Introducción **1**

1.1. Descripción del problema 1

1.2. Consideraciones y dificultades técnicas 3

1.2.1. Espacios faciales no-lineales y no-convexos altamente complejos 3

1.2.2. Variación en la apariencia facial 4

1.2.3. Alta dimensionalidad del espacio y conjunto de entrenamiento pequeño 6

1.3. Objetivo 6

1.4. Organización de la tesis 7

2. Antecedentes **9**

2.1. Representación de información en 3D 9

2.1.1. Nube de puntos 9

2.1.2. Modelo poligonal 10

2.1.3. Imagen de rango 10

2.2. Técnicas de adquisición de información en 3D 11

2.2.1. Limitantes y retos de la tecnología actual 12

2.3. Métricas de desempeño 14

2.3.1. Curvas CMC 14

2.3.2. Curvas ROC 14

2.4. Revisión del Estado del Arte 15

3. Síntesis de expresiones faciales **21**

3.1. Anatomía facial 21

3.1.1. La piel 21

3.1.2. El cráneo 22

3.1.3. Los músculos 23
3.1.4. Otros órganos 26
3.2. Preproceso de la cara base 26
3.3. El FACS 28
3.4. El método de los elementos finitos 30
3.4.1. El problema de elasticidad 30
3.4.2. Formulación con elementos isoparamétricos 32
3.5. Implementación y resultados 34

4. Ajuste de la cara base **39**
4.1. Eliminación de ruido 39
4.1.1. Algoritmo Laplaciano 39
4.1.2. Algoritmo de mediana 40
4.1.3. Algoritmo iterativo propuesto 41
4.1.4. Experimentos y resultados 42
4.2. Detección automática de puntos característicos 48
4.2.1. Descriptores de curvatura 49
4.2.2. Distancia estadística entre componentes faciales 50
4.2.3. Algoritmo heurístico propuesto 50
4.2.4. Experimentos y resultados 57
4.3. Ajuste 58
4.3.1. Alineación global 60
4.3.2. Deformación en el plano XY 64
4.3.3. Deformación en Z 66
4.3.4. Resultados 70

5. Reconocimiento facial **71**
5.1. Creación de las imágenes de rango 71
5.2. Técnicas de clasificación 74
5.2.1. Análisis de Componentes Principales 75
5.2.2. Análisis Discriminante Lineal 77
5.3. Implementación del sistema y resultados experimentales 79
5.3.1. Implementación 80
5.3.2. Experimentos 80
5.3.3. Discusión 83

6. Conclusiones y trabajo futuro **85**
6.1. Conclusiones 85
6.2. Trabajo futuro 85

A. La base de datos GavabDB **87**

B. FaceER **89**

Índice de figuras

1.1. Configuración genérica de un sistema de reconocimiento facial 3
1.2. Espacios faciales 4
1.3. No-linearidad y no-convexidad de los espacios faciales bajo translación, rotación, escalamiento y transformaciones Gamma 5
1.4. Variaciones intrasujeto 5
1.5. Variaciones intersujeto vs. Variaciones intrasujeto 6
1.6. Alta dimensionalidad del espacio vs. Conjunto de entrenamiento pequeño 7

2.1. Diferencias entre representaciones en 3D y 2.5D 10
2.2. Representaciones en 3D 11
2.3. Modelo de la misma persona bajo diferentes condiciones de iluminación 12
2.4. Errores en la captura de superficies en 3D 13
2.5. Ejemplo de curva CMC 15
2.6. Ejemplo de curva ROC 16
2.7. Efectos del cambio de expresión en la tasa de reconocimiento 17

3.1. Curva “Esfuerzo vs. Deformación” de la piel 22
3.2. Huesos del cráneo humano 23
3.3. Músculos faciales 24
3.4. Cara base 26
3.5. Capas de la cara base 27
3.6. Definición de un músculo en la cara base 28
3.7. Sólido sujeto a fuerzas aplicadas dentro del sólido y sobre la superficie del sólido 31

4.1. Escala *Shape Index* 50
4.2. Descriptores de superficie 51

4.3. Segmentación facial 53
4.4. Detección de la punta de la nariz 53
4.5. Detección de las esquinas internas de los ojos 54
4.6. Detección de las esquinas externas de los ojos 54
4.7. Detección de las esquinas de la nariz 55
4.8. Detección de las esquinas de la boca 56
4.9. Detección de los bordes de las mejillas 56
4.10. Detección del borde de la frente 57
4.11. Detección del borde de la barbilla 57
4.12. Puntos característicos detectados correctamente 58
4.13. Puntos característicos detectados incorrectamente 58
4.14. Puntos característicos detectados incorrectamente debido a las mallas 59
4.15. Puntos característicos de la cara base detectados manualmente 59
4.16. Alineación global 63
4.17. Alineación de las esquinas de los ojos 65
4.18. Alineación en el plano XY 67
4.19. Proyección cilíndrica de la cara base 68
4.20. Área del triángulo obtenida como el producto cruz de sus vértices . . 68
4.21. Deformación en el eje Z 69

5.1. Inclusión de muestras sintéticas en el espacio facial 72
5.2. Plano inicial para la creación de las imágenes de rango 73
5.3. Preproceso aplicado a las caras de prueba 73
5.4. Preproceso aplicado a las imágenes de rango 75
5.5. Problemas de la proyección con PCA para separación de clases 78
5.6. Imágenes en el conjunto de datos 80
5.7. Desempeño del sistema sin aumento de síntesis de expresiones 81
5.8. Desempeño del sistema con aumento de síntesis de expresiones 82

B.1. Ventana principal de *FaceER* 90
B.2. Editor de expresiones de *FaceER* 91
B.3. Ejemplo de descriptor de curvatura en falso color (*Shape Index*) . . . 93

Índice de tablas

1.1. Características biométricas y sus ventajas relativas 2

2.1. Algoritmos de reconocimiento facial en 3D 20

3.1. FACS Action Units 30
3.2. Expresiones faciales sintetizadas 37

4.1. Comparación del algoritmo propuesto 46
4.2. Comparación entre los mejores resultados de cada algoritmo 48
4.3. Modelo estadístico de la distancia entre componentes faciales 52
4.4. Ejemplos de expresiones faciales sintetizadas 70

5.1. Ejemplos de imágenes de rango usadas en la galería 74
5.2. Tabla comparativa de los experimentos realizados 83
5.3. Tiempos registrados en las etapas previas al reconocimiento 83

Capítulo 1

Introducción

El reconocimiento facial es una tarea cotidiana que los humanos hacemos sin siquiera percatarnos de ello. Los últimos avances en tecnología y la creciente demanda de aplicaciones en seguridad biométrica, control de acceso, realidad virtual, interacción humano-computadora, etc., han provocado que el reconocimiento automatizado de caras sea un área cada vez más atendida por investigadores en procesamiento de imágenes, reconocimiento de patrones, visión computacional, gráficas por computadora y psicología [73], [39], [4], [7].

Una de las principales aplicaciones del reconocimiento facial surge en el campo de seguridad biométrica debido a la necesidad de contar con mecanismos de autenticación que sean más robustos ante posibles ataques de robo de identidad y que a la vez permitan al usuario librarse de toda la maraña de claves de acceso necesarias para ingresar a diversos sistemas como cajeros bancarios, computadoras, sitios en Internet o cuentas de correo electrónico, sólo por mencionar algunos.

Los sistemas biométricos identifican o verifican de forma automática la identidad de un individuo por medio de características biológicas (la cara, la huella digital, la voz, la mano, el iris) o psicológicas (la firma autógrafa) asociadas a éste; de esta manera, el individuo cuenta con una *"clave de acceso biointegrada"* que no necesita recordar ni guardar. La cara posee ciertas ventajas sobre otras características biométricas: es fácil de obtener, no es intrusiva y no necesita cooperación o conocimiento por parte del usuario. En la tabla 1.1 se presenta un sumario de diferentes características biométricas y sus ventajas relativas.

1.1. Descripción del problema

En general, el problema de reconocimiento facial lo podemos enunciar como sigue:

"Dada una imagen fija o una secuencia de video, identificar o verificar una o más personas en la escena usando una base de datos facial almacenada"

	Cara 2D	Firma	Voz	Iris	Dedo	Retina	Paso	Mano	Cara 3D
Exactitud				✓	✓	✓			✓
Costo	✓		✓		✓				
Privacidad	✓			✓		✓	✓		✓
Integridad	✓		✓		✓				✓
Facilidad de uso	✓	✓	✓		✓			✓	✓
Desarrollo	✓	✓	✓		✓			✓	✓
Ocultamiento de identidad				✓		✓		✓	
Falsificación de identidad				✓	✓	✓		✓	✓

Tabla 1.1: Características biométricas y sus ventajas relativas. Aunque el reconocimiento facial en 2D y 3D no es tan exacto como el iris, su facilidad de uso y bajo costo los convierten en una opción preferible para algunos escenarios [4].

Es importante notar que el enunciado hace una clara diferencia entre *identificar* y *verificar*. En ambos escenarios, las imágenes (en 2D o 3D) de personas conocidas están almacenadas en la base de datos o *galería*; las imágenes que posteriormente llegan al sistema son usadas como *pruebas* para comparar contra las imágenes de la galería. La *identificación* se refiere al proceso de comparación uno-a-muchos, donde se compara la imagen de entrada o prueba contra toda la galería para determinar su identidad. En cambio, en la *verificación* se hace una comparación uno-a-uno entre la prueba y la imagen de la galería correspondiente a la identidad que se está demandando. Aunque las tareas básicas son las mismas tanto en identificación como en verificación, existe un mayor reto en la identificación puesto que hay mayores posibilidades de hacer identificaciones incorrectas al comparar una prueba contra la galería completa.

Para alcanzar el objetivo de identificación o verificación facial hay que pasar por varias etapas de procesamiento, las cuáles en conjunto conforman el sistema de reconocimiento facial (figura 1.1). Estas etapas son:

- **Detección facial:** Se encarga de segmentar la cara del resto de la imagen. Si la entrada es una secuencia de video, entonces se necesitará realizar un seguimiento de la cara en el video, lo que en el argot de visión computacional se conoce como *tracking*.

- **Normalización facial:** El objetivo de esta etapa es estandarizar la cara detectada para localizar y extraer características faciales con mayor facilidad. Dependiendo de la aplicación, la normalización puede darse con respecto a propiedades como tamaño y pose usando transformaciones geométricas (rotación, translación y escalamiento), o con respecto a propiedades fotométricas como normalización de la iluminación.

- **Extracción de características:** En esta etapa se trata de extraer un conjunto compacto de características que bajo cierta métrica permitan diferenciar entre

una cara y otra. Dichas características pueden ser geométricas, como la superficie facial y la distancia entre ciertas partes de la cara (ojos, nariz, boca); o fotométricas, como la textura y el color de la piel.

- **Comparación de características:** Aquí es donde se realiza el proceso de reconocimiento comparando los vectores de características de la cara de entrada contra los de las caras almacenadas en la galería, dependiendo del resultado de esta comparación se devolverá la identidad de la cara de entrada o se informará que la cara es desconocida para el sistema.

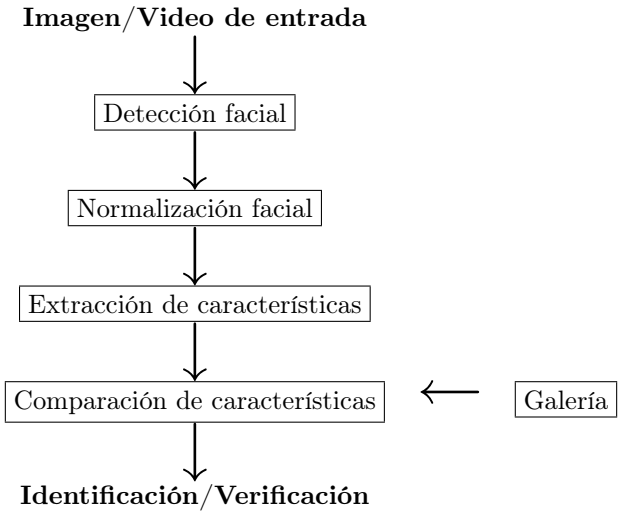


Figura 1.1: Configuración genérica de un sistema de reconocimiento facial.

1.2. Consideraciones y dificultades técnicas

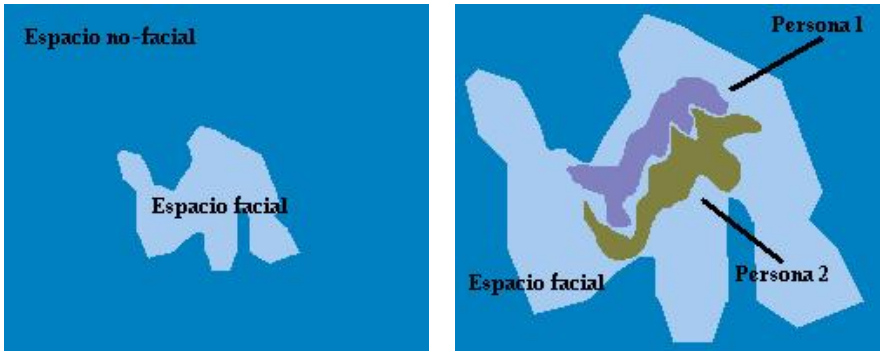
Como se mencionó al principio de este capítulo, los humanos podemos reconocer caras sin mayor esfuerzo; sin embargo, en los sistemas automáticos de reconocimiento facial aún hay grandes dificultades por resolver.

1.2.1. Espacios faciales no-lineales y no-convexos altamente complejos

Una imagen puede ser considerada como un punto o vector de un espacio de alta dimensión donde el número de coordenadas estará dado por el número de pixeles de la imagen. Supongamos que tenemos una imagen de 64×64 en escala de grises con 8 bits por pixel. Cada una de las 256^{4096} [1] posibles imágenes será un punto del espacio 4096-dimensional. Si tenemos una imagen facial, es natural pensar que todas las imágenes de caras con variaciones en tamaño, pose e iluminación estarán agrupadas

[1] aproximadamente 10^{9864}

en cierta forma y formarán un pequeño subespacio de este gran espacio. El problema de clasificación asociado radica en que tanto este subespacio como el subespacio de cada cara individual son altamente no-lineales y no-convexos [67], como se muestra en la figura 1.2.



(a) Espacio facial vs. Espacio no-facial. (b) Espacios faciales de diferentes individuos.

Figura 1.2: Espacios faciales [39].

Para ilustrar la no-linearidad y no-convexidad de los espacios faciales consideremos el experimento realizado en [39]. Se seleccionaron 64 imágenes faciales de tres individuos distintos. A cada imagen se aplican transformaciones de translación, rotación, escalamiento e iluminación con 11 variaciones graduales en los parámetros, produciendo una secuencia de 11 imágenes faciales por cada transformación. Después se aplica *Análisis de Componentes Principales* (PCA) [66] al conjunto de imágenes y se proyectan en el espacio generado por los primeros tres componentes principales. En la figura 1.3 se pueden ver los resultados obtenidos; cada curva en las gráficas representa las secuencias de 11 imágenes, de manera que se tienen 64 curvas por individuo (con un color distinto para cada individuo). El espacio tridimensional generado está proyectado en tres planos para facilitar su visualización. Como se puede apreciar, las trayectorias son no-lineales y no-convexas, de manera que se esperaría que en el espacio original las curvas sean aún más complejas.

1.2.2. Variación en la apariencia facial

Dependiendo de las circunstancias y del equipo con el que se obtiene la imagen, la cara de un mismo individuo puede presentar grandes variaciones de tamaño, pose, iluminación, oclusión debida a accesorios (por ejemplo lentes o bufandas), expresión facial, vello facial y envejecimiento. Estas variaciones son conocidas como *variaciones intrasujeto* (figura 1.4).

Por otra parte, las *variaciones intersujeto* pueden llegar a ser muy pequeñas (como en el caso de gemelos o entre padres e hijos). En conjunto, estos dos tipos de variaciones dificultan la tarea de reconocimiento facial, ya que las variaciones intrasujeto

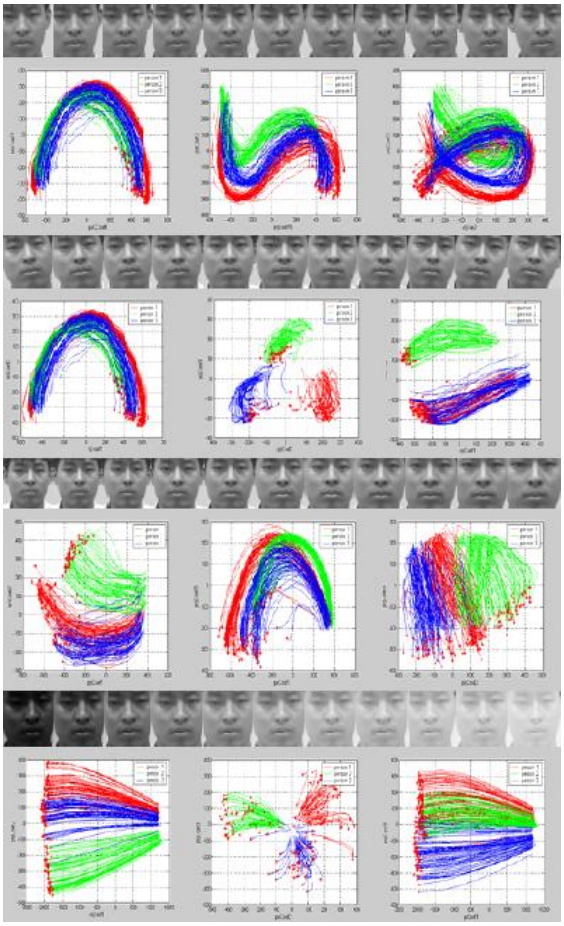


Figura 1.3: No-linearidad y no-convexidad de los espacios faciales (de arriba hacia abajo) bajo translación, rotación, escalamiento y transformaciones Gamma [39].

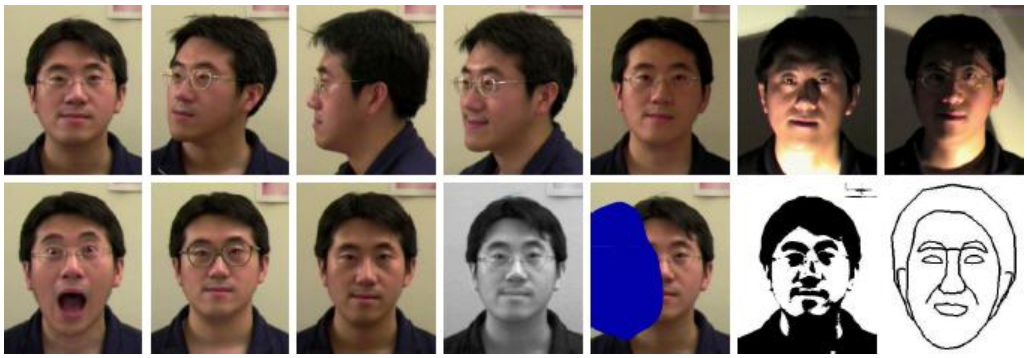


Figura 1.4: Variaciones intrasujeto en pose, iluminación, expresión, oclusión, accesorios (lentes), color y brillo [29].

a menudo llegan a ser mayores que las variaciones intersujeto [3] (figura 1.5).

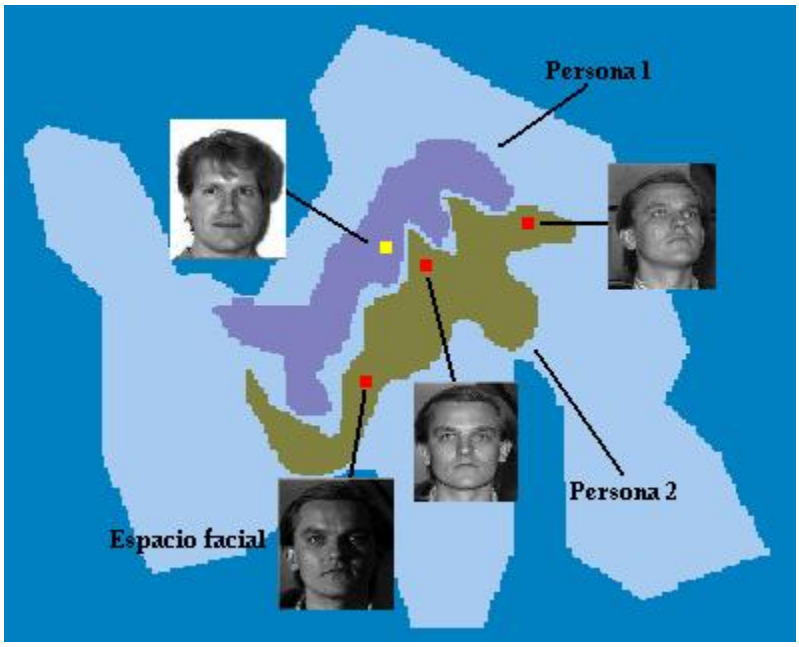


Figura 1.5: Variaciones intersujeto vs. Variaciones intrasujeto. En la figura se muestra cómo la variación de apariencia en el espacio facial entre la Persona 1 y la Persona 2 puede ser más pequeña que las variaciones intrasujeto de la Persona 2 [44]. Las imágenes faciales fueron tomadas de la base de datos de Yale [65].

1.2.3. Alta dimensionalidad del espacio y conjunto de entrenamiento pequeño

Otra dificultad surge al tratar de generalizar el espacio facial de una persona utilizando un conjunto pequeño de muestras. Como ya hemos visto, el espacio facial es demasiado grande (4096-dimensional para imágenes de 64×64 pixeles), sin embargo, el número de muestras por persona la mayor parte de las veces no llega siquiera a 10, por lo que es muy difícil que el espacio facial de la persona se pueda generalizar correctamente con esta pequeña cantidad de muestras de aprendizaje (figura 1.6).

1.3. Objetivo

Como vimos en la sección anterior, una de las dificultades para el reconocimiento facial automatizado radica en la variación de la apariencia facial, y aunque la información en 3D tiene potencial para enfrentar los cambios de iluminación, pose y tamaño, aún quedan los problemas por cambios en la expresión facial, oclusión y envejecimiento.

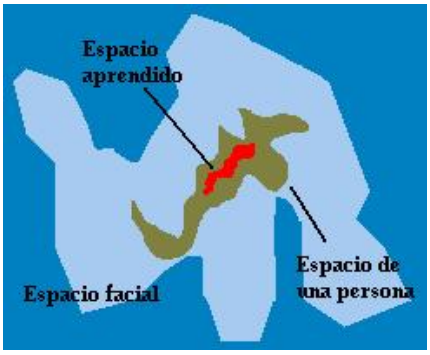


Figura 1.6: Alta dimensionalidad del espacio vs. Conjunto de entrenamiento pequeño. El subespacio aprendido es incapaz de caracterizar todas las imágenes posibles de una misma persona [39].

En este trabajo se pretende atacar el problema de variación en la expresión facial. Para esto, se buscará sintetizar expresiones faciales utilizando únicamente la imagen 3D, en posición frontal y con expresión neutral, de cada uno de los usuarios registrados en la galería. Una vez que se obtengan las imágenes sintetizadas con distintas expresiones, éstas serán agregadas a la galería para aumentar el número de muestras por usuario y tratar de mejorar la tasa de reconocimiento de los clasificadores basados en apariencia, como *Eigenfaces* y *Fisherfaces*.

1.4. Organización de la tesis

La tesis está dividida en seis capítulos:

El **Capítulo 2** es un breve repaso de conceptos de reconocimiento facial en 3D, pasando por las técnicas de adquisición y modelación de información tridimensional, así como una revisión del Estado del Arte.

En el **Capítulo 3** se abarca la parte de síntesis de expresiones faciales, desde la descripción de la anatomía del rostro humano hasta la generación de expresiones artificiales en una cara genérica utilizando el método de los elementos finitos.

El **Capítulo 4** trata sobre el registro entre la cara genérica y una cara de entrada. Aquí se describen los pasos intermedios necesarios para transformar la cara genérica en la cara de entrada: eliminación de ruido, detección automática de puntos característicos y transformaciones rígidas y no rígidas. También se presentan y comparan algunos de los resultados parciales de esta etapa.

El **Capítulo 5** describe la parte de reconocimiento facial en sí. Se detalla la construcción de la base de datos y la implementación de dos reconocedores: uno utilizando *Eigenfaces* y el otro con *Fisherfaces*. En este mismo capítulo se realizan los experimentos y se muestran los resultados de nuestra propuesta.

Finalmente, en el **Capítulo 6** damos las conclusiones del trabajo realizado junto con algunas ideas de trabajo futuro para ampliar y mejorar los resultados.

En los **Apéndices** se da una breve descripción de la base de datos utilizada en este trabajo de tesis (Apéndice A) y de la herramienta de software desarrollada (Apéndice B).

Capítulo 2

Antecedentes

2.1. Representación de información en 3D

En esta sección se revisan algunas de las formas más comunes de representar información en 3D. Algunos autores hacen diferencia entre representaciones en 3D y representaciones en 2.5D. Una representación en 3D incluye la información tridimensional del objeto vista desde todos sus ángulos (figuras 2.1(a) y 2.1(b)). En cambio, una representación en 2.5D solo muestra la información tridimensional visible al momento de digitalizar el objeto; por ejemplo, si se estuviera digitalizando una cabeza de frente, no se podría capturar la información de la parte trasera de la cabeza puesto que quedaría "tapada" por la parte frontal de la misma, como se ilustra en las figuras 2.1(c) y 2.1(d)[1]. En este trabajo usaremos el término 3D para referirnos indistintamente a ambas representaciones.

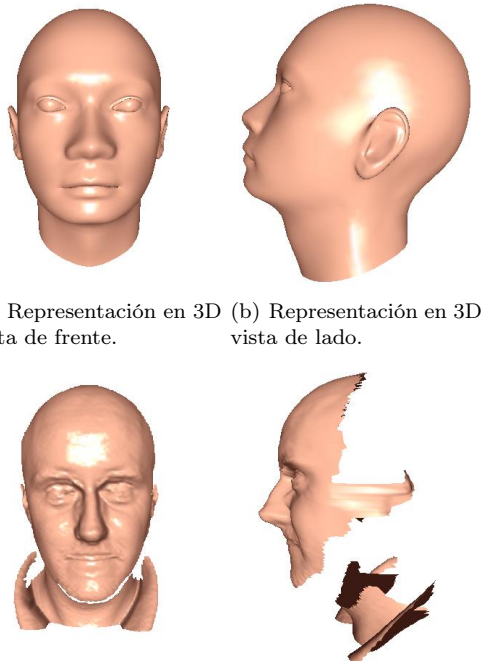
2.1.1. Nube de puntos

Una *nube de puntos* es un conjunto de puntos tridimensionales que describen la superficie de un objeto. Estos puntos p_i no siguen ningún orden y están representados por un vector que además de la información espacial (coordenadas x, y, z) puede contener información de color (valor rgb).

$$p_i = \{x_i, y_i, z_i, r_i, g_i, b_i\}\ ,\ 1 \leq i \leq n \tag{2.1}$$

La principal desventaja de esta representación es que carece de información de vecindades entre puntos, por lo que no se puede saber qué puntos están conectados entre sí. Sin esta información de vecindades se necesita una gran cantidad de puntos para aproximar la superficie real del objeto. En la figura 2.2(a) se puede ver un ejemplo de esta representación.

[1] Casi todas las caras en 3D mostradas en este trabajo, a excepción de la cara base [60], pertenecen a la base de datos GavabDB [48], descrita en el Apéndice A.



(a) Representación en 3D vista de frente.

(b) Representación en 3D vista de lado.

(c) Representación en 2.5D vista de frente.

(d) Representación en 2.5D vista de lado.

Figura 2.1: Diferencias entre representaciones en 3D y 2.5D. La representación en 2.5D se puede considerar como una representación parcial en 3D (2.1(a) y 2.1(b) disponible en [60], 2.1(c) y 2.1(d) disponible en [48]).

2.1.2. Modelo poligonal

El *modelo poligonal* además de la lista de n puntos tridimensionales contiene una lista de m polígonos donde se almacena la información de las vecindades. Cada polígono f_j está formado por un número variable (c_j) de vértices v que corresponden a los puntos tridimensionales originales:

$$p_i = \{x_i, y_i, z_i, r_i, g_i, b_i\}\ ,\ 1 \leq i \leq n \tag{2.2}$$

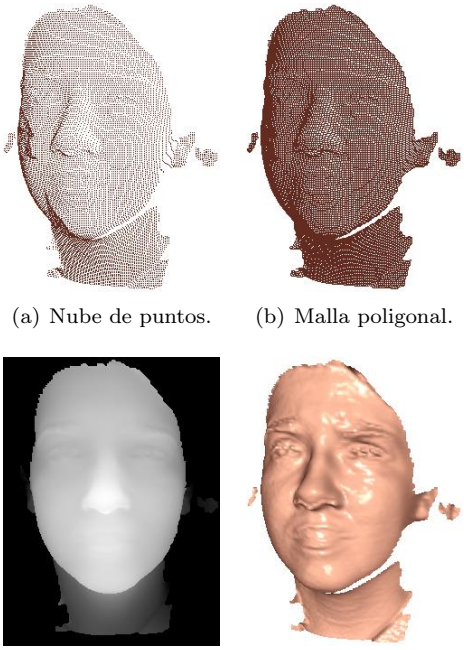
$$f_j = \left\{v_1, \ldots v_{c_j}\right\}\ ,\ 1 \leq j \leq m,\ c_j = \textit{número de vértices} \tag{2.3}$$

La ventaja del modelo poligonal sobre la nube de puntos radica en que cada polígono se puede descomponer en planos triangulares para representar un pedazo de la superficie total y por lo tanto se necesitarán menos puntos para representar el objeto (figura 2.2(b) y figura 2.2(d)).

2.1.3. Imagen de rango

Una *imagen de rango*, también conocida como *mapa de profundidad* o *imagen de intensidad*, es una imagen en 2D donde las intensidades de color en cada pixel indican

la distancia del sensor a la superficie del objeto representado. En la figura 2.2(c) los pixeles más claros, como los de la punta de la nariz, corresponden a puntos cercanos al sensor, mientras que los pixeles más oscuros corresponden a las zonas más alejadas.



(a) Nube de puntos. (b) Malla poligonal.

(c) Imagen de rango. (d) Superficie.

Figura 2.2: Representaciones en 3D del mismo modelo.

2.2. Técnicas de adquisición de información en 3D

Una parte muy importante en los sistemas de reconocimiento facial en 3D es la etapa de adquisición de la información tridimensional de la cara pues básicamente todo el sistema depende de esta información, a tal grado, que los avances en aplicaciones prácticas de reconocimiento facial en 3D están íntimamente ligados a los avances en tecnología de sensores tridimensionales.

Actualmente, los sensores tridimensionales se pueden clasificar en tres categorías [7], [4], [17]:

- **Visión estereoscópica:** Se basa en el principio del sistema visual humano, utilizando dos cámaras calibradas para capturar la imagen del objeto. Se resuelve el problema de correspondencia entre los puntos de las dos imágenes y por triangulación se obtiene la profundidad de cada punto. Aunque este método es barato y fácil de usar, el cálculo de la correspondencia suele tardar mucho y

puede arrojar resultados incorrectos ya que un punto de una imagen puede tener correspondencia con varios puntos de la otra imagen.

- **Proyección de luz blanca:** Como su nombre lo indica, proyecta un patrón de luz (comúnmente franjas) sobre el objeto. El sensor está conformado de un proyector de franjas y un sensor de luz (normalmente una cámara digital) que captura la imagen, donde la distorsión en el patrón de franjas es utilizada para calcular la profundidad de cada punto. Esta técnica es más rápida y permite capturar la información de color al mismo tiempo.

- **Sensores laser:** Aquí se sustituye la luz blanca por un laser (luz coherente) para proyectar un punto, una línea o múltiples patrones sobre el objeto. El principio sigue siendo el mismo: triangulación entre el laser, el sensor de luz y el punto del objeto. Cabe mencionar que se debe utilizar un laser que no sea dañino para los ojos. Una desventaja de esta técnica es que debido a que el dispositivo tiene que moverse alrededor del objeto para que el laser haga el barrido completo, el más mínimo movimiento puede generar errores en los datos.

2.2.1. Limitantes y retos de la tecnología actual

Como hemos mencionado anteriormente, la información en 3D, estrictamente hablando, no resulta afectada por las condiciones de iluminación. Una superficie tridimensional sigue siendo la misma independientemente de la forma en que se esté iluminando, cosa que no ocurre en imágenes en 2D donde la iluminación puede afectar severamente el color de la imagen. Puesto que el sensor con el que se obtiene la información tridimensional, sea de luz proyectada o de visión estereoscópica, en algún momento hace uso de imágenes en 2D, inherentemente se verá afectado por la iluminación y, en consecuencia, podemos decir que *los cambios en la iluminación de una superficie tridimensional pueden afectar la lectura hecha por el sensor al momento de la captura* [7], tal como se muestra en la figura 2.3.

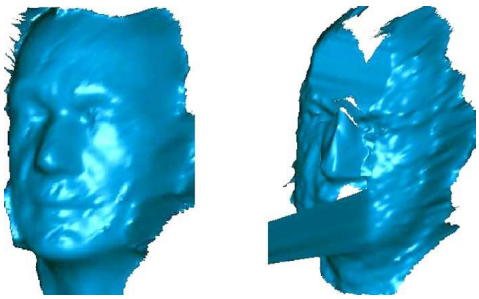
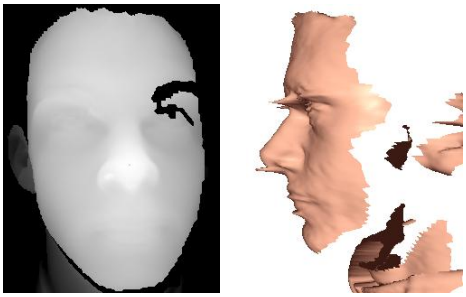


Figura 2.3: Modelo de la misma persona bajo diferentes condiciones de iluminación [7].

Sin importar el tipo de sensor que se esté usando para capturar la superficie facial, es común que aparezcan errores de captura, como *agujeros* y *picos*, en los globos

oculares y en regiones con vello, como las cejas, el bigote o la barba. Un agujero surge cuando el sensor no logra capturar los datos, dejando una región sin información (figura 2.4(a)). Un pico es un error en la medición, resultado de una inter-reflexión en un sensor de luz proyectada, o de un error de correspondencia en un sensor de visión estereoscópica (figura 2.4(b)).



(a) Agujero. (b) Pico.

Figura 2.4: Errores en la captura de superficies en 3D. 2.4(a): La región negra en el ojo y la ceja es un agujero. 2.4(b): Picos en las regiones de la nariz, las cejas y el labio inferior.

Otro factor importante a considerar es si el sensor es *activo* o *pasivo*, o dicho de otra forma, si el sensor proyecta o no luz hacia la escena. Dependiendo del tiempo que tarde un sensor activo en capturar la superficie facial surgirán errores debido al movimiento de la persona, por lo que es deseable que el sensor tarde poco tiempo en el proceso de captura; además, este tipo de sensores típicamente experimentan más problemas en regiones con vello facial. Los sensores pasivos por su parte, a menudo no logran capturar el suficiente número de puntos en la superficie facial, sobre todo en regiones donde no hay mucha textura, y generan superficies muy suavizadas en esos casos.

La profundidad de campo[2] y la resolución son otras limitantes muy importantes de los sensores de hoy en día, por lo que es necesario manejar campos de profundidad más grandes con resoluciones menores a $1mm$ [7].

A pesar de todas estas limitantes, la tecnología de sensores tridimensionales ya es lo suficientemente buena como para hacer investigación en sistemas de reconocimiento facial en 3D. Sin embargo, para contar con aplicaciones comerciales y sistemas de reconocimiento prácticos aún hay que avanzar un poco más en cuanto a sensores más robustos a los cambios de iluminación, con pocos errores de captura y que requieran menos cooperación del usuario.

[2]La profundidad de campo es el rango de distancias que logra captar un sensor.

2.3. Métricas de desempeño

Antes de revisar los avances realizados en el campo de reconocimiento facial en 3D es conveniente hacer una breve descripción de las métricas utilizadas para ilustrar su desempeño, de manera que se puedan hacer las comparaciones pertinentes entre los distintos métodos y resultados.

2.3.1. Curvas CMC

Las curvas CMC (*Cumulative Match Characteristic*) se usan para medir el desempeño con *conjuntos de prueba cerrados* en donde las muestras siempre pertenecen a algún usuario de la galería, es decir, no hay personas extrañas al sistema, todos son usuarios registrados.

Cuando una prueba p_j llega al sistema, ésta se compara contra toda la galería $\mathscr{G}$. Esta comparación entre la prueba p_j y una muestra g_i de $\mathscr{G}$ produce un *valor de semejanza* s_{ij}. Valores grandes de semejanza indican que las dos muestras en comparación son muy parecidas (una métrica de distancia se puede convertir a valor de semejanza tomando su valor negado). El valor de semejanza s_{ij} será un *acierto* (*match*) si p_j y g_i pertenecen a la misma persona, en caso contrario será un *no-acierto* (*nonmatch*). Denotemos con s_{*j} al valor de semejanza acertado para la prueba p_j. Para propósitos de identificación, se ordenan en forma descendente todos los valores de semejanza entre p_j y las muestras de $\mathscr{G}$. p_j tendrá *grado* (o *rank*) n si s_{*j} es el n-ésimo valor de semejanza más grande. Esto se denota como $rank(p_j) = n$. El *Rank 1* es conocido como *top match*.

Supongamos que

$$C\left(n\right) = \left|\{p_j : rank(p_j) \leq n\}\right| \tag{2.4}$$

es el número de pruebas con rank n o menor. La tasa de identificación de rank n es:

$$P_I\left(n\right) = \frac{\left|C\left(n\right)\right|}{\left|\mathscr{G}\right|} \tag{2.5}$$

La curva CMC muestra $P_I\left(n\right)$ como una función de rank n. En este tipo de curvas el *Rank 1* es el valor más utilizado para mostrar la tasa de reconocimiento de un sistema. En la figura 2.5 se puede ver una típica curva CMC.

2.3.2. Curvas ROC

Las curvas ROC (*Receiver Operator Characteristic*) son más utilizadas en pruebas de verificación y cuando se trabaja con *conjuntos de prueba abiertos*, donde el conjunto de prueba se divide en dos subconjuntos, el primer subconjunto ($P_{\mathscr{G}}$) contiene muestras de usuarios registrados en la galería $\mathscr{G}$, mientras que el segundo ($P_{\mathscr{N}}$) contiene muestras de personas que no aparecen en la galería.

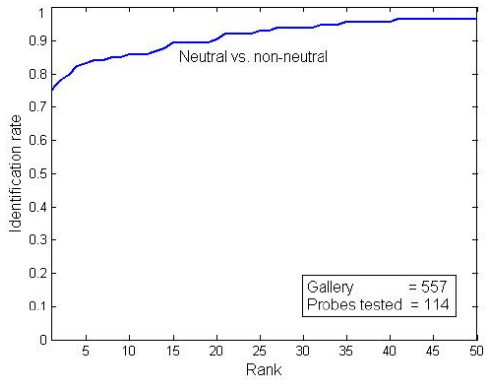


Figura 2.5: Ejemplo de curva CMC [46].

En una verificación, se presenta una prueba y el sistema la compara contra la muestra de la galería que corresponde a la identidad del individuo que se está demandando. El sistema acepta que la muestra de prueba pertenece a la identidad demandada si el valor de semejanza es mayor que cierto umbral de operación τ. La tasa de verificación se calcula con los valores de semejanza acertados entre $\mathscr{G}$ y $P_{\mathscr{G}}$, es decir:

$$P_V(\tau) = \frac{|\{p_j : s_{ij} \geq \tau, id(g_i) = id(p_j)\}|}{|P_{\mathscr{G}}|} \tag{2.6}$$

mientras que la tasa de *falsa alarma* o *falsos aceptados* (impostores aceptados incorrectamente por el sistema) se calcula con los valores de semejanza no acertados entre $\mathscr{G}$ y $P_{\mathscr{N}}$:

$$P_{FA}(\tau) = \frac{|\{s_{ij} : s_{ij} \geq \tau\}|}{|P_{\mathscr{N}}|\,|\mathscr{G}|} \tag{2.7}$$

Estos valores de semejanza no acertados son llamados también *impostores reales* puesto que las personas en $P_{\mathscr{N}}$ no aparecen en la galería.

La curva ROC muestra la relación entre la tasa de verificación P_V y la tasa de falsa alarma P_{FA}. Para mostrar la tasa de reconocimiento de un sistema, se reporta la tasa de verificación correspondiente a un cierto valor de la tasa de falsa alarma, como se muestra en la figura 2.6. También es común utilizar el *EER* (*Equal Error Rate*), que es el punto donde la tasa de falsos rechazados equivale a la tasa de falsos aceptados.

2.4. Revisión del Estado del Arte

Aunque los primeros trabajos de reconocimiento facial en 3D comenzaron a surgir a principios de los 90's, el trabajo reportado en esta área sigue siendo relativamente pequeño en comparación con otras áreas. En la tabla 2.1 se presenta un breve sumario de los avances en reconocimiento facial en 3D, basados en [7]; para profundizar más

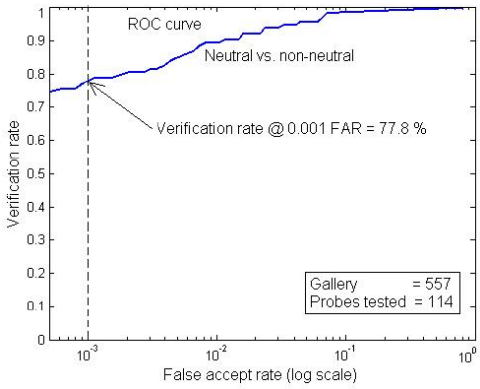


Figura 2.6: Ejemplo de curva ROC [46].

en un algoritmo revise la referencia correspondiente. Los algoritmos están ordenados cronológicamente de acuerdo al año de publicación.

La mayoría de los algoritmos comienzan haciendo una alineación facial por medio de transformaciones rígidas (rotaciones y translaciones) usando el centro de masa, la punta de la nariz, los ojos, o ajustando un plano a la cara y alineándolo con respecto a los ejes coordenados. Los puntos utilizados se detectan de forma automática o manual.

Como se puede observar en la tabla 2.1, la mayoría de los algoritmos reportan tasas de reconocimiento de 100 % en *Rank 1*, sin embargo, estos resultados deben interpretarse con mucho cuidado al intentar hacer comparaciones directas ya que únicamente [54] y [11] utilizan el mismo conjunto de datos.

El tamaño del conjunto de datos es un factor importante a considerar. Los resultados reportados por Xu [70] nos dan un ejemplo de como puede verse afectado el desempeño del sistema debido al tamaño del conjunto de datos. En un primer experimento con 30 personas en el conjunto de datos, reporta una tasa de reconocimiento de 96 % en *Rank 1*, pero al aumentar el conjunto de datos a 120 personas, la tasa decae hasta un 72 % de reconocimiento en *Rank 1*.

El desempeño de un sistema también depende en gran medida de la dificultad de los datos. Las variaciones en la expresión facial, la pose, la presencia de accesorios (lentes, bufanda, etc.), y el tiempo pasado entre las imágenes de la galería y las pruebas, son factores importantes que afectan el desempeño. De todos los algoritmos mostrados, solo [47], [41], [54], [11] y [8] utilizan conjuntos de datos que incluyen variaciones en pose y/o expresión facial.

Los algoritmos de reconocimiento en 3D que usan representaciones basadas en curvatura, como las imágenes Gaussianas extendidas (EGI: *Extended Gaussian Image*), son incapaces de distinguir entre caras iguales pero de distinto tamaño. Los algoritmos basados en PCA o ICP (*Iterative Closest Point*) pueden manejar el cambio en tamaño, pero tienen problemas con los cambios en expresión facial. La figura 2.7 muestra los resultados de un experimento donde se pusieron 70 personas con expre-

sión normal en la galería. Para la primera prueba se utilizaron imágenes tomadas una semana después, donde las 70 personas estaban sonriendo. En la segunda prueba se tomaron imágenes una semana después a las de la primera prueba; aquí las personas estaban con expresión normal. El reconocimiento se hizo utilizando PCA en 2D y 3D. Los resultados arrojados reflejan que el desempeño en 3D se ve severamente afectado por el cambio de expresión.

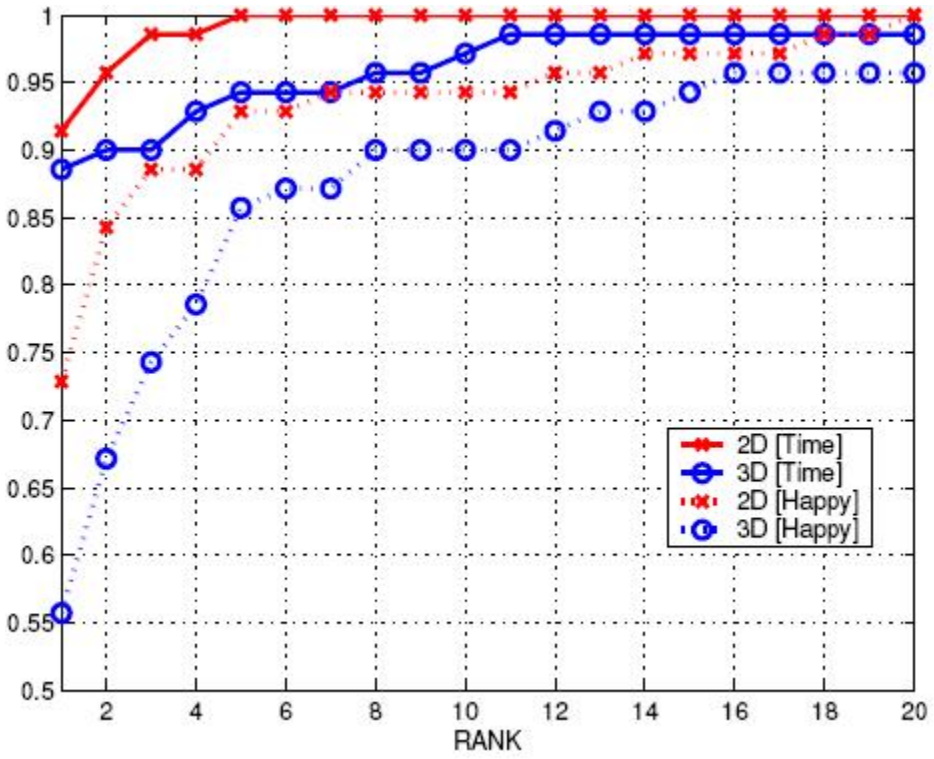


Figura 2.7: Efectos del cambio de expresión en la tasa de reconocimiento en 2D y 3D [7].

Para llevar a cabo comparaciones entre algoritmos y validaciones de experimentos correctas es necesario contar con conjuntos de datos que tengan las siguientes características:

- Un gran número de personas de distintos grupos étnicos.
- Imágenes de la misma persona tomadas en distintos lapsos de tiempo.
- Imágenes de la misma persona con cambios sustanciales en expresión facial.
- Alta resolución espacial ($1mm$ o mejor).
- Pocos errores (agujeros y picos) causados por los sensores.
- Amplia difusión y facilidad de adquisición dentro de la comunidad científica.

Hasta ahora, sólo el FRGC (*Face Recognition Grand Challenge*) [56] ha hecho esfuerzos significativos por proveer de todas estas características en un conjunto de datos, con el propósito de estandarizar la metodología de comparación entre los distintos algoritmos reportados por la comunidad científica. Sólo mediante el uso común de conjuntos de datos y metodologías estandarizadas se lograrán ver los avances reales en el campo de reconocimiento facial en 3D.

Autor y año	**Conjunto de datos**	**Representación**	**Algoritmo**	**Desempeño**
Cartoux [9], 1989	5 personas, 18 imágenes	Perfil, superficie	Distancia mínima	100 %
Lee [34], 1990	6 personas, 6 imágenes	EGI	Correlación	–
Gordon [22], 1992	26 galería, 24 prueba	Vector de características	Vector más cercano	100 %
Nagamine [49], 1992	16 personas, 160 imágenes	Múltiples perfiles	Vector más cercano	100 %
Achermann [1], 1997	24 personas, 240 imágenes	Imagen de rango	PCA, HMM (*Hidden Markov Model*)	100 %
Tanaka [63], 1998	37 personas, 37 imágenes	EGI	Correlación	100 %
Achermann [2], 2000	24 personas, 240 imágenes	Nube de puntos	Distancia de Hausdorff	100 %
Chua [12], 2000	6 personas, 24 imágenes	Nube de puntos	"*Point Signature*"	100 %
Hesher [26], 2003	37 personas, 222 imágenes	Imagen de rango	PCA	97 %
Lee [36], 2003	35 personas, 70 imágenes	Vector de características	Vector más cercano	94 % (*Rank 5*)
Medioni [45], 2003	100 personas, 700 imágenes	Nube de puntos	ICP	98 %
Moreno [47], 2003	60 personas, 420 imágenes	Vector de características	Vector más cercano	78 %
Pan [51], 2003	30 personas, 360 imágenes	Nube de puntos, imagen de rango	Distancia de Hausdorff, PCA	3–5 % EER 5–7 % EER
Lee [37], 2004	42 personas, 84 imágenes	Rango, curvatura	Distancia de Hausdorff ponderada	98 %

Lu [41], 2004	18 personas, 113 imágenes	Nube de puntos	ICP	96 %
Russ [58], 2004	200 personas, 468 imágenes (FRGC v1)	Imagen de rango	Distancia de Hausdorff	98 % verificación
Xu [70], 2004	120 (30) personas, 720 imágenes	Nube de puntos + Vector de características	Distancia mínima	96 % con 30 72 % con 120
Bronstein [8], 2005	30 personas, 220 imágenes	Nube de puntos	"Formas canónicas"	100 %
Chang [11], 2005	466 personas, 4007 imágenes (FRGC v2)	Nube de puntos	Multi-ICP	92 %
Gökberk [21], 2005	106 personas, 579 imágenes	Múltiple	Múltiple	99 %
Lee [38], 2005	100 personas, 200 imágenes	Vector de características	SVM (*Support Vector Machines*)	96 %
Lu [38], 2005	100 personas, 196 pruebas	Malla	ICP + TPS (*Thin Plate Spline*)	89 %
Pan [52], 2005	276 personas, 943 imágenes (FRGC v1)	Imagen de rango	PCA	95 %, 3 % EER
Passalis [54], 2005	466 personas, 4007 imágenes (FRGC v2)	Malla	Modelo deformable	90 %
Russ [59], 2005	200 personas, 398 imágenes (FRGC v1)	Imagen de rango	Distancia de Hausdorff	98.5 %

Tabla 2.1: Algoritmos de reconocimiento facial en 3D [7].

Capítulo 3

Síntesis de expresiones faciales

La animación facial ha recibido mucha atención en los últimos años debido al incremento en el poder computacional, la creciente demanda de personajes virtuales para la industria del cine y los videojuegos, los sistemas de videoconferencia, etc. Sin embargo, esta área no es nueva, ya que desde la década de los 70's los investigadores han tratado de generar y animar modelos faciales computarizados [53].

En este capítulo se detalla el método utilizado para sintetizar expresiones faciales. A groso modo, el método toma una cara base en 3D en la cuál se definen los músculos faciales y las distintas capas de la estructura facial para después resolver un problema de deformación utilizando el método de los elementos finitos. Cabe mencionar que existen otros métodos de animación que no serán revisados puesto que el objetivo principal de este trabajo es el de reconocimiento facial. Para profundizar más sobre el tema de animación facial revise las referencias [32], [55], [30] y [50].

3.1. Anatomía facial

Antes de describir el método de síntesis de expresiones faciales conviene hacer una revisión de la estructura anatómica de la cara para conocer mejor cómo interactúan los distintos componentes faciales en la realización de un gesto. Esta sección se basa en el material de las referencias [32], [55], [69] y [64].

3.1.1. La piel

La piel es un sistema complejo formado por varias estructuras que trabajan juntas para cumplir con varias funciones, entre las que se encuentran la regulación de la temperatura corporal, la protección contra agentes externos como bacterias y sustancias tóxicas, etc.; además es el órgano sensorial más grande, sensible a la temperatura, al tacto y al dolor. La piel también juega un papel importante en la comunicación por medio de la expresión facial, la textura, el color y el olor.

La piel está hecha de distintas capas con densidad y grosor variable, dependiendo de su función. La primer capa es la *epidermis*, una capa superficial protectora formada por células muertas que mide aproximadamente 1/10 del grosor total de la piel y está compuesta principalmente de keratina. Después se encuentra la *dermis* que está formada de fibras de colágeno y elastina inmersas en una sustancia gelatinosa. Finalmente, la *hipodermis* es la capa inferior; es una capa aislante formada por tejido graso que permite que la piel se deslice fácilmente por la fascia que conecta la superficie muscular a la piel.

Las propiedades biomecánicas de la piel se deben principalmente a la dermis. Se ha mostrado experimentalmente que la piel tiene una curva esfuerzo-deformación altamente no-lineal (figura 3.1): bajo poco esfuerzo y a medida que las fibras de colágeno comienzan a desenrollarse en la dirección del esfuerzo, la dermis ofrece poca resistencia al estiramiento, pero al aumentar el esfuerzo, las fibras totalmente desenrolladas presentan mayor resistencia al estiramiento. Por su parte, las fibras de elastina actúan como resortes que regresan las fibras de colágeno a su estado original cuando no hay esfuerzo. La incompresibilidad del agua en las células de grasa y el movimiento retardado de las fibras en este tejido provocan que la piel exhiba un comportamiento viscoelástico, es decir, se comporta en parte como un sólido elástico, y en parte como un fluido viscoso. Es este comportamiento no-lineal lo que dificulta la modelación computarizada de la piel.

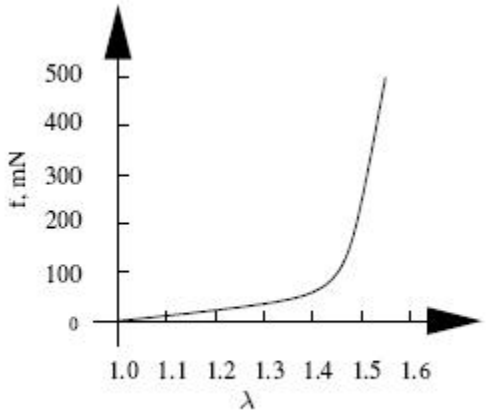


Figura 3.1: Curva "Esfuerzo vs. Deformación" de la piel [32].

3.1.2. El cráneo

El cráneo es la estructura ósea que brinda forma y estabilidad a la cabeza. Está formado por varios huesos, siendo la mandíbula la única estructura unida que rota sobre un eje horizontal cercano a la oreja. Anatómicamente, el cráneo se divide en dos partes: el *neurocráneo* que alberga al cerebro, y la cara o *viscerocráneo*. En la figura

3.2 se pueden ver los huesos del cráneo humano. A excepción de la nariz y la parte baja de las mejillas, la cara sigue el contorno trazado por del cráneo.

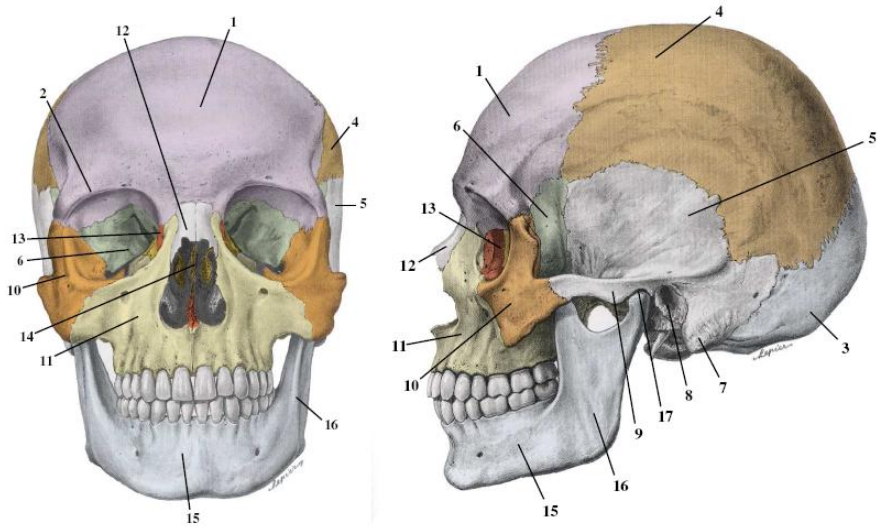


Figura 3.2: Huesos del cráneo humano: 1.frontal, 2.supraorbital, 3.occipital, 4.parietal, 5.temporal, 6.esfenoides, 7.mastoides, 8.agujero del oído, 9.zigoma, 10.zigomático, 11.maxilar, 12.nasal, 13.lacrimal, 14.etmoides, 15.mandíbula, 16.rama mandibular, 17.unión temporomandibular [61].

3.1.3. Los músculos

Las gran variedad en expresiones faciales es posible gracias a los músculos que se encuentran entre la piel y el cráneo (figura 3.3). Se pueden identificar más de 50 músculos faciales, los cuáles se clasifican en 2 categorías: los músculos de la expresión y los músculos de la masticación. La mayoría de los músculos faciales están pegados a la estructura ósea por un lado e insertados a la piel por el otro extremo, a excepción de los orbiculares de los ojos y la boca que solamente están pegados a la piel. Cuando un músculo se contrae, jala de la piel donde está insertado hacia el lugar donde éste emerge en el cráneo, provocando distorsiones en la piel cercana al área de inserción.

Los músculos de la expresión se clasifican de acuerdo a su región de influencia en:

- **Músculos del cuero cabelludo:** Los músculos del cuero cabelludo forman el *epicráneo*, que abarca toda la parte superior del cráneo y se encarga de mover la piel de la cabeza. Sus partes más grandes son el *occipital*, localizado en la parte posterior de la cabeza, y el *frontal* que cubre la frente. Estos dos músculos están conectados por la aponeurosis epicraneal, la cuál también es el origen del músculo *temporoparietal*. El occipital se origina detrás de los oídos, mientras que el frontal no está pegado al hueso, sino que se inserta en la piel de la frente y la región de las cejas, y se encarga de levantar las cejas y los párpados, así

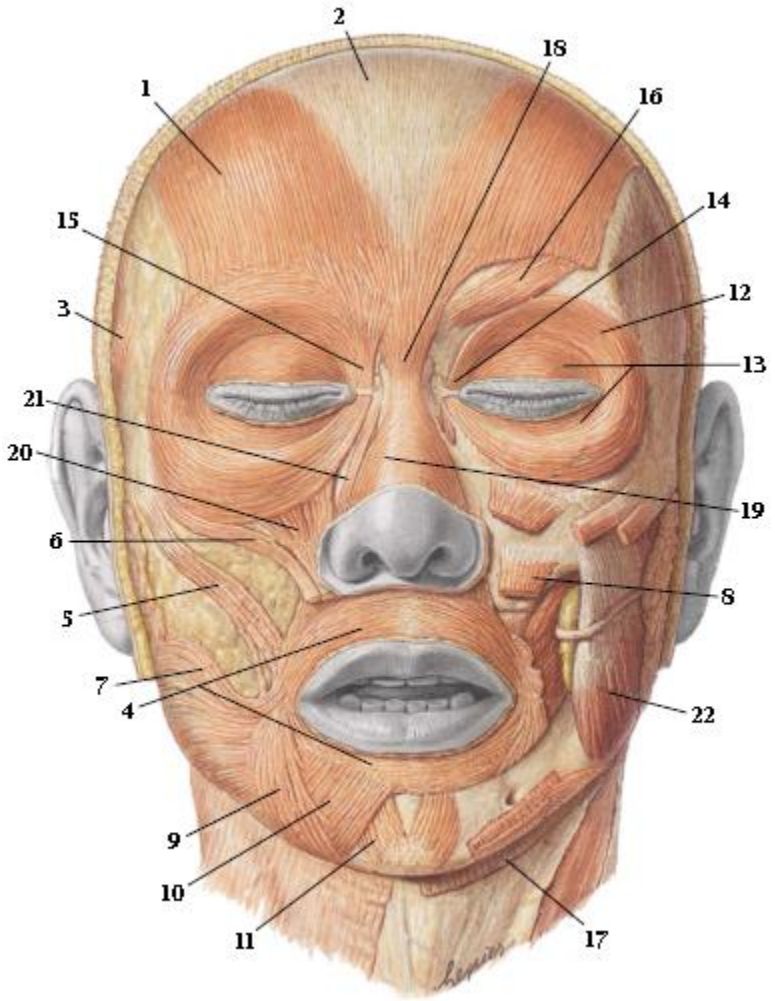


Figura 3.3: Músculos faciales: 1.frontal, 2.aponeurosis epicraneal, 3.temporoparietal, 4.orbicularis oris, 5.zigomático mayor, 6.zigomático menor, 7.risorio, 8.levator anguli oris, 9.triangularis, 10.depresor labii inferioris, 11.mentalis, 12.orbicularis oculi pars orbitalis, 13.orbicularis oculi pars palpebralis, 14.ligamento palpebral, 15.depresor supercilii, 16.corrugator supercilii, 17.platysma, 18.procerus, 19.nasalis, 20.levator labii superioris, 21.levator labii superioris alaeque nasi, 22.masetero [61].

como de producir las arrugas de la frente. El temporoparietal se extiende desde la aponeurosis epicraneal hasta la parte superior de los oídos, a cada lado de la cabeza. Finalmente, los músculos auriculares a un lado de los temporoparietales se encargan de mover la oreja.

- **Músculos oculares:** El músculo que se encuentra alrededor del ojo y sirve para abrirlo y cerrarlo se conoce como *orbicularis oculi*, está formado por el *pars orbitalis* que circula la órbita ocular, y el *pars palpebralis* en los párpados superior e inferior. En la esquina interna del ojo se encuentra el *ligamento palpebral* que

interrumpe las fibras circulares. El *depresor supercilii* se encuentra en la región superior del párpado y llega hasta la ceja. Por último, el *corrugator supercilii* penetra el orbicularis oculi y el frontal, y se encarga de jalar la esquina de la ceja hacia arriba y hacia abajo provocando arrugas en la parte superior de la nariz.

- **Músculos nasales:** El *procerus* cubre la parte inferior del hueso nasal y la parte alta del cartílago nasal, se inserta en la frente, entre las cejas, y jala la piel verticalmente produciendo una arruga horizontal en el puente nasal. El *nasalis* cubre el ala nasal y se encarga de abrir y cerrar el nostril. El *levator labii superioris alaeque nasi* se origina en el lado interno de la órbita ocular y se inserta en la piel del labio superior y el ala nasal, si se contrae fuertemente se produce un surco a un lado de la nariz. El *levator labii superioris* se origina en el maxilar, bajo la órbita ocular, y levanta la parte lateral del labio superior.

- **Músculos de la boca:** El *orbicularis oris* circula la boca y se extiende desde la parte inferior de la nariz hasta las mejillas; es una estructura compleja que actúa como un esfínter que permite abrir y cerrar la boca, además de que permite la inserción de otros músculos faciales en esta región. Los músculos *zigomáticos* (mayor y menor) van del hueso zigomático hasta las esquinas de la boca; estos músculos ayudan a crear las expresiones de sonrisa y felicidad levantando los labios superiores en las esquinas de la boca. El *risorio* es un músculo que jala las esquinas de la boca horizontalmente, no se origina en el hueso, sino en la fascia del músculo masetero. El *levator anguli oris* se inserta al orbicularis oris en la esquina de la boca y contribuye a las expresiones de sonrisa y felicidad jalando hacia arriba la piel, mientras que el *triangularis* hace lo opuesto, jalando la boca hacia abajo. El labio inferior es jalado hacia abajo por el *depresor labii inferioris*, el cuál se origina en la parte inferior de la mandíbula y está insertado al orbicularis oris y a la piel del labio inferior. El *buccinador* es un músculo que corre desde la mejilla hasta la esquina de la boca, contribuye a la acción de chupar e interviene en el proceso de masticación acercando las mejillas a los dientes, encerrando la comida en este pequeño espacio. Finalmente, el *mentalis* cubre la barbilla, se origina debajo de los dientes y se inserta a la piel de la barbilla, al jalar hacia arriba, el músculo eleva y saca el labio inferior.

- **Músculos del cuello:** De todos los músculos del cuello, solo el *platysma* es parte de los músculos faciales. Se origina en la base de la mandíbula y cubre la parte inferior de la quijada extendiéndose hacia abajo, a la parte frontal del cuello. Se encarga de estirar la piel del cuello.

Los músculos de la masticación, como su nombre lo indica, participan en el proceso de masticación y trituración de comida. El *masetero* es un músculo muy fuerte que se origina en el arco zigomático y corre diagonalmente sobre la mejilla para insertarse en el ángulo de la mandíbula. Cuando se contrae, aprieta los dientes, jugando el papel más importante en el proceso de trituración de comida. Los músculos *temporales* también se necesitan para cerrar la boca, se originan en el hueso temporal, corren

a través del arco zigomático y se insertan en la apófisis coronoide de la mandíbula. Los músculos *pterigoideos internos* se localizan a los lados de la mandíbula y ayudan a los maseteros y temporales a levantar la mandíbula para apretar los dientes. En contraste, los músculos *pterigoideos externos* ayudan a abrir la boca y a sacar la mandíbula.

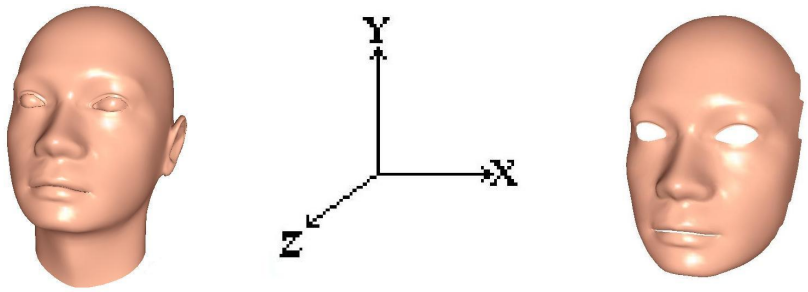
3.1.4. Otros órganos

La apariencia visual de la cara no está determinada solamente por la piel, el cráneo y los músculos. Hay otros órganos, como los ojos y la cavidad oral, que contiene los dientes y la lengua. Sin embargo, como en nuestro trabajo no simularemos estos órganos, sólo hacemos mención de ellos.

3.2. Preproceso de la cara base

Una vez revisados los conceptos necesarios, damos paso a la primera parte del proceso de animación. En esta parte dotaremos a nuestro modelo facial de la estructura anatómica necesaria para poder hacer expresiones. Con la breve descripción de anatomía facial dada en la sección anterior, es evidente que ésta tiene una estructura muy compleja que no intentamos simular con exactitud en este trabajo, sino más bien tratar de reproducir las características que nos permitan crear expresiones faciales con cierto grado de apego a la realidad.

La cara base es un modelo de muestra de la aplicación *FaceGen Modeller*, de *Singular Inversions Inc.*, disponible en http://www.facegen.com/downloads.htm [60]. El modelo original se puede ver en la figura 3.4(a), es una malla con $6,054$ elementos poligonales de 4 vértices ($6,105$ vértices en total). Sin embargo, puesto que sólo nos interesa la región de la cara, se editó la malla original dejando únicamente $2,676$ elementos y $2,777$ vértices, como se muestra en la figura 3.4(c).



(a) Malla original. (b) Sistema de referencia. (c) Malla editada.

Figura 3.4: Cara base.

Para dar grosor a la superficie de la cara base, de manera que simule las distintas capas de la estructura facial, utilizamos el vector unitario normal a cada vértice, multiplicado por un cierto factor, para generar una nueva capa de vértices bajo la capa original. Este proceso se repite dos veces, variando el factor por el que se multiplican los vectores normales, para así generar 3 capas de vértices y 2 capas de elementos hexaédricos de 8 nodos, como se puede ver en la figura 3.5. Como se mostrará más adelante, las capas de vértices tendrán distintas propiedades mecánicas para simular las propiedades del material del que están hechos los distintos componentes: la capa inferior simulará el cráneo, la capa intermedia será el tejido graso y la capa superior será la piel.

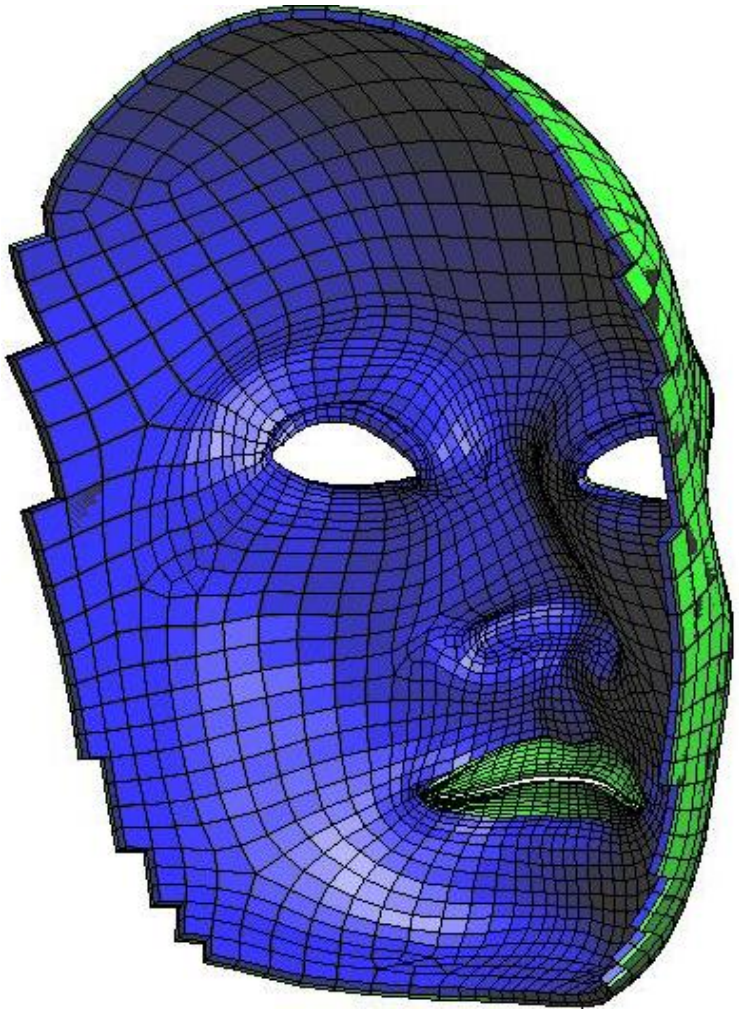
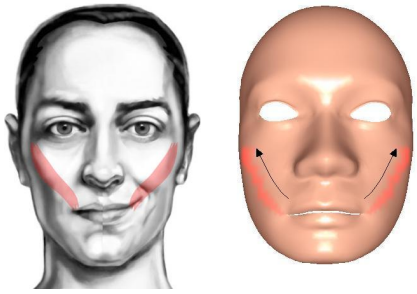


Figura 3.5: Capas de la cara base. En azul se muestra la capa inferior de elementos hexaédricos y en verde la capa superior.

Para dotar de músculos a la cara base, nos apoyamos en la herramienta de aprendizaje de anatomía facial desarrollada por Victoria Contreras Flores, *Artnatomy* [16].

Definimos manualmente cada músculo como un conjunto de vértices sobre los cuáles se ejerce fuerza en una dirección determinada, la cuál depende de la situación y de la forma del músculo. En la figura 3.6 se puede ver un músculo definido en la cara base y su dirección de acción.



(a) Artnatomy. (b) Cara base.

Figura 3.6: Definición de un músculo en la cara base. 3.6(a): El zigomático mayor mostrado en *Artnatomy* [16]. 3.6(b): El zigomático mayor en la cara base y su dirección de acción.

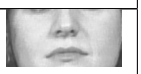
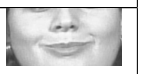
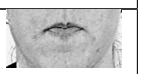
3.3. El FACS

Además de un modelo anatómico facial, la simulación de expresiones faciales requiere también de un mapeo entre la expresión deseada y las activaciones musculares requeridas. El *FACS* (*Facial Action Coding System*) [19] es el método más usado para medir y describir la conducta facial, determinando cómo la contracción de cada músculo facial cambia la apariencia de la cara. El sistema asocia cambios de expresión con la acción de los músculos que la producen, basándose en estudios de anatomía facial. Su meta es establecer un mecanismo confiable para determinar a qué categoría pertenece una expresión facial.

Las medidas del FACS se realizan en *Action Units* (AUs), no músculos, por dos razones. En primer lugar, más de un músculo interviene en una AU; en segundo, los cambios de expresión producidos por un músculo pueden ser separados algunas veces en dos o más AUs para representar acciones independientes de diferentes partes de un mismo músculo. Otro aspecto importante del FACS es que sólo provee información descriptiva, sin incluir el significado psicológico de la expresión.

El FACS tiene 44 AUs de las cuáles 30 están anatómicamente relacionadas con la contracción de un conjunto específico de músculos. Las bases anatómicas de las 14 restantes no están especificadas. Estas 14 AUs se conocen como acciones misceláneas. En la tabla 3.1 se muestran las primeras 30 AUs.

En la herramienta de software desarrollada para este trabajo (ver Apéndice B) se pueden definir expresiones utilizando la representación muscular o también por medio de AUs.

AU	Descripción	Músculo facial	Fig.
1	Elevar parte interior de cejas	Frontalis, pars medialis	
2	Elevar parte exterior de cejas	Frontalis, pars lateralis	
4	Bajar cejas	Corrugator supercilii, Depresor supercilii	
5	Elevar párpado superior	Levator palpebrae superioris	
6	Elevar mejillas	Orbicularis oculi, pars orbitalis	
7	Tensar párpados	Orbicularis oculi, pars palpebralis	
9	Fruncir nariz	Levator labii superioris alaquae nasi	
10	Elevar labio superior	Levator labii superioris	
11	Acentuar pliegue nasolabial	Zigomático menor	
12	Jalar comisuras	Zigomático mayor	
13	Inflar mejillas	Levator anguli oris	
14	Estrechar comisuras	Buccinador	
15	Bajar comisuras	Triangularis	
16	Bajar labio inferior	Depresor labii inferioris	
17	Levantar barbilla	Mentalis	
18	Fruncir labios	Incisivii labii sup. e inf.	
20	Estirar labios	Risorio y platysma	
22	Abrir labios	Orbicularis oris	
23	Apretar labios	Orbicularis oris	
24	Presionar labios	Orbicularis oris	

25	Separar labios	Depresor labii inferioris o mentalis u orbicularis oris	
26	Caída del mentón	Masetero, temporal, pterigoideo interno	
27	Abrir boca	Pterigoideos, digástricos	
28	Succionar labios	Orbicularis oris	
41	Bajar párpados	Levator palpebrae superioris	
42	Entrecerrar ojos	Orbicularis oculi	
43	Cerrar ojos	Levator palpebrae superioris; orbicularis oculi, pars palpebralis	
44	Estrechar párpados	Orbicularis oculi, pars palpebralis	
45	Parpadear	Levator palpebrae superioris; orbicularis oculi, pars palpebralis	
46	Guiñar	Levator palpebrae superioris; orbicularis oculi, pars palpebralis	

Tabla 3.1: FACS Action Units.

3.4. El método de los elementos finitos

En esta sección se dará una descripción del funcionamiento del *método de los elementos finitos* (*MEF* o *FEM* por sus siglas en inglés) aplicado al problema de elasticidad, ya que desde el punto de vista físico, el problema de simulación de expresiones faciales puede ser visto como un problema de mecánica de sólidos. La idea básica del método consiste en dividir el dominio, sobre el cuál están definidas las ecuaciones que caracterizan el comportamiento físico del problema, en muchos elementos (de ahí su nombre), para después realizar los cálculos en cada uno de estos elementos. La sección está basada en las referencias [74] y [40].

3.4.1. El problema de elasticidad

Consideremos un sólido elástico tridimensional con volumen V y superficie S, como se muestra en la figura 3.7. La superficie del sólido se divide en dos tipos: una superficie en la cuál las fuerzas externas están prescritas (S_f); y una superficie en la cuál los desplazamientos están prescritos (S_d). El sólido también puede estar sujeto a cargas en el cuerpo f_b y a cargas en la superficie f_s.

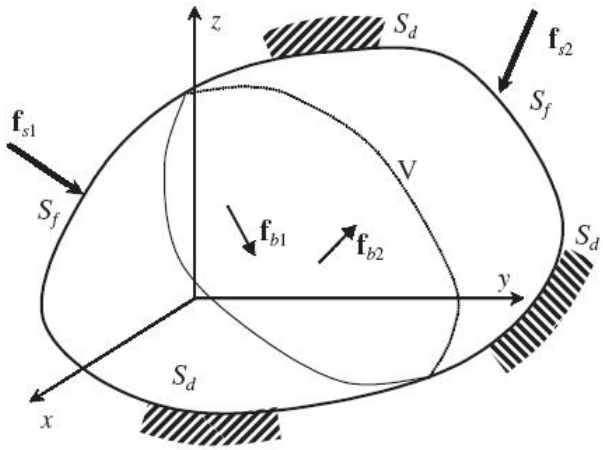


Figura 3.7: Sólido sujeto a fuerzas aplicadas dentro del sólido y sobre la superficie del sólido [40].

En cualquier punto del sólido, existirán 6 componentes de esfuerzo:

$$\sigma^T = \{\sigma_x \ \sigma_y \ \sigma_z \ \tau_{xy} \ \tau_{xz} \ \tau_{yz}\} \tag{3.1}$$

donde $\sigma_x, \sigma_y, \sigma_z$ son los esfuerzos normales sobre los ejes de referencia y $\tau_{xy}, \tau_{xz}, \tau_{yz}$ son los esfuerzos tangenciales. Del mismo modo, existirán también 6 componentes de deformación que se corresponderán con los componentes de esfuerzo:

$$\epsilon^T = \{\epsilon_x \ \epsilon_y \ \epsilon_z \ \gamma_{xy} \ \gamma_{xz} \ \gamma_{yz}\} \tag{3.2}$$

La deformación es el cambio de desplazamiento por unidad de longitud, y por tanto los componentes de deformación se pueden obtener a partir de las derivadas de los desplazamientos:

$$\epsilon_x = \frac{\partial u}{\partial x}; \ \epsilon_y = \frac{\partial v}{\partial y}; \ \epsilon_z = \frac{\partial w}{\partial z};$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}; \ \gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}; \ \gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \tag{3.3}$$

donde u, v, w son los componentes de desplazamiento en las direcciones x, y, z, respectivamente. Las seis relaciones deformación-desplazamiento en 3.3 se pueden rescribir en forma matricial como:

$$\epsilon = LU \tag{3.4}$$

donde U es el vector de desplazamientos y tiene la forma

$$U = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.5}$$

y L es la matriz de derivadas parciales obtenida por simple inspección de 3.3:

$$L = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \\ \partial/\partial y & \partial/\partial x & 0 \end{bmatrix} \tag{3.6}$$

La relación entre esfuerzos y deformaciones en un material sólido se deduce de la ecuación constitutiva o *Ley de Hooke*, que se enuncia de la siguiente forma:

$$\sigma = D\epsilon \tag{3.7}$$

donde D es una matriz de constantes de material:

$$D = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{(1-\nu)} & \frac{\nu}{(1-\nu)} & 0 & 0 & 0 \\ \frac{\nu}{(1-\nu)} & 1 & \frac{\nu}{(1-\nu)} & 0 & 0 & 0 \\ \frac{\nu}{(1-\nu)} & \frac{\nu}{(1-\nu)} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2(1-\nu)} \end{bmatrix} \tag{3.8}$$

en donde E y ν son el módulo de Young y el coeficiente de Poisson, respectivamente.

Haciendo uso del *Principio de los Trabajos Virtuales* (PTV) y teniendo en cuenta los esfuerzos y deformaciones que contribuyen al trabajo virtual del sólido, llegamos a la expresión integral de equilibrio en el problema de elasticidad:

$$\int_V \delta\epsilon^T \sigma dV = \int_V \delta U^T b dV + \int_s \delta U^T t ds + \sum_i \delta U_i^T q_i \tag{3.9}$$

donde V es el volumen de la región, b es el vector de fuerzas másicas por unidad de volumen, t es la carga vectorial en la superficie y q es la fuerza puntual:

$$b = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} ; \; t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} ; \; q_i = \begin{bmatrix} q_{xi} \\ q_{yi} \\ q_{zi} \end{bmatrix} \tag{3.10}$$

La ecuación 3.9 es el punto de partida para obtener las ecuaciones de discretización como se verá a continuación.

3.4.2. Formulación con elementos isoparamétricos

El término isoparamétrico se refiere a la utilización de las mismas funciones de forma para interpolar la geometría y los desplazamientos en un elemento. En este

apartado describiremos el problema anterior en términos de elementos isoparamétricos.

Un elemento finito típico, e, está definido por varios nodos y líneas rectas que conectan dichos nodos. Un desplazamiento u en cualquier punto dentro del elemento se puede aproximar como:

$$u \approx \hat{u} = \sum_k N_k a_k^e = [N_1 \ N_2 \ \ldots] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \end{bmatrix} = N a^e \tag{3.11}$$

en donde los componentes de N son las funciones de forma y a^e representa una lista de desplazamientos nodales para un elemento particular. En el problema que estamos analizando $a_i = [u_i, v_i, w_i]^T$

Las funciones de forma N tienen que escogerse de manera que regresen el desplazamiento nodal apropiado al sustituirse en la ecuación 3.11. En general:

$$N_i\left(x_i, y_i, z_i\right) = I \ \ (Identidad) \tag{3.12}$$

mientras que

$$N_i\left(x_j, y_j, z_j\right) = N_i\left(x_m, y_m, z_m\right) = 0 \tag{3.13}$$

dicho de otra manera, la función $N_i = 1$ para el nodo x_i, y_i, z_i y cero en cualquier otro nodo del elemento. En todos los libros de elementos finitos se dan las funciones de forma para distintos tipos de elementos, por lo que no las incluiremos aquí.

Con los desplazamientos conocidos en todos los puntos de un elemento, podemos determinar la deformación en cualquier punto de acuerdo a la ecuación 3.4, por lo tanto:

$$\epsilon = \hat{\epsilon} = Ba \tag{3.14}$$

con

$$B = LN \tag{3.15}$$

Una vez que se tienen las funciones de forma, la matriz B se puede obtener fácilmente.

Retomando la ecuación 3.9, podemos hacer las siguientes sustituciones:

$$\delta U = N \delta a^e$$

$$\delta \epsilon = B \delta a^e \tag{3.16}$$

El trabajo hecho por las fuerzas nodales es igual a la suma de los productos de los componentes individuales de la fuerza y sus correspondientes desplazamientos:

$$\delta a^{eT} q^e \tag{3.17}$$

De la misma manera, el trabajo hecho por los esfuerzos, las fuerzas másicas y las fuerzas de superficie es:

$$\delta\epsilon^T\sigma - \delta U^T b - \delta U^T t \tag{3.18}$$

o

$$\delta a^T \left(B^T\sigma - N^T b - N^T t\right) \tag{3.19}$$

Igualando el trabajo externo con el trabajo interno obtenido al integrar sobre el elemento, e, tenemos:

$$\delta a^{eT} q^e = \delta a^{eT} \left(\int_{V^e} B^T\sigma dV - \int_{V^e} N^T b dV - \int_{s^e} N^T t ds\right) \tag{3.20}$$

Y puesto que esta relación es válida para cualquier valor de desplazamiento virtual, los multiplicadores deben ser iguales, entonces:

$$q^e = \int_{V^e} B^T\sigma dV - \int_{V^e} N^T b dV - \int_{s^e} N^T t ds \tag{3.21}$$

Esta expresión es válida en general para cualquier relación esfuerzo-deformación, y la podemos escribir como:

$$q^e = K^e a^e + f^e \tag{3.22}$$

donde

$$K^e = \int_{V^e} B^T D B dV \tag{3.23}$$

y

$$f^e = -\int_{V^e} N^T b dV - \int_{s^e} N^T t ds \tag{3.24}$$

En este punto ya es posible aplicar técnicas de integración numérica para resolver la ecuación elemental (3.22). Para resolver el sistema completo se ensambla la serie de matrices y vectores de cada ecuación elemental, lo que conduce a un sistema de ecuaciones lineales cuya solución nos dará los desplazamientos en los nodos de cada elemento. Una vez que se conocen los desplazamientos en los nodos, se pueden conocer los campos de desplazamientos y los esfuerzos en cualquier punto de los elementos.

3.5. Implementación y resultados

Para implementar la parte correspondiente a la síntesis de expresiones faciales se adaptó el *Módulo de Aplicaciones de los Elementos Finitos* (*MEFi*) [6] para integrarlo al software desarrollado en este trabajo de tesis (*FaceER*). *MEFi* es un programa de

cómputo para análisis de estructuras en 2 y 3 dimensiones bajo problemas de esfuerzo y deformación; fue desarrollado en forma conjunta por la Universidad de Guanajuato, el CIMAT y el CIMNE (Centro Internacional de Métodos Numéricos en Ingeniería).

El problema de síntesis de expresiones faciales se abordó como un problema de estructuras tridimensionales sometidas a esfuerzo y deformación, donde la cara es la estructura tridimensional que sufre desplazamientos y deformaciones como consecuencia de las fuerzas ejercidas por los músculos faciales.

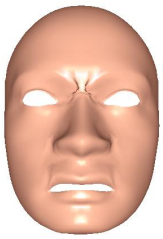
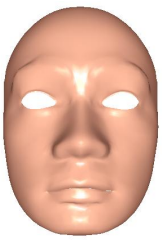
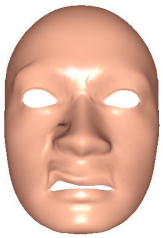
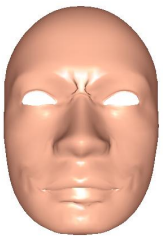
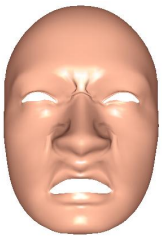
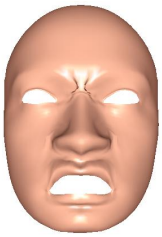
Los datos necesarios para resolver el problema son:

- **Propiedades del material:** Recordemos que se dotó de una estructura de 2 capas de elementos hexaédricos a la cara base para simular los distintos componentes faciales (figura 3.5). Basándonos en los experimentos realizados en [23], se dieron valores aproximados de módulo de Young: 5.6 kPa para la capa de la piel y 0.12 kPa para la capa de tejido graso. La cuasi-incompresibilidad de los tejidos se modeló con un coeficiente de Poisson igual a 0.45 para ambas capas.

- **Nodos prescritos:** Se impidió el movimiento de los nodos de la capa inferior para simular el cráneo. Además, como condición de contorno, se restringió el movimiento de todos los nodos en el borde de la cara.

- **Cargas:** Dependiendo de la expresión deseada, el editor de expresiones de *FaceER* (Apéndice B) tiene controles para aumentar o disminuir la fuerza de cada músculo (o de cada AU), lo que equivale a imponer cargas en los nodos que forman un músculo.

En el editor de expresiones de *FaceER* se afinaron manualmente los controles musculares para crear 18 expresiones faciales predefinidas, tomando como base las expresiones ilustradas en la herramienta *Artnatomy* [16]. Como veremos más adelante, estas expresiones predefinidas son las que se utilizarán para hacer el reconocimiento facial. En la tabla 3.2 se muestran los resultados del proceso, los cuáles pueden ser comparados con las ilustraciones de *Artnatomy*.

Artnatomy	**Cara base**	***Artnatomy***	**Cara base**
Amenaza		Atención	
			
Desprecio		Determinación	
			
Dolor		Duda	
			
Esfuerzo		Felicidad	
Ira		Ironía	
			

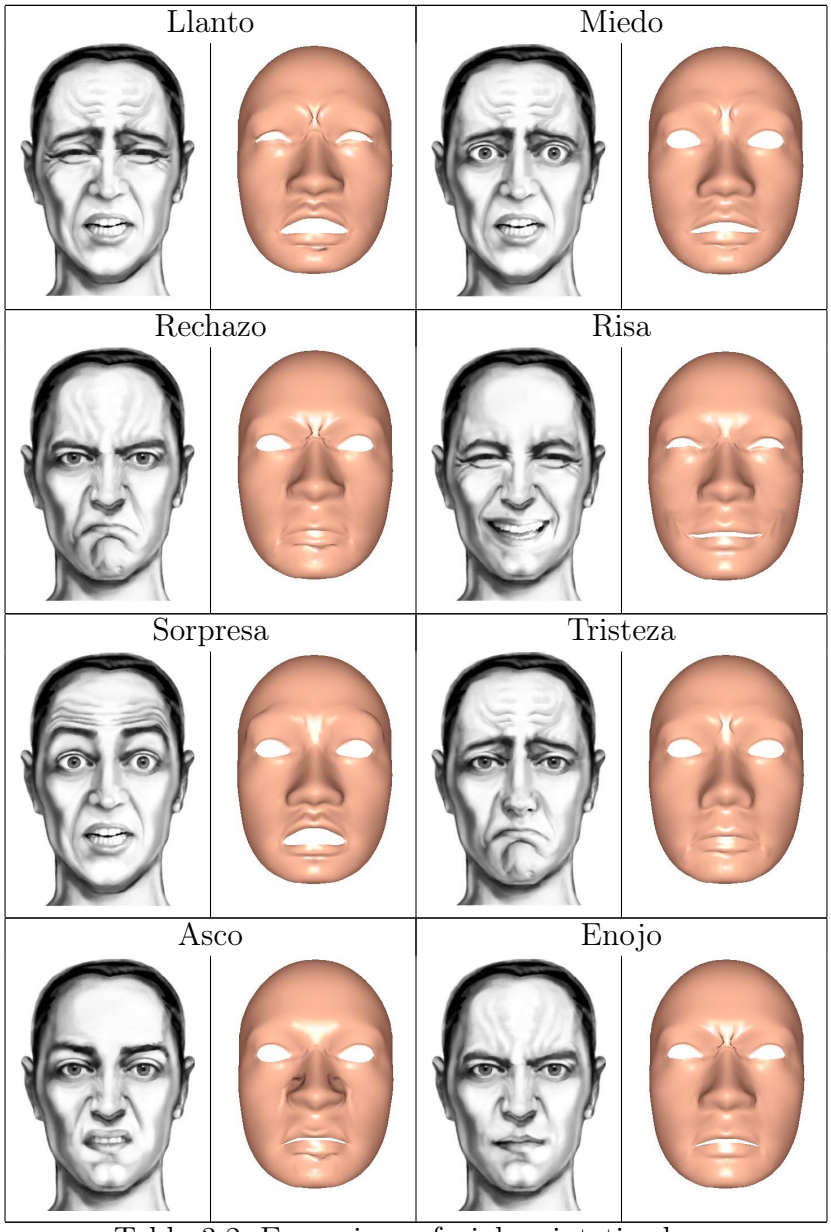


Tabla 3.2: Expresiones faciales sintetizadas.

Capítulo 4

Ajuste de la cara base

Con el trabajo realizado en el capítulo anterior contamos ya con una cara que tiene cierta estructura anatómica capaz de generar expresiones faciales. Sin embargo, lo que queremos es generar expresiones para todos los individuos que estarán en la base de datos del sistema de reconocimiento facial. Preprocesar la malla de entrada de cada uno de los individuos para sintetizar expresiones faciales sería una tarea impráctica y extremadamente difícil, ya que la estructura y la calidad de estas mallas dependerá de la técnica de adquisición que se esté utilizando. Es por ello que es necesario ajustar la cara base a cada una de las caras de entrada; dicho ajuste debe realizarse procurando distorsionar lo menos posible la estructura anatómica de la cara base para seguir contando con la funcionalidad de síntesis de expresiones que ya tenemos.

En este capítulo describiremos el proceso realizado para transformar la cara base en la cara de entrada de cada uno los individuos que están en la base de datos para después generar expresiones sintéticas de cada uno de ellos. El único requisito necesario es que las caras de entrada estén de frente y con expresión neutral.

4.1. Eliminación de ruido

De acuerdo a la revisión de las técnicas de adquisición de información tridimensional dada en el Capítulo 2, sabemos que, sin importar la técnica de adquisición utilizada, surgirán errores de captura en la malla. Para mejorar la calidad de la cara de entrada y hacer una detección de puntos característicos más exacta utilizamos dos algoritmos de eliminación de ruido: el primero es el suavizador Laplaciano clásico, el segundo es una propuesta de algoritmo iterativo que surge de combinar el suavizador Laplaciano con el suavizador de mediana.

4.1.1. Algoritmo Laplaciano

El suavizador Laplaciano aplicado a mallas tridimensionales es una extensión de la versión utilizada en procesamiento de imágenes, donde la idea básica es cambiar el

valor de un pixel por el promedio de los valores de los pixeles vecinos. El algoritmo Laplaciano aplicado a mallas es muy sencillo: la posición $p_i = [p_{ix}, p_{iy}, p_{iz}]$ de un vértice v_i se remplaza con el promedio de las posiciones de los vértices vecinos:

$$p_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} p_j, \tag{4.1}$$

donde $\mathcal{N}_i$ es la vecindad del vértice i.

Hay dos técnicas para calcular la nueva posición p_i. La primera es la llamada *versión simultánea*, y consiste en modificar todas las posiciones a la vez, de manera que toda nueva posición p_i, para $i = 1, \ldots, n$, depende completamente del mismo conjunto de posiciones, denominado q. La segunda variante es la *versión secuencial*, donde p_i no sólo depende del conjunto de posiciones anteriores, q, sino también de las nuevas posiciones p_j previamente calculadas. Aquí el resultado dependerá del orden en que se tomen los vértices.

La versión simultánea necesita más almacenamiento para mantener las posiciones q hasta que se calculen todas las nuevas posiciones, sin embargo, los resultados de esta técnica son mejores.

Un problema del suavizador Laplaciano es que la malla tiende a encogerse con cada iteración, de manera que si el número de iteraciones tiende a infinito, la malla se encogerá hasta convertirse en un punto. En la tablas 4.1 y 4.2 se muestran varias mallas faciales suavizadas con el algoritmo Laplaciano.

En este trabajo, el algoritmo Laplaciano se utilizó para suavizar las mallas antes de hacer la detección automática de puntos característicos. Debido a que este algoritmo suaviza mucho las regiones, deja una superficie donde es fácil localizar características de curvatura, como veremos más adelante.

4.1.2. Algoritmo de mediana

El suavizador de mediana aplicado a mallas, al igual que el suavizador Laplaciano descrito en el apartado anterior, es una extensión de la versión bidimensional utilizada en procesamiento de imágenes. La diferencia radica en que la posición p_i del vértice v_i se remplaza con la mediana[1] del vector formado por las posiciones de los vértices adyacentes.

Al igual que el suavizador Laplaciano, el filtro de mediana tiende a encoger y sobresuavizar las mallas conforme aumenta el número de iteraciones, sin embargo, preserva las características finas de la malla mejor que el suavizador Laplaciano. En las tablas 4.1 y 4.2 se muestran ejemplos de mallas suavizadas con el filtro de mediana.

[1] La mediana es el número que se encuentra a la mitad en un conjunto de n números ordenados. Si n es par, la mediana es el promedio de los 2 números que están en el centro.

4.1.3. Algoritmo iterativo propuesto

El algoritmo que se describirá a continuación surgió como una idea para mejorar la calidad de las mallas, eliminando los picos y preservando las características finas.

Se pensó en combinar el suavizador Laplaciano y el suavizador de mediana debido a que, por un lado, el suavizador Laplaciano es muy bueno para eliminar el ruido constituido por picos pequeños, mientras que el suavizador de mediana da buenos resultados con respecto al ruido formado por picos grandes.

La idea básica del algoritmo propuesto consiste en conservar la posición original de los vértices que tengan poco cambio entre iteraciones y suavizar los vértices que sufran grandes cambios. Para conseguir esto, se mantiene una copia extra, $\mathcal{Q}$, de la malla original $\mathcal{P}$, la cual se irá actualizando de la siguiente manera:

1. Se aplica un suavizador de mediana a la malla $\mathcal{Q}$.
2. Se aplica un suavizador Laplaciano a la malla suavizada en el paso anterior.
3. Para cada uno de los vértices v_i de la malla, se saca la distancia $d(p_i, q_i)$ entre la posición original p_i del vértice y la posición resultante después de aplicar los suavizadores, q_i.
4. Se define un umbral t multiplicando la distancia máxima d_{max} (obtenida en el paso anterior) por un cierto porcentaje.
5. Para cada uno de los vértices v_i de la malla, si $d(p_i, q_i)$ es mayor o igual a t se cambia la posición original p_i por la nueva posición q_i, en caso contrario q_i se regresa a la posición original p_i.
6. Se repite el proceso k veces.

En el algoritmo 4.1 se esquematiza la propuesta para eliminación de ruido en mallas.

Un problema del algoritmo, heredado de los suavizadores en los que se basa, es que tiende a encoger los bordes de la malla conforme aumenta el número de iteraciones. Sin embargo, un punto a favor del algoritmo es que tiene solamente dos parámetros, los cuáles son muy intuitivos. Uno es el número de iteraciones, que es un parámetro común en casi todos los algoritmos de eliminación de ruido. El número de iteraciones influye directamente en la eliminación de los picos grandes, es decir, entre más iteraciones se eliminarán mejor los picos, pero si se exagera en la cantidad de iteraciones también se puede encoger mucho la malla y comenzar a sobresuavizar, aunque en un grado menor a como lo hacen otros métodos. El otro parámetro es el porcentaje de cambio, relativo al vértice que tuvo el desplazamiento más grande en una iteración, a partir del cual se aplicarán los cambios de posición; por ejemplo, si fijamos un valor de 0.8, entonces estamos diciendo al algoritmo que aplique los cambios de posición en todos aquellos vértices cuyo desplazamiento haya sido mayor o igual al 80 % del tamaño del desplazamiento más grande. Entre más pequeño sea este porcentaje se harán

Algoritmo 4.1 Propuesta para eliminación de ruido en mallas.

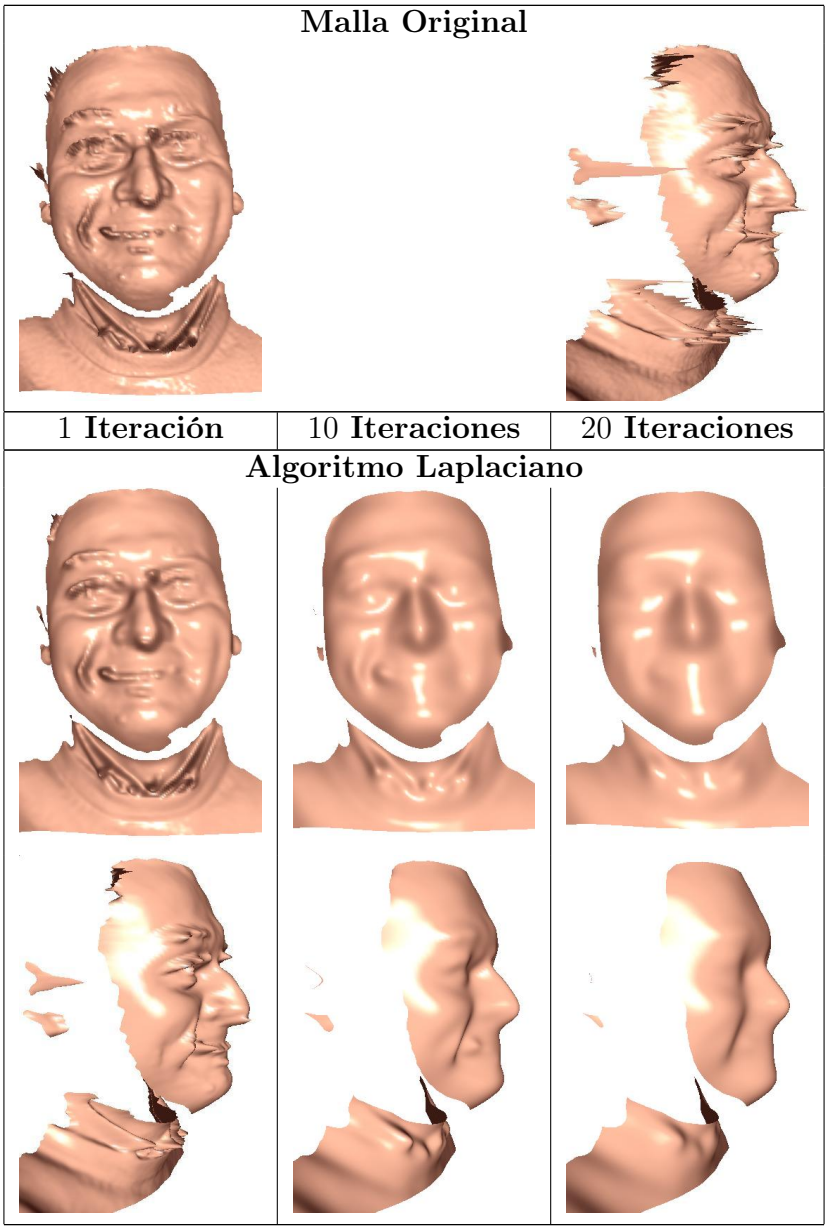
Datos: vértices $V = [v_1, \ldots, v_n]$, vecindades $\mathcal{N} = [\mathcal{N}_1, \ldots \mathcal{N}_n]$, número de iteraciones k y porcentaje w;

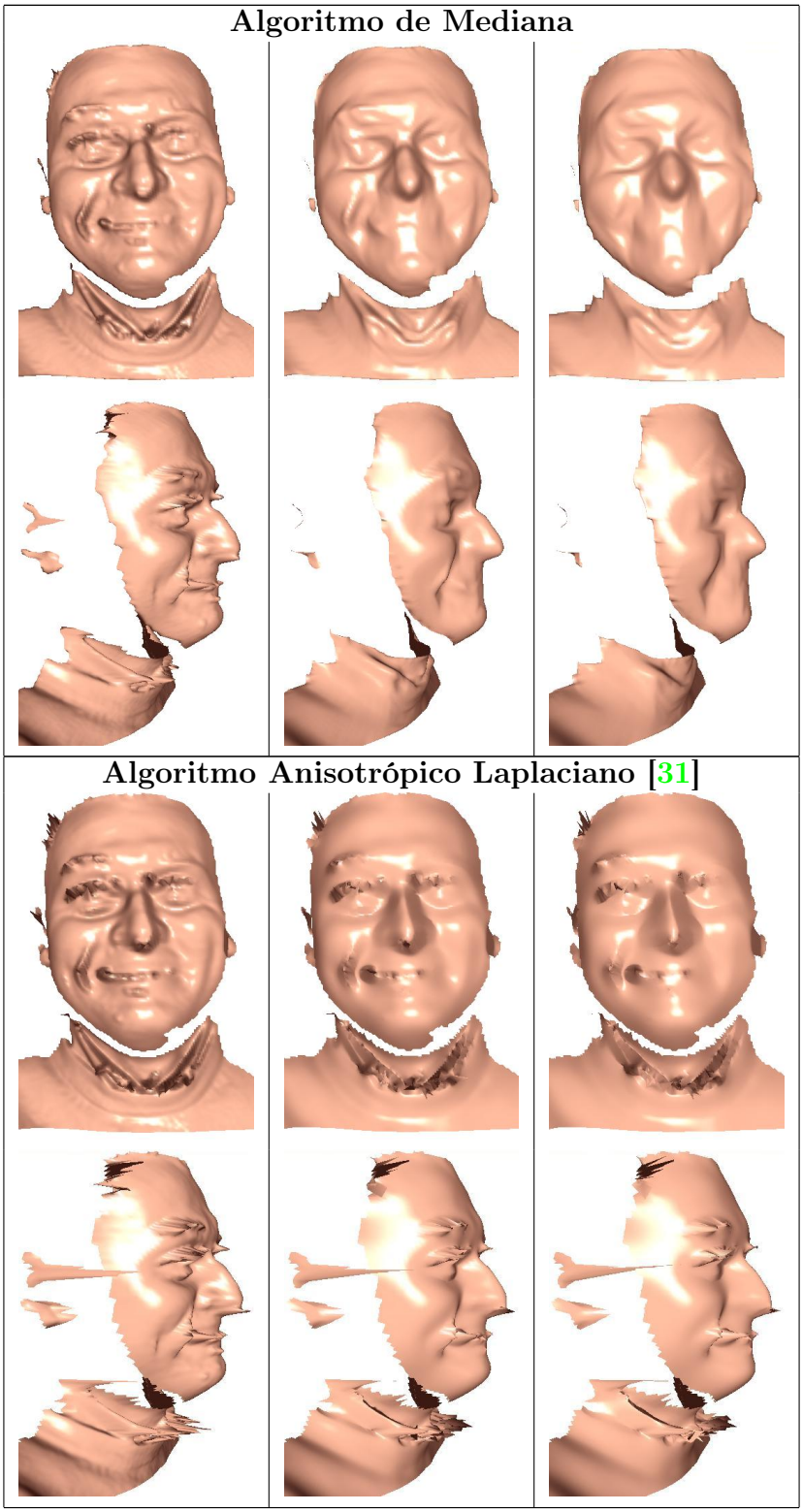
1: $\widehat{V} = V$
2: **para** $i = 0 : k - 1$ **hacer**
3: $\widehat{V} = medianSmooth(\widehat{V}, \mathcal{N})$;
4: $\widehat{V} = laplacianSmooth(\widehat{V}, \mathcal{N})$;
5: **para** $j = 0 : n - 1$ **hacer**
6: $d_j = \|v_j - \widehat{v}_j\|$;
7: **fin para**
8: $d_{max} = max(d_1, \ldots, d_n)$;
9: $t = w * d_{max}$;
10: **para** $j = 0 : n - 1$ **hacer**
11: **si** $d_j \geq t$ **entonces**
12: $v_j = \widehat{v}_j$;
13: **caso-contrario**
14: $\widehat{v}_j = v_j$;
15: **fin si**
16: **fin para**
17: **fin para**

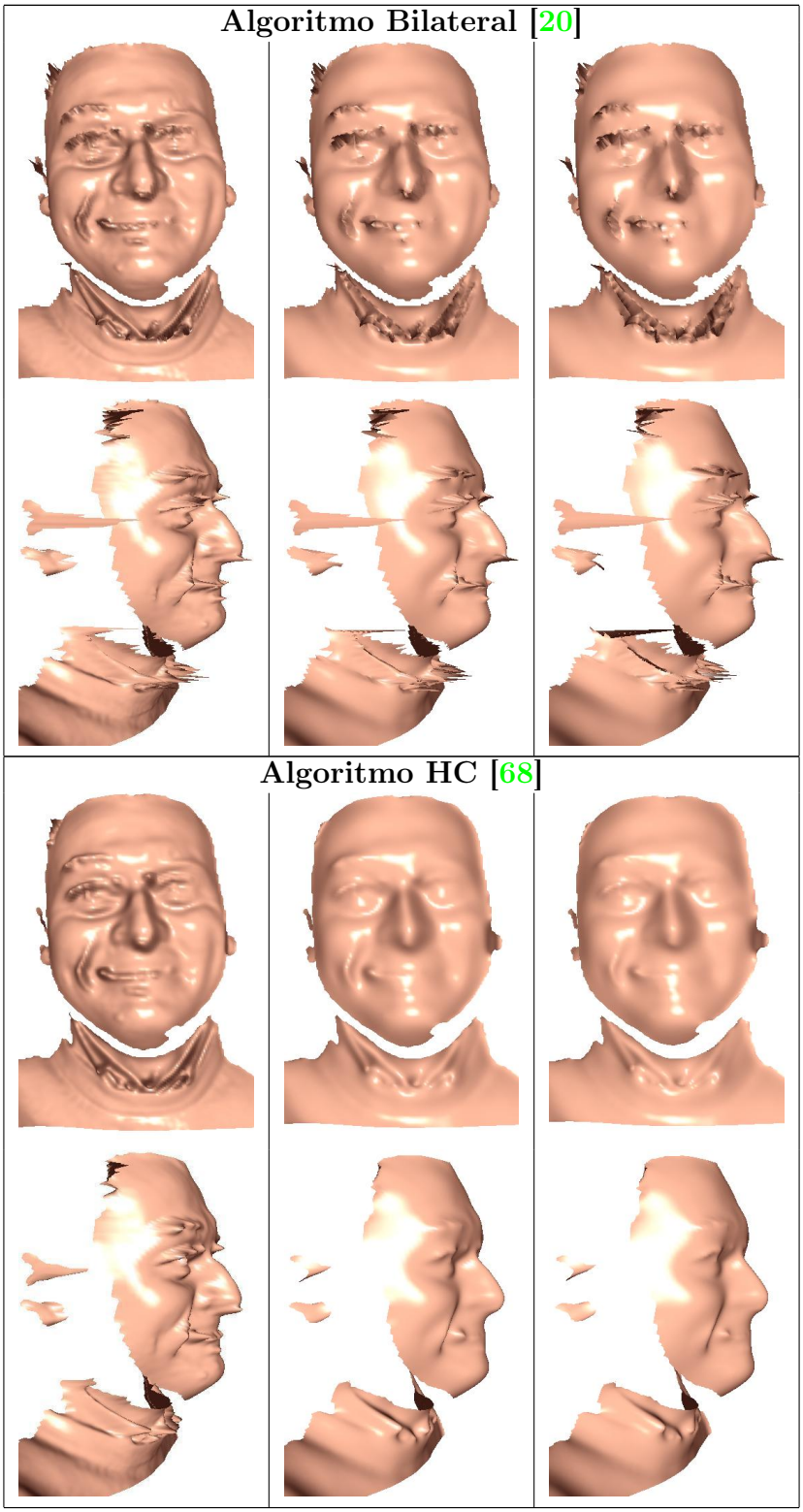
más cambios en una iteración, pero se corre el riesgo de sobresuavizar, en cambio si damos un porcentaje grande, se necesitarán más iteraciones pero los picos irán desapareciendo gradualmente sin perder otras características finas de la malla.

4.1.4. Experimentos y resultados

En la tabla 4.1 se compara el algoritmo propuesto contra los algoritmos individuales en los que se basa; se observa claramente que el algoritmo propuesto da mejores resultados en cuanto a eliminación de picos y preservación de características, obviamente será más lento ya que hace el doble del proceso al combinar los dos suavizadores. También se hace la comparación contra algunos algoritmos reportados en la literatura, los cuáles son buenos para eliminar el ruido pequeño, sin embargo, se puede ver que cuando hay picos grandes (muy frecuentes al capturar datos en 3D) el algoritmo presentado en este trabajo se desempeña mejor. En la tabla 4.2 se comparan los mejores resultados obtenidos con cada algoritmo, procurando que eliminen ruido pero sin sobresuavizar.


Malla Original
1 Iteración
10 Iteraciones
20 Iteraciones
Algoritmo Laplaciano


Algoritmo de Mediana
Algoritmo Anisotrópico Laplaciano [31]


Algoritmo Bilateral [20]
Algoritmo HC [68]

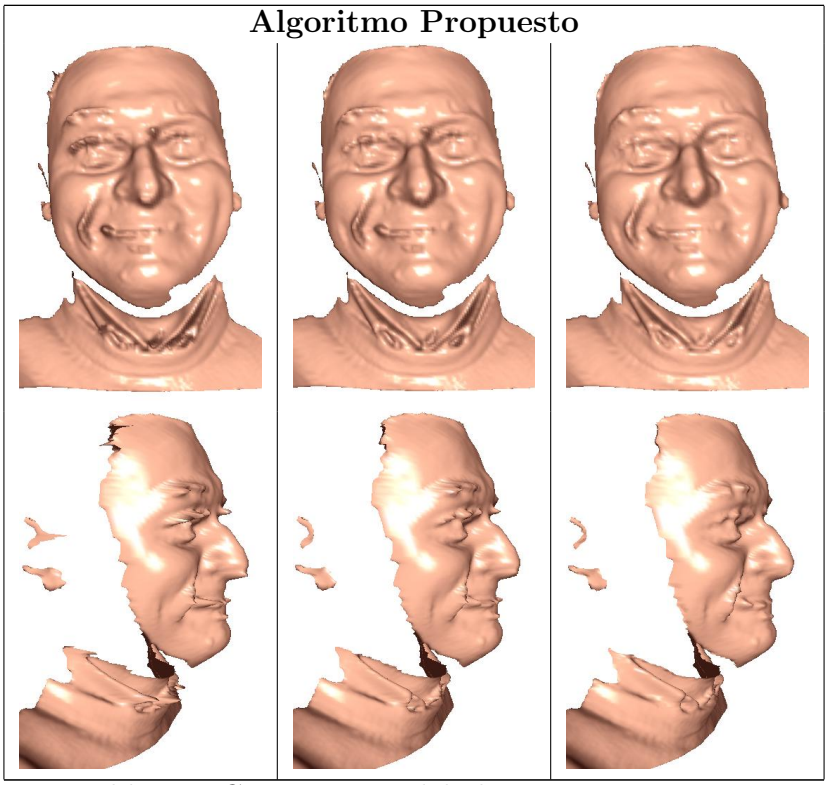
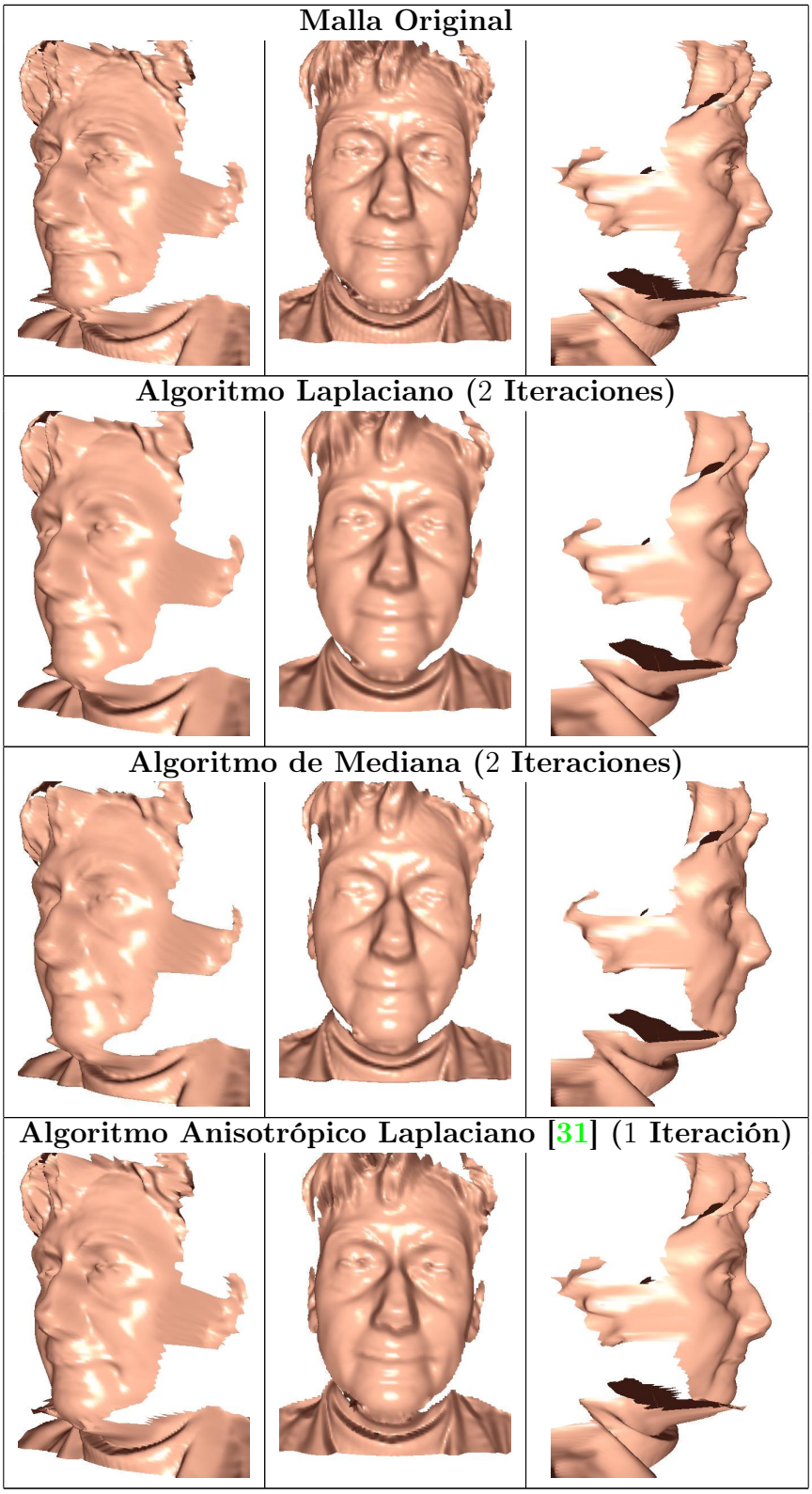


Tabla 4.1: Comparación del algoritmo propuesto.


Malla Original
Algoritmo Laplaciano (2 Iteraciones)
Algoritmo de Mediana (2 Iteraciones)
Algoritmo Anisotrópico Laplaciano [31] (1 Iteración)

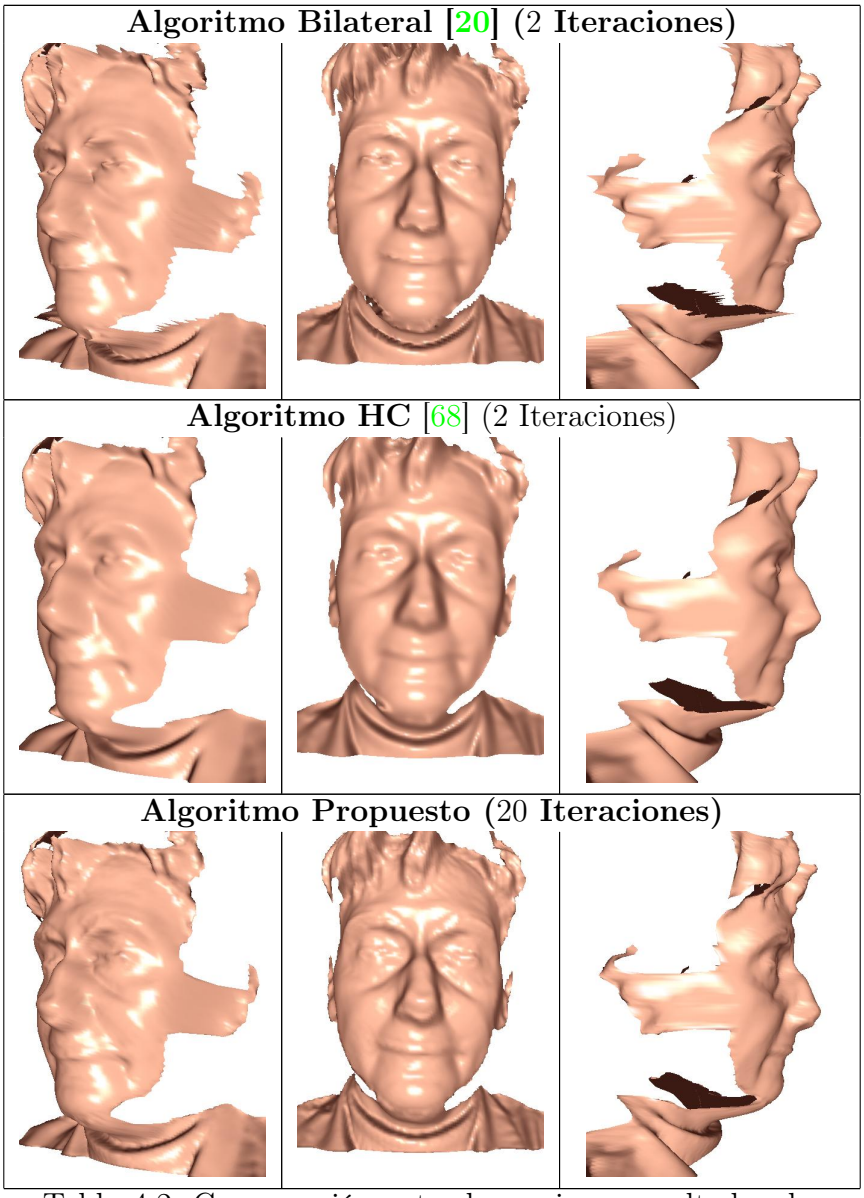


Tabla 4.2: Comparación entre los mejores resultados de cada algoritmo. Obsérvese como actúa cada uno de los algoritmos en la eliminación de los picos que están en el lado derecho de la barbilla y en los ojos.

4.2. Detección automática de puntos característicos

Para poder ajustar la cara base a cualquier cara de entrada, primero necesitamos una manera de hacer corresponder ciertos puntos de ambas caras entre sí. En esta

sección describiremos el algoritmo utilizado para detectar los puntos característicos necesarios para hacer dicha correspondencia. Se trata de un algoritmo heurístico que se basa en la información de curvatura de la superficie facial, así como en la información estadística de las distancias entre ciertos componentes faciales tales como los ojos, la nariz, la boca, las mejillas, la frente y la barbilla (tabla 4.3). El algoritmo es parecido al presentado en [13], la principal diferencia radica en que nuestra propuesta no utiliza información de color (imagen 2D), por lo que necesitamos hacer varias heurísticas sobre la información de curvatura para poder localizar los puntos característicos de una forma más exacta.

Antes de pasar a la descripción del algoritmo propuesto, daremos un breve repaso de los conceptos de curvatura que utiliza dicho algoritmo.

4.2.1. Descriptores de curvatura

La curvatura local de un punto en una superficie está caracterizada por la *curvatura máxima*, κ_1, y la *curvatura mínima*, κ_2. Las curvaturas máxima y mínima son ortogonales entre sí y se obtienen resolviendo el problema de eigenvalores $|\mathcal{H} - \kappa I| = 0$, donde I es la matriz identidad y $\mathcal{H}$ es la matriz Hessiana aproximada a partir de los vectores normales a la superficie:

$$\mathcal{H} = \begin{bmatrix} -\left(\frac{\partial \mathbf{n}}{\partial x}\right)_x & -\left(\frac{\partial \mathbf{n}}{\partial x}\right)_y \\ -\left(\frac{\partial \mathbf{n}}{\partial y}\right)_x & -\left(\frac{\partial \mathbf{n}}{\partial y}\right)_y \end{bmatrix} \tag{4.2}$$

donde $(\ldots)_x$ y $(\ldots)_y$ indican el componente x o y del vector, y $\mathbf{n}$ es el vector normal unitario a cada punto, el cuál es equivalente a:

$$\mathbf{n} = \begin{bmatrix} -\partial/\partial x \\ -\partial/\partial y \\ -\partial/\partial z \end{bmatrix} \tag{4.3}$$

A partir de las curvaturas máxima y mínima podemos obtener otros descriptores de curvatura:

- ***Shape Index* [18]:** Es una medida cuantitativa de la forma de la superficie en un punto p, se define como:

$$S(p) = \frac{1}{2} - \frac{1}{\pi} tan^{-1} \frac{\kappa_1(p) + \kappa_2(p)}{\kappa_1(p) - \kappa_2(p)} \tag{4.4}$$

  donde sus valores están en el rango $[0, 1]$. Dependiendo del valor, se distinguen nueve categorías de superficie, las cuales se muestran en la figura 4.1.

- **Curvatura Gaussiana:** La curvatura gaussiana de la superficie en un punto p es el producto de las curvaturas principales:

$$K(p) = \kappa_1(p)\,\kappa_2(p) \tag{4.5}$$

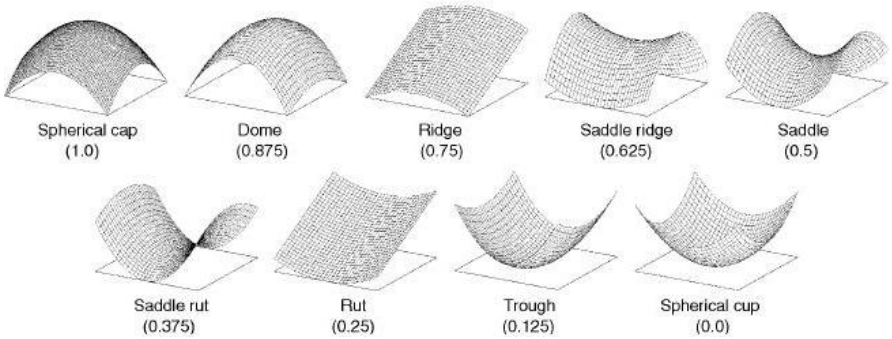


Figura 4.1: Escala *Shape Index*. Un valor de 1 representa una superficie convexa, mientras que un valor de 0 representa una superficie cóncava. El *Shape Index* es independiente del sistema de coordenadas utilizado [18].

Es positiva para esferas, negativa para hiperboloides de una hoja y cero para planos.

- **Curvatura Media:** La curvatura media de la superficie en un punto p es simplemente la media de las curvaturas principales:

$$H(p) = \frac{1}{2}\left(\kappa_1(p) + \kappa_2(p)\right) \tag{4.6}$$

En la figura 4.2 se muestra un ejemplo de estos descriptores sobre una superficie facial, también se muestra la magnitud del gradiente y las derivadas obtenidas directamente de los vectores normales unitarios, de acuerdo a la expresión 4.3.

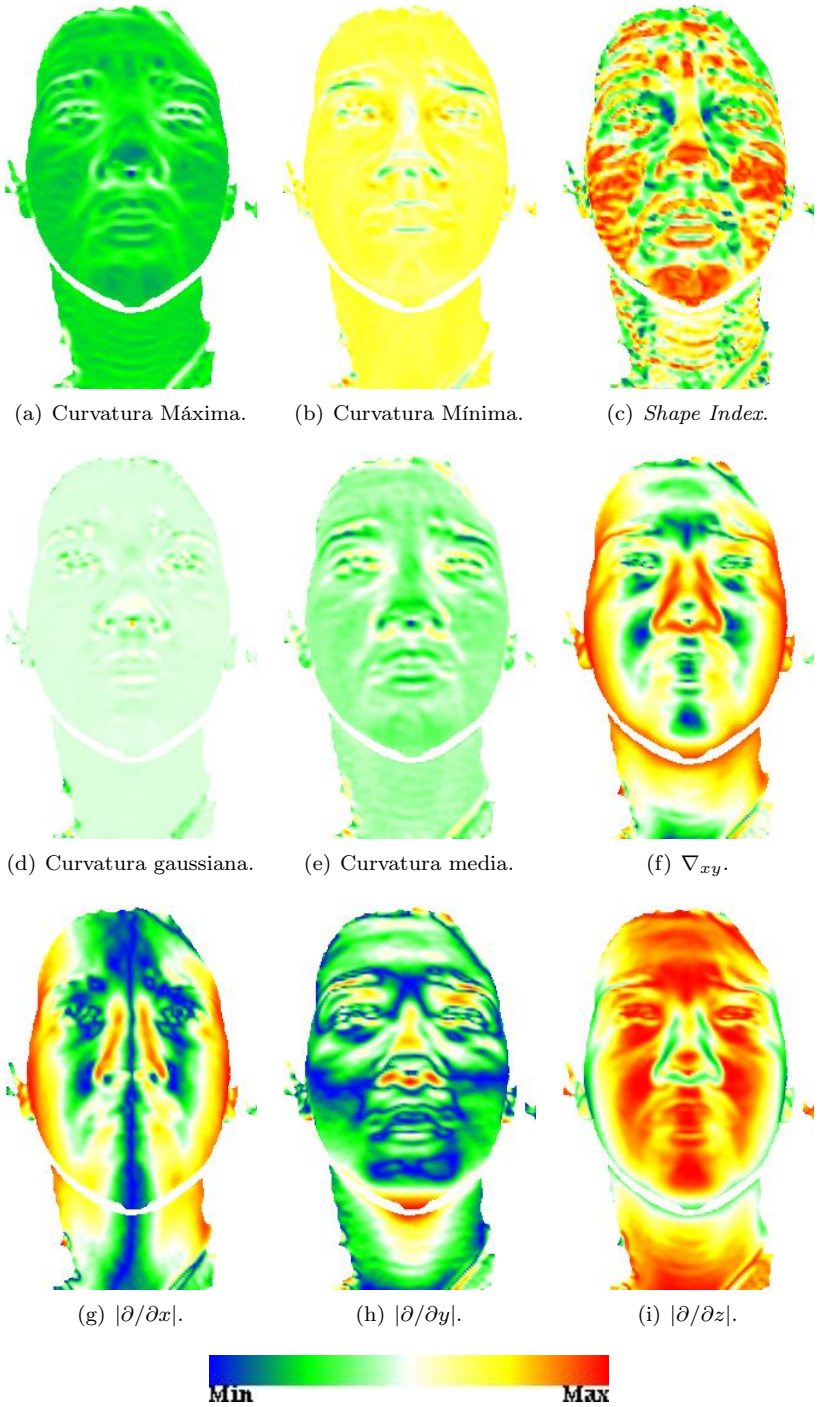
4.2.2. Distancia estadística entre componentes faciales

Las estadísticas de distancia entre componentes faciales mostradas en esta sección se basan en la referencia [13]. Fueron obtenidas de analizar una base de datos de aproximadamente $2,000$ imágenes de prueba. Para acotar la región de búsqueda de un punto, se incluye la distancia mínima, máxima y promedio de cada uno de los puntos (ver tabla 4.3).

Usando esta tabla de referencia, se pueden generar regiones de búsqueda para un determinado punto. Por ejemplo, una vez que se encuentra la punta de la nariz, se pueden acotar regiones para localizar otros puntos, lo que ayuda a mejorar la eficiencia del algoritmo en cuanto a tiempo y exactitud en la localización.

4.2.3. Algoritmo heurístico propuesto

Para que el algoritmo funcione correctamente es requisito indispensable que la cara de entrada esté en posición frontal y con una expresión neutral. Esta demanda



(a) Curvatura Máxima. (b) Curvatura Mínima. (c) *Shape Index*.

(d) Curvatura gaussiana. (e) Curvatura media. (f) ∇_{xy}.

(g) $|\partial/\partial x|$. (h) $|\partial/\partial y|$. (i) $|\partial/\partial z|$.

Figura 4.2: Descriptores de superficie. Con rojo aparece el valor máximo y con azul el valor mínimo de cada descriptor.

Punto A	**Punto B**	**Dirección**	**Min.** (mm)	**Avg.** (mm)	**Max.** (mm)
Frente	Punta de la nariz	Vertical	84.0	121.7	181.0
Punta de la nariz	Boca	Vertical	16.9	34.4	46.3
Punta de la nariz	Barbilla	Vertical	49.7	70.6	95.4
Punta de la nariz	Puente nasal	Vertical	21.2	32.2	48.4
Punta de la nariz	Esquina int. ojo	Horizontal	13.2	17.5	23.6
Esquina int. ojo	Esquina ext. ojo	Horizontal	15.9	30.6	39.7

Tabla 4.3: Modelo estadístico de la distancia entre componentes faciales [13].

no es muy exigente, ya que para enrolar un usuario en el sistema de reconocimiento facial se procederá de la misma forma que cuando uno se toma una fotografía para tramitar el pasaporte o el título profesional, es decir, las fotografías requeridas para esas situaciones son de frente y con expresión neutral; sólo que para el sistema de reconocimiento, en vez de capturar la imagen de color, se tomará la información tridimensional de la cara. Recordemos también que hay que aplicar un suavizador Laplaciano antes de aplicar el algoritmo para eliminar los picos de ruido pequeños que pueden afectar las características locales de curvatura de la malla.

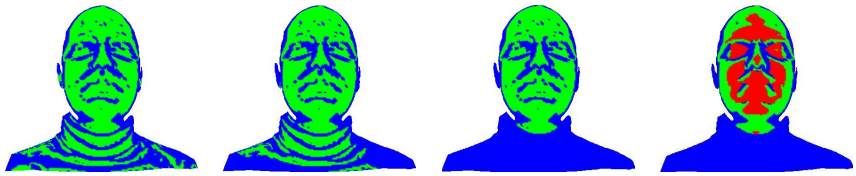
A continuación se enumeran las pasos que sigue el algoritmo heurístico para hacer una detección automática de puntos característicos:

1. **Segmentación facial:** Es común que la malla capturada contenga algunas otras partes del cuerpo, como el cuello y los hombros, a parte de la cara (ver figura 4.3), por lo que como primer paso es necesario segmentar la región facial. Para esto, partimos de la suposición de que la cara es una región que tiene un promedio de *Shape Index* ($avg\,(S)$) elevado, mayor a 0.7, tal como lo muestran las regiones amarillas y rojas de la figura 4.2(c).

   Entonces, lo que hacemos es filtrar la malla para quitar regiones con $S < 0.375$ y $|\partial/\partial z| < 0.5$ para tratar de separar varias regiones con S elevado (figura 4.3(a)). Después sacamos el promedio $avg\,(S)$ de cada región y quitamos aquellas regiones en las que $avg\,(S) \leq 0.7$ (figura 4.3(b)). La región facial será la región más grande (figura 4.3(c)).

   Aunque se supone que la cara de entrada está en posición frontal, es normal que aparezcan ligeras rotaciones en el plano XY, las cuáles son corregidas haciendo una alineación con respecto a los eigenvectores del conjunto de puntos en los que $\nabla_{xy} < 0.6$ (figura 4.3(d)).

2. **Detección de la punta de la nariz:** Varios algoritmos asumen que la punta

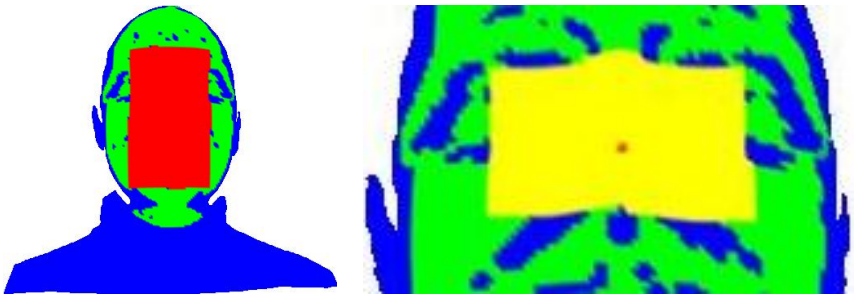


(a) En azul: $S < 0.375$ y $|\partial/\partial z| < 0.5$. (b) En verde: $avg\,(S) > 0.7$. (c) En verde: región facial segmentada. (d) Alineación utilizando $\nabla_{xy} < 0.6$ (puntos en rojo).

Figura 4.3: Segmentación facial.

de la nariz es el punto más cercano al sensor de captura. Sin embargo, esto no siempre es cierto, ya que hay personas que pueden tener los labios o la barbilla más prominente, o incluso, dependiendo del peinado, puede que el cabello esté más cerca del sensor que la punta de la nariz. Es por ello que desarrollamos la siguiente heurística para encontrar la punta nasal.

Primero se obtienen regiones con gradiente bajo ($\nabla_{xy} < 0.3$), las cuáles corresponderán a algunas áreas de la frente, mejillas y barbilla, como se ve en la figura 4.2(f). Tomamos los valores extremos en X y Y de estas áreas para formar un rectángulo donde se encontrará la nariz (figura 4.4(a)). En ese rectángulo se busca la región más grande con $|\partial/\partial z| \leq 0.5$ que pertenecerá a las alas nasales (figura 4.2(i)). Luego se buscan los valores extremos en Y de las alas nasales para recortar un poco el rectángulo y dentro de éste se busca la región con el punto máximo en Z y que tenga $\nabla_{xy} < 0.25$. La punta de la nariz será el centroide de esta región (figura 4.4(b)).

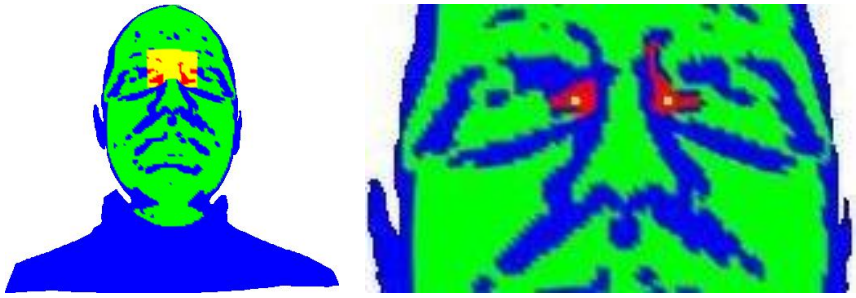


(a) En rojo: rectángulo de búsqueda nasal inicial. (b) En amarillo: rectángulo refinado, en rojo: punta de la nariz.

Figura 4.4: Detección de la punta de la nariz.

3. **Detección de las esquinas internas de los ojos:** Las esquinas internas de los ojos son regiones con valor de *Shape Index* bajo, como se ve en la figura 4.2(c). Para refinar y acelerar la detección de estos puntos utilizamos la información estadística entre la punta de la nariz y el puente nasal (tabla 4.3) para construir

un rectángulo donde se buscarán las regiones más grandes que tengan un valor de $S \leq 0.25$ (figura 4.5(a)). Una vez detectadas las dos regiones mayores se reduce el umbral de S y se aumenta el de $|\partial/\partial z|$ de forma gradual para reducir el tamaño de las regiones y obtener una ubicación más exacta de los puntos. Las esquinas internas de los ojos serán el centroide de cada una de estas dos regiones finales, respectivamente (figura 4.5(b)).

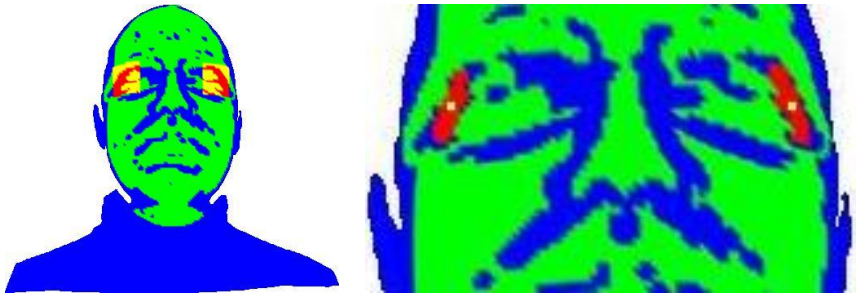


(a) En amarillo: rectángulo estadístico, en rojo: $S \leq 0.25$.

(b) En rojo: regiones mayores obtenidas, en amarillo: puntos encontrados.

Figura 4.5: Detección de las esquinas internas de los ojos.

4. **Detección de las esquinas externas de los ojos:** Para obtener estos puntos se construyen dos rectángulos estadísticos basándonos en la distancia que hay entre la esquina interna y externa del ojo (ver tabla 4.3). Dentro de cada rectángulo se buscan regiones con las siguientes características: $S < 0.5$, $|\partial/\partial x| < 0.9$ y $\nabla_{xy} > 0.5$ (figura 4.6(a)). Se toma la región más grande de cada rectángulo y se baja gradualmente el umbral de S para reducir el tamaño de las regiones. Las esquinas externas de los ojos serán el centroide de cada región (figura 4.6(b)).



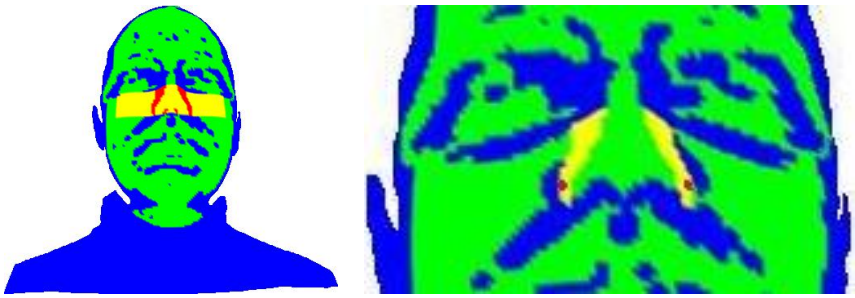
(a) En amarillo: rectángulos estadísticos, en rojo: $S < 0.5$, $|\partial/\partial x| < 0.9$ y $\nabla_{xy} > 0.5$.

(b) En rojo: regiones mayores obtenidas, en amarillo: puntos encontrados.

Figura 4.6: Detección de las esquinas externas de los ojos.

5. **Detección de las esquinas de la nariz:** Las esquinas de la nariz son regiones caracterizadas por tener $|\partial/\partial z| \leq 0.5$ y $|\partial/\partial x| > 0.65$, como se puede ver en las

figuras 4.2(i) y 4.2(g). Para encontrar estos puntos construimos un rectángulo usando los puntos recién encontrados: el rectángulo estará acotado en X por las esquinas externas de los ojos y en Y por la mitad de la distancia que hay entre la punta de la nariz y las esquinas internas de los ojos, en dirección vertical (figura 4.7(a)). Obtenemos las dos regiones mayores y las esquinas de la nariz serán los valores máximo y mínimo en X, dependiendo del lado en que esté cada región (figura 4.7(b)).



(a) En amarillo: rectángulo estadístico, en rojo: $|\partial/\partial z| \leq 0.5$ y $|\partial/\partial x| > 0.65$.

(b) En amarillo: regiones mayores obtenidas, en rojo: puntos encontrados.

Figura 4.7: Detección de las esquinas de la nariz.

6. **Detección de las esquinas de la boca:** Al igual que en puntos anteriores, para detectar las esquinas de la boca también construimos un rectángulo estadístico, tomando de la tabla 4.3 la distancia que hay entre la punta de la nariz y la boca, en dirección vertical, y acotando el rectángulo en X por las esquinas externas de los ojos. Disminuimos aún más el área de este rectángulo utilizando los extremos en Y de las regiones que estén dentro del ancho de la nariz y que tengan valores de $S > 0.75$ y $\nabla_{xy} < 0.35$ que son características de los labios, de manera que ahora el rectángulo se reduce en Y a la zona de los labios (figura 4.8(a)). Las esquinas de la boca se caracterizan por tener un valor de *Shape Index* bajo ($S < 0.25$) como se muestra en la figura 4.2(c). Por último, aplicamos la misma técnica de ir reduciendo el umbral, pero esta vez de la curvatura media H, hasta que las regiones sean muy chicas. De las regiones resultantes, las dos más grandes serán las esquinas de la boca (figura 4.8(b)).

7. **Detección de los bordes de las mejillas:** Antes de describir el proceso de detección de los bordes de las mejillas cabe aclarar que para hacer la detección de todos los bordes (mejillas, frente y barbilla), primero se hace una transformación rígida entre la cara de entrada y la cara base utilizando los puntos característicos que ya se tienen, tal como se describe en la siguiente sección. Esto se hace porque después de esta transformación el plano de los ojos quedará alineado con el eje X, lo que dará una detección de bordes más exacta.

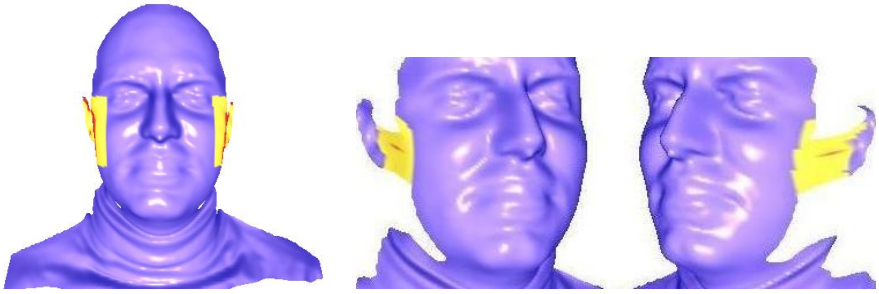
   Los bordes de las mejillas son regiones donde $|\partial/\partial z| < 0.4$, como se muestra en

(a) En rojo: rectángulo estadístico, en amarillo: $S < 0.25$.

(b) En amarillo: regiones mayores obtenidas, en rojo: puntos encontrados.

Figura 4.8: Detección de las esquinas de la boca.

la figura 4.2(i). Entonces, lo que hacemos en este caso es buscar estas regiones dentro de dos rectángulos delimitados por la boca y los ojos en Y y fuera de las esquinas externas de los ojos en X, tal como se ve en la figura 4.9(a). Los bordes de las mejillas serán el centroide de las dos regiones más grandes (figuras 4.9(b)).



(a) En amarillo: rectángulo estadístico, en rojo: $|\partial/\partial z| < 0.4$.

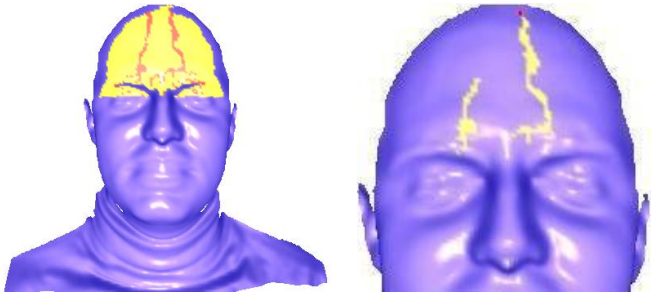
(b) En amarillo: regiones mayores obtenidas, en rojo: puntos encontrados.

Figura 4.9: Detección de los bordes de las mejillas.

8. **Detección del borde de la frente:** Si observamos la figura 4.2 podríamos pensar que el borde de la frente es fácil de obtener si lo tomamos como el valor máximo en Y. Sin embargo, este no es el caso general, ya que en ocasiones las mallas pueden traer información del cabello. Para detectar este punto hacemos el siguiente proceso.

   Primero se buscan regiones que estén arriba de los ojos y que tengan valores de $|\partial/\partial x|$ tal que $0.19 < |\partial/\partial x| < 0.25$. El borde superior del área de búsqueda estará delimitado por $|\partial/\partial z| > 0.6$ (figura 4.10(a)). Dentro de esta área se obtienen las dos regiones más largas en Y y el valor más alto en Y de cualquiera de estas regiones será el borde de la frente (figura 4.10(b)).

9. **Detección del borde de la barbilla:** Para encontrar el borde de la barbilla

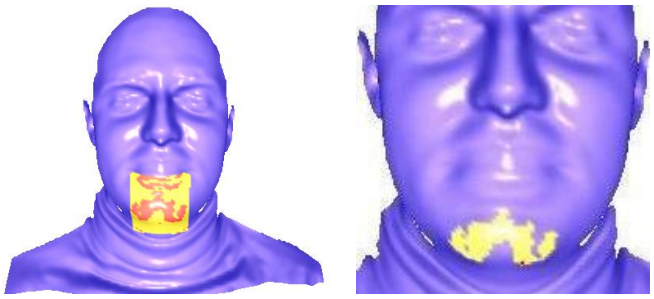


(a) En amarillo: área de búsqueda, en rojo: $0.19 < |\partial/\partial x| < 0.25$.

(b) En amarillo: regiones mayores obtenidas, en rojo: punto encontrado.

Figura 4.10: Detección del borde de la frente.

formamos un rectángulo con las esquinas de la boca y con el borde de la región facial detectada. Dentro de este rectángulo buscamos regiones donde $S > 0.85$ o $\nabla_{xy} < 0.3$, que son características de la barbilla (ver figura 4.2(c) y 4.2(f)). Luego se obtiene la región más grande dentro del rectángulo, y el valor mínimo en Y de esta región será el borde de la barbilla (figura 4.11).



(a) En amarillo: rectángulo de búsqueda, en rojo: $S > 0.85$ o $\nabla_{xy} < 0.3$.

(b) En amarillo: región mayor obtenida, en rojo: punto encontrado.

Figura 4.11: Detección del borde de la barbilla.

4.2.4. Experimentos y resultados

Para probar la exactitud de la detección automática de puntos con el algoritmo propuesto se utilizaron las dos mallas frontales de los 61 individuos (122 en total) de la base de datos GavabDB [48] (Apéndice A). Aunque en la base de datos se establece que las mallas frontales están en posición neutral, esto no es del todo cierto, ya que algunos individuos aparecen con algún esbozo de sonrisa o con la boca abierta y en algunos casos no están en posición frontal, sino rotados. Otro factor importante es

que hay mallas que están muy incompletas en regiones como ojos, cejas, mejillas y barbilla; o bien, hay mucha oclusión en la frente causada por el cabello.

En general, podemos decir que, para mallas faciales completas y sin oclusiones, y que cumplen con el requerimiento de estar de frente y con expresión neutral, el algoritmo hace una buena detección de puntos característicos, como se ve en los ejemplos de la figura 4.12. En las 122 pruebas realizadas se hizo una evaluación visual para determinar cuándo el algoritmo no hacía una detección correcta. Sólo en 8 ocasiones el algoritmo falló en detectar correctamente algunos puntos en mallas que cumplían con los requerimientos (ver ejemplos de la figura 4.13), mientras que en otras ocasiones (aproximadamente 15) el error se debió principalmente a la mala calidad de las mallas o a que no se cumplía con el requerimiento de entrada (ver figura 4.14).

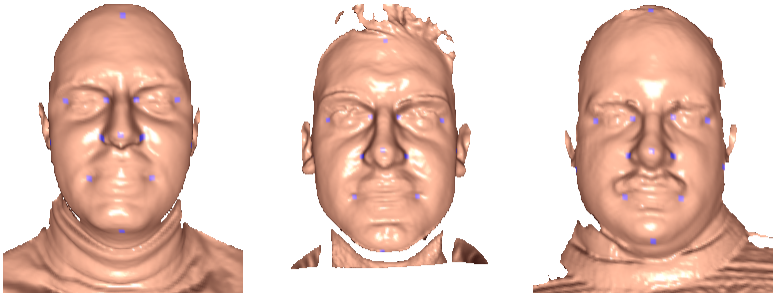
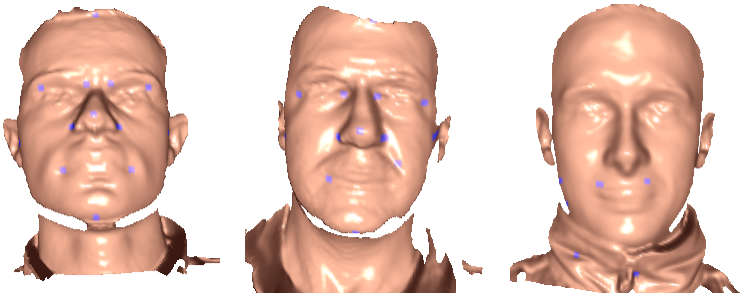


Figura 4.12: Puntos característicos detectados correctamente.

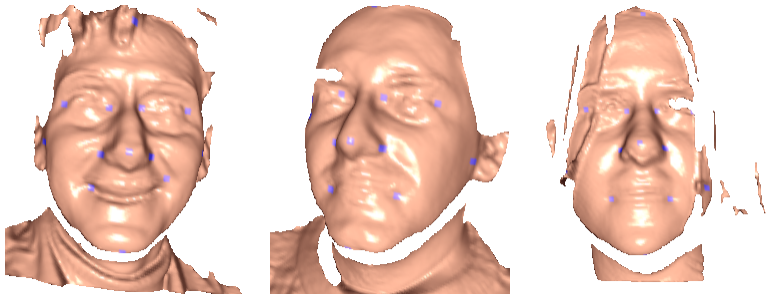


(a) Esquinas de los ojos mal detectadas. (b) Esquina izquierda de la boca mal detectada. (c) Región facial mal detectada.

Figura 4.13: Puntos característicos detectados incorrectamente.

4.3. Ajuste

Con el algoritmo de detección de puntos descrito en la sección anterior contamos ya con una base para ajustar la cara genérica a cualquier cara de entrada, haciendo



(a) Cara sonriendo, boca abierta, oclusión del cabello.

(b) Cara rotada e incompleta.

(c) Cara incompleta y oclusión del cabello.

Figura 4.14: Puntos característicos detectados incorrectamente debido a las mallas. El algoritmo no detecta bien los puntos debido a la mala calidad de las mallas o a que no cumplen con los requerimientos iniciales.

transformaciones para hacer corresponder los puntos característicos de ambas caras. En la cara base, estos puntos están marcados manualmente (figura 4.15), puesto que todos sus datos son conocidos.

Figura 4.15: Puntos característicos de la cara base detectados manualmente.

El proceso para convertir la cara base en la cara de entrada consiste básicamente en tres pasos:

1. Una alineación global o transformación rígida de la cara de entrada a la cara base.
2. Una alineación local entre los puntos característicos haciendo deformaciones a la cara base en el plano XY.
3. Una deformación a la cara base (proyectada cilíndricamente) en el eje Z para hacer corresponder todos sus vértices con la superficie de la cara de entrada.

Una vez realizado este proceso, la cara base será igual a la cara de entrada, y como la cara base ya tiene una estructura capaz de generar expresiones faciales, podremos generar expresiones sintéticas para cualquier cara de entrada.

4.3.1. Alineación global

Para encontrar la mejor transformación rígida de la cara de entrada que permitirá alinear y acercar sus puntos característicos con los de la cara base nos basamos en el método desarrollado en la referencia [28], que utiliza *Quaterniones Unitarios* (*Unit Quaternions*). Hablar de los quaterniones y sus propiedades queda fuera del alcance de este trabajo, si se desea ahondar más sobre el tema es conveniente revisar la referencia [28]. Para nuestros fines, un quaternion puede ser visto como un vector con cuatro componentes, de manera que si tenemos $p = [p_x \ p_y \ p_z]$, podemos representarlo como un quaternion imaginario:

$$\dot{p} = 0 + ip_x + jp_y + kp_z \tag{4.7}$$

Para obtener la transformación rígida que minimice la distancia entre los puntos característicos de ambas caras necesitamos minimizar la suma del error cuadrático:

$$\sum_{i=1}^{n} \|e_i\|^2 \tag{4.8}$$

con

$$e_i = p_{Bi} - sR(p_{Ei}) - p_0 \tag{4.9}$$

donde p_{Bi} es un punto en la cara base, p_{Ei} es un punto en la cara de entrada, s es un factor de escala, p_0 es la translación y $R\left(p_{Ei}\right)$ es la rotación aplicada al punto de la cara de entrada.

Antes de seguir avanzando, conviene referir todas las mediciones al centroide $\overline{p}$ de cada conjunto de puntos:

$$\overline{p_E} = \frac{1}{n}\sum_{i=1}^{n} p_{Ei} \qquad \overline{p_B} = \frac{1}{n}\sum_{i=1}^{n} p_{Bi} \tag{4.10}$$

las nuevas coordenadas estarán dadas por:

$$p'_{Ei} = p_{Ei} - \overline{p_E} \qquad p'_{Bi} = p_{Bi} - \overline{p_B} \tag{4.11}$$

Ahora la suma de errores cuadráticos (expresiones 4.8 y 4.9) es:

$$\sum_{i=1}^{n} \|p'_{Bi} - sR(p'_{Ei}) - p'_0\|^2 \tag{4.12}$$

donde

$$p'_0 = p_0 - \overline{p_B} + sR(\overline{p_E}) \tag{4.13}$$

y expandiendo la expresión tenemos:

$$\sum_{i=1}^{n} \|p'_{Bi} - sR(p'_{Ei})\|^2 - 2p'_0 \cdot \sum_{i=1}^{n} [p'_{Bi} - sR(p'_{Ei})] + n\,\|p'_0\|^2 \tag{4.14}$$

donde el término en el centro de la expresión es cero, ya que las mediciones están referidas al centroide; además, el primer término no depende de p'_0 y el último no puede ser negativo, por lo que el error total se minimiza con $p'_0 = 0$. Así pues:

$$p_0 = \overline{p_B} - sR(\overline{p_E}) \tag{4.15}$$

esto indica que la translación es sólo la diferencia entre el centroide de los puntos de la cara base y el centroide de los puntos rotados y escalados de la cara de entrada.

Puesto que $p'_0 = 0$ podemos escribir el error total a minimizar como:

$$\sum_{i=1}^{n} \|p'_{Bi} - sR(p'_{Ei})\|^2 \tag{4.16}$$

expandiendo la expresión obtenemos:

$$\underbrace{\sum_{i=1}^{n} \|p'_{Bi}\|^2}_{S_B} - 2s \underbrace{\sum_{i=1}^{n} p'_{Bi} \cdot R(p'_{Ei})}_{D} + s^2 \underbrace{\sum_{i=1}^{n} \|p'_{Ei}\|^2}_{S_E} \tag{4.17}$$

completando el cuadrado en s tenemos:

$$\left(s\sqrt{S_E} - D/\sqrt{S_E}\right)^2 + \left(S_B S_E - D^2\right)/S_E \tag{4.18}$$

esto se minimiza con respecto a la escala s cuando el primer término es igual a cero o $s = D/S_E$, esto es:

$$s = \frac{\sum_{i=1}^{n} p'_{Bi} \cdot R(p'_{Ei})}{\sum_{i=1}^{n} \|p'_{Ei}\|^2} \tag{4.19}$$

El resto del error se minimiza cuando D es lo más grande posible, por tanto tenemos que encontrar la rotación que maximice:

$$\sum_{i=1}^{n} p'_{Bi} \cdot R(p'_{Ei}) \tag{4.20}$$

Para encontrar la rotación que maximiza la expresión 4.20 se utilizan los quaterniones mencionados al principio del apartado. Sin entrar en detalle acerca de las propiedades de los quaterniones, se trata de encontrar el quaternion unitario $\dot{q}$ tal que maximice:

$$\sum_{i=1}^{n} \left(\dot{q}\dot{p}'_{Ei}\dot{q}^*\right) \dot{p}'_{Bi} \tag{4.21}$$

donde $\dot{q}^*$ es el conjugado de $\dot{q}$. De acuerdo a las propiedades de los quaterniones, la expresión 4.21 se puede escribir como:

$$\sum_{i=1}^{n} \left(\dot{q}\dot{p}'_{Ei}\right) \cdot \left(\dot{p}'_{Bi}\dot{q}\right) \tag{4.22}$$

donde

$$\dot{q}\dot{p}'_{Ei} = \begin{bmatrix} 0 & -p'_{Exi} & -p'_{Eyi} & -p'_{Ezi} \\ p'_{Exi} & 0 & p'_{Ezi} & -p'_{Eyi} \\ p'_{Eyi} & -p'_{Ezi} & 0 & p'_{Exi} \\ p'_{Ezi} & p'_{Eyi} & -p'_{Exi} & 0 \end{bmatrix} \dot{q} = \mathcal{R}_{Ei}\dot{q} \tag{4.23}$$

y de forma similar:

$$\dot{p}'_{Bi}\dot{q} = \begin{bmatrix} 0 & -p'_{Bxi} & -p'_{Byi} & -p'_{Bzi} \\ p'_{Bxi} & 0 & p'_{Bzi} & -p'_{Byi} \\ p'_{Byi} & -p'_{Bzi} & 0 & p'_{Bxi} \\ p'_{Bzi} & p'_{Byi} & -p'_{Bxi} & 0 \end{bmatrix} \dot{q} = \mathcal{R}_{Bi}\dot{q} \tag{4.24}$$

Rescribiendo 4.22 tenemos:

$$\sum_{i=1}^{n} (\mathcal{R}_{Ei}\dot{q}) \cdot (\mathcal{R}_{Bi}\dot{q}) = \sum_{i=1}^{n} \left(\dot{q}^T \mathcal{R}_{Ei}^T\right) \cdot (\mathcal{R}_{Bi}\dot{q}) \tag{4.25}$$

que es equivalente a:

$$\dot{q}^T \left(\sum_{i=1}^{n} \mathcal{R}_{Ei}^T \mathcal{R}_{Bi} \right) \dot{q} = \dot{q}^T N \dot{q} \tag{4.26}$$

donde

$$N = \begin{bmatrix} a & e & h & j \\ e & b & f & i \\ h & f & c & g \\ j & i & g & d \end{bmatrix} \tag{4.27}$$

y

$$\begin{aligned} a &= (S_{xx} + S_{yy} + S_{zz}) \\ b &= (S_{xx} - S_{yy} - S_{zz}) \\ c &= (-S_{xx} + S_{yy} - S_{zz}) \\ d &= (-S_{xx} - S_{yy} + S_{zz}) \end{aligned}$$

$$\begin{aligned} e &= S_{yz} - S_{zy} \\ f &= S_{xy} + S_{yx} \\ g &= S_{yz} + S_{zy} \\ h &= S_{zx} - S_{xz} \\ i &= S_{zx} + S_{xz} \\ j &= S_{xy} - S_{yx} \end{aligned}$$

con

$$S_{xx} = \sum_{i=1}^{n} p'_{Exi} p'_{Bxi} \qquad S_{xy} = \sum_{i=1}^{n} p'_{Exi} p'_{Byi} \tag{4.28}$$

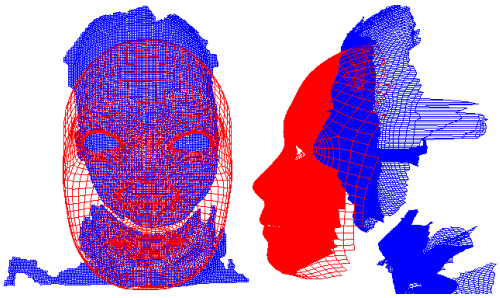
y así sucesivamente para el resto de los coeficientes S.

El quaternion unitario que maximiza $\dot{q}^T N \dot{q}$ es el eigenvector correspondiente al eigenvalor más grande de la matriz N. Una vez que se obtuvo el quaternion unitario se construye la matriz de rotación como:

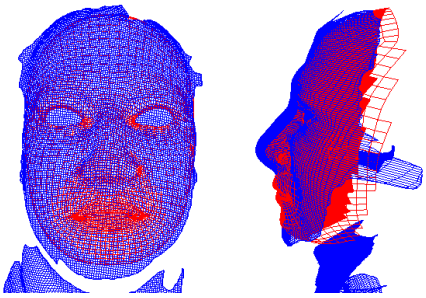
$$R = \begin{bmatrix} (q_0^2 + q_x^2 - q_y^2 - q_z^2) & 2\,(q_x q_y - q_0 q_z) & 2\,(q_x q_z + q_0 q_y) \\ 2\,(q_y q_x + q_0 q_z) & (q_0^2 - q_x^2 + q_y^2 - q_z^2) & 2\,(q_y q_z - q_0 q_x) \\ 2\,(q_z q_x - q_0 q_y) & 2\,(q_z q_y + q_0 q_x) & (q_0^2 - q_x^2 - q_y^2 + q_z^2) \end{bmatrix} \tag{4.29}$$

Para calcular la escala s sustituimos la matriz de rotación en la ecuación 4.19. Por último, se calcula la translación usando la ecuación 4.15

En la figura 4.16 se muestran los resultados de alinear una cara de entrada con la cara base. Para hacer la transformación rígida sólo se utilizaron los puntos característicos del interior de la cara (ojos, boca y nariz), tal como se mencionó al momento de describir el algoritmo de detección de puntos.



(a) Antes de alineación.



(b) Después de la alineación.

Figura 4.16: Alineación global. En rojo aparece la cara base y en azul la cara de entrada.

4.3.2. Deformación en el plano XY

Una vez realizada la alineación global de la cara de entrada, el siguiente paso consiste en deformar la cara base en el plano XY para hacer coincidir los puntos característicos de ambas caras. Este proceso de deformación consta de varios pasos, sin embargo, la idea subyacente es la misma: hacer un mapeo o re-escalamiento lineal de distancias, en dirección vertical u horizontal, de manera que la distancia entre cada par de puntos característicos de la cara base quede igual a las distancias correspondientes entre los mismos puntos en la cara de entrada. La función de mapeo utilizada es:

$$p_{Bi} = p_E^{min} + \left(p_E^{max} - p_E^{min}\right)\left(\frac{p_{Bi} - p_B^{min}}{p_B^{max} - p_B^{min}}\right) \tag{4.30}$$

donde p_{Bi} es el vértice i de la cara base que se está mapeando a su posición correspondiente en la cara de entrada, en dirección vertical u horizontal, mientras que p^{min} y p^{max} son los puntos característicos con valor mínimo y máximo, respectivamente, en la dirección de mapeo. Los subíndices $_B$ y $_E$ indican si el vértice pertenece a la cara base o a la cara de entrada, respectivamente. A continuación describimos cada uno de los pasos del proceso de deformación:

1. Se traslada la cara de entrada para hacer coincidir las puntas de la nariz de ambas caras con respecto al plano XY. Esto se expresa como:

$$p_{Exi} = p_{Exi} + (p_{Bx}^n - p_{Ex}^n)$$

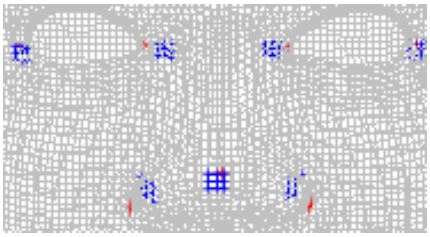
$$p_{Eyi} = p_{Eyi} + (p_{By}^n - p_{Ey}^n) \tag{4.31}$$

   donde p^n es la punta de la nariz; los índices x, y indican la dirección en la que se realiza la operación.

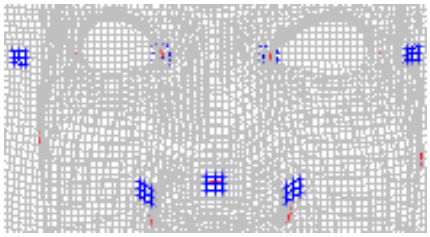
2. Se alinean las esquinas internas de los ojos en dirección horizontal, utilizando la función de mapeo (ecuación 4.30) con la punta de la nariz (p^n) y la esquina interna del ojo correspondiente. Para la esquina interna del ojo derecho (p^{ire}), $p^{min} = p_x^{ire}$ y $p^{max} = p_x^n$, mientras que para la esquina interna del ojo izquierdo (p^{ile}) se tiene que $p^{min} = p_x^n$ y $p^{max} = p_x^{ile}$. En la figura 4.17(b) se muestran los resultados de este mapeo.

3. Se alinean las esquinas externas de los ojos horizontalmente utilizando la ecuación 4.30, al igual que en el paso anterior. En este caso, para la esquina externa del ojo derecho (p^{ore}), $p^{min} = p_x^{ore}$ y $p^{max} = p_x^{ire}$, mientras que para la esquina externa del ojo izquierdo (p^{ole}) se tiene que $p^{min} = p_x^{ile}$ y $p^{max} = p_x^{ole}$. En la figura 4.17(c) se muestran los resultados.

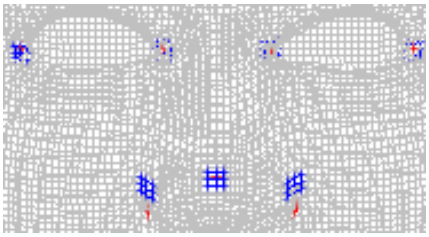
4. Se alinean los bordes de las mejillas en dirección horizontal siguiendo el mismo procedimiento con la ecuación 4.30. Para la mejilla derecha (p^{rc}), $p^{min} = p_x^{rc}$ y $p^{max} = p_x^{ore}$; para la mejilla izquierda (p^{lc}), $p^{min} = p_x^{ole}$ y $p^{max} = p_x^{lc}$.



(a) Antes del mapeo.



(b) Alineación de las esquinas internas.



(c) Alineación de las esquinas externas.

Figura 4.17: Alineación de las esquinas de los ojos. En rojo: puntos característicos de la cara base, en azul: puntos característicos de la cara de entrada.

5. Se alinean las esquinas de la nariz en dirección horizontal, pero esta vez cambiando un poco la mecánica. Aplicamos la misma ecuación 4.30, sólo que ahora movemos gradualmente una de las fronteras (p_x^{min} ó p_x^{max}) para disminuir la distancia de mapeo horizontal a medida que los vértices se alejan de las esquinas de la nariz, en dirección vertical, de manera que la distancia horizontal se hará cero al llegar a los ojos para evitar mover los vértices de los ojos que ya fueron alineados previamente, sucederá lo mismo para el caso de la boca.

   Para alinear la esquina derecha de la nariz p^{rn}, se toman los vértices que se encuentran entre la nariz y los ojos, en dirección vertical. Para los vértices que están entre la esquina derecha de la nariz y la punta de la nariz, en dirección horizontal, sustituimos p_E^{min} por p_i^{newrn} en la ecuación 4.30 y $p^{max} = p^n$.

$$p_i^{newrn} = p_{Bx}^{rn} + (p_{Ex}^{rn} - p_{Bx}^{rn}) \left(\frac{p_{Byi} - p_{By}^{ire}}{p_{By}^{rn} - p_{By}^{ire}} \right) \tag{4.32}$$

   Y para los vértices que están entre la esquina derecha de la nariz y la mejilla derecha, sustituimos p_E^{max} por p_i^{newrn} y $p^{min} = p^{rc}$. De forma similar, para los vértices entre la nariz y la boca:

$$p_i^{newrn} = p_{Bx}^{rn} + (p_{Ex}^{rn} - p_{Bx}^{rn}) \left(\frac{p_{Byi} - p_{By}^{rm}}{p_{By}^{rn} - p_{By}^{rm}} \right) \tag{4.33}$$

   donde p^{rm} es la esquina derecha de la boca.

   El caso de alineación de la esquina izquierda de la nariz p^{ln} se trata de la misma forma, sólo se cambian los puntos por los del lado izquierdo.

6. Se alinean las esquinas de la boca en dirección horizontal con el mismo proceso que se utilizó para detectar las esquinas de la nariz.

   Para alinear la esquina derecha de la boca p^{rm}, se toman los vértices que se encuentran entre la boca y la nariz, en dirección vertical. Para los vértices que están entre la esquina derecha de la boca y la esquina derecha de la nariz, en dirección horizontal, sustituimos p_E^{min} por p_i^{newrm} en la ecuación 4.30 y $p^{max} = p^n$.

$$p_i^{newrm} = p_{Bx}^{rm} + (p_{Ex}^{rm} - p_{Bx}^{rm}) \left(\frac{p_{Byi} - p_{By}^{rn}}{p_{By}^{rm} - p_{By}^{rn}} \right) \tag{4.34}$$

   Y para los vértices que están entre la esquina derecha de la boca y la mejilla derecha, sustituimos p_E^{max} por p_i^{newrm} y $p^{min} = p^{rc}$. De forma similar, para los vértices entre la boca y la barbilla:

$$p_i^{newrm} = p_{Bx}^{rm} + (p_{Ex}^{rm} - p_{Bx}^{rm}) \left(\frac{p_{Byi} - p_{By}^{c}}{p_{By}^{rm} - p_{By}^{c}} \right) \tag{4.35}$$

   donde p^c es el borde de la barbilla.

   El caso de alineación de la esquina izquierda de boca p^{lm} se maneja igual, sólo se cambian los puntos por los del lado izquierdo.

7. Se alinean los centroides de las esquinas de la boca, pero ahora en dirección vertical, utilizando la misma ecuación (4.30), sólo que esta vez $p^{min} = \left(p_y^{rm} + p_y^{lm}\right)/2$ y $p^{max} = p_y^n$

8. Se alinean los centroides de los ojos en dirección vertical, con $p^{min} = p_y^n$ y $p^{max} = \left(p_y^{ore} + p_y^{ire} + p_y^{ole} + p_y^{ile}\right)/4$.

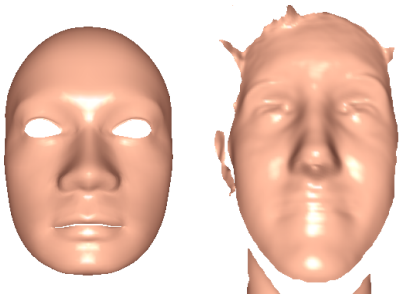
9. Se alinea el borde de la barbilla p^c en dirección vertical, con $p^{min} = p_y^c$ y $p^{max} = \left(p_y^{rm} + p_y^{lm}\right)/2$.

10. Se alinea el borde de la frente p^f en dirección vertical, utilizando para este caso $p^{min} = \left(p_y^{ore} + p_y^{ire} + p_y^{ole} + p_y^{ile}\right)/4$ y $p^{max} = p_y^f$.

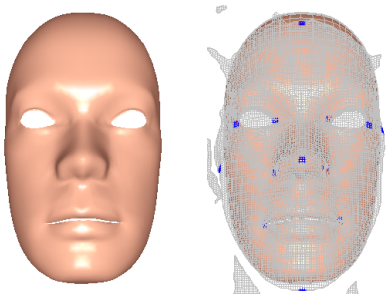
En la figura 4.18 se muestran los resultados de aplicar estos pasos para deformar la cara base y hacer una alineación de puntos característicos. Se puede ver que la estructura de la cara base se sigue conservando, sólo se modificaron las distancias entre sus puntos para adaptarse al tamaño y características de la cara de entrada.

4.3.3. Deformación en Z

Después de hacer la alineación global y la deformación en el plano XY, ahora lo único que falta es mover los vértices en el eje Z. Con esto conseguiremos que la cara base sea completamente igual a cualquier cara de entrada. El proceso de deformación en Z consiste en proyectar las caras cilíndricamente para después, por medio de



(a) Cara base. (b) Cara de entrada.



(c) Cara base deformada. (d) Puntos alineados.

Figura 4.18: Alineación en el plano XY. La cara base se deforma para adaptarse a los puntos de la cara de entrada.

interpolación lineal, localizar cada uno de los vértices de la cara base en la superficie de la cara de entrada.

Si tenemos un vértice $v = [v_x \ v_y \ v_z]$, calculamos sus coordenadas cilíndricas como:

$$v_x^c = \left(tan^{-1}\left(\frac{v_x}{v_z}\right)\right)\left(\frac{nc}{2\pi}\right)$$

$$v_y^c = v_y \tag{4.36}$$

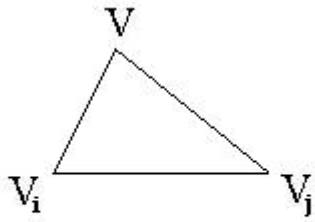
$$v_z^c = \sqrt{v_x^2 + v_z^2}$$

donde nc, es el ancho deseado para la malla, en nuestro caso 512. En la figura 4.19 se muestra la cara base proyectada cilíndricamente.

Para determinar a qué polígono (cuadrilátero o triángulo) de la malla de entrada proyectada en XY pertenecerá un vértice de la malla base se necesita hacer una prueba de membresía utilizando *funciones de borde* [27]. Una función de borde $E_{ij}(v)$ es positiva si v está del lado izquierdo de la línea $\overline{v_i v_j}$, la función más sencilla que cumple con esto es el *producto cruz*. Consideremos la figura 4.20, el producto cruz se puede utilizar para determinar el área del triángulo; si los vértices v_i, v_j, v están

Figura 4.19: Proyección cilíndrica de la cara base.

orientados en sentido contrario al de las manecillas del reloj entonces el signo del área será positivo, en caso contrario será negativo. Este signo sirve para determinar si el vértice v está a la izquierda (positivo) o a la derecha (negativo) de la línea $\overline{v_i v_j}$.



$$A(\Delta) = \tfrac{1}{2}\left|(\mathrm{V_j} - \mathrm{V_i}) \times (\mathrm{V} - \mathrm{V_i})\right|$$

Figura 4.20: Área del triángulo obtenida como el producto cruz de sus vértices.

Para un polígono triangular $[v_0, v_1, v_2]$, v estará adentro del triángulo si las tres funciones $E_{01}(v)$, $E_{12}(v)$ y $E_{20}(v)$ tienen signo positivo. En el caso de cuadriláteros habrá que evaluar cuatro funciones, $E_{01}(v)$, $E_{12}(v)$, $E_{23}(v)$ y $E_{30}(v)$, correspondientes a los lados de un cuadrilátero, más una quinta, $E_{20}(v)$, que hace la prueba contra una de las diagonales. El vértice estará dentro del cuadrilátero si se cumple cualquiera de estas expresiones lógicas:

$$E_{20} \wedge E_{01} \wedge E_{12} \wedge (E_{23} \vee E_{30})$$

$$(\neg E_{20}) \wedge E_{23} \wedge E_{30} \wedge (E_{01} \vee E_{12}) \tag{4.37}$$

Después de haber determinado el polígono al que pertenece el vértice v, el siguiente paso es calcular las *coordenadas baricéntricas* para interpolar la posición de v con

respecto al eje Z como:

$$v_z = \sum_{i=1}^{n} \beta_i v_{iz} \tag{4.38}$$

donde β_i son las coordenadas baricéntricas y n es el número de vértices v_i del polígono.

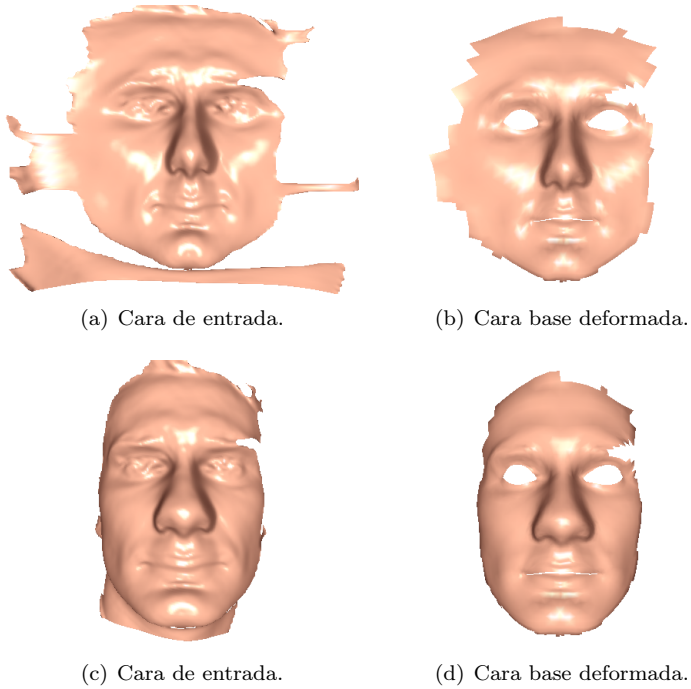
Para obtener las coordenadas baricéntricas utilizamos el algoritmo 4.2 extraido de [27].

Algoritmo 4.2 Coordenadas baricéntricas en 2D.

Datos: vértice v, polígono $V = [v_1, \ldots, v_n]$;

1: $s_i = v_i - v$
2: $A_i = s_i \times s_{i+1}$
3: $D_i = s_i \cdot s_{i+1}$
4: $r_i = \|s_i\|$
5: $t_i = (r_i r_{i+1} - D_i/A_i)$
6: $\mu_i = (t_{i-1} + t_i)/r_i$
7: $\beta_i = \mu_i / \sum_{j=1}^{n} \mu_j$

En la figura 4.21 se muestran los resultados de la deformación en el eje Z. Nótese que no se muestran los vértices de la cara base que no tienen correspondencia con algún polígono de la cara de entrada.



(a) Cara de entrada. (b) Cara base deformada.

(c) Cara de entrada. (d) Cara base deformada.

Figura 4.21: Deformación en el eje Z. 4.21(a) y 4.21(b) están proyectadas cilíndricamente.

4.3.4. Resultados

Después de esta etapa de ajuste ya podemos convertir la cara base en cualquier cara de entrada y generar expresiones faciales sintéticas, tal como se puede observar en la tabla 4.4.

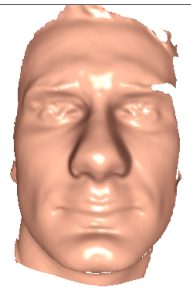
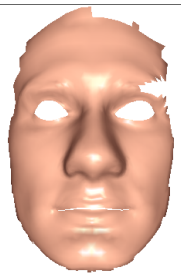
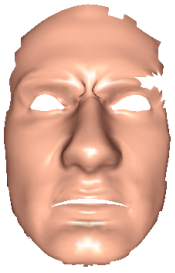
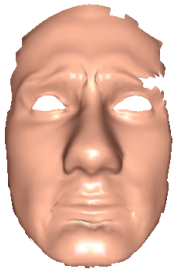
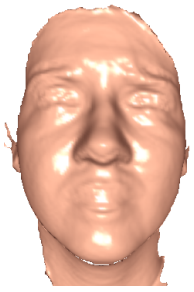
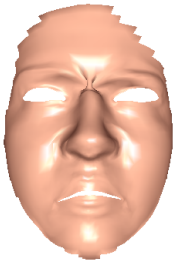
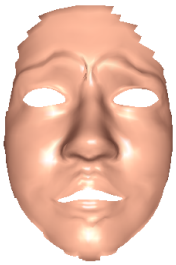
Entrada	**Ajustada**	***Enojo***	**Miedo**
			
			

Tabla 4.4: Ejemplos de expresiones faciales sintetizadas.

Capítulo 5

Reconocimiento facial

Finalmente, llegamos a la etapa más importante de nuestro trabajo: la construcción del sistema de reconocimiento facial en 3D. Después del trabajo desarrollado en capítulos previos contamos ya con todos los elementos necesarios para construir el sistema, el cuál describiremos en las secciones de este capítulo.

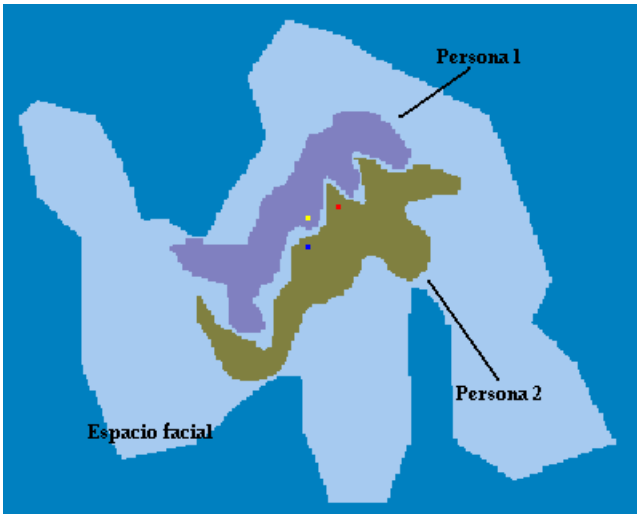
Como se estableció en el capítulo introductorio, el objetivo principal de este trabajo es atacar el problema de variación en la expresión facial para mejorar la tasa de reconocimiento. Para lograr esto "sembraremos" muestras sintéticas obtenidas a partir de la muestra neutral de los individuos de la galería con el propósito de abarcar un área más grande dentro del espacio facial de cada individuo, tal como se ilustra en la figura 5.1.

5.1. Creación de las imágenes de rango

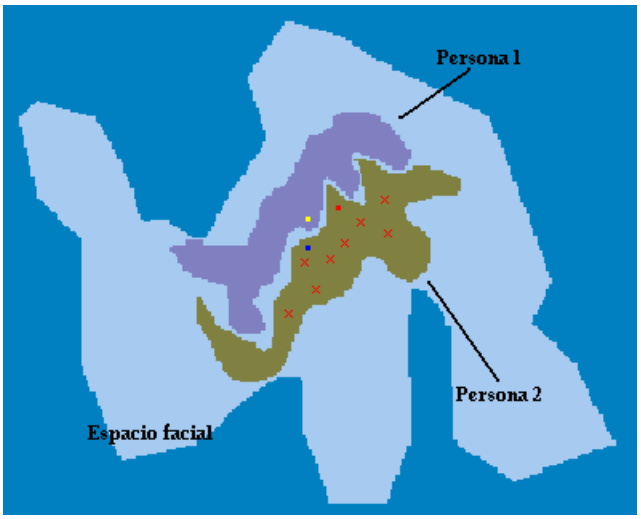
Para poder construir el sistema de reconocimiento facial aplicando técnicas de clasificación y aprendizaje estadístico, tales como PCA (*Eigenfaces*) y LDA (*Fisherfaces*), necesitamos que todas las mallas faciales (pruebas y galería) tengan la misma representación, ya que estas técnicas operan directamente en la representación de la imagen. A continuación se describe el proceso para generar esta representación como imagen de rango.

Comenzamos generando $9,891$ vértices acomodados en un plano de 130×150 mm (aproximadamente el tamaño promedio de la región facial) centrado en $Z = 0$, como se muestra en la figura 5.2. Los vértices de este plano se moverán únicamente en el eje Z para adaptarse a la superficie de la malla facial que se vaya a representar, implementando las mismas técnicas de funciones de borde y coordenadas baricéntricas que se utilizaron en la Sección 4.3.3 para deformar la cara base en Z.

Antes de mover los vértices del plano en dirección Z primero se alinea la punta de la nariz con el centro del plano, de esta manera todas las caras tendrán la punta de la nariz en la misma coordenada del plano. Para las caras de la galería, este punto



(a) Espacio original.



(b) Espacio "sembrado".

Figura 5.1: Inclusión de muestras sintéticas en el espacio facial. 5.1(a): La prueba perteneciente a la persona 2 (en azul) está más cerca de la muestra de la persona 1 (en amarillo) que de la muestra perteneciente a su misma clase (en rojo), por lo que cualquier métrica de distancia hará un reconocimiento incorrecto. 5.1(b): Al agregar las muestras sintéticas (cruces rojas) ahora la prueba está más cerca de una muestra sintética perteneciente a su misma clase, por lo que se incrementa la probabilidad de hacer un reconocimiento correcto.

ya fue determinado en la etapa de detección de puntos característicos. Sin embargo, para las caras que se utilizarán como pruebas no hay necesidad de aplicar síntesis de expresiones y por lo tanto no tendrán puntos característicos detectados. Dado que en este trabajo no nos enfocamos al problema de detección facial automática, haremos el proceso manualmente y supondremos que la cara que entra al sistema

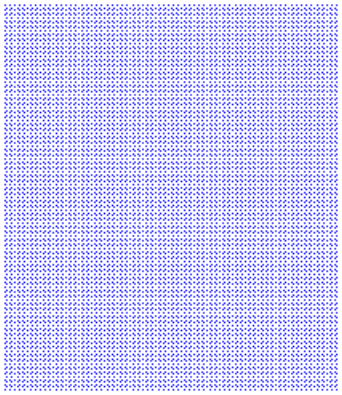
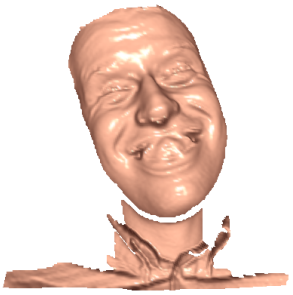


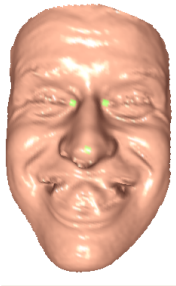
Figura 5.2: Plano inicial para la creación de las imágenes de rango.

para ser identificada ya pasó por un proceso de detección que nos entrega únicamente la región facial y 3 puntos clave: las dos esquinas internas de los ojos y la punta de la nariz. Utilizando el plano generado por estos 3 puntos se alineará la cara de prueba con respecto a los ejes, como se muestra en la figura 5.3.



(a) Cara de entrada.

(b) Cara después del preproceso.



(c) Cara alineada.

(d) Imagen de rango.

Figura 5.3: Preproceso aplicado a las caras de prueba. Se eliminan las regiones que no pertenezcan a la cara (5.3(b)) y se marcan las esquinas internas de los ojos y la punta de la nariz (en verde) para alinearla 5.3(c). 5.3(d): La esquina inferior negra da una idea de cuánto se deformó el plano inicial en en eje Z.

En la tabla 5.1 se muestran algunas imágenes creadas para la galería. Es importante mencionar que, en las imágenes de rango de la galería, la parte de los ojos se tomó de la imagen original, ya que si recordamos, la cara base utilizada para sintetizar expresiones tiene la región de los ojos hueca.

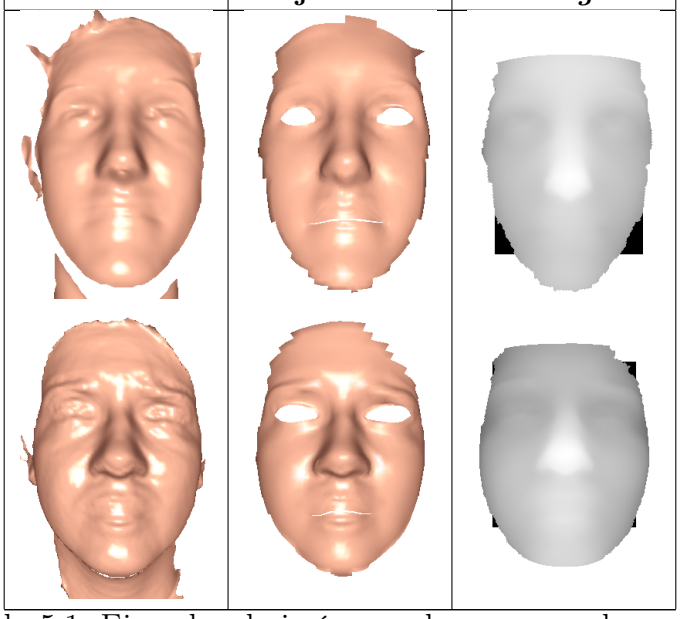
Entrada	**Ajustada**	***Rango***
		

Tabla 5.1: Ejemplos de imágenes de rango usadas en la galería.

5.2. Técnicas de clasificación

Una vez que se tiene la representación adecuada para las imágenes, la tarea restante consiste en utilizar técnicas de análisis de imagen para hacer el reconocimiento facial. Lo que queremos hacer es extraer la información relevante de la imagen facial, codificarla de la mejor manera posible, y hacer comparaciones en una base de datos con caras codificadas de la misma forma. Para este trabajo utilizaremos *Análisis de Componentes Principales* (PCA) y *Análisis Discriminante Lineal* (LDA).

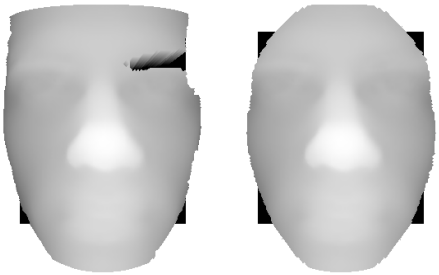
Las técnicas mencionadas son muy sensibles al ruido, por lo que implementamos dos pasos adicionales en las imágenes de rango para disminuir el ruido:

1. **Aplicación de una máscara elíptica:** Se utiliza una máscara elíptica para cortar las imágenes de rango y dejar únicamente las características centrales de la imagen quitando el ruido de la periferia.

2. **Aplicación de parches en agujeros:** Cualquier agujero que haya quedado dentro de la región delimitada por la máscara elíptica se parcha de la siguiente

manera: para cada vértice sin información se cambia su valor por el promedio de los vecinos que sí tienen información.

En la figura 5.4 se muestran los resultados de aplicar estos dos pasos en una imagen.



(a) Imagen original. (b) Imagen modificada.

Figura 5.4: Preproceso aplicado a las imágenes de rango. (5.3(b)): Imagen de rango original donde se puede apreciar el agujero en la ceja izquierda . 5.3(d): La misma imagen de rango después de aplicar la máscara elíptica y parchar el agujero.

5.2.1. Análisis de Componentes Principales

El *Análisis de Componentes Principales*(*PCA: Principal Components Analysis*) es una técnica que ha sido muy utilizada para reducción de la dimensionalidad en visión computacional, especialmente en reconocimiento facial [66], [5], [26], [24], [71]. Esta técnica, aplicada a reconocimiento facial, también se conoce como *Eigenfaces*.

En términos matemáticos, se trata de encontrar los componentes principales de la distribución facial, o los eigenvectores de la matriz de covarianza del conjunto de imágenes faciales de entrenamiento, tratadas como un punto o vector de un espacio de muy alta dimensión.

Estos eigenvectores o *eigenfaces* pueden ser vistos como un conjunto que caracteriza la variación entre imágenes faciales. Cada cara individual puede ser representada exactamente como una combinación lineal de *eigenfaces*; o bien, puede ser aproximada usando sólo los *eigenfaces* que corresponden a los eigenvalores más grandes, los cuáles representan la mayor cantidad de variabilidad en el conjunto de imágenes. Los M *eigenfaces* mayores generan un espacio M-dimensional (el espacio facial) de todas las posibles imágenes.

Supongamos que tenemos un conjunto de mallas faciales de entrenamiento Γ_1, Γ_2, $\ldots$, Γ_M, cada una de las cuáles está representada por un vector de dimensión N, donde N es el número de nodos ($9,891$ en nuestro caso, de acuerdo a la sección anterior) que contienen el valor de la coordenada Z. La malla facial promedio se define como

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n \tag{5.1}$$

Cada malla difiere de la malla promedio por el vector

$$\Phi_n = \Gamma_n - \Psi \tag{5.2}$$

Para aplicar PCA, buscamos un conjunto de M vectores ortonormales, u_n, los cuáles describan mejor la distribución de los datos. El k-ésimo vector, u_k, se escoge de manera que

$$\lambda_k = \frac{1}{M} \sum_{n=1}^{M} \left(u_k^T \Phi_n \right)^2 \tag{5.3}$$

es un máximo, sujeto a

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1, & si\ l = k \\ 0, & otro\ caso \end{cases} \tag{5.4}$$

Los vectores u_k y escalares λ_k son los eigenvectores y eigenvalores, respectivamente, de la matriz de covarianza

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T \tag{5.5}$$

donde $A = [\Phi_1\ \Phi_2\ \Phi_3\ \ldots\ \Phi_M]$. La matriz C es de tamaño $N \times N$, y calcular los N eigenvalores y eigenvectores es una tarea extremadamente demandante, por lo que necesitamos un método computacionalmente más accesible para encontrar estos eigenvectores.

Si el número de puntos en el espacio es menor que la dimensión del espacio ($M < N$), habrá solamente $M - 1$ eigenvectores significativos, en lugar de N (los eigenvectores restantes tendrán un eigenvalor asociado igual a cero). Afortunadamente podemos resolver para los N eigenvectores resolviendo primero para los eigenvectores de una matriz de $M \times M$ y después tomando la combinación lineal apropiada de las mallas faciales Φ_i. Consideremos los eigenvectores v_i de $A^T A$ tal que

$$A^T A v_i = \mu_i v_i \tag{5.6}$$

Premultiplicando ambos lados por A, tenemos

$$AA^T A v_i = \mu_i A v_i \tag{5.7}$$

donde podemos ver que Av_i son los eigenvectores de $C = AA^T$

Así pues, construimos la matriz $L = A^T A$ de tamaño $M \times M$, donde $L_{mn} = \Phi_m^T \Phi_n$, y encontramos los M eigenvectores, v_l de L. Estos vectores determinan las combinaciones lineales del conjunto M de mallas faciales para formar los *eigenfaces* u_l

$$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k, \quad l = 1,\ \ldots,\ M \tag{5.8}$$

Con este análisis los cálculos se reducen enormemente, del orden del número de nodos en las mallas (N), al orden del número de mallas en el conjunto de entrenamiento (M).

Finalmente, para hacer el reconocimiento facial, se proyectan las mallas faciales (galería y pruebas) en el espacio facial generado por los M *eigenfaces* con una simple multiplicación matricial:

$$\omega_k = u_k^T (\Gamma - \Psi), \quad k = 1, \ldots, M \tag{5.9}$$

donde u_k es el k-ésimo eigenvector y ω_k es el k-ésimo peso en el vector $\Omega^T = [\omega_1 \ \omega_2 \ \ldots \ \omega_M \ ]$. El vector Ω se toma como la "*clave facial*" que representa la cara de una persona en el espacio facial.

Para reconocer una malla facial se comparan todas las claves faciales de la galería contra la clave facial de la prueba utilizando la *distancia euclidiana* o la *distancia del coseno*,

$$d_{euclidiana} = \|\Omega_{prueba} - \Omega_{galeria}\| \tag{5.10}$$

y

$$d_{coseno} = 1 - \frac{\Omega_{prueba}^T \Omega_{galeria}}{\|\Omega_{prueba}\| \, \|\Omega_{galeria}\|} \tag{5.11}$$

respectivamente. Para determinar si una malla de prueba corresponde o no a un individuo de la galería se aplica un umbral a la distancia calculada. Cualquier comparación que produzca un valor de distancia menor al umbral se considera una aceptación.

5.2.2. Análisis Discriminante Lineal

Aunque la proyección de las mallas con PCA es adecuada para reducir la dimensionalidad del problema, las características que produce no necesariamente son buenas para discriminar entre clases distintas en un conjunto de muestras. Puesto que en nuestro sistema agregaremos mallas sintéticas para aumentar el conjunto de muestras por individuo, necesitaremos de una proyección que produzca una buena separación entre clases. La figura 5.5 muestra un ejemplo en dos dimensiones donde la proyección con PCA no puede distinguir entre clases de una población [62]. En esta figura, la proyección en el componente principal Y_1 es incapaz de separar dos clases obvias. Una proyección en Z_1 daría una separación clara. Esta separación se logra por medio del *Análisis Discriminante Lineal* (*LDA: Linear Discriminant Analysis*) o *Discriminante Lineal de Fisher* (*FLD: Fisher's Linear Discriminant*) que logra extraer las características óptimas para discriminar entre las clases representadas en el conjunto de entrenamiento.

El método de LDA, a diferencia de PCA, trata de generar la matriz de proyección de tal manera que se minimice la variación entre múltiples muestras de una misma

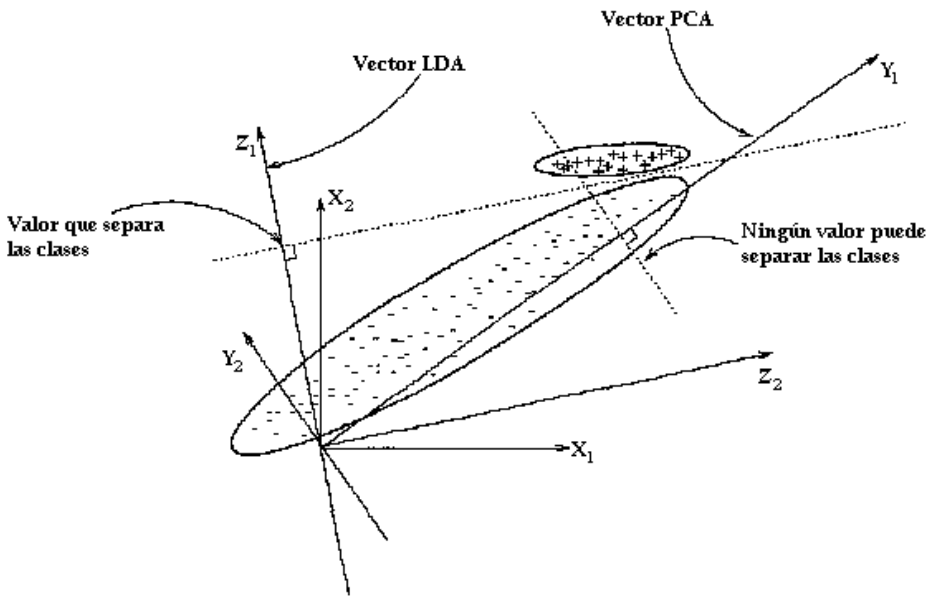


Figura 5.5: Problemas de la proyección con PCA para separación de clases [62].

persona (dispersión intraclase), al mismo tiempo que maximiza la separación entre clases (dispersión interclase).

Consideremos un conjunto de M muestras de entrenamiento Γ_1, Γ_2, ..., Γ_M en un espacio de N dimensiones, donde cada muestra pertenece a una de c clases distintas X_1, X_2, ..., X_c. Definamos la matriz de dispersión interclase como

$$S_B = \sum_{n=1}^{c} |X_n| \left(\Psi_n - \Psi\right) \left(\Psi_n - \Psi\right)^T \tag{5.12}$$

y la matriz de dispersión intraclase como

$$S_W = \sum_{n=1}^{c} \sum_{i=1}^{|X_n|} \left(\Gamma_{ni} - \Psi_n\right) \left(\Gamma_{ni} - \Psi_n\right)^T \tag{5.13}$$

donde X_n es la n-ésima clase formada por las muestras pertenecientes a la n-ésima persona:

$$X_n = [\Gamma_{n1}\ \Gamma_{n2}\ \Gamma_{n3}\ \ldots] \tag{5.14}$$

y Ψ, Ψ_n son el promedio total y el promedio por clase, respectivamente:

$$\Psi = \frac{1}{\sum_{m=1}^{c} |X_m|} \sum_{n=1}^{c} \sum_{i=1}^{|X_n|} \Gamma_{ni} \tag{5.15}$$

$$\Psi_n = \frac{1}{|X_n|} \sum_{i=1}^{|X_n|} \Gamma_{ni} \tag{5.16}$$

Si S_W es no singular, la proyección óptima U_{opt} es escogida como la matriz con columnas ortonormales que maximiza la razón entre el determinante de la matriz de dispersión interclase y el determinante de la matriz de dispersión intraclase:

$$U_{opt} = \underset{U}{\text{máx}} \frac{\left|U^T S_B U\right|}{\left|U^T S_W U\right|} \tag{5.17}$$

y U_{opt} resulta ser la matriz de eigenvectores de $S_W^{-1} S_B$:

$$S_B u_i = \lambda S_W u_i \quad \Rightarrow \quad S_W^{-1} S_B u_i = \lambda u_i \tag{5.18}$$

Sin embargo, en el problema de reconocimiento facial, la matriz S_W siempre es singular, debido a que el rango de S_W es cuando mucho $M - c$, y, en general, el número de muestras en el conjunto de entrenamiento es mucho más pequeño que la dimensión de los datos ($M < N$). Para superar esta complicación se proyecta el conjunto de entrenamiento a un espacio de dimensión menor de manera que la matriz intraclase resultante sea no singular. Esto se consigue aplicando PCA para reducir la dimensión del espacio a $M - c$, y después se aplica LDA para reducir la dimensión a $c - 1$. Este método se conoce como *Fisherfaces* [5], [25]. De manera formal, U_{opt} está dada por

$$U_{opt} = U_{fld} U_{pca} \tag{5.19}$$

$$U_{pca} = \underset{U}{\text{máx}} \left(\left|U^T S_T U\right|\right) \tag{5.20}$$

$$U_{fld} = \underset{U}{\text{máx}} \frac{\left|U^T U_{pca}^T S_B U_{pca} U\right|}{\left|U^T U_{pca}^T S_W U_{pca} U\right|} \tag{5.21}$$

donde S_T es la matriz de dispersión total:

$$S_T = \sum_{n=1}^{c} \sum_{i=1}^{|X_n|} \left(\Gamma_{ni} - \Psi\right) \left(\Gamma_{ni} - \Psi\right)^T \tag{5.22}$$

Para hacer el reconocimiento facial, se sigue la misma mecánica que con *eigenfaces*, se proyectan las mallas faciales (galería y pruebas) en el espacio facial generado por los $c - 1$ *fisherfaces* con una multiplicación matricial:

$$\Omega = \left(\Gamma - \Psi\right)^T U_{opt} \tag{5.23}$$

y la "*clave facial*" Ω se compara utilizando la distancia euclidiana (ecuación 5.10) o la distancia del coseno (ecuación 5.11).

5.3. Implementación del sistema y resultados experimentales

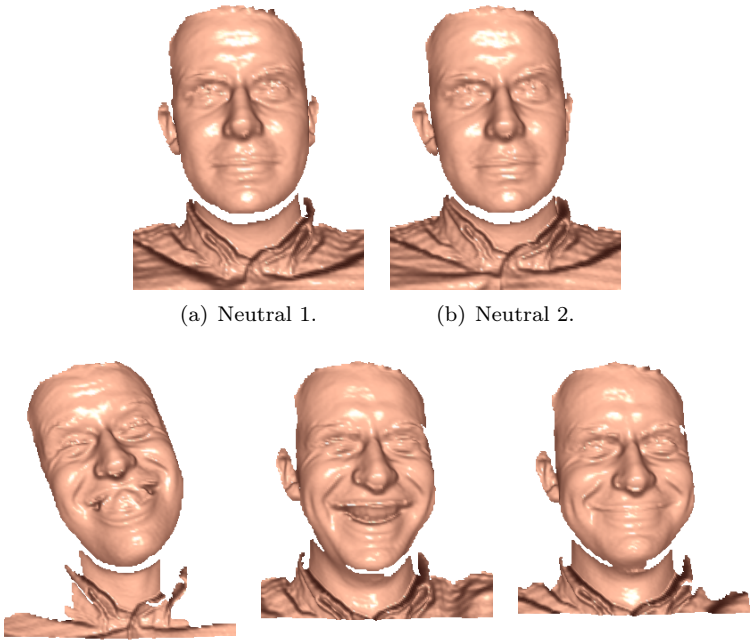
A continuación presentamos los resultados obtenidos para evaluar la efectividad de nuestro sistema de reconocimiento facial.

5.3.1. Implementación

Se trabajó en una computadora portátil con un procesador *Intel Pentium 4-m* a 1.8 GHz y 512 MB de memoria RAM. La aplicación desarrollada, *FaceER* (**Face** **E**xpression and **R**ecognition), se programó en C++ con la herramienta *Borland C++ Builder 5.0* y haciendo llamadas a funciones de *OpenGL* y *Matlab*. Para una descripción más detallada revise el Apéndice B.

5.3.2. Experimentos

Para hacer los experimentos de identificación se crearon dos conjuntos de imágenes: uno de entrenamiento y el otro de prueba. Las mallas faciales usadas en la generación de estos conjuntos pertenecen a la base de datos GavabDB [48] (ver Apéndice A). Se tomaron las 5 imágenes frontales de cada individuo, de las cuales 2 tienen expresión neutral y las 3 restantes tienen variaciones en la expresión, como se muestra en la figura 5.6. Para el conjunto de entrenamiento se tomó una de las dos imágenes con expresión neutral y las 4 restantes se tomaron para el conjunto de prueba. A ambos conjuntos se aplicó el procedimiento descrito en la Sección 5.1 para crear las imágenes de rango.



(a) Neutral 1. (b) Neutral 2.

(c) Gesto. (d) Risa. (e) Sonrisa.

Figura 5.6: Imágenes en el conjunto de datos.

En total, el conjunto de datos consta de 285 imágenes de 57 personas distintas (57 imágenes para entrenamiento y 228 para el conjunto de prueba). La base de datos

tiene 61 personas distintas, sin embargo, se quitaron 4 personas porque no se logró hacer una buena detección de puntos característicos debido a la mala calidad de las mallas con expresión neutral.

En un primer experimento y para propósitos de comparación, aplicamos *eigenfaces* a las 57 imágenes de entrenamiento, sin aumentar el conjunto con síntesis de expresiones. En este caso sólo se tiene una muestra por individuo, por lo que *eigenfaces* y *fisherfaces* producen el mismo resultado, ya que maximizar la variabilidad del conjunto equivale a maximizar la distancia entre clases de una sola muestra. Para proveer de un mecanismo de comparación directa entre *eigenfaces* y *fisherfaces*, en todos los experimentos tomamos los $(c-1)$ vectores más grandes para generar el espacio facial y utilizamos las dos métricas de distancia: la distancia euclidiana y la distancia del coseno. En la figura 5.7 se muestra la curva CMC para medir el desempeño del sistema al tratar de identificar las 228 imágenes del conjunto de prueba. Como se puede apreciar, a pesar de que estamos trabajando con un conjunto de datos distinto, el comportamiento de la curva es muy similar al de la figura 2.7, donde el desempeño del sistema de reconocimiento resulta afectado por los cambios de expresión.

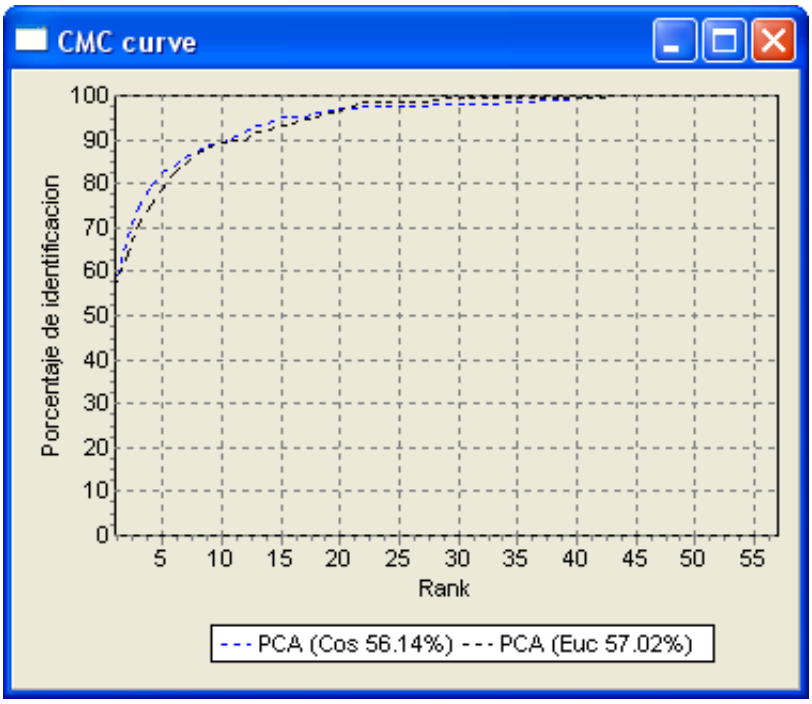


Figura 5.7: Desempeño del sistema sin aumento de síntesis de expresiones.

En otro experimento se agregaron 18 imágenes sintéticas por individuo al conjunto de entrenamiento. Estas 18 imágenes corresponden al número de expresiones sintéticas predefinidas que *FaceER* puede generar (ver tabla 3.2). El conjunto de entrenamiento constará de 1083 imágenes en total. Para manejar los ligeros cambios de pose que surgen en imágenes frontales, las 18 imágenes fueron agregadas con variaciones en la

pose, de la siguiente manera:

- 6 imágenes frontales (sin rotación)
- 4 imágenes rotadas en el plano en 5° y 10°: 2 hacia la derecha y 2 hacia la izquierda.
- 4 imágenes rotadas en 5° y 10°, 2 hacia el lado izquierdo y 2 hacia el lado derecho.
- 4 imágenes rotadas en 5° y 10°, 2 hacia arriba y 2 hacia abajo.

Los resultados de este experimento para las mismas 228 imágenes de prueba se muestran en la figura 5.8. Como se puede apreciar, utilizando *fisherfaces* y agregando expresiones sintetizadas al conjunto de entrenamiento, el sistema mejora el desempeño sustancialmente, de 56.14 % a 84.65 %.

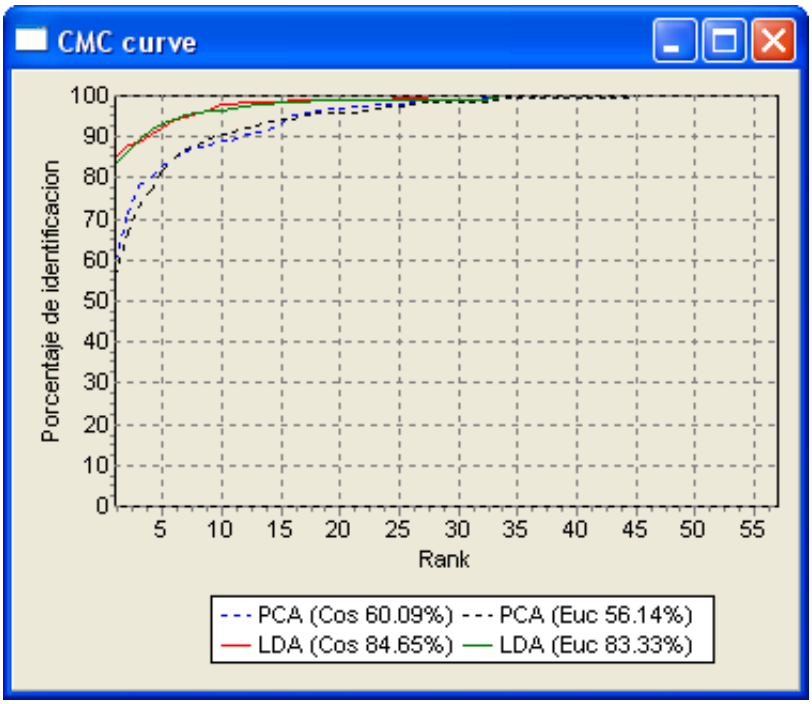


Figura 5.8: Desempeño del sistema con aumento de síntesis de expresiones. Al agregar imágenes con expresiones sintetizadas y tomar en cuenta la separación entre clases (LDA), el desempeño del sistema mejora sustancialmente.

En la tabla 5.2 se resumen los resultados de estos experimentos. En la última columna podemos ver que el método de *Fisherfaces* con aumento de síntesis de expresiones consume más tiempo de procesamiento en la etapa de entrenamiento, sin embargo, este incremento en el tiempo no es tan importante, ya que para una aplicación real, la etapa de entrenamiento es una tarea que normalmente se realiza fuera de

Método	Reconocimiento (%)		Tiempo de entrenamiento
	D. Cos.	D. Euc.	
Eigenfaces (sin aumento)	56.14	57.02	30 seg.
Eigenfaces (con aumento)	60.09	56.14	4.5 min.
Fisherfaces (con aumento)	84.65	83.33	13 min.

Tabla 5.2: Tabla comparativa de los experimentos realizados.

Proceso	Tiempo
Detección de puntos	0.5 seg
Eliminación de ruido (20 iteraciones del algoritmo propio)	Entre 2 y 4 seg.
Ajuste de la cara base	Entre 7 y 12 seg.
Síntesis de expresiones (Procesamiento con FEM)	2 min.
Síntesis de expresiones (Mapeo de desplazamientos)	0.5 seg.
Generar conjunto de entrenamiento	Entre 60 y 80 seg.
Generar imagen para reconocimiento	Entre 15 y 20 seg.

Tabla 5.3: Tiempos registrados en las etapas previas al reconocimiento. Los tiempos están calculados con respecto a una cara de entrada.

línea (*off-line*). Con este mismo enfoque, la tabla 5.3 muestra los tiempos consumidos en cada etapa o proceso previo al reconocimiento.

Cabe mencionar que el tiempo consumido por los procesos de la tabla 5.3 depende en gran medida del número de vértices que tenga la cara de entrada. Por otra parte, todos los procesos, a excepción del último (generar imagen para reconocimiento) se pueden realizar fuera de línea, por lo que para una aplicación real, sólo habría que mejorar el tiempo del último proceso.

5.3.3. Discusión

Aunque el sistema de reconocimiento facial presentado maneja efectivamente las variaciones en expresión facial y mejora la tasa de reconocimiento al incluir imágenes sintéticas en el conjunto de entrenamiento, aún es necesario ampliar el método para manejar variaciones pronunciadas en pose, ya que en este trabajo solo utilizamos mallas faciales de frente y con ligeras variaciones en pose.

Para llevar a cabo una comparación con otros métodos reportados en la literatura es necesario hacer una evaluación en igualdad de condiciones. Sin embargo, como se mencionó en la Sección 2.4, aún hay muchas ambigüedades a la hora de hacer comparaciones, ya que cada autor evalúa su método utilizando conjuntos de datos distintos en tamaño y complejidad, por lo que no se puede hacer una comparación directa entre métodos.

La única comparación aproximada de nuestro método la podemos hacer contra el trabajo de Moreno et. al. [47], puesto que utilizan la misma base de datos; aún así, recordemos que nosotros utilizamos solamente 57 de las 61 personas distintas en el conjunto. Ellos reportan una tasa de reconocimiento de 78 % para las imágenes frontales con expresión neutral, 62 % para las que tienen expresión de sonrisa, y un porcentaje más bajo para las imágenes en las que la persona presenta un gesto o la risa más pronunciada, por lo que, incluso tomando su mejor porcentaje reportado (78 %), nuestro método arroja mejores resultados (84.65 %).

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

En este trabajo se presentó un método para hacer reconocimiento facial en 3D utilizando un esquema de síntesis de expresiones para aumentar la galería a partir de una sola muestra de entrenamiento por persona. Dentro del trabajo realizado se pueden destacar los siguientes puntos:

- Se utilizó el método de los elementos finitos para simular la acción de los músculos en la generación de expresiones faciales.
- Se propuso un algoritmo iterativo simple y efectivo, para eliminación de ruido en mallas tridimensionales.
- Se propuso un algoritmo heurístico para la detección automática de puntos característicos en caras frontales utilizando únicamente descriptores de curvatura e información estadística.
- Se presentó un proceso de ajuste por etapas para transformar la cara base y sintetizar expresiones faciales en cualquier cara de entrada.
- Se probó un sistema de reconocimiento facial en 3D basado en *Fisherfaces* que efectivamente mejora la tasa de reconocimiento al aumentar el conjunto de entrenamiento con expresiones sintetizadas.

6.2. Trabajo futuro

Para completar y mejorar el trabajo realizado, aún quedan varias cosas por hacer:

- Crear el módulo de detección facial en 3D para completar el sistema de reconocimiento.

- Afinar el modelo anatómico utilizado para crear expresiones sintéticas más realistas.
- Seguir trabajando en el algoritmo iterativo de eliminación de ruido para reducir el número de iteraciones necesarias para obtener buenos resultados.
- Idear alguna forma para disminuir el tiempo de creación de imágenes de rango para el conjunto de entrenamiento.
- Buscar otras bases de datos de uso más difundido, como el *FGRC v2*, para poder realizar más pruebas y comparaciones del método propuesto.
- Incorporar información en 2D (color).

Apéndice A

La base de datos GavabDB

GavabDB [48] es una base de datos facial creada por el Departamento de Informática, Estadística y Telemática de la Universidad Rey Juan Carlos, en España, la cuál se construyó para ser usada en experimentos de reconocimiento facial que requieran de imágenes tridimensionales. La base de datos contiene imágenes en 3D de la superficie facial representada como una malla tridimensional. El objetivo era construir una base de datos lo suficientemente grande con respecto al número de individuos y que a la vez tuviera variaciones acentuadas de la misma persona.

Durante el proceso de adquisición se tomaron 9 imágenes distintas de cada persona:

- 2 vistas frontales de la cara con expresión neutral.
- 2 vistas de perfil (izquierdo y derecho), donde el individuo rotó su cabeza aproximadamente 90° sobre el eje vertical en ambas direcciones.
- 2 rotaciones en dirección vertical, aproximadamente 35° hacia arriba y 35° hacia abajo, respectivamente.
- 3 vistas frontales con variaciones en la expresión: sonrisa, risa pronunciada y un gesto elegido al azar por la persona.

No se consideraron variaciones con respecto a la presencia de lentes, sin embargo se permitieron oclusiones con la mano o con la lengua, además de que había individuos con barba, bigote, etc. Por otra parte, se omitió la información de textura (color), dejando únicamente las mallas tridimensionales.

En total, la base de datos contiene 549 imágenes de 61 personas distintas (45 hombres y 16 mujeres). Todas las personas son de raza caucásica y su edad oscila entre los 18 y los 40 años.

Para obtener las imágenes se utilizó un digitalizador *Minolta* $VI - 700$, un sensor laser que captura la imagen de rango y la información de textura en menos de un segundo. Las imágenes fueron tomadas directamente del digitalizador, es decir que no se construyeron mediante ningún proceso de registro de varias imágenes capturadas

en distintos ángulos. No se hizo ningún esfuerzo para controlar las condiciones de iluminación durante el proceso de captura, la luz predominante fue la luz solar, aunque había también una luz artificial, típica de los ambientes de oficina.

Los individuos fueron colocados a 1.5 ± 0.5 metros de distancia del digitalizador y podían mover su cabeza normalmente. Por esta razón hay cambios en la resolución entre las imágenes. La altura de la silla podía ser cambiada para que la cabeza quedara a la altura del digitalizador. Para capturar las distintas poses, los individuos dirigieron su vista hacia ciertos puntos marcados, pero sin forzarlos a mantener un ángulo preciso o su posición fija.

Las mallas están formadas por puntos de la superficie facial, los cuales están conectados formando cuadriláteros. Las coordenadas (x, y, z) de cada punto están referidas al sistema coordenado cuyo origen está en el dispositivo digitalizador al momento de la captura. Existen varios errores en las mallas debido a agujeros, oclusiones, contorno facial incompleto, ruido, etc., los cuáles no fueron removidos para dar un panorama más real del proceso de adquisición con los dispositivos actuales.

La base de datos se puede obtener en http://gavab.escet.urjc.es.

Apéndice B

FaceER

FaceER (**Face E**xpression and **R**ecognition) es una aplicación desarrollada para servir como herramienta básica en este trabajo de tesis. Se programó en C++ utilizando *Borland C++ Builder 5.0* y haciendo llamadas a funciones de *OpenGL* y *Matlab*.

FaceER es una herramienta muy completa que realiza todo el procesamiento requerido para el método de reconocimiento facial desarrollado en esta tesis. Como se puede ver en la figura B.1, el área principal de la aplicación es un entorno tridimensional hecho en *OpenGL*, el cuál podemos manipular fácilmente con los botones del ratón. Arrastrando el botón izquierdo podemos rotar el entorno, mientras que con el botón derecho podemos alejar o acercar los objetos.

A continuación mostraremos cada una de las funciones de la aplicación describiendo brevemente las opciones de su menú principal.

Menú Archivo

El menú Archivo contiene los submenús clásicos presentes en casi todas las aplicaciones, los cuales son:

- **Abrir:** Abre un cuadro de diálogo donde se puede seleccionar la malla facial que se desea abrir. Los formatos de archivo para mallas tridimensionales que maneja *FaceER* son: *VRML 1.0* (*.vrml), *GiD mesh ascii* (*.msh), *Wavefront OBJ* (*.obj) y un formato propio (*.rif) para imágenes de rango creadas en la misma aplicación.
- **Cerrar:** Cierra cualquier archivo que haya sido abierto previamente.
- **Guardar, Guardar como...:** Guarda cualquier cambio que haya sido hecho en la cara base en un archivo con alguno de los formatos descritos, excepto el formato propio, que se utiliza para generar las imágenes de rango utilizadas en el módulo de reconocimiento. Es importante mencionar que no guarda los cambios

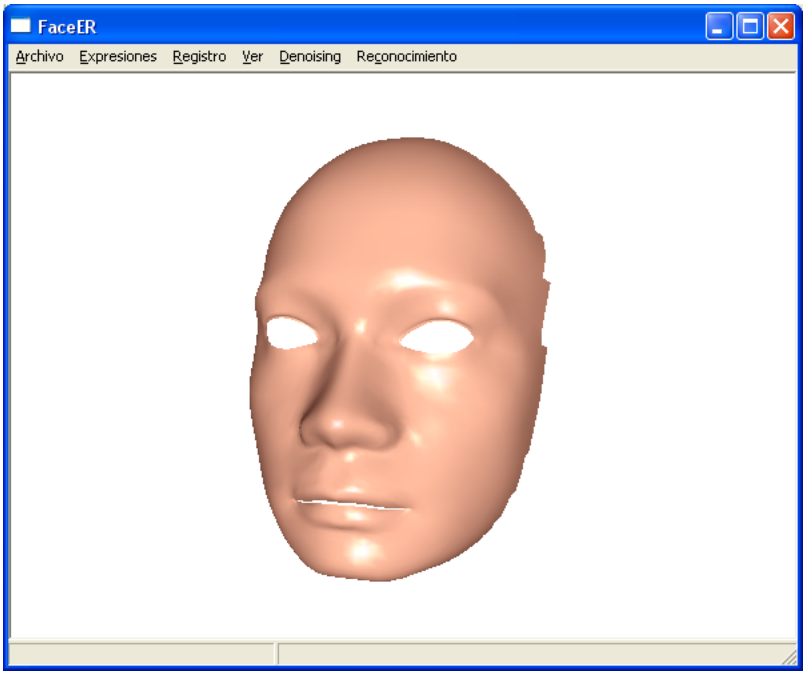


Figura B.1: Ventana principal de *FaceER*.

hechos en las caras de entrada, ya que todo el procesamiento lo realiza en la cara base que viene por default en la aplicación.

- **Salir:** Sale de la aplicación.

Menú Expresiones

Este menú contiene básicamente toda la funcionalidad de la parte de síntesis de expresiones faciales, convirtiendo a *FaceER* en un editor de expresiones. Consta de dos opciones:

- **Editar expresión:** Esta opción abre la ventana de edición de expresiones (figura B.2). Al pasar el ratón sobre el nombre de un músculo o una *Action Unit*, dependiendo de la **Vista** seleccionada, se marcará en rojo el músculo o músculos correspondientes. Las barras de desplazamiento sirven para editar la cantidad de fuerza (normalizada a un valor entre 0 y 1) que aplicará cada músculo para generar una expresión. En la caja de selección rotulada como **Expresión Predefinida** se puede seleccionar una de las 18 expresiones predefinidas para obtener los parámetros de fuerza en forma automática. Una vez que se definieron los parámetros de fuerza de cada músculo, el botón **Aceptar** inicia el procesamiento con elementos finitos para generar la expresión deseada en la cara base. Dicho procesamiento tarda aproximadamente 2 minutos.

- **Expresiones predefinidas:** Al seleccionar esta opción se despliega un submenú con las 19 expresiones predefinidas en la aplicación. La ventaja de esta opción es que genera la expresión casi en forma instantánea, pues en vez de aplicar todo el procesamiento con elementos finitos, únicamente agrega a la cara base el campo de desplazamientos predefinido para cada expresión. Estos desplazamientos se encuentran almacenados en archivos de la aplicación.

Como ya dijimos, todo el procesamiento se realiza en la cara base, por lo que si se desea sintetizar expresiones a una cara de entrada, primero habrá que transformar la cara base en la cara de entrada utilizando las opciones del menú **Registro** que se explicará a continuación.

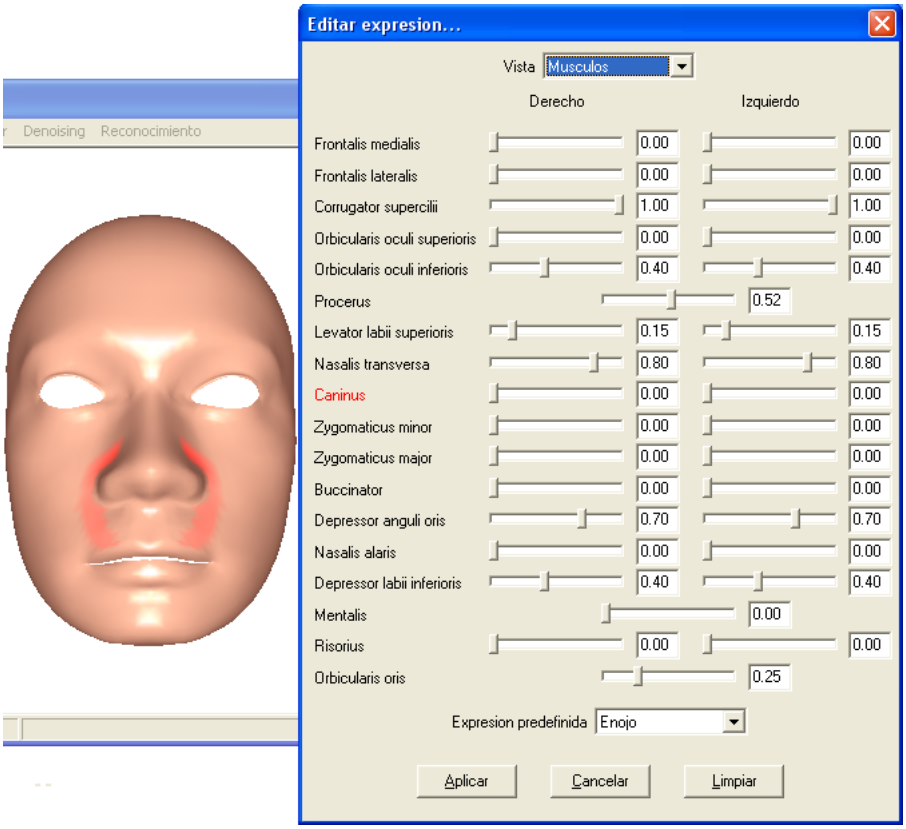


Figura B.2: Editor de expresiones de *FaceER*.

Menú Registro

El menú **Registro** contiene las opciones necesarias para transformar la cara base en la cara de entrada.

- **Puntos característicos:** Detecta los puntos característicos en la cara de entrada siguiendo el algoritmo descrito en la Sección 4.2.3.
- **Registrar cara:** Aplica el proceso mostrado en el Capítulo 4 para ajustar la cara base a la cara de entrada.

Menú Ver

Este menú contiene varias opciones de visualización y personalización del entorno:

- **Puntos, Alambrado, Sólido:** Estas 3 opciones son excluyentes entre sí y sirven para mostrar la cara de entrada como un conjunto de puntos, una malla alambrada, o un sólido, respectivamente.
- **Iluminación:** Sirve para quitar o poner iluminación en las caras.
- **Smooth shading:** Si está activado muestra los polígonos suavizados, en caso contrario quita el efecto de suavidad mostrando los bordes entre los polígonos.
- **Color de fondo, Color de luz y Color del objeto:** Estas opciones abren un cuadro de diálogo que permite cambiar el color del fondo, de la fuente de iluminación, y del objeto, respectivamente.
- **Cara base, Cara de entrada:** Muestran u ocultan la cara respectiva.
- **Vista Inicial:** Pone el entorno en estado inicial, quitando cualquier rotación o acercamiento realizado con el ratón.
- **Operaciones en falso color:** Esta opción despliega un submenú con varios descriptores de curvatura. Al aplicar alguno de estos descriptores, las caras se muestran en una escala de falso color que indica el valor del descriptor de curvatura en cada vértice (figura B.3). También aparece una ventana con controles que permiten aplicar umbrales para ocultar o mostrar ciertas características de la cara basándose en el descriptor de curvatura aplicado.

Menú Denoising

Las opciones del menú **Denoising** sirven para aplicar algún algoritmo de eliminación de ruido a la cara de entrada. Estos algoritmos son:

- **Algoritmo Laplaciano**
- **Filtro de mediana**
- **Algoritmo propio**

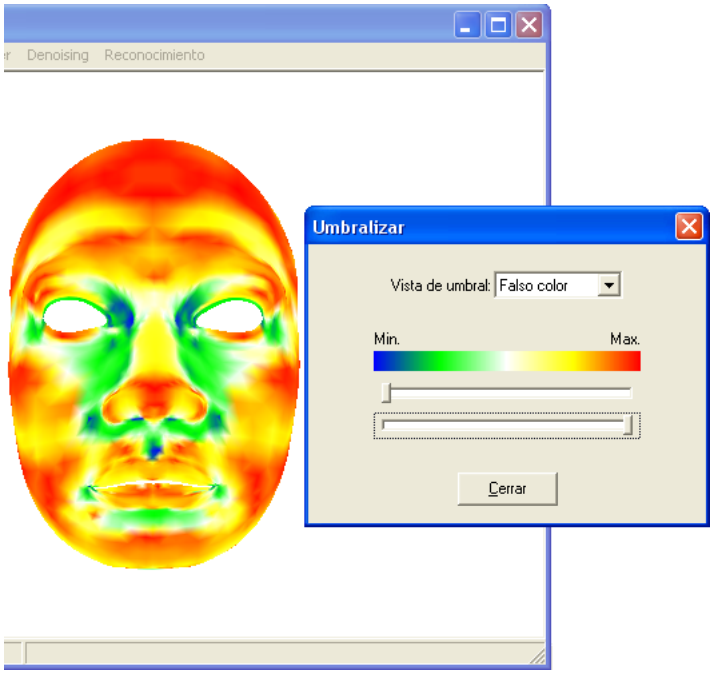


Figura B.3: Ejemplo de descriptor de curvatura en falso color (*Shape Index*).

- **Algoritmo anisotrópico Laplaciano**
- **Algoritmo bilateral**
- **Algoritmo HC**

Todas las opciones aplican una iteración del algoritmo respectivo, únicamente la opción **Algoritmo propio** aplica 20 iteraciones al presionar la opción una sola vez, puesto que se observó que el algoritmo arroja buenos resultados con 20 iteraciones.

Menú Reconocimiento

Este menú contiene toda la funcionalidad de la parte de reconocimiento facial. Sus opciones son:

- **Generar conjunto de entrenamiento:** Esta opción permite agregar las 19 imágenes de rango de una cara de entrada, previamente ajustada a la cara base, al conjunto de entrenamiento del sistema de reconocimiento.
- **Entrenar clasificador**: Esta opción abre una ventana donde hay que seleccionar el directorio raiz de la base de datos facial (dbRoot). Una vez hecho esto, el sistema de reconocimiento será entrenado con el conjunto que se encuentra almacenado en el directorio dbRoot/FaceER/trainSet

- **Alinear cara:** Este opción alinea una cara de entrada para agregarla al conjunto de prueba del sistema de reconocimiento. Para alinear la cara hay que seleccionar las esquinas internas de los ojos y la punta de la nariz en el siguiente orden: esquina interna del ojo derecho, esquina interna del ojo izquierdo, punta de la nariz. Los puntos se ubican arrastrando el botón central del ratón y se marcan con el botón derecho del ratón, mientras se sigue presionando el botón central.
- **Agregar cara al conjunto de prueba:** Esta opción genera la imagen de rango de la cara de entrada, previamente alineada con la opción anterior, para agregarla al conjunto de prueba del sistema de reconocimiento.
- **Reconocer conjunto de prueba:** Esta opción abre una ventana donde hay que seleccionar el directorio raiz de la base de datos facial (dbRoot). Una vez hecho esto, el sistema de reconocimiento será probado con el conjunto que se encuentra almacenado en el directorio dbRoot/FaceER/testSet

Sólo resta mencionar que para el entrenamiento del sistema de reconocimiento facial se hicieron llamadas a *Matlab* para agilizar los cálculos en los métodos de *Eigenfaces* y *Fisherfaces*.

Bibliografía

[1] B. Achermann, X. Jiang, H. Bunke. *Face Recognition using Range Images.* International Conference on Virtual Systems and Multimedia, 129-136, 1997.

[2] B. Achermann, H. Bunke. *Classifying Range Images of Human Faces with Hausdorff Distance.* 15th International Conference on Pattern Recognition (ICPR'00), 2:809-813, 2000.

[3] Y. Adini, Y. Moses, S. Ullman. *Face Recognition: The Problem of Compensating for Changes in Illumination Direction.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):721-732, 1997.

[4] L. Akarun, B. Gökberk, A. Salah. *3D Face Recognition for Biometric Applications.* 13th European Signal Processing Conference (EUSIPCO), September 2005.

[5] P. Belhumeur, J. Hespanha, D. Kriegman. *Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):711-720, 1997.

[6] S. Botello, H. Esqueda, F. Gómez, M. Moreles, E. Oñate. *MEFi. Módulo de Aplicaciones del Método de los Elementos Finitos.* Monografía M-AUGTO-2, Guanajuato, Gto., Noviembre 2004.

[7] K. Bowyer, K. Chang, P. Flynn. *A Survey of Approaches and Challenges in 3D and Multi-modal 3D+2D Face Recognition.* Computer Vision and Image Understanding, 101(1):1-15, 2006.

[8] A. Bronstein, M. Bronstein, R. Kimmel. *Three-Dimensional Face Recognition.* International Journal of Computer Vision, 64(1):5-30, 2005.

[9] J. Cartoux, J. LaPreste, M. Richetin. *Face Authentication or Recognition by Profile Extraction from Image Ranges.* Proceedings of the Workshop on Interpretation of 3D Scenes, 194-199, 1989.

[10] K. Chang. *New Multi-Biometric Approaches for Improved Person Identification..* Ph.D. thesis, University of Notre Dame, 2004.

[11] K. Chang, K. Bowyer, P. Flynn. *Adaptive Rigid Multi-Region Selection for Handling Expression Variation in 3D Face Recognition.* 2005 IEEE Computer

Society Conference on Computer Vision and Pattern Recognition (CVPRW'05) - Workshops, 3:157, 2005.

[12] C. Chua, F. Han, Y. Ho. *3D Human Face Recognition using Point Signature.* Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 233-238, 2000.

[13] D. Colbry, G. Stockman, A. Jain. *Detection of Anchor Points for 3D Face Verification.* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005.

[14] D. Colbry. *Human Face Verification by Robust 3D Surface Alignment.* Ph.D. thesis, Michigan State University, 2006.

[15] A. Colombo, C. Cusano, R. Schettini. *3D Face Detection using Curvature Analysis.* Pattern Recognition, 39(3):444-455, 2006.

[16] V. Contreras. *Artnatomy. Anatomical Basis of Facial Expression Learning Tool.* Spain, 2005. Disponible en http://www.artnatomia.net

[17] N. D'Apuzzo. *Overview of 3D Surface Digitization Technologies in Europe.* Three-Dimensional Image Capture and Applications VI, Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 6056, San Jose (CA), USA, 2006.

[18] C. Dorai, A. Jain. *COSMOS - A Representation Scheme for 3D Free-Form Objects.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(10):1115-1130, 1997.

[19] P. Ekman, W. Friesen. *Manual for the Facial Action Coding System.* Consulting Psychologists Press, Palo Alto, 1977.

[20] S. Fleishman, I. Drori, D. Cohen-Or. *Bilateral Mesh Denoising.* International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2003, 22(3):950-953, 2003.

[21] B. Gökberk, A. Salah, L. Akarun. *Rank-Based Decision Fusion for 3D Shape-Based Face Recognition.* International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA'05), LNCS 3546:1019-1028, 2005.

[22] G. Gordon. *Face Recognition based on Depth and Curvature Features.* 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92), 108-110, 1992.

[23] F. Hendriks, D. Brokken, J. van Eemeren, C. Oomens, F. Baaijens, J. Horsten. *A Numerical-Experimental Method to Characterize the Non-Linear Mechanical Behaviour of Human Skin.* Skin Research and Technology 9(3):274-283, 2003.

[24] T. Heseltine, N. Pears, J. Austin. *Three-Dimensional Face Recognition: An Eigensurface Approach.* International Conference on Image Processing (ICIP'04), 2:1421-1424, 2004.

[25] T. Heseltine, N. Pears, J. Austin. *Three-Dimensional Face Recognition: A Fisher-surface Approach.* Proceedings of the International Conference on Image Analysis and Recognition, LNCS 3212:684-691, 2004.

[26] C. Hesher A. Srivastava, G. Erlebacher. *A Novel Technique for Face Recognition using Range Imaging.* Seventh International Symposium on Signal Processing and Its Applications, 2:201-204, 2003.

[27] K. Hormann, M. Tarini. *A Quadrilateral Rendering Primitive.* Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware, 7-14, 2004.

[28] B. Horn. *Closed-Form Solution of Absolute Orientation using Unit Quaternions.* Journal of the Optical Society of America, 4(4):629-642 1987.

[29] R. Hsu. *Face Detection and Modeling for Recognition.* Ph.D. thesis, Michigan State University, 2002.

[30] C. Huang. *A Solution for Computer Facial Animation.* Thesis, National Taiwan University, 2000.

[31] H. Huang, U. Ascher. *Fast Denoising of Surface Meshes with Intrinsic Texture.* Preprint, 2006.

[32] K. Kähler. *A Head Model with Anatomical Structure for Facial Modeling and Animation.* Ph.D. thesis, Universität dees Saarlandes, 2003.

[33] R. Koch, M. Gross, A. Bosshard. *Emotion Editing using Finite Elements.* Computer Graphics Forum, 17(3):295-302, 1998.

[34] J. Lee, E. Milios. *Matching Range Images of Human Faces.* Third International Conference on Computer Vision, 722-726, 1990.

[35] Y. Lee, D. Terzopoulos, K. Waters. *Constructing Physics-Based Facial Models of Individuals.* Proceedings of Graphics Interface '93, 1-8, 1993.

[36] Y. Lee, K. Park, J. Shim, T. Yi. *3D Face Recognition using Statistical Multiple Features for the Local Depth Information.* 16th International Conference on Vision Interface, 2003.

[37] Y. Lee, J. Shim. *Curvature-Based Human Face Recognition using Depth-Weighted Hausdorff Distance.* International Conference on Image Processing (ICIP), 3:1429-1432, 2004.

[38] Y. Lee, H. Song, U. Yang, H. Shin, K. Sohn. *Local Feature Based 3D Face Recognition.* International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA'05), LNCS 3546:909-918, 2005.

[39] S. Li, A. Jain (Editors). *Handbook of Face Recognition.* Springer, 2005.

[40] G. Liu, S. Quek. *The Finite Element Method: A Practical Course.* Butterworth-Heinemann, 2003.

[41] X. Lu, D. Colbry, A. Jain. *Matching 2.5D Scans for Face Recognition.* International Conference on Biometric Authentication (ICBA'04), LNCS 3072:30-36, 2004.

[42] X. Lu, R. Hsu, A. Jain, B. Kamgar-Parsi, B. Kamgar-Parsi. *Face Recognition with 3D Model-Based Synthesis.* International Conference on Biometric Authentication (ICBA'04), LNCS 3072:139-146, 2004.

[43] X. Lu, A. Jain. *Deformation Analysis for 3D Face Matching.* Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05), 1:99-104, 2005.

[44] X. Lu. *3D Face Recognition across Pose and Expression.* Ph.D. thesis, Michigan State University, 2006.

[45] G. Medioni, R. Waupotitsch. *Face Recognition and Modeling in 3D.* IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG'03), 232-233, 2003.

[46] A. Mian, M. Bennamoun, R. Owens. *Automatic 3D Face Detection, Normalization and Recognition.* Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), 2006.

[47] A. Moreno, A. Sánchez, J. Vélez, F. Díaz. *Face Recognition using 3D Surface-Extracted Descriptors.* Iris Machine Vision and Image Processing Conference (IMVIPC 2003), September 2003.

[48] A. Moreno, A. Sanchez. *GavabDB: A 3D Face Database.* Proceedings of the Second COST275 Workshop on Biometrics on the Internet, Vigo (Spain), 2004.

[49] T. Nagamine, T. Uemura, I. Masuda. *3D Facial Image Analysis for Human Identification.* International Conference on Pattern Recognition (ICPR'92), 324-327, 1992.

[50] J. Noh, U. Neumann. *A Survey of Facial Modeling and Animation Techniques.* Technical Report 99–705, University of Southern California, 1998.

[51] G. Pan, Z. Wu, Y. Pan. *Automatic 3D Face Verification from Range Data.* International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), 3:193-196, 2003.

[52] G. Pan, S. Han, Z. Wu, Y. Wang. *3D Face Recognition using Mapped Depth Images.* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPRW'05) - Workshops, 3:175, 2005.

[53] F. Parke. *Computer Generated Animation of Faces.* Proceedings of the ACM Annual Conference, 1:451-457, 1972.

[54] G. Passalis, I. Kakadiaris, T. Theoharis, G. Toderici, N. Murtuza. *Evaluation of 3D Face Recognition in the Presence of Facial Expressions: An Annotated Deformable Model Approach.* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPRW'05) - Workshops, 3:171, 2005.

[55] C. Pelachaud, S. Prevost. *Talking Heads: Physical, Linguistic, and Cognitive Issues in Facial Animation.* Course Notes, Computer Graphics International, 1995.

[56] P. Phillips, P. Flynn, T. Scrugss, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek. *Overview of the Face Recognition Grand Challenge.* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 1:947-954, 2005.

[57] R. Ribo, M. Riera, M. Esolano, J. Pérez, L. Font, E. Güeto. *GiD. The Personal Pre and Postprocessor.* CIMNE. Disponible en http://www.gidhome.com

[58] T. Russ, M. Koch, C. Little. *3D Facial Recognition: A Quantitative Analysis.* International Carnahan Conference on Security Technology, 338-344, 2004.

[59] T. Russ, M. Koch, C. Little. *A 2D Range Hausdorff Approach for 3D Face Recognition.* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPRW'05) - Workshops, 2005.

[60] *Singular Inversions Inc., FaceGen Modeller.* Disponible en http://www.facegen.com/downloads.htm

[61] J. Sobotta. *Atlas of Human Anatomy. Volume 1: Head, Neck, Upper Limb.* Lippincot Williams & Wilkins, 13th english/english edition, 2001.

[62] D. Swets, J. Weng. *Using Discriminant Eigenfeatures for Image Retrieval.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(8):831-836, 1996.

[63] H. Tanaka, M. Ikeda, H. Chiaki. *Curvature-Based Face Surface Recognition Using Spherical Correlation. Principal Directions for Curved Object Recognition.* Third International Conference on Automatic Face and Gesture Recognition, 372-377, 1998.

[64] D. Terzopoulos, K. Waters. *Physically-Based Facial Modeling, Analysis and Animation.* Journal of Visualization and Computer Animation, 1:73-80, 1990.

[65] *The Yale Face Database.* Disponible en http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[66] M. Turk, A. Pentland. *Eigenfaces for Recognition.* Journal of Cognitive Neuroscience, 3(1):71-86, 1991.

[67] M. Turk. *A Random Walk through Eigenspace.* IEICE Transactions on Information and Systems, E84-D(12):1586-1595, 2001.

[68] J. Vollmer, R. Mencl, H. Müller. *Improved Laplacian Smoothing of Noisy Surface Meshes.* Computer Graphics Forum, 18(3):131-138, 1999.

[69] K. Waters. *A Muscle Model for Animating Three-Dimensional Facial Expression.* 14th Annual Conference on Computer Graphics and Interactive Techniques, 17-24, 1987.

[70] C. Xu, Y. Wang, T. Tan, L. Quan. *Automatic 3D Face Recognition Combining Global Geometric Features with Local Shape Variation Information.* Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 308-313, 2004.

[71] C. Xu, Y. Wang, T. Tan, L, Quan. *A New Attempt to Face Recognition using 3D Eigenfaces.* 6th. Asian Conference on Computer Vision, 2:884-889, 2004.

[72] Y. Zhang, T. Sim, C. Tan. *Rapid Modeling of 3D Faces for Animation using an Efficient Adaptation Algorithm.* Proceedings of the 2nd. International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, 173-181, 2004.

[73] W. Zao, R. Chellappa, P. Phillips, A. Rosenfeld. *Face Recognition: A Literature Survey.* ACM Computing Surveys, 35(4):399-458, 2003.

[74] O. Zienkiewicz, R. Taylor. *The Finite Element Method. Volume 1: The Basis.* Butterworth-Heinemann, 5th. Edition, 2000.