

Proyecto Aguará

Reconocimiento de Caras

Documentación de Proyecto de Grado
Ingeniería Eléctrica

Cecilia Aguerrebere Otegui
Germán Capdehourat Longres
Mauricio Delbracio Bentancor
Matías Mateu Graside

Tutores: Alicia Fernández, Federico Lecumberry

Facultad de Ingeniería
Universidad de la República

Montevideo, Uruguay
Octubre 2006

Agradecimientos

Queremos agradecer ...

... profunda y eternamente a nuestros familiares y amigos, quienes a lo largo de estos años han sido muy generosos y pacientes en todo sentido.

... a nuestros tutores, Alicia Fernández y Federico Lecumberry, quienes depositaron desde el vamos una gran confianza en nuestro trabajo.

... a docentes y compañeros, quienes nos han acompañado y orientado a lo largo de nuestra carrera.

... a todas las personas que se prestaron para participar en este proyecto colaborando en la creación de una base de caras.

... a la gente del IIE y al GTI, en particular a Pablo Cancela, Juan Cardelino, Julia Demasi, Alvaro Gómez e Ignacio Ramírez cuyos aportes fueron de gran utilidad.

Tabla de contenidos

I	Acercamiento al tema	1
1.	Introducción	3
1.1.	Motivación	3
1.2.	Objetivos	5
1.3.	Resumen del Proyecto	5
1.3.1.	Acercamiento al Tópico	5
1.3.2.	Análisis y Diseño del Sistema	6
1.3.3.	Implementación y Evaluación	8
1.4.	Organización de la Documentación	10
2.	Sistemas Biométricos	11
2.1.	Introducción	11
2.2.	Breve Reseña Histórica	11
2.3.	Definición	12
2.4.	Modos de Operación	13
2.5.	Clasificación	14
2.6.	Características de Comparación	15
2.7.	Distintos Identificadores Biométricos	15
2.8.	Conceptos Básicos de Desempeño	18
2.9.	Evaluación	21
3.	Reconocimiento de Caras	23
3.1.	Introducción	23
3.2.	Antecedentes del Reconocimiento de Caras	25
3.3.	Detección de Caras	30
3.3.1.	Métodos Basados en Rasgos Faciales	32
3.3.2.	Métodos Basados en la Imagen	34
3.4.	Reconocimiento de Caras	36
3.4.1.	Métodos Holísticos	36
3.4.2.	Métodos Basados en Características Locales	43

II	Análisis y Diseño	51
4.	Detección de Caras y Normalización	53
4.1.	Introducción	53
4.2.	esCARAbajo: Una Primera Experiencia	53
4.2.1.	Segmentación utilizando información de color	53
4.2.2.	Validación de los candidatos a caras	55
4.2.3.	Evaluación del algoritmo	57
4.3.	Algoritmo de Detección Utilizado en el Sistema	58
4.3.1.	Aspectos Generales	58
4.3.2.	Descripción Funcional	58
4.3.3.	Evaluación	62
4.3.4.	Propuestas de mejora	65
4.4.	Normalización	66
5.	Descriptores de Caras	69
5.1.	Introducción	69
5.2.	Características Globales	70
5.2.1.	Principal Components Analysis: PCA	70
5.2.2.	Independent Components Analysis: ICA	71
5.2.3.	Linear Discriminant Analysis: LDA	73
5.2.4.	Ensayos y Resultados	75
5.2.5.	Conclusiones	79
5.3.	Características Locales	80
5.3.1.	Wavelets de Gabor	80
5.3.2.	Medidas de Similitud	87
5.3.3.	Ensayos y Resultados	89
5.3.4.	Conclusiones	91
5.4.	Elección de Descriptores para el Sistema	92
5.5.	Estudio del Banco de Filtros de Gabor	94
5.5.1.	Análisis de la Expresión <i>base</i>	94
5.5.2.	Estudio Estadístico del Espectro de Imágenes de Caras	95
6.	Localización de Puntos Principales	101
6.1.	Introducción	101
6.2.	Active Appearance Models	102
6.2.1.	Introducción	102
6.2.2.	Aplicación a Caras: Modelado de la Apariencia Facial	102
6.2.3.	Búsqueda del Active Appearance Model	105
6.3.	Elastic Bunch Graph Matching	106
6.3.1.	Aspectos fundamentales de la localización de puntos	106
6.3.2.	Elección del Algoritmo de Localización	109
6.4.	EBGM: Una Implementación	110
6.4.1.	Estimación Inicial de la Posición	110

6.4.2.	Refinamiento de la Posición: Estimación de Desplazamiento	112
6.5.	Alternativas en el Ajuste de Puntos de EBGM	117
6.5.1.	Estimación Inicial de las Coordenadas del Grafo	117
6.5.2.	Refinamiento del Ajuste de Grafo	121
6.6.	Generalización de la Localización de Puntos	126
6.6.1.	Detección de la Pose	126
6.6.2.	Localización de puntos considerando la pose	132
6.6.3.	Conclusiones	134
7.	Medidas de Similitud	137
7.1.	Introducción	137
7.2.	Medidas de Similitud	137
7.2.1.	Similitudes Promedio	138
7.2.2.	Similitudes Ponderadas	139
7.2.3.	Alternativa aplicando LDA	145
7.3.	Fusión de Referencias	149
III	Implementación y Evaluación	153
8.	Sistema	155
8.1.	Introducción	155
8.2.	Definición de Sistema	155
8.3.	Especificación Funcional	158
8.4.	Configuración del Sistema	158
9.	Evaluación	163
9.1.	Introducción	163
9.2.	Estimación de Intervalos de Confianza	164
9.2.1.	Paramétrica	164
9.2.2.	No Paramétrica	166
9.2.3.	Intervalos de Confianza en los Ensayos	166
9.3.	Ensayos y Resultados	167
9.3.1.	Evaluación Modular del Sistema	168
9.3.2.	Elección del Sistema Óptimo	179
9.3.3.	Comparación con otros algoritmos	181
9.3.4.	Evaluación del Sistema Automático	185
10.	Conclusiones y Trabajo Futuro	191
10.1.	Conclusiones	191
10.2.	Trabajo a Futuro	192

IV	Apéndices	201
A.	Biblioteca Desarrollada	203
A.1.	Detección: <code>libdetect</code>	204
A.2.	Preprocesado y Normalización: <code>libPreprocessNormalize</code>	205
A.3.	Localización y Extracción de Jets: <code>libcsuFaceGraph</code>	206
A.4.	Manejo de Identidades: <code>libidentidad</code>	208
A.5.	<code>csu2magick</code>	209
A.6.	<code>csuCommon</code>	209
B.	Interfaz de Usuario	211
B.1.	Versión de Línea de Comando	211
B.2.	Interfaz de Usuario: Manual Básico	212
B.2.1.	Elementos que Componen la Interfaz Gráfica	213
B.2.2.	Ejemplos de Uso de la Interfaz	215
B.2.3.	Requerimientos del Sistema	216
C.	Bases de Caras	217
C.1.	Base IIE	217
C.2.	Base FERET	221
C.3.	Bases Estándar	223
D.	Contenido del CD Adjunto	225
E.	Cálculos Estimación del Desplazamiento	227
F.	Dedicación	229
F.1.	Planificación	230
F.2.	Ejecución Real	231
F.3.	Evaluación de la Planificación y Conclusiones	232

Resumen

En este proyecto se presenta un sistema biométrico de reconocimiento, orientado a una aplicación de control de acceso, utilizando como característica biométrica una imagen digital del rostro de la persona. En la primera etapa se aborda el área de reconocimiento de caras y se estudia las técnicas existentes para resolverlo. En la segunda se analiza un sistema basado en características locales que utiliza filtros de Gabor como descriptores de la cara inspirado en el algoritmo EBGM. Se adquirió una base de caras en circunstancias similares a las de una aplicación de control de acceso. Se implementa una biblioteca que permite el desarrollo de aplicaciones. Con la misma se desarrolla un prototipo que integra desde la adquisición en tiempo real hasta la decisión de reconocimiento. Por último se evalúa el sistema y distintas alternativas estudiadas con la base estándar FERET. El desempeño obtenido es comparable con los resultados más significativos con esta base.

Palabras Claves: biometría, reconocimiento de caras, EBGM, wavelets de Gabor, FERET.

Lista de Personas

Figura	Persona
3.1	Valeria Meilán
3.2	Ian McKellen
3.3	Mauricio Delbracio
4.9	Maximiliano Magallanes
4.9	Pedro Casas
4.9	Laura Landin
4.9	Juan Piquinela
4.9	Martín Rocamora
4.9	Pablo Monzón
4.11	Ignacio Ramírez
4.12	Curly Howard
4.12	Larry Fine
4.12	Moe Howard
5.10	Cecilia Aguerrebere
6.1	Federico Lecumberry
6.2	Federico Lecumberry
6.3	Federico Lecumberry
6.5	Alicia Fernández
6.7	Rafael Potrie
6.11	Federico Lecumberry
6.12	Federico Lecumberry
6.13	1035 de la Base FERET
6.14	1035 de la Base FERET
6.15	1035 de la Base FERET
7.1	736 de la Base FERET
7.2	Germán Capdehourat
B.1	Matías Mateu
B.2	Matías Mateu
B.3	Matías Mateu
C.2	Cecilia Aguerrebere
C.3	0001, 0038, 0468, 1013 de la Base FERET
C.4	1133 de la Base FERET

Prefacio

”... una clase especial de cambio que abarca cierta índole de reconstrucción de los compromisos de cada grupo.”

Tomas S. Khun

Los autores de este proyecto entendemos necesario expresar ciertos preceptos filosóficos en los que se basó el grupo, para que el lector se haga eco de los mismos. Más allá de los objetivos académicos intrínsecos, cualquier proyecto de estas características, debe trascender de su lugar de origen y debe dejar en claro cuál es el alcance y cuáles son los medios para que el resto de los actores de la sociedad puedan hacer uso provechoso del mismo.

En ese sentido se debe mencionar el poder que implica el desarrollo y la utilización de sistemas para la identificación de personas mediante la imagen de la cara. Un sistema de estas características podría permitir el reconocimiento de un individuo sin su consentimiento, convirtiéndose así en una potencial arma. Hoy en día existen aplicaciones, fomentadas por determinados grupos de interés, que en pos de la seguridad pública violan la privacidad de las personas. En nuestra opinión, creemos que las aplicaciones del reconocimiento de caras deben de seguir una filosofía de respeto a la privacidad, es decir contar con la autorización de todos los individuos involucrados, en particular aquellos a ser identificados.

Además, se decidió desarrollar una aplicación basada en software libre con la convicción de que estas herramientas han de vehicular de manera transparente y justa el acceso a implementaciones de sistemas, al uso de técnicas o algoritmos, en este caso en el área del Reconocimiento de Caras. La utilización del software desarrollado, así como también la investigación realizada, están sujetas a la filosofía de uso mencionada anteriormente.

Esperemos la lectura sea de su agrado.

Parte I

Acercamiento al tema

Capítulo 1

Introducción

1.1. Motivación

Desde las últimas décadas cada vez más individuos, instituciones y empresas han incorporado a sus actividades y procesos rutinarios una estrecha interacción con sistemas informáticos y con equipamiento automatizado. Esta vinculación ha reemplazado la interacción humana por lo que denominamos interacción individuo-máquina. En este escenario se vuelven cada vez más necesarios los sistemas amigables para el usuario que aseguren su privacidad e integridad de su información, así como sistemas automáticos que optimicen el tiempo de las operaciones repetitivas. Las soluciones actuales implican que el usuario deba administrar una cantidad considerable de números y claves. Se utiliza un PIN¹ para extraer dinero, una contraseña para usar una computadora, otra serie de claves para acceder a los servicios que brinda Internet y así se podría seguir enumerando ejemplos de una extensa lista de opciones. Por otro lado, aplicaciones de interés nacional e internacional en el dominio público como son la identificación civil, el control de migraciones, la expedición de documentos como la libreta para conducir, son problemas que requieren del manejo de gran cantidad de información, por lo que sería muy útil automatizar los procesos involucrados para disminuir los tiempos que se emplean en la actualidad.

En los últimos años el desarrollo de una nueva tecnología denominada *Biometría Informática* hace pensar que muchos de estos problemas podrían ser resueltos en un futuro no muy lejano. De hecho, ya existe una oferta de productos importante (véase por ejemplo [1]), la cual proyecta un horizonte de mediano plazo en el que algunas tecnologías se generalicen y se vuelvan de uso cotidiano. La *Biometría Informática* [2] es la aplicación de técnicas matemáticas y estadísticas sobre los rasgos físicos o de conducta de un individuo para su clasificación. La principal línea de investigación y desarro-

¹Personal Identification Number.

llo de esta disciplina es el reconocimiento de individuos. El mismo consiste en la automatización de los métodos utilizados desde hace mucho tiempo para representar la identidad de las personas, aprovechando el gran avance en capacidad de procesamiento y baja de costos de la tecnología informática.

Dentro de las especializaciones de la *Biometría Informática* se encuentra el *Reconocimiento Automático de Caras*. Los desafíos que propone la implementación de un sistema artificial de reconocimiento de caras son recogidos por neurofisiólogos, psicólogos cognitivos, psicólogos sociales y científicos de la informática. Una de las metas más importantes que persiguen los investigadores es comprender el complejo sistema visual humano y cómo éste logra almacenar la información de los rostros humanos y utilizarla para representar y discriminar entre identidades con un alto grado de exactitud. Es natural aceptar que el sistema visual humano es capaz de reconocer individuos mediante la visualización de sus rostros incluso en condiciones muy variadas y hostiles como rotaciones de la pose o niveles no uniformes de iluminación. A lo largo de estas tres últimas décadas se han hecho esfuerzos para intentar imitar e interpretar lo creado por la naturaleza, con diversidad de resultados.

En el caso de una de las ciencias interesadas en la temática, la neurociencia, las consideraciones pueden ser del estilo ¿es el reconocimiento de caras un proceso con especificidades? [3]. Se ha encontrado que existen células especializadas en el sistema nervioso humano para el reconocimiento de caras. El área que estudia los procesos cognitivos del ser humano sugiere que existe una zona dedicada exclusivamente para esta tarea, aunque los procesos son similares a los que ocurren en el reconocimiento de objetos en general. Se ve cómo funciona entonces el aspecto interdisciplinar del problema. Un esquema posible es que la neurociencia modele un sistema de neuronas dedicadas al reconocimiento de individuos y luego los ingenieros y programadores simulen dicho sistema.

Es importante destacar el estado del trabajo en esta temática a nivel de la Facultad de Ingeniería de la Universidad de la República. El GTI² ha realizado investigaciones con huellas dactilares [4, 5]. No existen antecedentes utilizando rostros. Dado que es la primera experiencia con caras, el proyecto no solo involucra a los participantes directos, estudiantes y tutores, sino también al GTI en su conjunto, abriendo una nueva rama de investigación que puede ser continuada por futuros proyectos. Además el resultado de este trabajo servirá como insumo para un proyecto CSIC³ titulado Fusión Biométrica donde se busca estudiar las mejoras que se logran mediante la fusión de huellas con caras para realizar la identificación de individuos.

²Grupo de Tratamiento de Imágenes.

³Comisión Sectorial de Investigación Científica de la Universidad de la República.

En el presente trabajo se muestra el desarrollo de un sistema de Reconocimiento de Caras en el que se describe las ideas en las que se ha inspirado el mismo. Se pone en relieve algunas diferencias conceptuales y alternativas que existen a la hora de llevar a un lenguaje computacional una solución al problema, resuelto de manera tan brillante por la naturaleza en el caso de la raza humana.

1.2. Objetivos

El objetivo central de este proyecto es la implementación de un sistema biométrico de autenticación, orientado a una aplicación de control de acceso, utilizando como característica biométrica una imagen digital del rostro de la persona. La propuesta de investigación y desarrollo en este tema surge en el marco de nuestro proyecto de fin de carrera. Se convierte así en nuestro primer proyecto de ingeniería de magnitud considerable. Se debe diferenciar dos metas específicas a partir de la propuesta planteada.

Por un lado, uno de los objetivos de este proyecto es la acumulación de experiencia y de conocimiento en una área nueva para el GTI. Es por esto que nos enfocamos en realizar un trabajo riguroso y metodológico que pueda ser objeto de futuras extensiones. Por otro lado, se plantea un objetivo práctico: el desarrollo de una aplicación de autenticación de personas enfocada al control de acceso donde la autenticación se implementa mediante la extracción de características biométricas, específicamente los rasgos faciales.

1.3. Resumen del Proyecto

Se puede concebir el proyecto en tres grandes instancias. La primera, en la que se realiza el acercamiento al reconocimiento de caras y los problemas asociados. La segunda, en la que se analiza los bloques funcionales de un sistema de reconocimiento automático de caras y se estudian alternativas para cada uno de ellos. Por último se sintetiza la implementación del sistema buscando óptimos de funcionamiento en términos de performance y tiempos de ejecución.

1.3.1. Acercamiento al Tópico

Cuando se plantea un proyecto de estas características, en el que la información proviene de diversas fuentes y está dispersa en bibliotecas físicas y virtuales, el primer desafío es realizar una búsqueda que permita obtener una visión general, el estado del arte y las tendencias de investigación en el mundo científico. La búsqueda bibliográfica cumple al menos dos funciones. La primera función es la que, acompañada de la documentación, logra tener

un registro que abarque la diversidad de enfoques para resolver el problema. La segunda, que se desprende de la anterior, es la de generar una síntesis y modelado del problema que se está estudiando, de modo de conocer su lenguaje específico, sus códigos, nomenclaturas y tecnicismos.

La primera parte de la documentación recoge el estudio bibliográfico en tres áreas fundamentalmente; los sistemas biométricos, la detección de caras y el reconocimiento de caras. El primero de estos temas se desarrolla en el capítulo 2 y es el que da un marco teórico general, en el que se encuentran en particular los sistemas de reconocimiento de caras. Los dos últimos tratados en el capítulo 3, presentan las técnicas y trabajos más representativos de cada área a nuestro criterio.

Es importante resaltar la metodología utilizada para evaluar los algoritmos en condiciones objetivas a lo largo del presente trabajo. Un insumo fundamental para lo anterior es contar con al menos un banco de datos que permita procesar en cantidad y en calidad las imágenes para poder extraer conclusiones del desempeño de los algoritmos y de los sistemas de reconocimiento que hacen uso de tales algoritmos. Es así que se define como tareas prioritarias y se ejecutan en la primera etapa del proyecto la obtención de la base de caras estándar FERET⁴ [6] y la adquisición de una base de caras propia, la base IIE, que también fue diseñada con criterios específicos orientados a los requerimientos del presente trabajo. Ambas se detallan en el apéndice C.

1.3.2. Análisis y Diseño del Sistema

En esta etapa se define el problema a abordar y se analizan los módulos que componen el sistema desarrollado. El diagrama 1.1 explica a nivel de bloques las diferentes etapas estudiadas durante el proyecto. Básicamente existen dos grandes modalidades de uso del sistema. La primera es la modalidad de registrado, donde se ingresa al sistema la información de un nuevo usuario. Para esto se necesita contar con N imágenes del usuario a registrar. La segunda es la modalidad de confrontación, donde un usuario ya registrado se presenta al sistema (por ejemplo a través de un PIN), y en base a una única imagen del individuo, el sistema confronta contra la información de la base de datos y valida su identidad. Existe además de la autenticación recién descrita, la funcionalidad de identificación, donde el sistema decide solamente a partir de la imagen del individuo a qué usuario corresponde.

Como se observa en la figura 1.1, el primer bloque es el encargado de la

⁴Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office.

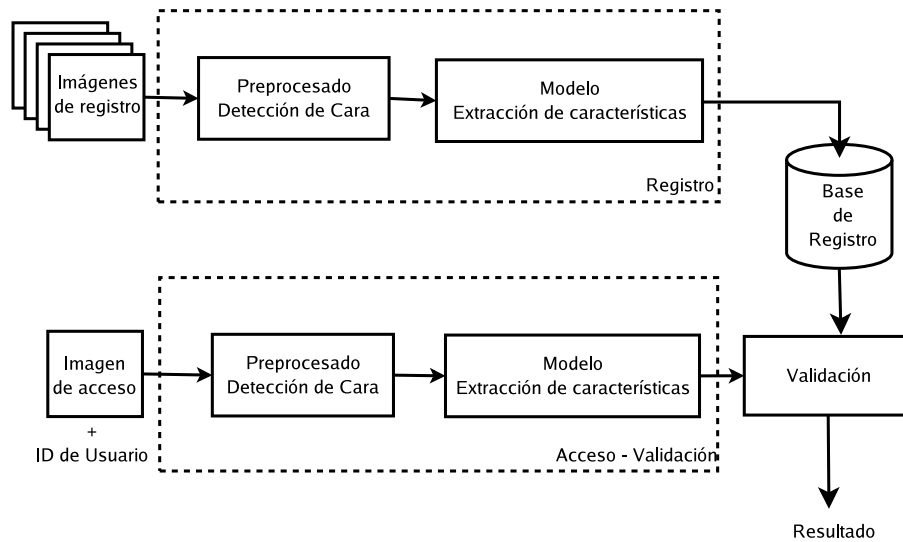


Figura 1.1: Diagrama de bloques de un sistema de autenticación de caras.

detección de la cara en la imagen y el preprocesado de la misma. En este proyecto se trabajó en primera instancia con un algoritmo de detección de caras basado en la información del color [7]. Si bien el uso de la información del color mostró propiedades interesantes, los resultados obtenidos no permitían utilizar la implementación desarrollada de este algoritmo en el sistema de reconocimiento de caras. Fue por esto que se decidió utilizar un detector de caras de mejor desempeño, disponible en la biblioteca de código libre MPT, *Machine Perception Toolbox* [8]. La misma fue desarrollada en el *Machine Perception Laboratory* de la Universidad de San Diego, California.

El módulo de detección, además de localizar la cara en la imagen, se encarga de encontrar las coordenadas de los ojos en la misma, funcionalidad que cumplían ambos algoritmos estudiados. Esta información se utiliza para la normalización geométrica de la imagen, de manera de contemplar traslaciones, rotaciones y cambios de escala. Además se estudiaron ajustes del nivel de gris de la imagen, así como máscaras que eliminan los efectos del fondo de la escena en la posterior confrontación. Estas dos primeras etapas del sistema se describen en el capítulo 4.

Una decisión crítica para lograr un buen desempeño del sistema fue la elección del descriptor de cara a utilizar. El mismo define qué características se consideran relevantes a la hora de la discriminación de los individuos. De la gran variedad de algoritmos encontrados en la búsqueda bibliográfica se opta por comparar dos posibilidades. En un caso se estudian descriptores globales que consideran la cara como un todo, sin tener en cuenta a priori

los rasgos presentes en la misma. Un ejemplo de éstos es la aplicación de análisis de componentes principales (PCA), como en el clásico trabajo de Turk y Pentland [9] de 1991. Por otro lado se estudian descriptores locales, en este caso la respuesta a un banco de filtros de Gabor en ciertos puntos principales definidos en la cara. Los mismos se utilizan en el algoritmo EBGGM introducido en 1997 por Wiskott *et al.* [10]. Finalmente, a partir de los estudios realizados se decidió trabajar con características locales puesto que se obtuvo mejor desempeño y mayor robustez con las mismas.

Luego de elegido el descriptor de cara a utilizar se profundizó en el estudio del mismo, en este caso los wavelets de Gabor. Se realizó un análisis de la densidad espectral de las imágenes de caras, el cual determinó la potencia que debían tener los núcleos de Gabor a utilizar. Este estudio fue decisivo para obtener un mejor desempeño en el sistema desarrollado. La elección del descriptor de cara y el estudio posterior a la misma se encuentra en el capítulo 5.

La elección de características locales para representar una cara agregó una necesidad extra en el bloque de extracción de características, la localización de los puntos principales en la cara. Aquí se trabajó a partir de una implementación del algoritmo EBGGM desarrollada en la Universidad Estatal de Colorado (CSU) por David Bolme en su tesis de maestría [11]. El código fuente de dicha implementación está disponible en [12]. Éste hecho permitió invertir más tiempo en el análisis y la búsqueda de mejoras en las distintas etapas del algoritmo, que en principio se iba a dedicar a la implementación del mismo. En particular se estudió la elección de la posición inicial de los puntos y diversas técnicas para el refinamiento de la posición de los mismos. Además se trabajó en la localización de puntos para el caso de poses con rotaciones. Este módulo se trata en el capítulo 6.

Por último se analizó las distintas opciones para comparar los vectores de características de dos caras, a los efectos de tener una medida de similitud para la etapa de confrontación. Se utilizaron algunas que surgen a partir del descriptor elegido para representar la cara. Por otro lado se trabajó con técnicas generales de reconocimiento de patrones aplicadas al caso particular de las caras con las características utilizadas en este proyecto. Este etapa se describe en el capítulo 7.

1.3.3. Implementación y Evaluación

En la etapa de implementación se integran los bloques funcionales para dar como resultado un sistema de reconocimiento automático de caras. A partir del modelo conceptual del sistema y del código disponible en los distintos bloques se diseñó una biblioteca detallada en el apéndice A. La misma

consta de los módulos que realizan cada una de las etapas del sistema como son: la detección automática de la cara, la normalización, la localización de puntos principales, la extracción de características, la confrontación y el manejo de la base de datos de los individuos registrados.

Con las alternativas estudiadas en cada módulo del sistema de reconocimiento de caras, se establecen distintas configuraciones del mismo, detalladas en el capítulo 8. Estas configuraciones son evaluadas en el capítulo 9 mediante un sistema de evaluación estandarizado que permite comparar la performance en autenticación e identificación respecto de otros sistemas actuales y preexistentes. Lo alentador de esta comparación son los resultados que ubican al sistema en una posición relativa favorable frente a otros algoritmos actuales. Finalmente se presentan conclusiones y posibles temas para futuras investigaciones en el capítulo 10.

También se debe indicar que el diseño del sistema se basa en una biblioteca que permitió el desarrollo de un prototipo con dos aplicaciones descritas en el apéndice B; una aplicación de línea de comando y una aplicación con interfaz gráfica. Las dos implementan las funcionalidades de un sistema de reconocimiento automático de caras. La aplicación de interfaz gráfica resuelve la adquisición de imágenes en tiempo real como etapa necesaria pensando en una aplicación de control de acceso. Por lo tanto a lo largo de los capítulos se observa que además de la búsqueda de la mejora de performance también se busca la disminución de tiempos debido a la importancia de este segundo parámetro en las aplicaciones en tiempo real. Este último elemento es en definitiva el mayor compromiso con el que se ven enfrentados los sistemas de reconocimiento de caras pensados para la vida real.

1.4. Organización de la Documentación

Los próximos dos capítulos corresponden al marco teórico del tema.

Capítulo 2 - Sistemas Biométricos Se introduce el mundo de los sistemas biométricos.

Capítulo 3 - Reconocimiento de Caras Se presenta el caso particular de los sistemas biométricos que utilizan la cara para representar una identidad.

Los cuatro capítulos siguientes se enfocan en los bloques del sistema y los descriptores de cara elegidos.

Capítulo 4 - Detección y Normalización Se describe el algoritmo utilizado para la detección y normalización en el sistema.

Capítulo 5 - Descriptores de caras Se presentan los descriptores utilizados para la representación de una cara.

Capítulo 6 - Localización de puntos principales Se detalla la localización de puntos principales utilizada en el sistema.

Capítulo 7 - Medidas de similitud Se define cómo confrontar dos caras y obtener una medida de similitud entre ellas.

En los últimos tres capítulos se detalla, evalúa y analiza el sistema resultante.

Capítulo 8 - Sistema Se explica de manera integral el sistema y sus variantes de configuración.

Capítulo 9 - Evaluación Se describe el método de evaluación utilizado y se muestran los resultados obtenidos.

Capítulo 10 - Conclusiones y trabajo futuro Se analiza el trabajo realizado y se sugieren posibles caminos de investigación futura.

Capítulo 2

Sistemas Biométricos

2.1. Introducción

Retomando los conceptos introducidos en el capítulo anterior, la *biometría*¹ es el estudio de métodos para la clasificación de seres humanos basados en una o más características físicas o de comportamiento intrínsecas. La principal aplicación de esta disciplina es el reconocimiento único de los individuos. Un sistema biométrico realiza el reconocimiento automático de personas basado en estas características. Se dice que el reconocimiento de un individuo es por lo que *es*, a diferencia de los métodos tradicionales que reconocen por algo que el individuo *posee* (tarjeta, llave) o *recuerda* (contraseña, PIN).

Las huellas dactilares, la retina, el iris, los patrones faciales, las venas de la mano o la geometría de la palma de la mano, representan ejemplos de características físicas, mientras que entre los ejemplos de características del comportamiento se incluye la firma, el paso y el tecleo. La voz se considera una mezcla de características físicas y del comportamiento. En la práctica, todos los rasgos biométricos comparten aspectos físicos y del comportamiento, puesto que para medirlos siempre se produce una interacción entre la persona y el sensor correspondiente.

En este capítulo se describe los aspectos fundamentales de los sistemas biométricos en general, basados en [2, 13, 14].

2.2. Breve Reseña Histórica

En las culturas occidentales la *biometría* no se puso en práctica hasta finales del siglo XIX. Sin embargo, la misma surge mucho antes de que se conociera en Occidente. Existen registros de que ya era utilizada en China

¹El término se deriva de las palabras griegas *bios* de vida y *metron* de medida.

desde el siglo XIV. El explorador y escritor portugués Joao de Barros escribió que los comerciantes chinos estampaban las huellas de la palma de las manos de los niños en papel con tinta como método para distinguirlos.

Hasta fines del siglo XIX, la identificación en Occidente confiaba simplemente en la *memoria fotográfica*. Fue recién en 1882 que el jefe del departamento fotográfico de la Policía de París, Alphonse Bertillon, desarrolló el sistema antropométrico conocido luego como *Bertillonage*. Éste fue el primer sistema preciso y de carácter científico, utilizado ampliamente para identificar criminales y que convirtió a la *biometría* en un campo de estudio. Este sistema funcionaba con medidas precisas de ciertas longitudes y anchos de la cabeza y del cuerpo, así como el registro de marcas individuales como tatuajes y cicatrices. El *Bertillonage* fue adoptado extensamente en Occidente hasta que se detectaron defectos en el mismo, principalmente problemas como cambios en los individuos que afectaban las medidas y distintos métodos para obtener las mismas. A partir de este hecho, las fuerzas policiales occidentales comenzaron a utilizar la huella dactilar, esencialmente el mismo sistema visto en China cientos de años antes.

Ivan Vučetić [15], nacido en 1858 en Lessina, Croacia, quien a los 24 años emigrara junto a su familia a la República Argentina, creó el primer sistema moderno de clasificación de huellas digitales. Ya era sabido que no existen dos individuos que tengan dibujos idénticos en las yemas de los dedos, pero hasta ese momento, no se había logrado implementar un sistema universalmente reconocido para la identificación de personas. Es así que en 1891 en Buenos Aires, en el ámbito de la investigación policial, se implementó el primer sistema de identificación a través de impresiones de huellas dactilares con el denominado *Método Vucetich*.

En el siglo XX la *biometría* creció desde la simple utilización de la huella dactilar, hasta el empleo de múltiples métodos teniendo en cuenta varias medidas físicas y de comportamiento. Con el avance de la informática comenzó el desarrollo de sistemas automáticos y un área nueva de investigación tecnológica. Esto último, junto con los problemas de seguridad de la actualidad, hace que las aplicaciones de la biometría también hayan aumentado. Hoy en día se utiliza en distintas actividades, desde la identificación civil hasta sistemas de control de acceso como puede ser el marcado de hora del personal de una empresa.

2.3. Definición

En términos formales, un sistema biométrico es un sistema de reconocimiento de patrones. Un sistema de reconocimiento de patrones consta básicamente de tres etapas. En la primera de ellas se realiza la adquisición de

datos sensoriales, obteniéndose una representación de un objeto del universo que se denomina patrón. En la segunda etapa se realiza la extracción de características, en la que se guarda una descripción concisa y representativa del universo observado, descartando información redundante. Por último se tiene la etapa de clasificación donde se da la toma de decisiones y se asigna al patrón una clase específica. En la figura 2.1 se resume las tres etapas anteriores.

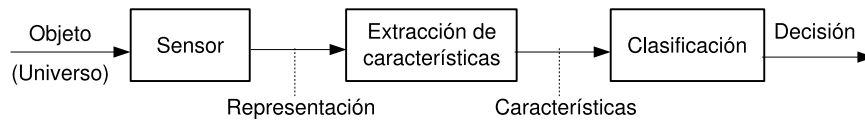


Figura 2.1: Sistema de Reconocimiento de Patrones.

En [16] se mencionan ejemplos muy variados, desde reconocer una cara visualmente hasta identificar las llaves del auto en el bolsillo por el tacto. Mediante procesos perceptuales lo que hace el ser humano es reconocer patrones.

Formalmente entonces se puede definir un sistema biométrico como:

Un sistema de reconocimiento de patrones que clasifica una persona a partir de la autenticidad de un rasgo físico específico o de una característica de comportamiento.

2.4. Modos de Operación

Existen distintos modos de operación claramente diferenciados según la forma de trabajo de los sistemas biométricos. A continuación se muestra la clasificación más importante según el modo en que opera el sistema:

Verificación El sistema valida la identidad de una persona comparando la característica sensada con el registro que se tiene del usuario en la base de datos. El individuo que desea ser reconocido indica su identidad a través de un número de PIN, nombre de usuario o una tarjeta inteligente y el sistema realiza una comparación entre la característica sensada y el registro en la base de datos para asignarle cierta validez a la identidad proclamada. En términos concretos, responde a la pregunta, *¿soy quien digo ser?*

Identificación Ingresa una característica sensada de un individuo al sistema y este reconoce la identidad haciendo la confrontación contra

toda la base de datos y decide si el usuario está en la base de datos y qué identidad posee. Responde a la pregunta, *¿quién soy?*

Otras variantes según el modo de operación del sistema son las siguientes:

- ● *On-line* : Se denomina sistema *on-line* cuando el mismo trabaja en tiempo real. Típicamente se trata de sistemas automáticos con exigentes requerimientos de tiempo. Ejemplo: control de acceso.
- *Off-line* : Un sistema *Off-line* o fuera de línea no trabaja en tiempo real. En general son sistemas semi-automáticos con menores requerimientos de tiempo. Ejemplo: aplicaciones forenses.
- ● *Reconocimiento positivo* : El reconocimiento por la positiva responde a la pregunta, *¿la persona es quien dice ser?*. Ejemplo: control de acceso.
- *Reconocimiento negativo* : El reconocimiento por la negativa responde a la pregunta, *¿la persona es quien niega ser?*. Ejemplo: infidelidad.

2.5. Clasificación

A continuación se muestra una clasificación según Wayman [14], basada en distintas características comunes a todos los sistemas biométricos.

- *Cooperativo vs. No Cooperativo* : Un impostor coopera o no con el sistema para romperlo. Está ligado con el concepto de reconocimiento positivo o negativo. En el primer caso el impostor coopera con el sistema para ser aceptado como un usuario válido. En el segundo caso el impostor no coopera con el sistema de manera de que el mismo no lo reconozca.
- *Explícito vs. Encubierto* : La persona sabe o no que está siendo reconocida por el sistema.
- *Habitual vs. No Habitual* : Según qué tan a menudo se utiliza el sistema. Ejemplos: renovación de la cédula o contraseña de red.
- *Supervisado vs. No Supervisado* : Existe o no la supervisión humana mientras se utiliza el sistema.
- *Ambiente controlado o variable* : Según qué condiciones de temperatura, presión, etc. se tengan. Depende por ejemplo de si es un lugar al aire libre o un lugar cerrado puesto que cambia mucho la iluminación.
- *Público vs. Privado* : Es utilizado por los clientes de la empresa o por los empleados de la empresa. Ejemplos: cliente de un banco o contraseña de red de un empleado.

- *Abierto vs. Cerrado* : Se trabaja con formatos de público conocimiento o con formatos propietarios.

Los sistemas más comúnmente utilizados son: cooperativo, explícito, habitual, de enrolamiento supervisado, de reconocimiento no supervisado, de ambiente controlado, privado y cerrado.

2.6. Características de Comparación

Un rasgo físico o característica de comportamiento puede ser usado como identificador biométrico para reconocer una persona si satisface los siguientes requerimientos:

Universalidad Todo ser humano debe tenerla.

Distinción Debe identificar a las personas unívocamente.

Permanencia Debe ser invariante con el tiempo y perdurar durante toda la vida de las personas.

Mensurabilidad Debe ser fácil de medir cuantitativamente, por lo que se necesitan sensores apropiados.

Además otros temas deben ser considerados en la práctica, incluyendo:

Performance Refiere a la precisión, velocidad y robustez que se puede alcanzar y los requerimientos necesarios para ello, así como otros factores operacionales o ambientales que afecten el funcionamiento del sistema.

Aceptación La gente debe acostumbrarse a utilizarlo en su vida cotidiana.

Seguridad Apunta a qué tan fácil es violar los sistemas por métodos fraudulentos.

2.7. Distintos Identificadores Biométricos

A continuación se listan distintos identificadores biométricos. No es una lista cerrada e incluye características que no han sido del todo exploradas por la comunidad científica, ni validadas para la implementación de sistemas de reconocimiento de personas hasta el momento.

ADN Es un código uni-dimensional que identifica a una persona, excepto por el hecho de que los gemelos tienen patrones de ADN idénticos. Sin embargo, hoy día es el más usado en el contexto de aplicaciones forenses.

Oreja Es sabido que la forma de la oreja y la estructura cartilaginosa de la pinna² son distintivas. De todas formas, las características de una oreja no se espera que sean únicas.

Cara Es uno de los más aceptados, puesto que es el que utilizamos los humanos en las interacciones visuales. Además, la adquisición de las imágenes del rostro es no intrusiva.

Termogramas (cara, mano, venas de la mano) El patrón de radiación de calor generado por el cuerpo humano es una característica de cada individuo y puede ser capturada por una cámara infrarroja.

Andar El andar es la forma peculiar de cómo uno camina y es un complejo identificador biométrico espacial-temporal. No se supone que sea muy distintivo, pero sí suficiente para permitir verificación en aplicaciones de bajos requerimientos de seguridad.

Mano Algunas características de la mano (ejemplo: largo de los dedos) son relativamente invariantes y peculiares (aunque no muy distintivas) para un individuo.

Iris La textura del iris está determinada por procesos morfogénéticos caóticos durante el desarrollo embrionario y es distinta para cada persona y cada ojo. Capturar la imagen del iris requiere cooperación por parte del usuario.

Tecleado Existe la hipótesis de que cada persona teclea de forma característica. Este comportamiento no se espera que sea único pero ofrece suficiente información discriminatoria para permitir verificación de la identidad.

Olor Es sabido que cada objeto exuda un olor que es característico de su composición química y puede ser usado para distinguir varios objetos. No está claro si la invariancia del olor del cuerpo humano puede ser detectada a pesar de los aromas artificiales y la variación en la composición química del ambiente circundante.

Retina La vasculatura retinal es rica en estructuras y se supone una característica de cada individuo y cada ojo. Se dice que es uno de los identificadores biométricos más seguros porque no es fácil de cambiar o replicar.

Huellas Dactilares Se producen tantas variaciones durante la formación de las huellas dactilares que es virtualmente imposible que dos huellas sean idénticas. Se trata de patrones característicos de crestas y surcos únicos para cada individuo. Además existen configuraciones con

²Parte visible de la oreja que reside fuera de la cabeza.

Identificador biométrico	Univ.	Dist.	Perm.	Mens.	Perf.	Acep.
ADN	A	A	A	B	A	B
Oreja	M	M	A	M	M	A
Rostro	A	B	M	A	B	A
Termograma facial	A	A	B	A	M	A
Huella dactilar	M	A	A	M	A	M
Andar	M	B	B	A	B	A
Geometría de la mano	M	M	M	A	M	M
Venas de la mano	M	M	M	M	M	M
Iris	A	A	A	M	A	B
Tecleo	B	B	B	M	B	M
Olor	A	A	A	B	B	M
Retina	A	A	M	B	A	B

Tabla 2.1: Distintos identificadores biométricos y sus características.

variación limitada lo que permite la clasificación. Es una de las tecnologías biométricas más maduras y es aplicable en un gran número de aplicaciones de reconocimiento.

Firma La manera en la que una persona firma su nombre es una característica individual. Hoy en día es utilizada en transacciones gubernamentales, comerciales y legales como método de verificación. Tiene la contra de que varía con el paso del tiempo y los expertos saben reproducirla con facilidad.

Voz Es una técnica aceptada en casi todas las sociedades. La voz puede ser el único identificador biométrico usado en aplicaciones por vía telefónica. No se espera que sea suficientemente distintivo para permitir identificación en grandes bases de datos. La señal vocal usada para reconocimiento se degrada por el micrófono, el canal de comunicación y la digitalización y puede variar según la salud, el estrés y las emociones de la persona. Además, existen personas expertas en imitar voces de otras personas.

Se puede apreciar en el cuadro 2.1 distintos identificadores biométricos y sus características según la siguiente convención: A significa Alto, B significa bajo, M significa medio para la Universalidad, Distinción, Permanencia, Mensurabilidad, Performance y Aceptación.

2.8. Conceptos Básicos de Desempeño

En los sistemas de reconocimiento de patrones se presentan dos tipos de errores, independientemente de la aplicación que se trate:

Falsos Positivos - FMR (False Match Rate)

Corresponde a *sí* clasificar un patrón en una clase que *no* le corresponde.

Falsos Negativos - FNMR (False Non-Match Rate)

Corresponde a *no* clasificar un patrón en una clase que *sí* le corresponde.

En general en sistemas biométricos se utilizan términos específicos de la aplicación como son:

Falsas Aceptaciones - FAR (False Acceptance Rate)

Corresponde a *sí* dar por válida una identidad que *no* es la correcta.

Falsos Rechazos - FRR (False Rejection Rate)

Corresponde a *no* dar por válida una identidad que *sí* es la correcta.

Los sistemas biométricos en general obtienen un *score* de similitud en base al cual se establece si la persona es o no quien dice ser. Esto hace que los errores antes mencionados dependan de un umbral de decisión. Al definir el umbral de decisión quedan definidos los dos tipos de errores que se cometen: FAR y FRR. A continuación se describen los errores basados en [17] pensando en un sistema de control de acceso, pero sin pérdida de generalidad. Si variamos el umbral de decisión desde el valor mínimo que toma el *score* al máximo posible, se tendrá en el primer extremo que todos los impostores ingresarán al sistema y que en el otro extremo ninguno de ellos lo logrará. La fracción definida por la cantidad de impostores aceptados dividido la cantidad de impostores define el FAR que significa tasa de falsa aceptación. Si el $FAR = 1$ entonces todos los impostores logran violar el sistema, en cambio si el $FAR = 0$ ningún impostor logra violarlo. En la figura 2.2 se ilustra este concepto.

Análogamente, trabajando con las personas que son clientes o usuarios genuinos del sistema, se sufrirán cada vez más los falsos rechazos a medida que aumento el umbral de decisión. Esto se observa en la figura 2.3.

Por último, se observa que si las distribuciones del *score* de las dos clases de personas sobre un mismo eje se superponen, también lo hacen las curvas de FAR y de FRR asociadas. El punto definido por $FAR = FRR$ es el que se denomina EER (Equal Error Rate), es decir tasa igualitaria de error. Lo

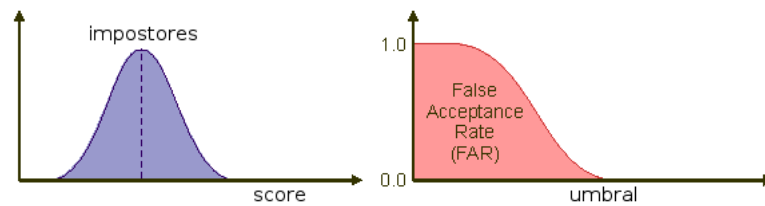


Figura 2.2: Distribución del *score* de impostores del sistema a la izquierda y FAR en función del umbral de decisión a la derecha.

anterior se observa en la figura 2.4.

Si bien el EER es un buen indicador para comparar dos sistemas y presenta la ventaja de ser un valor escalar, no se debe perder de vista que se trata de un único punto de funcionamiento del sistema. Según la aplicación puede ser más importante el FAR o el FRR, por lo que no se trabajará en el punto que ambos coinciden. Por ejemplo, en el caso de la identificación civil, es más importante tener un bajo FAR, puesto que es muy grave expedir un documento para una persona equivocada. En el otro extremo, las aplicaciones forenses trabajan con un bajo FRR, porque aunque se tengan más falsos positivos lo importante es no descartar al individuo buscado.

Existe un estricto compromiso entre un error y el otro, por lo que el mejor punto de trabajo depende de la aplicación específica. Por esta razón, para caracterizar un sistema biométrico se utiliza la curva ROC (Receiver Operating Characteristic), donde se grafican los distintos puntos de funcionamiento del sistema. Esta curva grafica de manera paramétrica como función del umbral de decisión, el porcentaje de falsos positivos (en la figura 2.5 en el eje de las ordenadas), versus el correspondiente porcentaje de falsos rechazos (en la figura 2.5 en el eje de las abscisas). Pueden haber variantes tales como graficar el porcentaje de aceptaciones correctas en vez

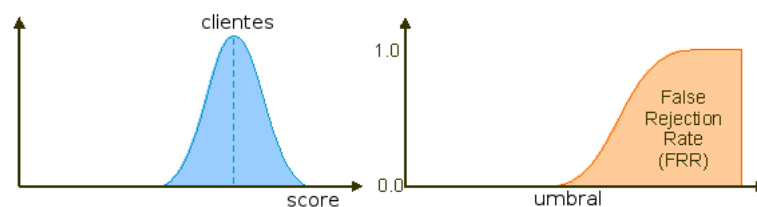


Figura 2.3: Distribución del *score* de los clientes del sistema a la izquierda y FRR en función del umbral de decisión a la derecha.

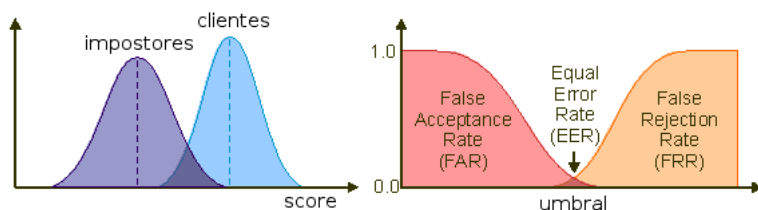


Figura 2.4: Se observan las dos distribuciones superpuestas y la interpretación gráfica del EER.

de los falsos rechazos o el uso intercambiado de los ejes coordenados, pero en todos los casos lo que se observa es el compromiso entre ambas tasas de error. Lo interesante de la ROC es que es independiente de los valores que toma el umbral, lo que permite la comparación de performance de diferentes sistemas bajo las mismas condiciones.

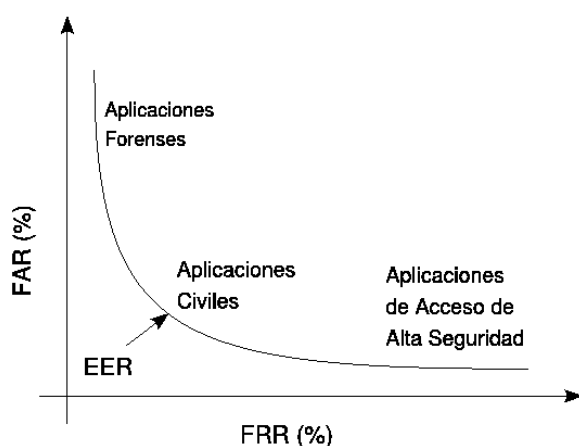


Figura 2.5: ROC - Receiver Operating Characteristic.

Cabe aclarar que la ROC sirve para caracterizar un sistema trabajando en modo verificación. Para ver el desempeño del sistema en identificación se utiliza la denominada curva de rango o CMC (Cumulative Match Characteristic). Se define una identificación correcta de rango N cuando el usuario figura entre las N identidades más cercanas de la base. Esta curva muestra el porcentaje de identificación correcta de rango N en función del rango, como se observa en la figura 2.6. En el caso $N = 1$ se obtiene la tasa de reconocimiento correcto RR (Recognition Rate).

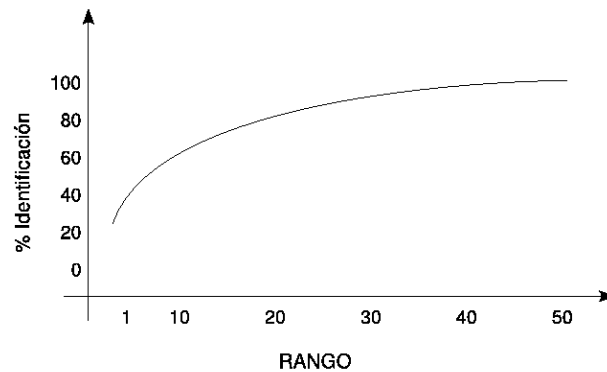


Figura 2.6: Curva de Rango o CMC - Cumulative Match Characteristic.

2.9. Evaluación

Se definen tres tipos de evaluación de sistemas biométricos:

Tecnológica - Algoritmos

El objetivo de la evaluación tecnológica es comparar distintos algoritmos de una misma tecnología. Algoritmos que cumplen con un protocolo de entrada/salida son testeados (sensores y otros aspectos se dejan de lado). El test de todos los algoritmos es realizado con una o más bases de datos. Como ejemplo se tiene el FVC³ [18] que se hace cada dos años con huellas dactilares y su símil para caras, el FRVT⁴ [19].

Operacional - Prototipo

El objetivo de la evaluación operacional es determinar la performance del sistema punta a punta con un prototipo o una aplicación simulada. El test se realiza en un ambiente que modela la aplicación del mundo real.

De campo - Prueba real

El objetivo de la evaluación de campo es determinar la performance de un sistema biométrico completo en el ambiente de una aplicación específica con una población objetivo específica. En general, los resultados no son repetibles debido a diferencias desconocidas o no documentadas entre ambientes operacionales.

³Fingerprint Verification Competition.

⁴Face Recognition Vendor Test.

Capítulo 3

Reconocimiento de Caras

3.1. Introducción

Dentro del área del Tratamiento de Imágenes por Computadora, el Reconocimiento de Caras ha recibido especial atención principalmente desde fines de la década del ochenta hasta la actualidad. Al menos dos razones justifican este hecho; la primera es el amplio rango de aplicaciones, tanto comerciales como legales, que crecen y se refuerzan día a día; la segunda es la disposición de la tecnología necesaria para alcanzar rendimientos antes insospechados.

Si bien es cierto que los sistemas actuales de reconocimiento han alcanzado un nivel de madurez aceptable en varias de sus aplicaciones, su éxito se ve limitado en situaciones donde se requiere abarcar la variabilidad de condiciones que impone el mundo real. Por ejemplo, para el reconocimiento utilizando imágenes de exterior con cambios en la iluminación y/o la pose, la performance es aún mala. En otras palabras, los sistemas de reconocimiento automático actuales siguen siendo poco robustos y están muy lejos de las capacidades del sistema perceptual humano.

Una de las principales razones para que el reconocimiento de caras sea uno de los sistemas biométricos más estudiados, es que los seres humanos nos reconocemos de esta manera. El reconocimiento de caras se puede ver como la conjunción de varias disciplinas, entre ellas, la psicología, la neurociencia y la visión por computador. En los últimos treinta años han existido importantes esfuerzos en comprender el funcionamiento del Sistema Visual Humano (SVH), que junto con el sistema nervioso, forman el dispositivo reconocedor por naturaleza. Algunos trabajos [20] han impulsado la tesis de que los seres humanos contamos con un proceso exclusivo para la identificación de caras, mientras otros se aferran a la idea de que es un comportamiento general de un proceso mayor de reconocimiento de todo tipo de objetos.



Figura 3.1: Ejemplos de algunas dificultades presentes a la hora de realizar un sistema de reconocimiento de caras.



Figura 3.2: Pequeños cambios hacen que el sistema visual se vea confundido.

Si bien los seres humanos podemos reconocer caras en distintas poses, expresiones, tamaños, condiciones de iluminación como en la figura 3.1, es muy fácil engañar al sistema visual humano. En la figura 3.2 se muestra un ejemplo donde manteniendo determinadas características, ojos, nariz, cejas, y cambiando otras como el pelo, barba, bigote, se puede confundir al observador. En las últimas décadas diferentes estudios muestran que el reconocimiento de caras es resultado de factores holísticos¹ y de configuración, pero también de análisis de características locales. Un ejemplo que muestra que el reconocimiento de caras se puede deber más a factores de configuración que a locales es *The Thatcher Illusion* [21]. En la figura 3.3 se muestra un ejemplo de una cara donde sus ojos y boca fueron invertidos. La inversión hace que la imagen resulte grotesca en su vista normal. Sin embargo si la imagen es rotada 180 grados la cara parece igual. En las siguientes secciones se introduce una clasificación de los métodos computacionales existentes para el reconocimiento de caras, clasificándolos en holísticos o basados en características locales.

Un planteo general del problema del Reconocimiento de Caras a través de imágenes se puede formular de la siguiente manera: dadas imágenes fijas o de video de una escena, el objetivo del Reconocimiento de Caras es identificar o verificar, una o más personas en la escena, utilizando una base de datos de caras, es decir un registro de individuos. Aplicando lo visto en el Capítulo 2 al Reconocimiento de Caras, en problemas de identificación, la entrada al sistema es una imagen de un rostro desconocido y el sistema reporta la identidad a partir de una base de datos de caras conocidas. Por otro lado, en problemas de verificación, la entrada al sistema es una imagen y una identidad proclamada (por medio de un PIN por ejemplo) y el sistema debe confirmar o rechazar la identidad propuesta para la imagen de rostro de entrada.

Un sistema de reconocimiento de caras se puede dividir conceptual y funcionalmente en tres bloques. En la primera etapa se realiza la detección del rostro en la imagen, en la segunda se realiza la extracción de características y en la tercera se aplica algún algoritmo de reconocimiento. En la figura 3.4 se esquematiza esta idea.

3.2. Antecedentes del Reconocimiento de Caras

Se puede situar al reconocimiento de caras en la misma escena de tiempo que el área de investigación denominada Visión por Computador, la cual

¹De la Real Academia, *holístico* 1. adj. Fil. Perteneciente o relativo al holismo. *Holismo* 1. m. Fil. Doctrina que propugna la concepción de cada realidad como un todo distinto de la suma de las partes que lo componen.



(a) Imágenes rotadas 180 grados



(b) Imágenes vista normal

Figura 3.3: *The Thatcher Illusion*. En la imagen superior, no se logra visualizar lo grotesco del cambio producido al invertir los ojos y la boca. En la imagen inferior, en la vista normal se ve el cambio efectuado.

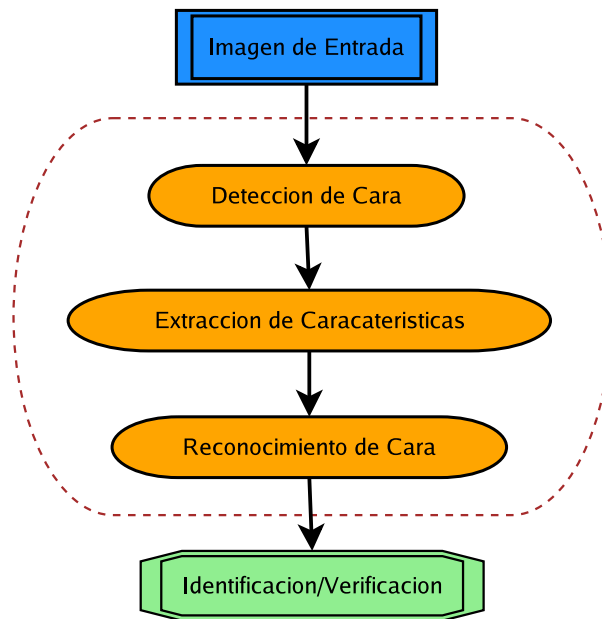


Figura 3.4: Diagrama de bloques representativo del problema de reconocimiento automático de caras.

comienza a desarrollarse fuertemente a partir de 1960. El reconocimiento automático de caras se ha preservado a lo largo del tiempo y tiene hoy en día total vigencia si bien no logra la confiabilidad y seguridad de sistemas basados en otras características biométricas como por ejemplo los de huellas digitales. Es probable que ayuden factores tales como su naturaleza no invasiva o que es un área interdisciplinaria y que interesa *per se* al ser este un método primario de identificación con que cuentan los seres humanos.

Los primeros intentos de implementación de sistemas de reconocimiento de caras datan de la década de 1960. Estos sistemas eran semiautomáticos. Sus algoritmos se basaban en ideas intuitivas de realizar el reconocimiento de caras, observando las características más sobresalientes de las caras y comparándolas entre diversos patrones. En general estos métodos tomaban fotografías a las cuales le realizaban marcas (por ejemplo en los ojos, las orejas, la nariz, la boca) y luego computaban las distancias relativas entre pares de puntos y comparaban contra una referencia. En la década de los setenta Goldstein, Harmon y Lesk [22] sistematizan el marcado de 21 medidas subjetivas, entre ellas el color del pelo y el ancho de los labios. Sin duda que por esta época la mayor limitante de los sistemas se debía a la dificultad extrema en automatizar el proceso de extracción y confrontación de características.

A fines de los años setenta Fisher y Elschlagerb introducen un sistema de reconocimiento con un grado de automatización superior al de sus antecesores. Se utilizaban características modelo (*templates*) de diferentes partes del rostro humano y luego se mapeaban sobre un template de la cara. De todas maneras el enfoque adoptado no tuvo mayor éxito principalmente debido al avance tecnológico de la época.

Kohonen [23] es recordado por su sistema de reconocimiento de caras que demuestra el uso de redes neuronales para reconocimiento en base a imágenes de caras normalizadas. Incluyó el análisis de componentes principales (PCA) mediante el uso de los vectores propios de la matriz de autocorrelación de la imagen de cara normalizada. Las complicaciones de alineación y normalización de la cara en la imagen impidieron que el sistema de Kohonen fuera un éxito práctico.

En años venideros, los investigadores en el área comenzaron a estudiar esquemas de reconocimiento de caras basados en bordes, distancias entre características, regiones y nuevos enfoques de redes neuronales. Todos los sistemas de esa época sufrieron el problema de no ser escalables a tamaños de bases de datos grandes. La localización y normalización también representaron problemas de gran dificultad en esta etapa de evolución de las técnicas. El primer sistema totalmente automático introdujo un enfoque estadístico que se apoyaba en histogramas y niveles de gris. Dado un modelo genérico de características estadísticas se creaba un conjunto de patrones para una imagen y se lo comparaba con el modelo genérico.

En 1989 Kirby y Sirovich [24] introducen un método matemático que simplifica el cálculo de las *eigenfaces*². Muestran la eficiencia de la representación de las imágenes de caras alineadas y normalizadas mediante bases de dimensión menor a 100. Ya en 1991 Turk y Pentland [9] utilizan el hecho de que la representación mediante *eigenfaces* minimiza el error en media cuadrática para detectar y localizar caras en imágenes naturales y a distintas escalas. Con estas ideas logran una aplicación confiable en tiempo real. La simpleza del sistema y su desempeño dan un espaldarazo definitivo al área y consolidan la temática que al día de hoy sigue proponiendo nuevos desafíos.

A mediados de la década del noventa, aparecen una gran cantidad de métodos que intentan relajar las hipótesis de funcionamiento, condiciones de iluminación, pose, expresión, etc. Principalmente fue impulsado por un fuerte crecimiento del área de reconocimiento de patrones y un marcado avance tecnológico que permitió la implementación de técnicas en base a ideas teóricas ya existentes. Entre los métodos más importantes se encuen-

²Vectores propios de la matriz de covarianza de las imágenes de caras

tran generalizaciones al método planteado por Turk y Pentland como LDA, ICA y KPCA. Además surgieron métodos del tipo de ataque local, basados en texturas como EBGm o AAM e incluso métodos basados en modelos 3D. En la sección 3.4 se describe algunas de estas técnicas.

Con el nuevo siglo surgieron aplicaciones comerciales que tuvieron y siguen teniendo gran éxito en el mercado. Junto con el avance de los productos comerciales, surge un debate entre la seguridad pública y la privacidad individual y social. Algunos gobiernos han impulsado el desarrollo de aplicaciones de reconocimiento de caras en aeropuertos y fronteras, como cámaras de vigilancia o pasaportes electrónicos. Existen opiniones encontradas sobre si el grado de madurez de la tecnología permitirá el correcto funcionamiento en este tipo de aplicaciones. Por otro lado, se discute la legitimidad de la utilización en casos donde no existe consentimiento por parte del individuo a ser identificado.

3.3. Detección de Caras

Uno de los principales problemas en el área del Reconocimiento de Caras es la propia detección de las mismas dentro de la imagen. Es imprescindible para el correcto funcionamiento de un algoritmo de reconocimiento, que la cara sea localizada de manera precisa.

Si bien la principal aplicación de la detección de caras es una de las etapas de un algoritmo de identificación y/o verificación de personas, también se aplica en otras como: identificación de estados de ánimo, compresión en tiempo real (videoconferencia), seguimiento, sistemas de vigilancia, interfaz computadora-hombre, etc.

Al igual que en otros problemas de procesamiento de imágenes, muchos investigadores concentran sus esfuerzos en entender el funcionamiento del sistema visual humano, para así poder aplicarlo a este problema en particular. Cabe destacar que si bien se ha avanzado mucho en el área todavía se está muy lejos de resolver el problema de manera general. Los seres humanos tenemos una increíble capacidad para detectar caras, incluso hasta en lugares donde no existen. Como ejemplo se plantea la figura 3.5.

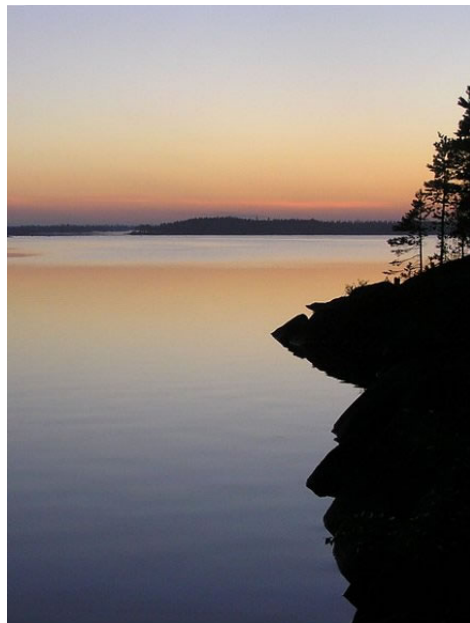


Figura 3.5: Ejemplo de una “cara” que no es cara.

Podríamos formular el problema de la detección de caras de la siguiente

manera: dada una imagen, el objetivo es detectar la presencia de caras y en caso afirmativo dar las coordenadas de las caras presentes. Claro está que el problema planteado en dichos términos es muy complejo. A continuación se listan algunas de las condiciones y dificultades a considerar:

- Pose y orientación de la cara.
- Tamaño de la cara.
- Presencia de lentes, barba, gorro, etc.
- Expresión de la cara.
- Problemas de iluminación (por ejemplo: no uniformidad de la misma).
- Condiciones generales de la imagen (por ejemplo: ruido, fondo complejo).
- Cantidad desconocida de caras en la imagen.

En la figura 3.3 se muestra un ejemplo de lo complicado que puede ser encontrar caras en un escenario real cualquiera.



Figura 3.6: Ejemplo de las complicaciones a tener en cuenta al localizar caras en un escenario real.

Además de la localización de caras en imágenes existen otros problemas relacionados con este:

- Localización de características relevantes: ojos, boca, cejas, mentón, orejas, etc.
- Reconocimiento de expresiones: triste, alegre, enojado, etc.
- Estimación de la pose y seguimiento.

En la actualidad existe una gran cantidad de algoritmos de detección de caras con desempeño variado y dependiente de los escenarios a considerar. Para una descripción de alto nivel de los mismos se pueden consultar los *surveys* de Yang et al. [25] y de Hjelmsås y Low [26]. Una posible clasificación de los algoritmos de detección de rostros es:

Métodos Basados en Rasgos Faciales: buscan encontrar aquellas características presentes en cualquier rostro: ojos, cejas, labios, boca, mentón, líneas de contorno, etc.

Métodos Basados en la Imagen: aplican herramientas generales de reconocimiento de patrones para sintetizar un modelo a partir de un conjunto de imágenes de entrenamiento. Trabajan con la imagen completa o una región de ésta sin buscar rasgos faciales de forma localizada.

La anterior no es la única clasificación posible. Otras clasificaciones posibles son:

- Métodos basados en color vs. Métodos basados en tonos de gris.
- Métodos holísticos vs. Métodos locales.
- Métodos de localización en tiempo real vs. Métodos offline.

3.3.1. Métodos Basados en Rasgos Faciales

Sobre el comienzo de los años setenta comenzó el desarrollo de esta área, motivada por ideas intuitivas de la representación de un rostro. Los primeros métodos buscaban resolver el problema de encontrar los rasgos de una cara en una situación totalmente controlada (fondo blanco, cara de frente, expresión neutra). La mayoría de ellos se basaban en buscar relaciones geométricas entre puntos característicos hallados mediante heurística³. A lo largo de los años, y en particular en la última década creció de manera sustancial el interés por resolver este problema, logrando así un desarrollo muy importante del área. Se pueden definir tres ramas dentro del conjunto de métodos basados en rasgos faciales:

³De la Real Academia, *heurística*: 4. f. En algunas ciencias, manera de buscar la solución de un problema mediante métodos no rigurosos, como por tanteo, reglas empíricas, etc.

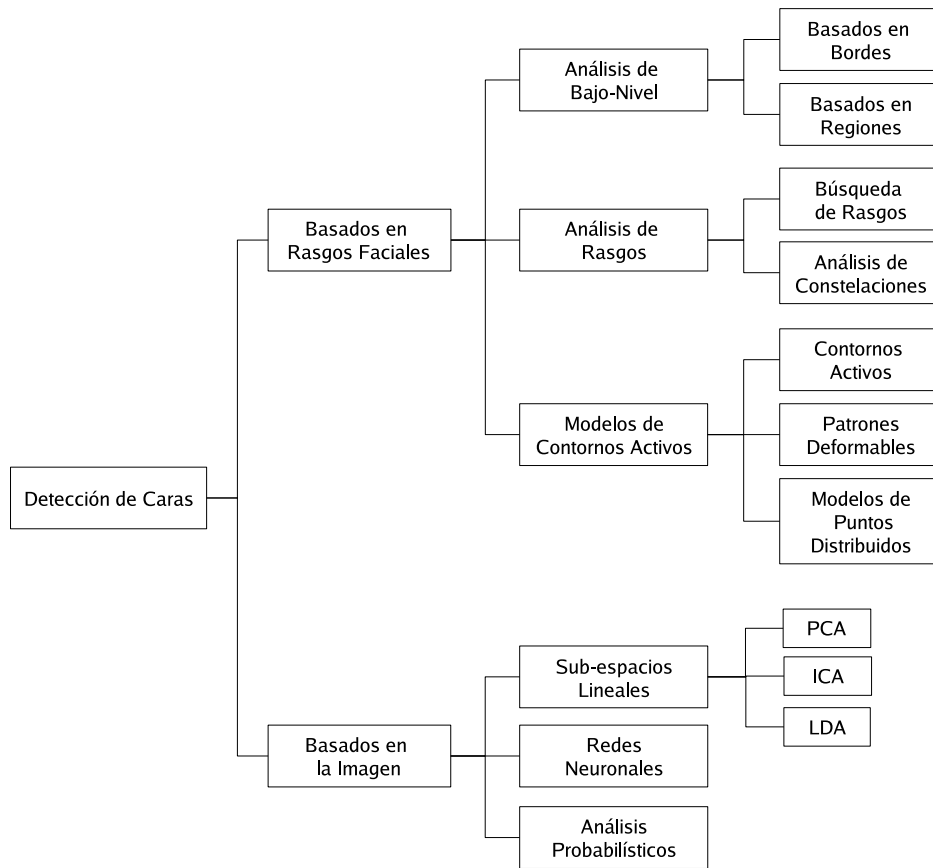


Figura 3.7: Una posible clasificación de los métodos de Detección de Caras.

Análisis de bajo nivel

Son técnicas que trabajan directamente con los píxeles, principalmente hay dos:

- *Basados en bordes*: buscan bordes, los afinan, etiquetan y finalmente buscan estructuras similares a las de una cara.
- *Basados en regiones*: aprovechan el hecho de que hay zonas más oscuras que el resto de la cara (cejas, pupilas, etc). Umbralizan y separan la imagen en regiones. Localizan la cara comparando la distribución de las regiones presentes con la distribución de regiones *tipo* de una cara.

Análisis de rasgos faciales

Dado que el análisis a bajo nivel puede brindar información parcialmente correcta o incompleta, esta familia de métodos busca encontrar implícita-

mente los rasgos faciales. Se basan fuertemente en las relaciones geométricas que cumplen los diferentes rasgos presentes en una cara. Existen dos grandes aproximaciones al respecto:

- *Búsqueda de rasgos*: intentan realizar una búsqueda ordenada de los rasgos característicos de una cara. Por ejemplo, primero buscan la frente, luego los ojos, continúan con la nariz, etc. Se basan en hipótesis sobre la pose y orientación de la cara y utilizan heurística.
- *Análisis de Constelaciones*: buscan levantar algunas de las hipótesis de los métodos anteriores sobre la pose y orientación de la cara. Se basan en un análisis probabilístico de la ubicación de ciertos puntos característicos (constelación) de una cara.

Análisis mediante modelos de contornos activos

Los métodos basados en modelos de contornos activos buscan adaptar un modelo genérico de un rasgo (ojo, boca, contorno de la cara) a la imagen o porción de imagen en cuestión. Para esto, buscan iterar deformando el modelo hasta adaptarlo al rasgo buscado. Se basan fuertemente en la información local de la imagen (bordes, nivel de gris). Existen tres grandes técnicas:

- *Snakes o contornos activos*: son comúnmente utilizados para encontrar el contorno de la cara. Se basan en la minimización de una función de energía para adaptar el modelo.
- *Patrones deformables*: buscan adaptar modelos paramétricos de cada rasgo. Al igual que los Snakes se basan en minimizar una función de energía para la adaptación.
- *Modelos de puntos distribuidos (PDM)*: son una manera compacta de representar de manera paramétrica las formas buscadas. El ajuste de estos modelos se basa en discretizar el contorno PDM en diferentes puntos etiquetados. Las variaciones posibles de estos puntos se guardan en un modelo estadístico realizado cuidadosamente a partir de un conjunto de entrenamiento.

3.3.2. Métodos Basados en la Imagen

La familia de métodos presentada anteriormente se caracteriza por ser fuertemente sensible a variaciones impredecibles como pueden ser variaciones en la aparición de las caras (poses, orientaciones, oclusiones) así como también variaciones del entorno (fondo complejo, iluminación no uniforme). Por esta razón se vio la necesidad de atacar el problema de una manera holística, es decir aplicar una técnica a la imagen como una unidad. La idea principal de estos métodos es transformar el problema de la detección de

caras, en un problema de reconocimiento de patrones y así utilizar toda la batería de herramientas disponibles en el área. En raras ocasiones se va a utilizar de manera explícita el hecho de que lo que se está buscando es una cara. Existe una gran cantidad de técnicas distintas entre ellas destacamos las siguientes:

Métodos de sub-espacios lineales

Esta gama de métodos fue la primera en surgir dentro de esta sub-familia. Se basan en representar las imágenes de caras en espacios lineales. Para esto buscan encontrar el espacio lineal al que pertenecen mediante un análisis estadístico. Existen diferentes aproximaciones entre ellas: PCA (Análisis de Componentes Principales), ICA (Análisis de Componentes Independientes), LDA (Análisis de Discriminante Lineal). Cada uno de estos métodos se diferencia del otro en la forma de hallar el subespacio de caras. Al día de hoy son los métodos más explorados, posiblemente debido a su generalidad para aplicarlos a toda clase de problema de reconocimiento de patrones. En el presente trabajo se ha implementado y estudiado algunos de estos algoritmos pero aplicándolos directamente al problema de reconocimiento de caras. Los resultados obtenidos se muestran en el capítulo 5.

Redes neuronales

Las redes neuronales surgen en los años cincuenta con la intención de simular el comportamiento del sistema nervioso humano. Se basan en conexiones entre pequeñas y sencillas estructuras (neuronas). En los últimos años ha habido una fuerte tendencia a aplicar este tipo de estructuras a problemas de reconocimiento de patrones. Las redes neuronales necesitan de una etapa inicial de aprendizaje con conjuntos representativos de entrenamiento, para que luego el funcionamiento sea el deseado. Existen modelos de red de distinta complejidad (cantidad de neuronas, tipo de neuronas, cantidad de capas de neuronas, conexiones entre neuronas) con las que se obtienen distintas performances. También existen aproximaciones locales de redes neuronales (no holísticas), cuyo objetivo es encontrar diferentes estructuras locales y luego en base a ellas reconocer el objeto en cuestión.

Análisis probabilísticos

Además de los dos métodos vistos anteriormente, existe otra serie de métodos estadísticos para la detección de caras. Muchos de ellos se basan en principios fundacionales del reconocimiento de patrones como pueden ser el principio de máxima verosimilitud o la distancia de Kullback-Leibler. Esta clase de métodos busca estimar distribuciones de probabilidad mediante histogramas y luego comparar los mismos frente a histogramas medios aprendidos estadísticamente de imágenes con caras.

3.4. Reconocimiento de Caras

Una gran variedad de métodos de reconocimiento de caras han sido propuestos durante los últimos treinta años. El reconocimiento de caras es un tema desafiante que ha capturado el interés de muchos grupos de investigadores en las más variadas áreas y disciplinas tales como la psicología, reconocimiento de patrones, redes neuronales, visión por computador, entre otros. Es por esto que la literatura en este tema es realmente vasta y diversa. Se toma en este trabajo la siguiente categorización de alto nivel [27]:

Métodos holísticos Utilizan toda la imagen de la cara como entrada al sistema de reconocimiento y la utilizan como la unidad básica de procesamiento.

Métodos basados en características locales Se extraen características locales, como ser ojos, nariz, boca, etc. Sus posiciones y estadísticas locales constituyen la entrada al sistema de reconocimiento.

Además, existen métodos híbridos que combinan técnicas holísticas y locales. En esta sección se describe algunas técnicas para el reconocimiento, que a su vez están estrechamente relacionadas con el modelo de cara que se adopta. El mismo determina las características a extraer de la imagen y las posibles medidas de similitud para la etapa de confrontación. En la figura 3.8 se muestra una clasificación de las técnicas descritas en las secciones siguientes.

3.4.1. Métodos Holísticos

A. Análisis de Componentes: PCA, ICA, LDA

Análisis de Componentes Principales (PCA) Considerando la distribución de imágenes que contienen caras, se intenta capturar la variabilidad de estas imágenes, buscando la independencia de cualquier cara o característica particular. Esto se logra aplicando la transformada de Karhunen-Loeve al espacio de caras. Se denomina PCA [9] dado que se busca extraer, de un conjunto de imágenes de entrenamiento, un subespacio cuya base maximice la varianza del espacio original. A estos vectores se les denomina *Eigenfaces* dado que son los vectores propios correspondientes a los valores propios más grandes de la matriz de covarianza de las señales imagen. Se logra así reducir considerablemente la dimensión del problema pasando a trabajar con las proyecciones de las caras originales en el subespacio generado. Fijando luego una métrica, se implementa el reconocimiento encontrando la distancia entre el vector de características de la imagen de entrada y los vectores almacenados en la base de datos.

Análisis de Componentes Independientes (ICA) Este método intenta representar el espacio de caras en un subespacio que minimice la dependencia de

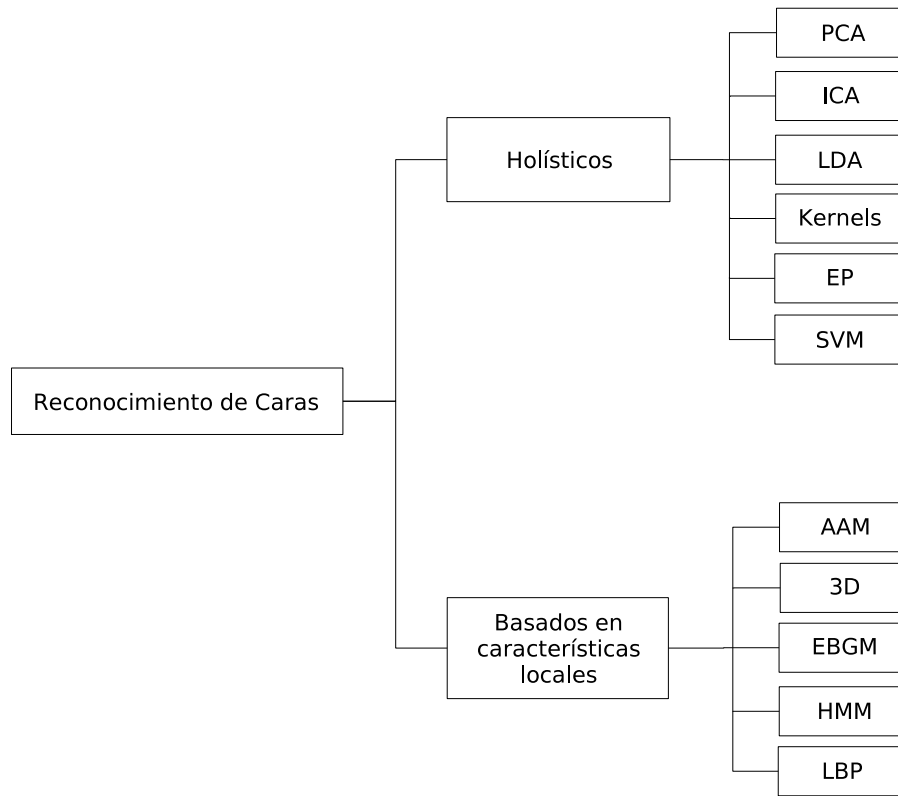


Figura 3.8: Clasificación de algunos de los métodos de Reconocimiento de Caras.

segundo y de mayor orden entre sus componentes. Se asume que las señales de entrada son combinaciones de fuentes no observables estadísticamente independientes [28]. Si la combinación es lineal, se puede definir una matriz de combinación cuyos coeficientes son los que definen la combinación lineal. Entonces ICA estima la matriz inversa de la matriz de combinación. Vale acotar que para aplicar ICA se utiliza previamente PCA para reducir la dimensionalidad del espacio original de caras para disminuir el costo computacional.

A.1. Análisis de Discriminante Lineal (LDA) Este algoritmo intenta llevar el espacio de caras a un subespacio de baja dimensionalidad que aumente la separabilidad de las clases presentes [29]. La idea del algoritmo es encontrar la base de vectores en un subespacio que mejor discrimine entre las diferentes clases, en el caso del reconocimiento las identidades. Se utilizan todas las muestras de todas las clases y se calcula la matriz de dispersión entre clases distintas (inter-clase) y la matriz de dispersión en la misma clase (intra-clase). Se busca maximizar la relación entre el determinante de la matriz inter-clase y el determinante de la matriz intra-clase. Los elementos de la

base que maximiza la relación anterior, se denominan *Fisherfaces* en honor a Robert Fisher que en 1936 propuso una solución a este método clásico de reconocimiento de patrones denominado *Fisher's Linear Discriminant (FLD)*.

Existen varias publicaciones que intentan poner en igualdad de condiciones estos métodos para poder evaluarlos y compararlos. En un estudio muy interesante [30] se compara los tres algoritmos utilizando diferentes nociones de distancia. Lo sorprendente, es que se concluye que ninguno de los algoritmos es superior y que deberá tenerse en cuenta la aplicación específica en la que se esté trabajando para seleccionar el adecuado. De todas formas, el algoritmo que obtiene un desempeño superior al resto en la mayoría de las aplicaciones es LDA, el único de los tres que realiza un aprendizaje supervisado. En la sección 5.2 se profundiza en el estudio de estas técnicas y se analiza su desempeño con la base IIE.

Los algoritmos antes mencionados se toman como base para la comparación del resto de los algoritmos descritos en esta sección. En esencia, todos los métodos comparten la hipótesis de que el complejo fenómeno visual humano en cuanto al reconocimiento de caras es intrínsecamente de baja dimensionalidad.

B. Métodos basados en Kernels

Este grupo de métodos son una generalización a los métodos de análisis de componentes (PCA, ICA, LDA) [31, 32, 33]. En los métodos de componentes, se construye un subespacio que cumpla determinadas restricciones y luego se elige una base que lo genere de alguna manera particular (componentes de mayor potencia, mejor discriminación, etc). La novedad introducida por los métodos de Kernels es poder tener en cuenta momentos de mayor orden sin tener un costo computacional excesivamente grande. Buscan llevar el problema de clasificación a un espacio de mayor dimensión donde las clases sean linealmente separables. Para esto se realiza lo siguiente:

1. Se mapean los vectores de entrenamiento a través de una función no lineal que lleva los puntos a un espacio de mayor dimensión.
2. Se plantea un problema equivalente al problema de PCA en dicho espacio. Existen algunas variantes que intentan resolver el problema de ICA o LDA en su lugar.
3. Se resuelve el problema equivalente, utilizando el *Kernel Trick*, que es una manera simplificada de resolver el problema de PCA en el espacio de mayor dimensión. Si el mapeo no lineal cumple determinadas condiciones particulares, se pueden realizar todos los cálculos de la resolución del problema equivalente, sin necesidad de mapear los vectores en el espacio de mayor dimensión. Para esto existen diferentes

funciones, llamadas *núcleos* que posibilitan dicho *atajo*. Entre ellas se destacan los núcleos gaussianos y polinómicos.

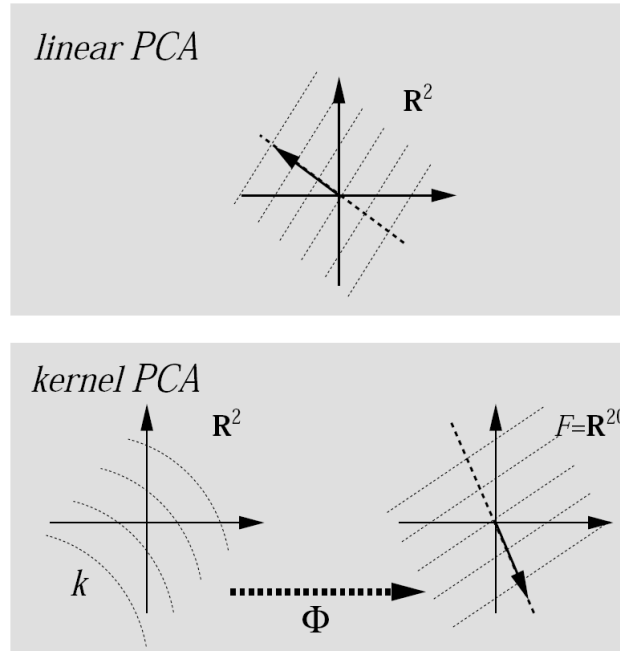


Figura 3.9: Ejemplo de aplicación de KPCA. Curvas de nivel de distancia constante al vector propio principal. Arriba se observa las curvas para el vector propio principal en el espacio de entrada (PCA). Abajo se muestran las curvas para el vector propio principal en el espacio de alta dimensión (KPCA).

En la figura 3.9 se puede ver un ejemplo de aplicación de los núcleos para el caso particular de KPCA (Kernel PCA). Se observa como quedan las curvas de nivel de proyección constante al vector propio principal. En el caso de PCA, las curvas son paralelas entre sí. En cambio en KPCA, las curvas en el espacio original son no lineales, buscando así poder clasificar de mejor manera los datos. Lo interesante es que existe una manera de realizar dicha clasificación sin necesidad de realizar el mapeo no lineal y así evitar costos computacionales enormes.

Dentro de las ventajas de estos métodos podemos destacar que obtienen un mejor desempeño que los métodos de Análisis de Componentes. Además intentan capturar la información presente en los momentos estadísticos de mayor orden. Como contrapartida requieren un mayor costo computacional que los métodos de subespacios lineales y además presentan los mismos problemas frente a iluminación no uniforme y variación de pose.

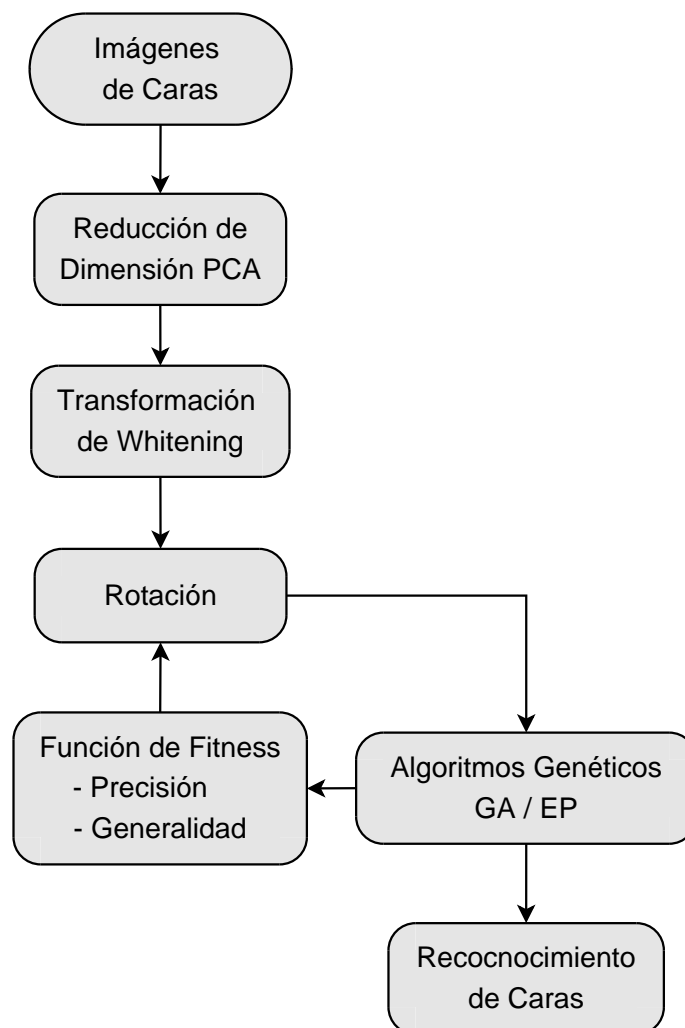
C. Evolutionary Pursuit (EP)

Este método, al igual que PCA, ICA y LDA se basa en el análisis de componentes. EP plantea una manera novedosa de obtener una base de vectores *eficiente* para la representación de las imágenes de caras [34, 35]. Para encontrar la base, se realiza una búsqueda de manera de maximizar una función *fitness* que mide al mismo tiempo la precisión de la clasificación y la habilidad de generalización del sistema. Como el problema de buscar la base óptima es un problema de dimensión alta se utiliza un tipo de Algoritmos Genéticos (GA) al cual se le llama Evolutionary Pursuit.

Para encontrar las EP-caras se realiza lo siguiente:

1. Se reduce la dimensión de los datos mediante PCA.
2. Se transforma el espacio anterior mediante una transformación de Whitening.
3. En el espacio Whitened-PCA, se realiza el siguiente procedimiento hasta cumplir un número máximo de iteraciones o la performance buscada.
 - a) Se realizan varias rotaciones entre pares de vectores de una base del espacio y luego se selecciona un conjunto de ellos. Se codifica la rotación efectuada mediante representación en palabras de bits, utilizando 10 bits para cada ángulo rotado (precisión de 0.09 grados).
 - b) Se calcula la función *fitness* para medir la precisión y la generalización del sistema. Precisión hace referencia al buen desempeño del algoritmo hasta ahora, mientras que por generalización se entiende la performance esperada en pruebas futuras.
 - c) Se calculan los ángulos y los vectores que maximizan la función *fitness* y se guardan como la mejor solución hasta el momento.
 - d) Se itera a un nuevo subconjunto de ángulos y por lo tanto de vectores rotados, mediante Algoritmos Genéticos.
4. Con la base *óptima* hallada se realiza reconocimiento de caras siguiendo un procedimiento análogo al de los métodos por subespacios lineales.

En la figura 3.10 se muestra un diagrama resumiendo el procedimiento descrito. Esta técnica novedosa obtiene una mejor performance que PCA, ICA o LDA debido a que es más general, es decir aprenden mejor que los anteriores con las mismas imágenes de entrenamiento. La desventaja principal es que se debe resolver un problema de búsqueda en un espacio de muy alta dimensión.

Figura 3.10: Diagrama de bloques del algoritmo *Evolutionary Pursuit*.

Clasificación Bayesiana

La clasificación Bayesiana presenta una nueva medida de similitud probabilística que utiliza las diferencias de intensidad de las imágenes ($\Delta = I_1 - I_2$) como características de las variaciones típicas en la apariencia de un individuo [36]. Se definen dos tipos de variaciones entre las imágenes de caras:

- Variaciones Intrapersonales (Ω_I) - correspondientes a cambios debido a la diferentes expresiones faciales del mismo individuo.
- Variaciones Interpersonales (Ω_E) - correspondientes a las diferencias entre distintos individuos.

De esta forma la medida de similitud queda expresada en términos de probabilidades como:

$$S(I_1, I_2) = P(\Delta \in \Omega_I) = P(\Omega_I|\Delta)$$

Utilizando la regla de Bayes se puede escribir de la forma:

$$P(\Omega_I|\Delta) = \frac{P(\Delta|\Omega_I)P(\Omega_I)}{P(\Delta|\Omega_I)P(\Omega_I) + P(\Delta|\Omega_E)P(\Omega_E)}$$

donde las probabilidades a priori $P(\Omega)$ están dadas por las condiciones de trabajo (ej: número de imágenes de test vs. total en la base) o por otras fuentes de información a priori de las imágenes a confrontar.

Para tomar la decisión se utiliza la regla de *Maximum a posteriori* (MAP), donde se considera válida la cara si se cumple $S(I_1, I_2) > \frac{1}{2}$. Existe una medida de similitud más simple llamada *Maximum Likelihood* (ML) dada por $S'(I_1, I_2) = P(\Delta|\Omega_I)$ donde solo se considera variaciones intrapersonales.

Para estimar las densidades de probabilidad $P(\Delta|\Omega_I)$ y $P(\Delta|\Omega_E)$ se utilizan técnicas que resuelven el problema de realizar el entrenamiento con pocas muestras para la gran dimensión de los datos. Estas herramientas utilizan PCA para reducir la dimensión de los datos. Si bien este método no utiliza información a priori dado por el hecho que esté trabajando con caras, sí agrega un marco teórico para modelar las posibles variaciones que se presentan. Los índices de reconocimiento son buenos. La principal dificultad de este método es la estimación de las probabilidades a priori, dada por las pocas muestras disponibles para el entrenamiento.

D. Support Vector Machine (SVM)

Es una herramienta genérica para resolver problemas de reconocimiento de patrones planteada en la década pasada por Cortes y Vapnik [37]. Dado un conjunto de puntos en un determinado espacio que pertenecen a dos clases distintas, SVM encuentra el hiperplano que separa la mayor cantidad de puntos de la misma clase del mismo lado. Esto se realiza maximizando la distancia de cada clase al hiperplano de decisión denominado OSH (*Optimum Separating Hyperplane*). Los puntos más cercanos al hiperplano, de cada conjunto en cuestión, son los llamados vectores soporte (*support vectors*). En [38, 39] se intenta resolver el problema de reconocimiento de caras mediante SVM. Para aplicar SVM a este problema particular es necesario reducir la dimensionalidad de los datos para lo que se utilizan herramientas como PCA o LDA. En la figura 3.11 se observa distintos planos de decisión y el plano de decisión obtenido mediante SVM.

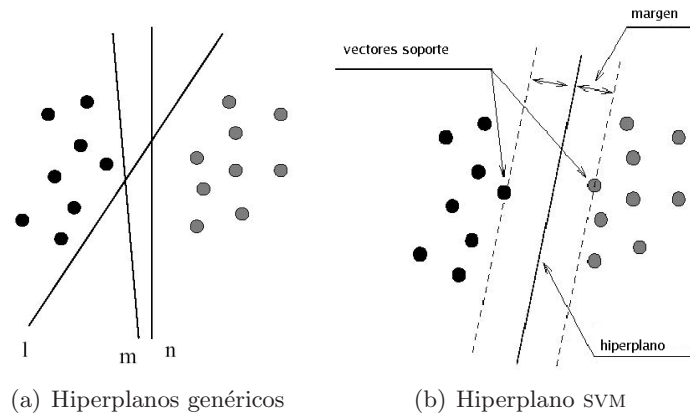


Figura 3.11: Hiperplanos de decisión para un problema de clasificación de dos clases. A la izquierda posibles hiperplanos de decisión, a la derecha hiperplano de decisión obtenido mediante SVM.

En general se obtienen resultados superiores a los métodos clásicos de análisis de componentes y también se logra cierta robustez frente a cambios en la iluminación.

3.4.2. Métodos Basados en Características Locales

A. Active Appearance Model (AAM)

Active Appearance Model (AAM) es un modelo estadístico de la forma y la apariencia en niveles de gris del objeto de interés que se puede generalizar a casi cualquier ejemplo válido de dicho objeto [40, 41]. Ajustar dicho modelo a una imagen implica encontrar los parámetros del modelo que minimizan la diferencia entre la imagen y una síntesis del modelo, proyectado en la imagen. La gran cantidad de parámetros presentes en el modelo hacen de éste un problema complicado. El modelo se genera combinando modelos de variación de la forma y la apariencia de la cara. Estos se construyen a partir de imágenes de prueba donde se marcan puntos de interés (puede ser en forma manual o automática).

Los modelos de forma y apariencia, se generan considerando distintos puntos marcados en la imagen como vectores y aplicando PCA a dicha información. Cada ejemplo de forma y apariencia puede entonces ser representado por los coeficientes correspondientes. Dado que existe correlación entre la variación de la forma y la apariencia, se aplica PCA a la concatenación de los vectores de los modelos anteriores y con ello se construye el modelo final.

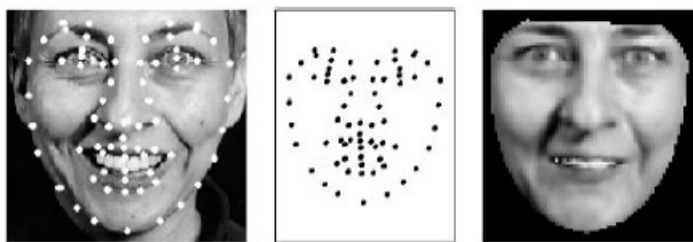


Figura 3.12: Ejemplos de puntos de interés.

Figura 3.13: Ejemplos de los resultados obtenidos en el ajuste del modelo utilizando *Active Appearance Model*.

Teniendo un modelo combinado de forma y apariencia y una posición inicial aproximada, se propone un algoritmo para ajustar los parámetros del modelo y sintetizar una cara lo más similar posible a la original. La figura 3.13 muestra los resultados obtenidos. Aplicado al Reconocimiento de Caras para cada imagen conteniendo un rostro se ajusta un modelo de forma y apariencia y los coeficientes resultantes se toman como representantes de la identidad tanto para el registrado como para el reconocimiento. Se utiliza para dicho ajuste información extra, proveniente de la relación existente entre la imagen error y el error en los parámetros del modelo.

Dentro de las ventajas se destaca poder obtener representación de vistas no conocidas de la cara a partir de la interpolación de poses conocidas. Como aspecto negativo se menciona la complejidad en la implementación. Además los resultados son fuertemente sensibles a la correcta selección de los puntos principales.

B. Modelado 3D

La idea central de esta familia de técnicas es ajustar un modelo genérico 3D a cada imagen en cuestión [42, 43]. Para esto es necesario contar con imágenes 3D (de rango, por ejemplo). Existen distintas técnicas de adquisición de imágenes (o reconstrucción) 3D, entre ellas: cámaras estereo, escaners láser, SLS (*Structure Light System*), secuencia de imágenes 2D.

Primero mediante una serie de imágenes 3D de unos pocos individuos se construye un modelo de cara genérico 3D (por ejemplo un modelo de malla triangular). Luego dicho modelo es ajustado de manera iterativa, para cada individuo de la base completa. Finalmente se registra las variaciones respecto al modelo genérico y a su vez alguna información de textura.

Los métodos basados en modelos 3D son menos sensibles a variaciones en la pose o en cambios en la iluminación que otros métodos. Su principal contrapartida es que necesitan imágenes de adquisición compleja (Adquisición 3D), además de ser extremadamente costos computacionalmente.

C. Elastic Bunch Graph Matching (EBGM)

Este método consiste en extraer una representación de la cara en forma de grafo y el reconocimiento se realiza comparando los grafos correspondientes a las distintas imágenes [44, 10]. Se define un conjunto de puntos principales, como por ejemplo las pupilas, esquinas de la boca, etc. Un grafo etiquetado representando una cara consiste en lo siguiente: N nodos ubicados en los puntos principales y las aristas que se forman entre parejas de nodos. Cada nodo es etiquetado con los denominados *jets* y cada arista es etiquetada con la distancia entre los nodos correspondientes.

Los *jets* se basan en una transformada wavelet definida como la convolución de la imagen con una familia de núcleos de Gabor con distintas frecuencias y orientaciones. Una representación de esto se muestra en la figura 3.14.

Para encontrar los puntos principales en nuevas caras se necesita una descripción general del objeto cara. Dicha descripción debe cubrir una amplia gama de variaciones como son diferentes formas de ojos, boca, diferentes sexos, razas, etc. Para ello se combina un grupo representativo de grafos modelo en una estructura del tipo *stack* llamada *Face Bunch Graph* (FBG). Cada grafo modelo tiene la misma estructura y los nodos refieren a los mismos puntos principales. La figura 3.15 muestra la representación gráfica del FBG. El objetivo en una imagen de prueba es encontrar los puntos principales y obtener el grafo que maximice la similitud con el FBG considerado. El grafo resultante es la representación de la cara presente en la imagen.

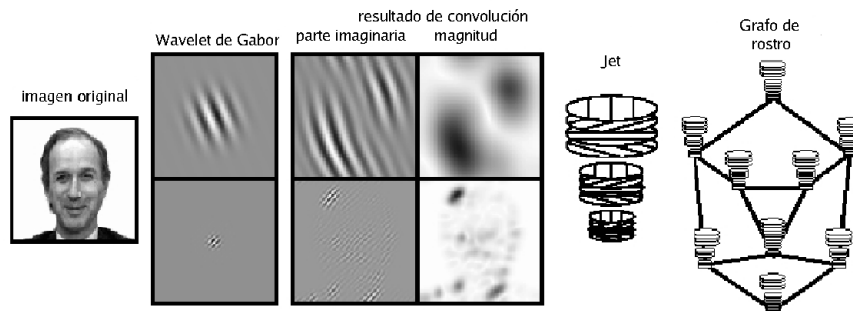


Figura 3.14: Representación de la cara en forma de grafo.

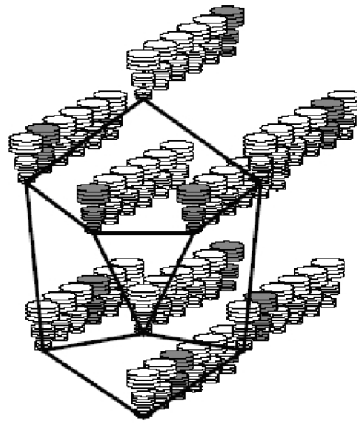


Figura 3.15: *Face Bunch Graph*.

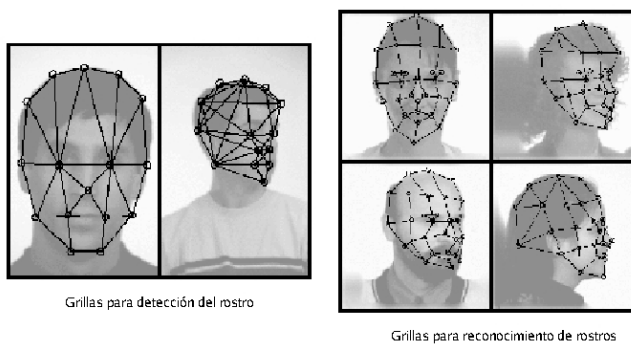


Figura 3.16: Ejemplos de grafos obtenidos utilizando EBGm.

Luego de extraído el grafo de la imagen de prueba, se utiliza una función de similitud para evaluar la similitud de dicho grafo con los grafos de la base de datos. Un criterio de decisión puede ser tomar el máximo valor de similitud como cara encontrada.

Si bien es necesaria una etapa de marcado manual de puntos para la generación del modelo estadístico, el algoritmo logra buenos resultados con una única imagen por persona. El problema principal de este algoritmo es el costo computacional de convolucionar a la imagen con una familia de núcleos de Gabor, lo cual implica tiempos de procesamiento elevados.

D. Modelos de Markov escondidos (HMM)

Se trata de un conjunto de modelos estadísticos utilizados para caracterizar las propiedades estadísticas de una señal [45]. Estos modelos son de gran utilidad para la representación de dependencias estadísticas en problemas que tienen una temporalidad inherente, por lo que han tenido éxito en aplicaciones como reconocimiento de voz y de gestos [46]. Para el caso de imágenes, las dependencias temporales se transforman en dependencias espaciales, por lo que es necesario encontrar una forma de recorrer una señal que en vez de ser unidimensional, como es en el caso de la voz, sea de dos dimensiones.

Formalmente un HMM consiste en dos procesos interrelacionados:

1. un proceso subyacente, modelado por una cadena de Markov no observable con un número finito de estados, una matriz de transición de estados y una distribución de probabilidad de estados inicial.
2. un proceso observable dado por los vectores de observación y relacionado por un conjunto de densidades de probabilidad asociadas a cada estado.

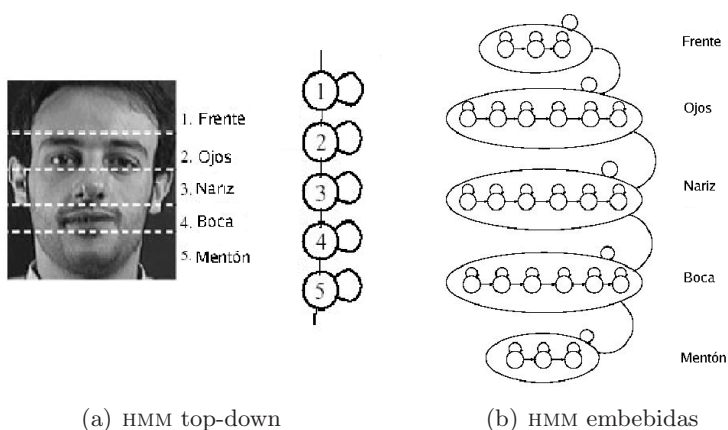
Para aplicar HMM al reconocimiento de caras, lo que se hace es dividir la imagen en bloques y se aplica la DCT o KLT. Luego ciertos coeficientes de la transformada del bloque se utilizan como los vectores de observación.

Se modela la cara por una cadena de Markov [16], la cual se entrena a los efectos de encontrar los parámetros, utilizando el algoritmo EM maximizando $P(O/\lambda)$, siendo O el vector de observación y λ la cadena de Markov. De esta forma se obtiene una cadena λ para cada individuo de la base. Luego para tomar la decisión se elige la cadena λ^* que maximice $P(O/\lambda_i)$. Existen distintos modelos para representar la cara, a los efectos de recorrer la señal bidimensional:

HMM top-down Se toman bloques que se corresponden con las distintas

regiones horizontales de la cara desde arriba hacia abajo. De esta manera los bloques corresponden a la frente, los ojos, la nariz, la boca y el mentón. En la figura 3.17(a) se puede observar esta idea.

HMM embebidas Se hace una cadena de cadenas, donde una de nivel inferior recorre la imagen en sentido horizontal y luego una de nivel superior recorre la imagen en sentido vertical. Esta idea se puede ver en la figura 3.17(b).



(a) HMM top-down

(b) HMM embebidas

Figura 3.17: A la izquierda, división en franjas de un rostro para el algoritmo HMM. A la derecha, las cadenas de nivel inferior y superior para el algoritmo HMM.

Existen otras formas de modelar la cara, pero todas mantienen la idea de dividir la misma en bloques y tomar como vectores de observación algunos coeficientes de la DCT o KLT. En la figura 3.18 se muestra otro ejemplo de modelado.

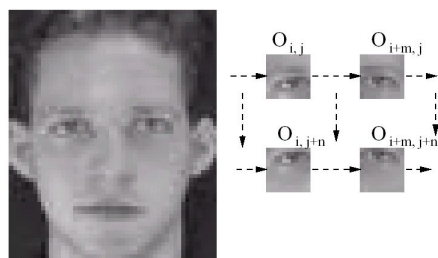


Figura 3.18: Alternativa de vectores de observación del algoritmo HMM.

La dificultad en este tipo de algoritmos se centra en el entrenamiento de las cadenas. Además es costoso desde el punto de vista computacional.

E. Local Binary Patterns (LBP)

El operador original LBP [47] es una herramienta interesante como descriptor de textura. Este operador recorre la imagen y etiqueta los píxeles de la misma umbralizando una vecindad de 3×3 respecto al valor del píxel en el que se encuentra, considerando el resultado como un número binario. Luego el histograma de las etiquetas puede utilizarse como descriptor de textura. En la figura 3.19 se ilustra el funcionamiento del operador LBP básico.

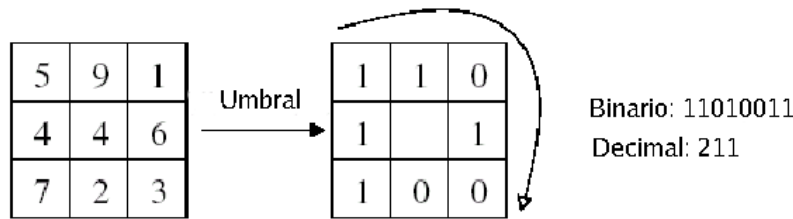


Figura 3.19: Ejemplo del operador LBP.

Algunas variantes al operador básico fueron introducidas posteriormente [48]. Una de ellas es el hecho de considerar entornos de tamaño variable. Usando vecindades circulares e interpolando los valores de los píxeles en donde cualquier radio es válido. Otra extensión es el tomar en cuenta solamente los patrones uniformes, siendo éstos los que tienen al menos dos transiciones 0-1 o 1-0. Esta idea se apoya en el hecho de que la mayor parte de los patrones en una imagen son de tipo uniforme. En [49] se puede ver más en detalle este operador.

El histograma obtenido mediante el procedimiento descrito contiene información acerca de la distribución local de los micropatrones, como bordes, puntos y otros, sobre la imagen completa. Para tener una representación eficiente de la cara es necesario además almacenar información espacial. Por lo tanto, la imagen se divide en regiones uniformes mediante una grilla cuadrangular (bloques 7×7). Luego se calculan los histogramas para cada bloque y se concatenan todos para obtener el patrón de la cara. De esta forma se logra efectivamente una descripción de la cara en tres escalas o niveles: las etiquetas contienen información de los patrones a nivel de los píxeles, los histogramas en una pequeña región producen información a nivel regional y los histogramas regionales son concatenados para crear una descripción global de la cara.

Como aspecto más positivo se resalta que es una representación simple y eficiente de la cara la cual logra obtener buenos resultados.

Parte II

Análisis y Diseño

Capítulo 4

Detección de Caras y Normalización

4.1. Introducción

Como se describe en el capítulo anterior, la detección de caras es una etapa crucial en el proceso de reconocimiento. El módulo utilizado para completar esta tarea debe ofrecer una elevada performance que garantice la correcta localización de las mismas para el posterior reconocimiento. En este capítulo se describe un primer acercamiento al trabajo con caras, se detalla el algoritmo de detección utilizado en el sistema final y por último se describe la etapa de normalización aplicada a las imágenes.

4.2. esCARAbajo: Una Primera Experiencia

En el marco del proyecto final del curso Tratamiento de Imágenes por Computadora¹ se implementa el software denominado esCARAbajo. El mismo implementa un detector de caras en imágenes de fondo complejo, utilizando información de color. Las técnicas utilizadas se basan fuertemente en el artículo *Face Detection in Color Images* [7].

Las distintas tonalidades de piel se modelan a través de un *cluster* de piel definido en un espacio de color. Luego, dada una imagen de fondo complejo, se segmentan las regiones pertenecientes al *cluster* de piel y se etiquetan como posibles candidatos a caras. Una segunda etapa del algoritmo utiliza técnicas de localización de características, en este caso ojos y boca, para validar los distintos candidatos.

4.2.1. Segmentación utilizando información de color

La apariencia del color de piel en la imagen depende de las condiciones de iluminación existentes. Se utiliza entonces una técnica de compensación

¹Curso dictado por GTI, IIE.

de iluminación que busca normalizar dicha apariencia. Para esto se define un blanco de referencia considerando los píxeles que pertenecen al 5% superior de valores de luminancia. Luego, los canales R, G y B de la imagen se ajustan de manera que el valor medio de los píxeles pertenecientes al blanco de referencia sea escalado linealmente a 255.

El siguiente paso consiste en segmentar las regiones de la imagen que pertenecen al *cluster* de piel elegido. Se trabaja en esta implementación con dos *clusters* diferentes, uno definido en el espacio RGB y otro definido en el espacio YCbCr. En la figura 4.1 se observan ejemplos de resultados obtenidos utilizando los distintos *clusters*. Se encuentra diferencias entre las regiones clasificadas como piel dependiendo del *cluster* utilizado. El *cluster* YCbCr clasifica correctamente todas las regiones de piel a costa de permitir muchos falsos positivos. Por otro lado el *cluster* RGB presenta pocos falsos positivos siendo más restrictivo en la clasificación.



Figura 4.1: Ejemplos de los resultados obtenidos en la segmentación utilizando distintos *clusters*. En primer lugar se observa la imagen original, luego los resultados para el *cluster* RGB y finalmente los correspondientes al *cluster* YCbCr.

Se obtiene así una imagen binaria, a la cual se aplican técnicas de morfología para eliminar regiones no deseadas y etiquetar posteriormente las regiones conexas resultantes. Por último se implementa un mecanismo para determinar, entre los candidatos resultantes, cuáles tienen mayor probabilidad de ser caras. Para esto se utilizan una serie de descriptores que, combinados, dan un *score* al candidato que lo habilita o no a pasar a las siguientes etapas. Los descriptores utilizados son:

Área: Se considera que el área ocupada por la cara no puede ser menor que 400 píxeles.

Factor de forma: Cuantifica la regularidad de una región. Toma el valor 1 para círculos y tiende a cero mientras más alargada sea la región. El valor elegido para el factor de forma mínimo es de 0.3.

Solidez: Indica qué porcentaje de la región está dentro del cerco convexo de la misma. El cerco convexo es la mínima región convexa que contiene a la región.

Extensión: Da una medida del área que ocupa una región en el cerco *boundingBox* correspondiente. El *boundingBox* es el rectángulo de menor tamaño que encierra a la región de interés. También se asume que las caras tendrán un valor alto de extensión.

Relación de aspecto: Relación entre el diámetro mayor y el diámetro menor de la región. Con este parámetro se tiene una medida cuantitativa de la regularidad de la región.

Es claro el compromiso existente entre la rigidez de los umbrales utilizados y la cantidad de candidatos descartados.

4.2.2. Validación de los candidatos a caras

Luego de obtenidos los candidatos a caras se procede a la validación de los mismos a través de la búsqueda de elementos característicos, como son los ojos y la boca. Para encontrar la ubicación de los ojos y la boca en un candidato se realizan *mapas* derivados directamente de la luminancia y las cromas de la imagen original. Un *mapa* es una imagen en tonos de gris que resalta determinadas características relevantes de la imagen para facilitar la detección de elementos concretos, en este caso ojos y boca.

Mapa de Ojos: EyeMap

La construcción del mapa de ojos se basa fundamentalmente en dos aspectos:

- Los ojos presentan baja intensidad de rojo (bajo Cr) y alta intensidad de azul (alto Cb).
- Los ojos presentan píxeles muy claros así como píxeles muy oscuros.

Considerando el primer aspecto se construye el *EyeMap de cromas* que resalta los ojos utilizando información de las cromas (Cr y Cb). Tomando el segundo aspecto se crea el *EyeMap de luminancia*, que resalta aquellas zonas que tienen altos valores de luminancia y que a su vez tienen vecinos cercanos con bajos valores de luminancia. Es decir, se resaltan los altos contrastes. Combinando los anteriores, *EyeMap de cromas* y *EyeMap de luminancia*, se obtiene el *EyeMap* final. Por detalles en la implementación de ambos mapas se recomiendan los trabajos de Hsu [7] y Jackway [50].

Con el objetivo de definir cuales son las regiones de ojos dentro del *EyeMap* obtenido, se utiliza un umbral adaptivo. El umbral de decisión es una combinación lineal entre el promedio del mapa y el máximo valor del mapa. Finalmente obtenemos un mapa binario con todos los píxeles que están por encima del umbral. Para terminar esta etapa se realiza un filtrado

morfológico que elimina aquellas regiones que no tienen un tamaño mínimo, aplicándole primero apertura y luego cerradura. La figura 4.2 muestra un diagrama representativo de estas últimas etapas.

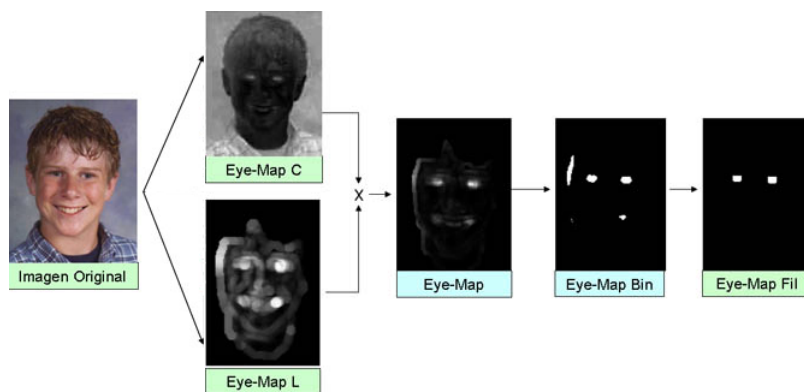


Figura 4.2: Diagrama de bloques de creación del mapa de ojos.

Mapa de boca: MouthMap

La construcción del mapa de boca se basa en un único aspecto:

- En la boca, la relación *intensidad de rojo / intensidad de azul* es mayor que en cualquier otra región de la cara.

Se genera entonces el *MouthMap* a partir de la información de cromas. Se aplican luego técnicas de dilatación, umbralizado y filtrado morfológico para regularizar el mapa y eliminar regiones que no alcanzan un tamaño mínimo. En la figura 4.3 se muestra un diagrama de creación del mapa de boca.

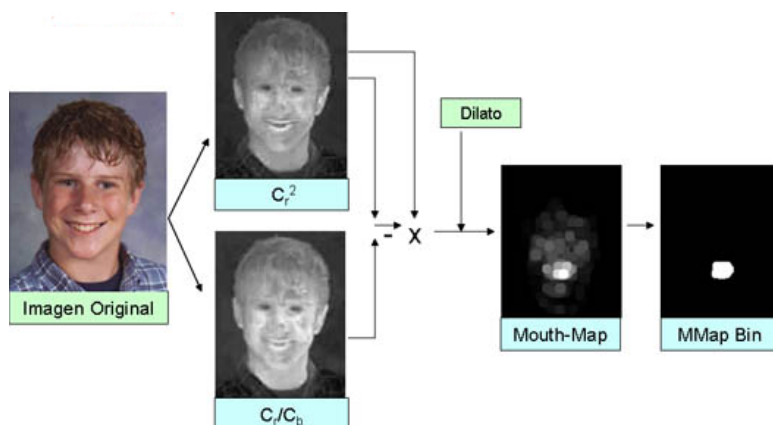


Figura 4.3: Diagrama de bloques de creación del mapa de boca.

El proceso de validación de candidatos a caras se puede resumir en los siguientes pasos:

1. Para cada candidato a cara, segmentado a partir de la información de color, se generan los correspondientes *EyeMap* y *MouthMap*.
2. Se toman todas las combinaciones posibles de dos ojos y una boca, presentes en los mapas, a lo que se denomina *candidato*.
3. Para cada *candidato*, que supere un conjunto de restricciones mínimas, se calcula un score que contempla qué tan buen candidato es. Dicho score considera aspectos como simetría, orientación, simetría en el tamaño de los ojos, coherencia con el tamaño de la cara.
4. El *candidato* que arroja el máximo score es seleccionado para dar las coordenadas de los ojos y boca en la cara localizada. En caso de no encontrarse ojos y boca, o si el score de mejor *candidato* no supera un umbral mínimo, se descarta la presencia de una cara.

La figura 4.4 muestra un diagrama de bloques general del sistema punta a punta.

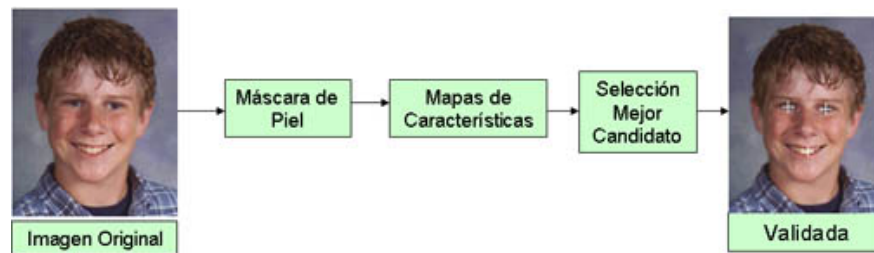


Figura 4.4: Diagrama de bloques del algoritmo punta a punta. Se observan los ojos y la boca correctamente localizados en la imagen de salida.

4.2.3. Evaluación del algoritmo

Se evalúa el algoritmo con diferentes bases de imágenes. Las estadísticas se toman para una base de 300 imágenes tipo foto-carnet [51], con poses de frente o levemente rotadas, de tamaño 150×200 píxeles. Una cara es correctamente detectada si la ubicación encontrada para los ojos y la boca difiere de la real dentro de cierta tolerancia. De otra manera se considera la detección como un falso positivo.

Resultados Obtenidos	
Detección Correcta	55 %
Falsos Positivos	27 %

Como se aprecia en la tabla anterior los resultados obtenidos no son buenos. Por la mala performance del algoritmo se descarta su utilización como bloque detector de caras. Sería necesario continuar trabajando en las distintas etapas del mismo para mejorar su desempeño. Se decide no continuar en el desarrollo de esta implementación, con lo cual se da paso a la búsqueda de algoritmos alternativos para la detección.

4.3. Algoritmo de Detección Utilizado en el Sistema

En esta sección se describen los principios básicos del algoritmo de detección de caras elegido. El mismo cuenta con la funcionalidad de localizar los ojos en la cara detectada, elemento necesario en la posterior normalización de la imagen.

4.3.1. Aspectos Generales

Retomando la clasificación de algoritmos de detección del capítulo 3, el algoritmo seleccionado pertenece al grupo de los basados en la imagen, implementando redes neuronales. La documentación principal del mismo se encuentra en [52]. Se resaltan cuatro características principales:

- Utiliza como características los wavelets denominados *Haar-like features*, también utilizados en [53].
- Realiza búsqueda multiescala piramidal similar a la descrita en [54].
- Implementa una cascada de clasificadores como en [53], consumiendo más tiempo en procesar las subimágenes que contienen caras.
- Entrena basado en una variante del algoritmo Adaboost llamada Gentleboost introducida en [55].

La implementación utilizada fue desarrollada en el *Machine Perception Laboratory* de la Universidad de San Diego, California. La misma forma parte del MPT, *Machine Perception Toolbox* [8], biblioteca de código libre. A los efectos de nuestra aplicación esta herramienta se utiliza para cumplir con los dos objetivos esenciales de esta etapa:

- Verificar que la imagen contenga una y sólo una cara.
- Determinar la posición de los ojos en la imagen.

4.3.2. Descripción Funcional

La implementación considerada consta de un módulo principal, entrenado con imágenes de tamaño fijo (24×24 píxeles), que decide si la ventana que se analiza es o no el objeto buscado. Por tratarse de un clasificador

para un tamaño fijo, el detector lo utiliza como un filtro de ventana móvil realizando una búsqueda espacial. Esta búsqueda se hace a distintas escalas, para poder encontrar objetos de distintos tamaños.

El clasificador es genérico, con lo cual el objeto a localizar queda definido por las imágenes con que se realiza el entrenamiento. Es así que se utiliza un módulo entrenado con imágenes de caras para la localización de caras y luego un módulo entrenado con imágenes de ojos para detectar ojos en las regiones clasificadas como caras.

Haar-like features

Las características elegidas se denominan *Haar-like features*, derivadas de las funciones base de los wavelets de Haar, que son simplemente ondas cuadradas. En la figura 4.5 podemos ver un ejemplo de las *Haar-like features* utilizadas. Cada característica corresponde a la diferencia de las sumas de los píxeles en los rectángulos blancos y los rectángulos grises.

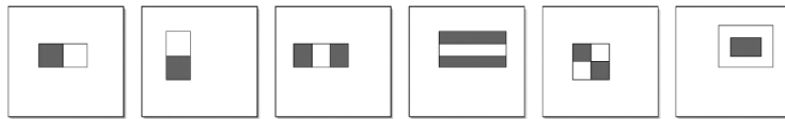


Figura 4.5: Ejemplo de Haar-like features.

Estas características pueden tener distinta ubicación, forma y tamaño. En este caso se trabaja con un tamaño de imagen de 24×24 píxeles, teniendo más de 170.000 características, por lo que se tiene sobredeterminado el espacio, aumentando considerablemente la dimensión. La razón principal de su uso es que se calculan de manera muy rápida a partir de una imagen auxiliar denominada *integral-image* [53], que corresponde a la integral doble de la imagen.

Dada la gran cantidad de características generadas, son fundamentales los algoritmos de entrenamiento que seleccionan las características más importantes, a los efectos de obtener la mejor performance posible. Con cada una de estas características es posible crear un clasificador *débil* como el siguiente:

$$h_k(x) = \begin{cases} +1 & \text{si } z_k(x) > \tau_k \\ -1 & \text{otro caso} \end{cases}$$

donde x es la sub-imagen analizada, $z_k(x)$ es la k -ésima *feature* y τ_k es el umbral correspondiente.

Si bien estos clasificadores *débiles* no son capaces de obtener la performance buscada, la combinación de varios de éstos permite llegar a clasificadores de buen desempeño.

Entrenamiento de clasificadores fuertes

A partir de los clasificadores *débiles* correspondientes a cada *característica* se construyen clasificadores *fuertes* de mayor complejidad y mejor desempeño. Un clasificador *fuerte* se crea a partir de la combinación lineal de clasificadores débiles,

$$H_M(x) = \frac{\sum_{m=1}^M \alpha_m h_m(x)}{\sum_{m=1}^M \alpha_m}$$

donde los $\alpha_m \geq 0$ corresponden a los pesos de cada clasificador *débil* y la decisión se toma según el $\text{Signo}[H_M(x)]$.

El clasificador *fuerte* es entrenado con un algoritmo llamado *Gentle-Boost*, que es una variante del *AdaBoost*, donde en lugar de realizar una optimización exacta para minimizar el error, se utiliza Newton-Raphson en cada paso de la optimización. Estos algoritmos de entrenamiento pertenecen a una familia reciente (1995) de algoritmos de entrenamiento denominada *boosting*. Buscan, a partir de la combinación de varios clasificadores *débiles*, obtener un clasificador potente de gran performance. Para más detalle ver [55].

Cascada de clasificadores

Si bien es posible obtener clasificadores *fuertes* con altos porcentajes de detección, aún podemos tener un número alto de falsos positivos. Una solución es obtener el *AND* de varios clasificadores para disminuir las falsas alarmas. Una variante de esto, introducida en [53], es entrenar clasificadores *fuertes* de distintos grados de complejidad, y aplicarlos en cascada como muestra la figura 4.6.

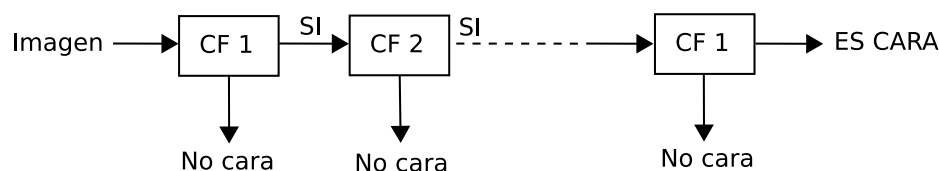


Figura 4.6: Cascada de clasificadores *fuertes*.

Cada clasificador es más complejo que en la etapa anterior (utiliza más *features*) y se entrena con muestras *no-cara* que fueron falsos positivos en

dicha etapa, lo que le permite resolver casos más difíciles. Esta técnica resulta muy eficiente puesto que en cada etapa el clasificador decide si la ventana en proceso sigue siendo analizada o debe ser descartada. Descartar ventanas tempranamente permite invertir más tiempo en las que realmente contienen caras. Cabe resaltar que es posible descartar aproximadamente el 60 % de los falsos candidatos con sólo dos *features*, logrando 100 % de detección correcta.

Búsqueda multiescala piramidal y post-procesado

Como el entrenamiento se realiza para un tamaño fijo (24×24 píxeles) es necesario realizar la búsqueda por ventanas en diferentes escalas. Para esto se realiza una búsqueda piramidal donde se recorre la imagen original y copias más pequeñas de la misma, reduciendo en cada paso en $1,2\times$, hasta llegar al tamaño mínimo (24×24 píxeles). Una ventaja de las características elegidas, descrita en [53], es el hecho de poder calcular las mismas a otra escala, sin tener que realizar el escalado previo de la imagen y luego el cálculo, sino operando directamente sobre las características ya obtenidas. Luego de aplicado el detector a distintas escalas, es factible que una cara que esté presente en la imagen se detecte en varias escalas simultáneamente, por lo que existe una etapa de post-procesado que se encarga de unir todas las detecciones solapadas como se muestra en la figura 4.7. De esta manera se eliminan todos los falsos positivos.

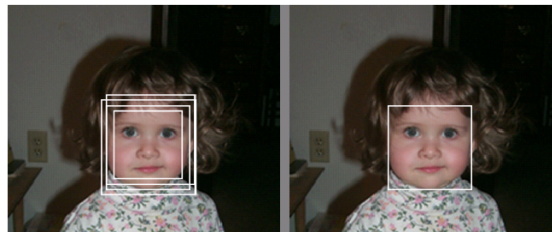


Figura 4.7: Post-procesado para unir detecciones múltiples.

Localización de ojos

En la figura 4.8 se resume el funcionamiento del detector de caras con una imagen de ejemplo. Allí se puede ver la detección inicial de la cara y la posterior localización de los ojos en la misma. El algoritmo utilizado para localizar los ojos es similar al del detector de caras. La diferencia radica en que se utiliza un solo clasificador *fuerte*, en lugar de varios en cascada. El mismo se entrena con imágenes de *ojos* y *no-ojos* e incluye información a priori: se sabe que sólo hay dos ojos por cara. Como muestras de entrenamiento positivas se toman ojos con distintas aperturas para dar robustez al detector. Utiliza además la estadística de la posición de los ojos en una cara, realizando la búsqueda únicamente hasta cierta distancia de la posición

media, tomando un intervalo de confianza del 99,9%.

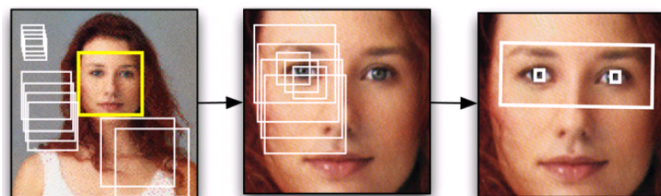


Figura 4.8: Ejemplo de funcionamiento del detector.

4.3.3. Evaluación

Se efectuaron dos evaluaciones distintas del detector, enfocadas a las dos funciones principales que éste debe cumplir. La primera mide la performance del mismo como detector de caras y la segunda evalúa la precisión que se logra al detectar los ojos en la cara.

Evaluación como detector de caras

Se utiliza para la evaluación la base de imágenes adquirida en este proyecto, denominada base IIE, detallada en el apéndice C. Los resultados obtenidos fueron ciertamente satisfactorios y se resumen en la gráfica de la figura 4.9.

Se observa una muy buena performance para los casos de tomas frontales, con lentes comunes y con iluminación no uniforme. El algoritmo demuestra ser robusto ante las diferentes poses y la presencia de lentes negros. No ocurre lo mismo en el caso de las oclusiones, como podemos apreciar en la toma con bufanda.

Evaluación como localizador de los ojos

En esta etapa se evaluó la precisión en la localización de los ojos, realizando la comparación de las coordenadas obtenidas automáticamente frente a las mismas registradas manualmente. La evaluación se realizó en aquellas fotos en las que solamente se encontró una cara. Se considera una detección correcta si ambos ojos están bien localizados, es decir si ninguna de sus coordenadas supera una tolerancia a convenir respecto del marcado manual. En la figura 4.10 se muestran los resultados sobre un total de 1250 fotos (Si consideramos solo las tomas de frente el total baja a 328). La tabla siguiente muestra un resumen de los resultados obtenidos:

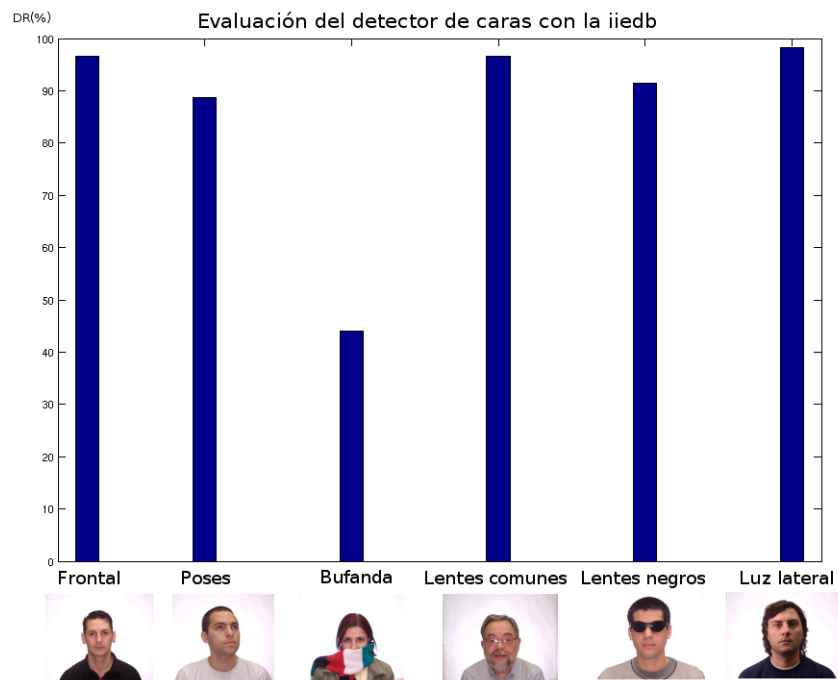


Figura 4.9: Evaluación del detector de caras.

Tolerancia	Frente	Todas
3 píxeles	30 %	19 %
6 píxeles	79 %	67 %
9 píxeles	96 %	88 %
12 píxeles	98 %	96 %

Se observa que el detector tiene una performance del 90% al considerar una tolerancia de 9 píxeles. Cabe destacar que las imágenes utilizadas tienen un tamaño de 640×480 píxeles, donde podríamos estimar el tamaño del ojo en: 18×18 píxeles. Observando la figura 4.11 se puede ver que un error de 3 píxeles es totalmente despreciable, un error de 6 píxeles es prácticamente despreciable y un error de 9 píxeles es pequeño, por lo que consideramos que la performance del detector es buena.

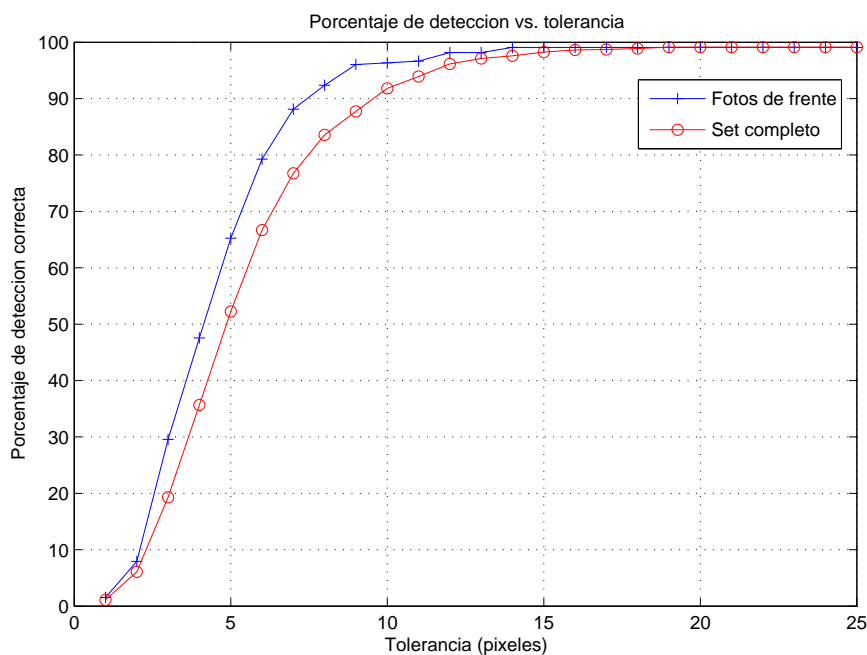


Figura 4.10: Evaluación de la precisión en la localización de los ojos considerando el set completo y solamente las fotos de frente.



Figura 4.11: Ejemplo de la precisión en la localización de los ojos. A la derecha, vemos tres máscaras de diferentes tamaños sobre el ojo izquierdo de un individuo de la base. Las máscaras son de radio: 9 píxeles (superior), 6 píxeles (media) y 3 píxeles (inferior).

4.3.4. Propuestas de mejora

Puesto que se encuentra que la performance del detector es adecuada para esta aplicación, se buscan mejoras al mismo apuntando a la eficiencia. La primera propuesta consiste en aplicar a la imagen, antes de comenzar la búsqueda, un filtro binario que segmente las regiones de color piel. El posterior procesamiento se realiza solamente en las ventanas que contengan más del 50% de los píxeles de dicho color, logrando descartar rápidamente gran parte de la imagen. Esto debería reflejarse en un incremento en la velocidad de detección. Este beneficio se ve opacado en caso de imágenes con una sola cara ocupando la mayor parte de la misma. Es claro que apunta a la detección de caras en el caso general, considerando imágenes de fondo complejo. Por otro lado se encuentra que la eficiente estructura en cascada del clasificador hace que más de la mitad de las ventanas se descarten con simplemente dos *features de Haar*, lo que implica menor costo que el cálculo de los píxeles de color piel de cada ventana. Finalmente se descarta el filtrado de píxeles en tonalidades de piel.

Una segunda propuesta plantea reducir la búsqueda multiescala a un rango de escalas menor, en torno al tamaño esperado de la cara en la imagen. Esto permite exigir que la cara ocupe un cierto tamaño mínimo en la imagen, de forma tal de asegurarse determinada resolución para la misma. Se trabaja a partir de un factor de escala de 8, lo que implica una ventana de tamaño 80×60 para imágenes de 640×480 . Esto repercute en un algoritmo de detección más rápido puesto que tenemos que clasificar menor cantidad de ventanas por imagen. Para las imágenes de la base IIE, de tamaño 640×480 , se reduce el tiempo de búsqueda a la cuarta parte. Cabe aclarar que la modificación introducida no cambia la performance del detector de caras obtenida con la base IIE. La variante refiere a limitar las escalas en las que se busca y se cumple que las escalas a las que se encuentran las caras de la base permanecen dentro del rango de búsqueda establecido.

4.4. Normalización

La etapa de normalización pretende uniformizar las imágenes de entrada al sistema luego de detectada la cara. En este proceso se aplican distintas técnicas que logran, entre otras cosas, reducir el tamaño de las imágenes y mover la posición de los ojos del individuo a coordenadas predeterminadas.

A continuación se describen los pasos seguidos para completar la etapa de normalización.

1. Se implementa una transformación geométrica que ubica las coordenadas de los ojos en un valor predeterminado $\{(52, 64); (76, 64)\}$ y se realiza un rescalado de la imagen para llevarla a un tamaño dado (128×128 píxeles).
2. Se realiza un ajuste de los píxeles de manera de obtener una señal de media nula y desviación estándar 1.
3. Los bordes de la imagen son suavizados en un margen de 30 píxeles. Esto se aplica para evitar posibles efectos de borde introducidos durante la transformación geométrica. El valor de los píxeles es pesado por el factor $\frac{d}{30}$, siendo d la distancia desde el borde. Los píxeles que distan más de 30 píxeles del borde no son modificados.

Opcionalmente, entre los puntos 2 y 3, se normaliza la iluminación de la imagen mediante ecualización de histograma. En las evaluaciones del sistema final se incluyen ensayos con y sin ecualización de histograma. También de forma opcional, se agrega una máscara elíptica que rescata la cara del individuo eliminando el fondo de la imagen. La misma presenta un decaimiento exponencial a la media de la imagen, que la hace suave en la eliminación del fondo. La utilización de la máscara no cambia la performance del algoritmo con la base IIE pero podría solucionar inconvenientes ocasionados por la influencia del fondo de la escena en casos más complejos. La figura 4.12 muestra un ejemplo de fondo complejo en el cual se pueden apreciar los resultados con y sin máscara elíptica. Así como en el caso de la ecualización de histograma, el uso de la máscara fue evaluado en los ensayos del sistema punta a punta en el capítulo 9.



Figura 4.12: A la izquierda la imagen original. En el borde superior derecho se observa la imagen normalizada sin utilizar máscara. En el borde inferior derecho se aprecia el efecto de la máscara elíptica.

Capítulo 5

Descriptores de Caras

5.1. Introducción

Un aspecto clave, en todo sistema de reconocimiento son los descriptores utilizados en la representación del objeto de interés. El descriptor es el nexo entre el objeto real y la representación que el sistema tiene del mismo. Cuanto mejor sea dicha representación, mejor será la performance del sistema a la hora de reconocer objetos. Dependiendo del enfoque elegido, los descriptores o características tomadas pueden ser más adecuados para la diferenciación *intra* o *inter* clase. Es decir, una característica puede ser suficiente para clasificar un objeto como perteneciente a una clase, pero puede no bastar para diferenciar entre elementos de la misma clase. Por ejemplo, en el caso de las caras, la presencia de la nariz puede ser útil para clasificar un objeto como cara pero no es suficiente para diferenciar entre distintos individuos.

La elección de las características se ve sesgada por la aplicación o el problema a resolver. Retomando el ejemplo antes citado, la presencia de nariz puede resolver el problema de detección de caras pero no resuelve el problema de reconocimiento de individuos. Al igual que la clasificación de los métodos de reconocimiento propuesta en el capítulo 3, en este capítulo se presenta una clasificación de descriptores globales o locales. Descriptores globales son aquellos que toman la imagen como un todo y no utilizan información a priori sobre el objeto a representar. Descriptores locales son los que extraen información de puntos o regiones particulares de la imagen, considerando el hecho de que se trabaja con un objeto dado.

En las secciones siguientes se realiza un estudio de distintos ejemplos de descriptores globales y locales aplicados al reconocimiento de caras.

5.2. Características Globales

Se trabaja con tres algoritmos clásicos de extracción y selección de características utilizados para problemas de reconocimiento de patrones: Principal Components Analysis (PCA), Independent Component Analysis (ICA) y Linear Discriminant Analysis (LDA). Se estudia la performance de los mismos al aplicarlos al reconocimiento de caras.

5.2.1. Principal Components Analysis: PCA

PCA [9] es una técnica tradicional de proyección sobre un subespacio para reconocimiento de caras, es probablemente la más utilizada. A continuación se describe las ideas fundamentales. Se tiene un set de imágenes de entrenamiento I . Como primer paso se computa la imagen promedio de I y se resta a cada una de las imágenes de entrenamiento, obteniendo el set de datos $i_1, i_2, \dots, i_n \in I - \bar{I}$. Luego se compone una matriz X tal que cada columna es una imagen de muestra. XX^T es la matriz de covarianza de las muestras de entrenamiento y las componentes principales de la matriz de covarianza se computan resolviendo:

$$R^T(XX^T)R = \Lambda$$

donde Λ es la matriz diagonal de valores propios y R es la matriz de vectores propios ortonormales.

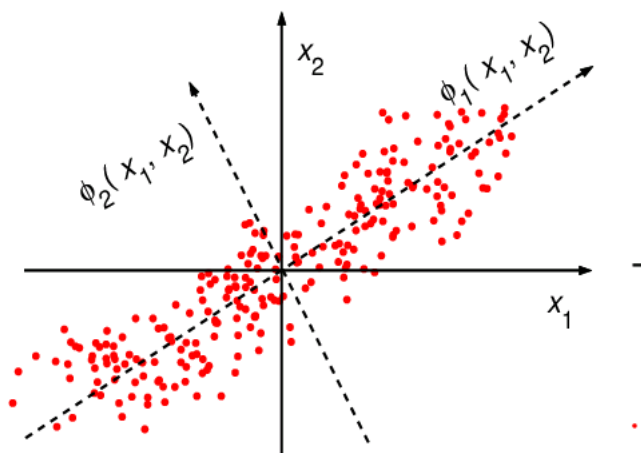


Figura 5.1: Ejemplo 2D de PCA, se observa la base PCA ortogonal en las direcciones de mayor varianza.

Se puede ver geoméricamente que R es una matriz de cambio de base que rota los antiguos ejes a los ejes propios, donde el vector propio con

valor propio más grande se corresponde con el eje de máxima varianza, el asociado con el segundo más grande se corresponde con la segunda mayor varianza y es ortogonal con el anterior y así sucesivamente. En la figura 5.1 se puede observar lo detallado anteriormente, para un ejemplo sencillo de baja dimensión.

Finalmente, se eligen los n vectores propios de mayor valor propio asociado. El parámetro de compresión en este caso es precisamente n dado que indica la dimensión del vector de características que va a representar a la imagen original. Cada coeficiente de la representación de la imagen se obtiene proyectándola sobre cada uno de los vectores de la base PCA.

5.2.2. Independent Components Analysis: ICA

ICA [28] es una herramienta de análisis cuyo objetivo es descomponer una señal observada (imagen de una cara) en una combinación lineal de fuentes independientes. Surge de la técnica conocida por su sigla BSS, o *Blind Source Separation*, que intenta obtener las fuentes independientes a partir de combinaciones de las mismas. Mientras que PCA decorrelaciona las señales de entrada utilizando estadísticos de segundo orden (minimizando el error cuadrático medio de proyección, i.e.: KLT), ICA minimiza mayores órdenes de dependencia.

El número de observaciones $N(1 \leq i \leq N)$ debe ser mayor o igual al número de fuentes originales $M(1 \leq j \leq M)$. En general se utiliza $N = M$. Asumiendo que cada X_i es una combinación desconocida y diferente de los “vectores fuentes” originales, ICA expande cada señal X_i en una suma ponderada de vectores fuente. Se encuentra aquí una fuerte similitud con PCA.

Sea \mathbf{S} la matriz de señales independientes y \mathbf{X} la matriz de observación. Si A es la matriz de combinación desconocida, el modelo de combinación se puede escribir como:

$$\mathbf{X} = A.\mathbf{S}$$

Asumiendo que las señales fuente son independientes unas de las otras y que la matriz A es invertible, el algoritmo ICA encuentra una matriz de separación W , tal que:

$$\mathbf{U} = W.\mathbf{X} = W.A.\mathbf{S}$$

donde \mathbf{U} : estimación de las componentes independientes. La figura 5.2 muestra un esquema de lo anterior.

La esencia de los algoritmos que implementan ICA es la búsqueda de la matriz W según cierto método iterativo de optimización. Para una matriz

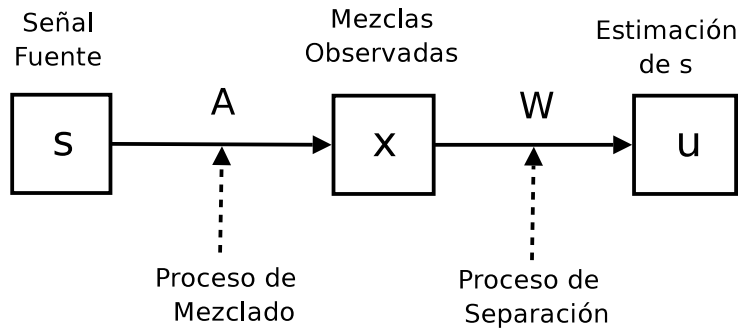


Figura 5.2: Esquema Blind Source Separation

U vista como arreglo de vectores, los vectores son estadísticamente independientes cuando:

$$f_U(U) = \prod_i f_{U_i}(U_i)$$

En la implementación utilizada de ICA se utiliza el algoritmo FAST-ICA [56], probablemente uno de los algoritmos más generales, el cual pretende maximizar:

$$J(y) \simeq C[E\{G(y)\} - E\{G(v)\}]^2$$

donde G es una función no cuadrática, v es la densidad de probabilidad gaussiana y C es una constante mayor a cero. Se puede demostrar que maximizando una función de estas características se obtiene un óptimo en el sentido de independencia buscado. FAST-ICA es un algoritmo ampliamente explorado en esta área. Desde el punto de vista de la performance de los algoritmos que implementan ICA se ha demostrado empíricamente que existen diferencias muy pequeñas y que todos obtienen un máximo muy similar de componentes independientes.

En la literatura de esta temática se argumenta que ICA no tiene ventajas como algoritmo de clasificación *per se* [28, 57]. Se suele utilizar PCA como preprocesador: proyectar las observaciones en un subespacio de dimensión menor. De esta forma se controla el número de componentes o dimensión de la base producida por ICA.

En este trabajo se utiliza una implementación de ICA clásica y una alternativa:

- En la implementación clásica se alimenta el algoritmo FAST-ICA con las primeras m componentes de las imágenes de entrenamiento proyectadas en la base PCA. A la salida, el algoritmo devuelve la matriz de separación W . Para obtener el representante de una imagen de prueba,

primero se proyecta sobre la base PCA y se obtienen los primeros m coeficientes, luego se proyecta sobre la matriz entrenada W . Con este representante de la imagen original es que se alimenta el algoritmo de clasificación.

- En la implementación alternativa se alimenta el algoritmo FAST-ICA directamente con los primeros m vectores de la base PCA. A la salida se recoge la matriz U que está compuesta en sus columnas por las coordenadas independientes de la base PCA. Luego las imágenes de prueba se proyectan sobre la matriz U . El vector de dimensión m resultante es lo que se toma como el representante de las imágenes originales.

5.2.3. Linear Discriminant Analysis: LDA

LDA o *Linear Discriminant Analysis* [29] es una técnica de aprendizaje supervisado para clasificar datos. La idea central de LDA es obtener una proyección de los datos en un espacio de menor (o incluso igual) dimensión que los datos entrantes, con el fin de que la separabilidad de las clases sea la mayor posible. Es una técnica supervisada ya que para poder buscar esa proyección se debe entrenar el sistema con patrones etiquetados. Es importante aclarar que LDA no busca en ningún momento minimizar el error de representación cometido, como sí lo hace PCA. Existen varias implementaciones de LDA, entre ellas se encuentra FISHER-LDA [58]. Para explicarlo se considera la versión más simple del problema:

Encontrar el vector w de proyección, que proyecte los datos a un espacio uni-dimensional de manera de obtener la mayor separabilidad entre sus clases.

Formalizando, tenemos $\{x_1..x_n\}$ patrones d -dimensionales etiquetados en c clases. Cada clase cuenta con N_c patrones. Se busca w , para obtener $y_i = w^T x_i$ proyecciones uni-dimensionales de los patrones. Lo que busca FISHER-LDA es maximizar la siguiente función objetivo:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \quad (5.1)$$

donde S_B es la matriz de dispersión inter-clase y S_W es la matriz de dispersión intra-clase. Más precisamente:

$$S_B = \sum_c N_c (\mu_c - \mu)(\mu_c - \mu)^T \quad S_W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T$$

siendo μ_c la media de cada clase, μ la media de todos los datos, N_c la cantidad de patrones de la clase c . Se puede ver [16] que el w que maximiza

la función objetivo debe cumplir:

$$S_B w = \lambda S_W w$$

Si S_W es no singular es posible resolver el clásico problema de valores propios para la matriz $S_W^{-1} S_B$:

$$S_W^{-1} S_B w = \lambda w$$

Si ahora se sustituye la solución en (5.1) se obtiene lo siguiente:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} = \lambda_k \frac{w_k^T S_B w_k}{w_k^T S_W w_k} = \lambda_k \text{ con } k = 1..d$$

siendo w_k vector propio k de valor propio λ_k . En consecuencia, para maximizar la solución se debe considerar el vector propio de $S_W^{-1} S_B w$ con mayor valor propio asociado. Claro está que este desarrollo tiene sentido para el caso en que se quiere proyectar los datos sobre un espacio uni-dimensional. Se puede ver [16] que para el caso m -dimensional, se debe resolver el mismo problema y elegir los m vectores propios con valores propios asociados más grandes.

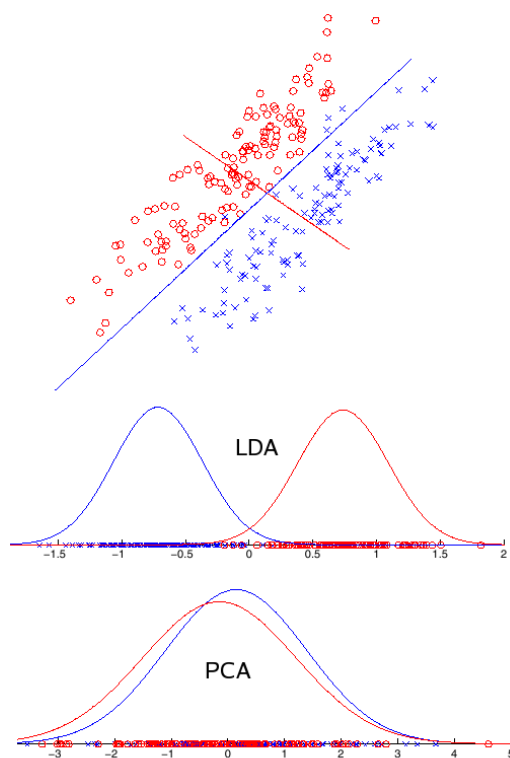


Figura 5.3: Ejemplo 2D de LDA y PCA, se observa la proyección de los datos sobre el vector de mayor valor propio asociado para el caso de PCA y el caso de LDA.

En la figura 5.3 se observa para un ejemplo particular, las diferencias en las proyecciones de los datos sobre un espacio unidimensional considerando la base PCA y LDA obtenidas. Se destaca la importancia de considerar un método como el de LDA si el interés radica en la diferenciación de las clases.

En este caso particular donde se trabajó con imágenes (datos de alta dimensión) se aplicó, como en el caso de ICA, una primera etapa de PCA para reducir la dimensionalidad de los datos. Los datos son reducidos a dimensión 100.

5.2.4. Ensayos y Resultados

En la evaluación de los algoritmos propuestos se utiliza la base IIE. Las imágenes consideradas son preprocesadas según la normalización descrita en el capítulo 4. Se aplican los mencionados como opcionales, ecualización de histograma y máscara elíptica. Para la clasificación se utiliza el algoritmo de los k vecinos más cercanos (K-NN), tomando como distancia la norma euclideana.

Se define tres clases de ensayos que se listan a continuación. Se detalla en cada caso las tomas utilizadas de la base IIE especificadas en el apéndice C.

- **Test I:** Entrenamiento: 4 imágenes de frente (tomas a, d, j, g). Evaluación: 2 imágenes de frente tomadas una semana después (tomas u, x).
- **Test II:** Entrenamiento: 2 imágenes de frente (tomas a, d). Evaluación: 2 imágenes de frente tomadas en el mismo momento (tomas j, g).
- **Test III:** Entrenamiento: 4 imágenes de frente (tomas a, d, j, g). Evaluación: 8 imágenes de varias expresiones (tomas m, n, o, p), partes ocluídas (tomas q, r, s) o problemas de iluminación (toma t).

Luego de proyectar los datos en los diferentes subespacios, se aplica K-NN con métrica euclideana. Se estudia la performance del mismo al aplicarlo con parámetro $k = 1, 3, 5$. A continuación se muestra algunos de los resultados obtenidos.

Test I $k = 1$	Dimensión				max	
	10	30	50	70	RR _{max}	dim
PCA	71.43 %	86.73 %	88.78 %	88.78 %	88.78 %	40
ICA	69.39 %	86.73 %	91.84 %	91.84 %	91.84 %	50
LDA	86.73 %	95.92 %	95.92 %	92.86 %	96.94 %	39

Test I $k = 3$	Dimensión				max	
	10	30	50	70	RR _{max}	dim
PCA	70.40 %	84.69 %	86.73 %	86.73 %	87.75 %	42
ICA	68.36 %	85.71 %	90.81 %	91.83 %	91.83 %	70
LDA	85.71 %	95.91 %	95.91 %	92.85 %	97.95 %	56

Las tablas anteriores muestran la performance (RR) de los algoritmos frente al Test I al variar la dimensión del subespacio de proyección. La primera tabla clasifica utilizando el algoritmo K-NN con $k = 1$ (vecino más cercano). La segunda utiliza el mismo algoritmo pero con $k = 3$. Ambos casos dan resultados muy similares. Se observa que a partir de 40 coeficientes los resultados prácticamente no varían. En ambos casos el algoritmo que mejor se desempeña es LDA logrando un altísimo porcentaje de reconocimiento. En la figura 5.4 se muestra la evolución de los algoritmos al considerar mayor número de componentes.

La tabla siguiente muestra resultados obtenidos para el Test II. Se observa mejor performance que en el caso anterior. En este Test, todos los algoritmos tienen un muy buen desempeño. Cabe acotar que las imágenes utilizadas para el entrenamiento y para la validación fueron adquiridas en la misma sesión fotográfica.

Test II $k = 1$	Dimensión				max	
	10	30	50	70	RR _{max}	dim
PCA	86.73 %	94.89 %	95.91 %	95.91 %	96.93 %	33
ICA	88.77 %	97.95 %	97.95 %	94.89 %	97.95 %	29
LDA	78.57 %	94.89 %	97.95 %	93.87 %	98.97 %	41

Según la literatura existen variantes de los métodos explicados anteriormente. Por ejemplo, para el método PCA, hay autores que sostienen que las primeras componentes no sirven para diferenciar las imágenes de la base. Durante este estudio se implementa la clasificación sin tomar en cuenta los primeros dos componentes (PCA2). Respecto a ICA, se obtienen los resultados mediante el método original y su variante (ICA2). En todos estos casos el algoritmo de clasificación utilizado es 1-NN. Para el algoritmo LDA se muestra los resultados de aplicar la clasificación K-NN, con $k = 1, 3, 5$. En la figura 5.5 se observa algunos resultados.

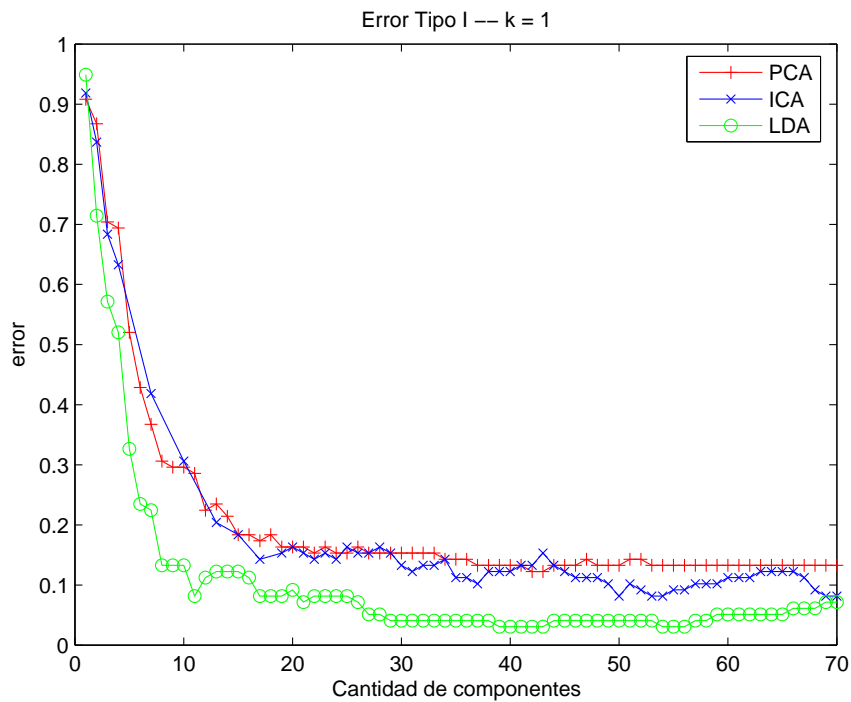


Figura 5.4: Evolución del error cometido al clasificar el Test I en función de la cantidad de componentes tomados. Clasificación mediante 1-NN.

Finalmente se aplican los algoritmos al Test III - el más difícil - donde los resultados no son alentadores. La codificación del tipo de foto en una letra es la descrita en el apéndice C. En este caso se evalúa con las tomas: sonriente (m), sorprendido (n), enojado (o), guiñada (p), con bufanda (q), con lentes normales (r), lentes negros (s) e iluminación no uniforme (t). Seguidamente se presenta un resumen de lo obtenido frente al Test III. Los resultados mostrados corresponden a representaciones de dimensión 50.

Test III Método	Tipo de Foto			
	m	n	o	p
PCA	81.6 %	67.3 %	71.4 %	79.5 %
PCA2	77.5 %	61.2 %	65.3 %	69.3 %
ICA	73.4 %	73.4 %	73.4 %	73.4 %
ICA2	81.6 %	67.3 %	71.4 %	79.5 %
LDA	87.7 %	81.6 %	83.6 %	83.6 %

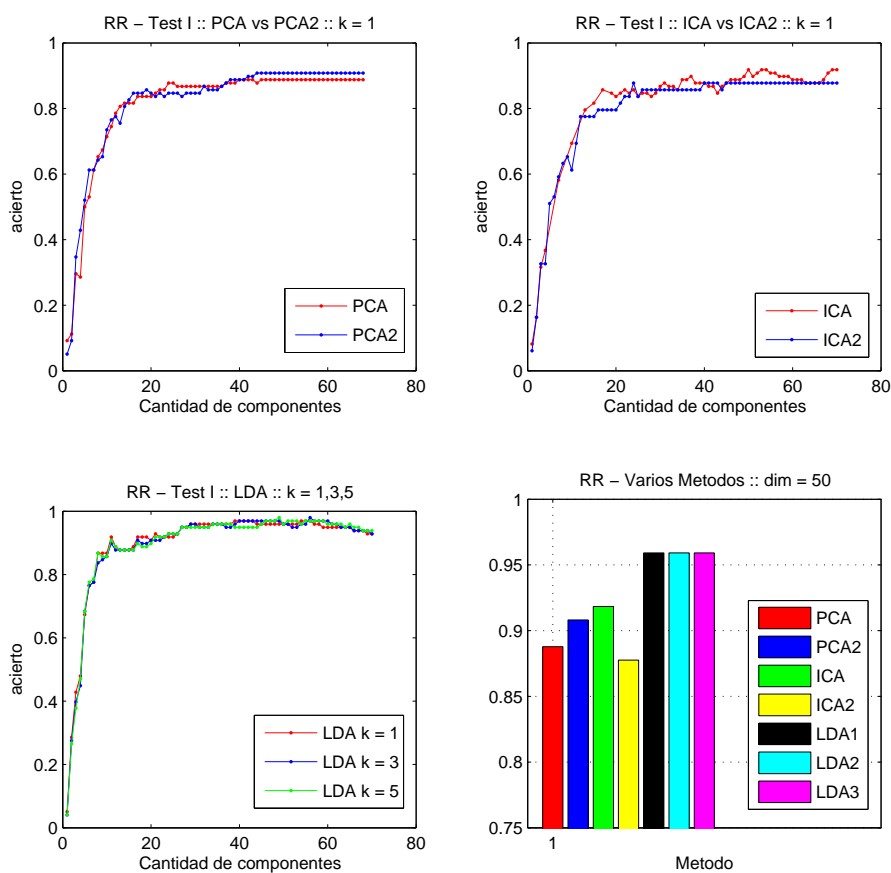


Figura 5.5: Resultados varios para el Test I. PCA2: PCA quitando los dos primeros componentes para la clasificación. ICA2: Arquitectura ICA entrenada directamente mediante la base PCA. LDA para $k = 1, 3, 5$. En los tests ICA y PCA se utilizó 1-NN para clasificar.

Test III Método	Tipo de Foto			
	q	r	s	t
PCA	28.5 %	59.1 %	18.3 %	20.4 %
PCA2	28.5 %	61.2 %	12.2 %	20.4 %
ICA	20.4 %	57.1 %	24.4 %	2.0 %
ICA2	28.5 %	61.2 %	18.3 %	20.4 %
LDA	65.3 %	81.6 %	32.6 %	26.5 %

Culminando, en la figura C se muestra un ejemplo de bases de caras utilizadas para proyectar las imágenes originales en espacios de menor dimensión.

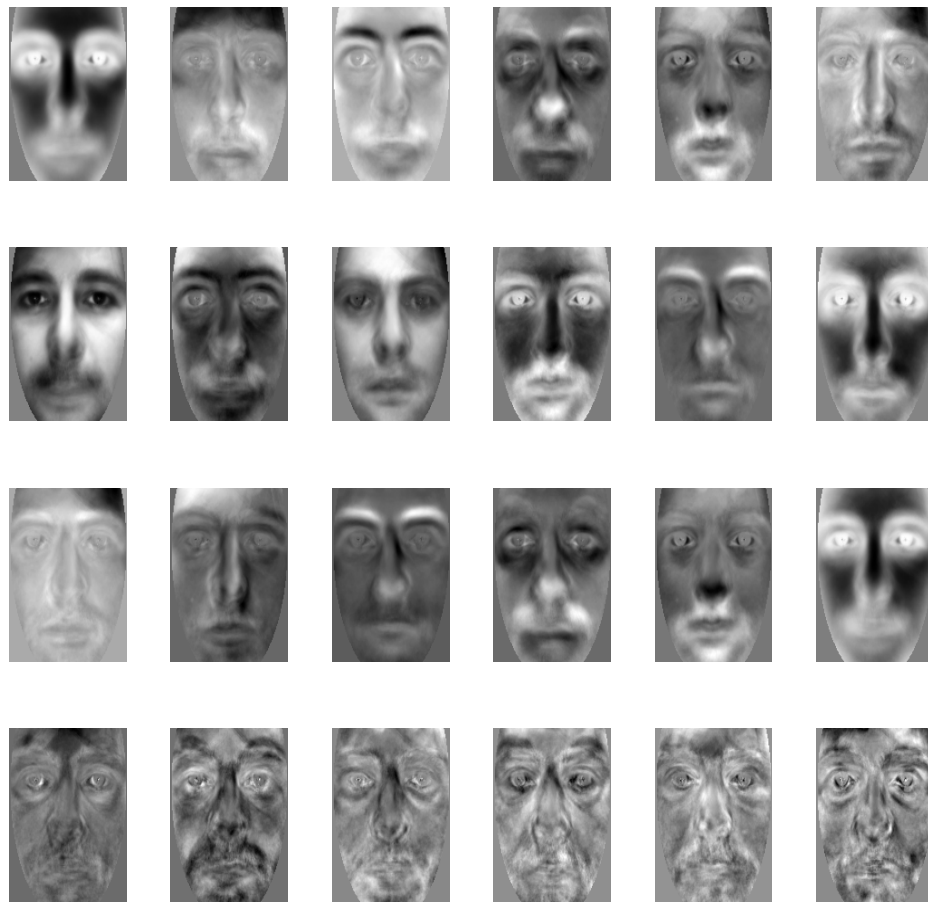


Figura 5.6: Imágenes generadoras de los diferentes subespacios de dimensión 6. En orden descendente: base PCA, ICA, ICA2, LDA.

5.2.5. Conclusiones

Se estudia la performance de tres algoritmos bien conocidos en el mundo del Reconocimiento de Caras. Se encuentra que todos ellos son fuertemente sensibles a las imágenes con las que son entrenados. Su performance baja significativamente frente a oclusiones y a problemas de iluminación.

De manera global se puede concluir que el algoritmo que obtiene mejores resultados al clasificar es LDA, el único de los cuales se entrena de forma supervisada. En el caso de ICA se concluye que mejora visiblemente los resultados frente a PCA, aún estando lejos de la performance de LDA. Se observa que PCA es un buen reductor de dimensión (útil como compresor), que por sí solo no logra obtener buenos resultados de clasificación. LDA logra obtener

un buen RR frente a variaciones de expresiones, como pueden ser una sonrisa o un enojo. Frente a oclusiones se encuentra que ninguno de los algoritmos logra una performance aceptable. Esto es debido al carácter holístico de los mismos.

Respecto a las variantes de los algoritmos utilizados, se concluye que ICA2 no logra obtener una performance tan buena como la de ICA - clásico - obteniendo valores similares a los de PCA. En cuanto a PCA2 se obtiene una mejora sensible frente a PCA en la mayoría de los casos. Se observa que la performance de los algoritmos no tiene un cambio significativo al variar el parámetro k del clasificador por vecino más cercano.

5.3. Características Locales

Dado que existen una gran cantidad de descriptores locales, se concentró esfuerzos en estudiar uno en particular, los wavelets de Gabor. En esta sección se estudia el desempeño de los mismos como descriptores locales, en particular su potencial discriminador en el reconocimiento de caras.

Como fuese expuesto anteriormente, la representación de la cara enfocada a características locales, apunta a utilizar explícitamente el hecho de estar trabajando con una cara. Es decir, las características no se extraen a partir de la imagen como un todo sino en puntos particulares. En este caso se eligen puntos característicos que representan un rostro, por ejemplo las pupilas y la punta de la nariz.

Los puntos principales elegidos son marcados de forma manual sobre las imágenes utilizadas en la evaluación, la figura 5.10 muestra un ejemplo de esto. Luego se procede a la extracción de los wavelets de Gabor en dichos puntos. Cada cara es representada por los wavelets resultantes en sus puntos característicos. Las medidas de similitud utilizadas son detalladas en la sección 5.3.2. Finalmente, para la clasificación se utilizan técnicas de K-NN para $k=1$ y 3.

5.3.1. Wavelets de Gabor

Introducción

Una de las posibilidades para analizar el espectro de una imagen son los wavelets de Gabor. Al igual que el análisis tradicional de Fourier, permite conocer el contenido espectral de una imagen. Sin embargo, a diferencia de este último, permite el estudio localizado del espectro, centrandolo en una región particular. Los wavelets de Gabor se basan en una familia de filtros que permiten ser sintonizados en determinada orientación, en determinada frecuencia y además pueden limitar su rango de acción a determinada

región de la imagen.

Los wavelets de Gabor están formados por un banco de filtros lineales e invariante temporales, por lo que se los caracteriza por su respuesta al impulso. La misma se basa en un núcleo gaussiano modulado mediante una senoide:

$$W(x, y, \theta, f, \phi, \sigma, \gamma) = e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} \cos(2\pi f x' + \phi) \quad (5.2a)$$

con

$$\begin{cases} x' = x \cos(\theta) + y \sin(\theta) \\ y' = -x \sin(\theta) + y \cos(\theta) \end{cases} \quad (5.2b)$$

A la familia de filtros según (5.2a) la denominamos banco de filtros de gabor *base*. En la expresión anterior, se reconocen algunos parámetros de esta clase de filtros:

Frecuencia : Es el inverso de la distancia en píxeles entre máximos locales contiguos. En la ecuación (5.2a) es representado por f , frecuencia de la senoide. También se puede caracterizar por su período λ , siendo este $\lambda = \frac{1}{f}$.

Selectividad espacial : Representa qué tan local es el filtro en el dominio espacial. En (5.2a) es representado por el radio de la gaussiana σ . Para valores mayores de σ menor es la selectividad espacial. Con el objetivo de que todos los wavelets sean escalas y/o rotaciones de un par de filtros *madre*, se deberá cumplir que $\sigma = k\lambda$.

Selectividad frecuencial : Representa qué tan local es el filtro en el dominio de la frecuencia. En (5.2a) es representado indirectamente por $\frac{1}{\sigma}$. Para valores mayores de σ mayor es la selectividad frecuencial. Observando el punto anterior se aprecia que cuanto mayor es la selectividad espacial menor es la selectividad frecuencial.

Relación de Aspecto : La relación de aspecto de la gaussiana es representada en (5.2a) por γ .

Orientación : Es el ángulo que forma el centro del filtro en el dominio frecuencial con el eje horizontal. En (5.2b) es representado por θ .

Fase : Es la fase de la senoide que modula la gaussiana. En (5.2a) es representada por ϕ . Por lo general se utiliza dos fases que distan $\pi/2$ entre sí. La dupla resultante de aplicar esta pareja de filtros puede ser vista como un número complejo.

En la figura 5.7 se muestra 3 ejemplos para entender la influencia de los distintos parámetros. En todos los casos se toma $\sigma = \frac{1}{f}$.

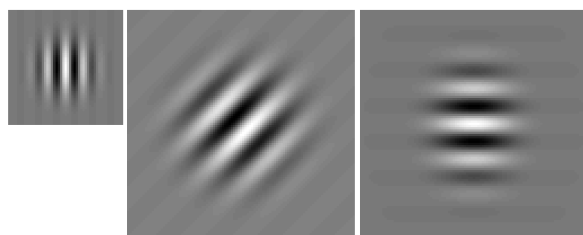


Figura 5.7: Ejemplos de wavelets de Gabor

- *Ejemplo 1* : $f = \frac{1}{8}, \theta = 0^\circ, \phi = 0$ y $\gamma^2 = 1$
- *Ejemplo 2* : $f = \frac{1}{16}, \theta = 45^\circ, \phi = \frac{\pi}{2}$ y $\gamma^2 = 1$
- *Ejemplo 3* : $f = \frac{1}{16}, \theta = 90^\circ, \phi = 0$ y $\gamma^2 = 2$

Un parámetro adicional que es necesario definir para poder implementar correctamente los filtros anteriores, es el tamaño de máscara del filtro. El mismo va a estar estrictamente relacionado con el soporte del filtro. Para el caso de un filtro de Gabor de relación de aspecto unitaria, se sabe que en un círculo de radio 2σ con centro en el origen, está concentrado el 95 % de la potencia total del filtro. A su vez, si se aumenta el radio a 3σ la potencia encerrada en dicha región aumenta al 99 %. Por otro lado, la transformada de Fourier de un filtro de Gabor son dos campanas gaussianas centradas a más/menos la frecuencia del mismo. En la figura 5.8 se observa la distorsión introducida al considerar una máscara de tamaño finito. La distorsión introducida en el espectro, se debe al efecto de enventanar la respuesta al impulso del filtro. Se observa que para máscaras de radio 3σ dicho efecto es casi despreciable. A lo largo del proyecto se adopta como criterio para definir el tamaño de la máscara que la misma logre cubrir el 99 % de la potencia del filtro. En el caso de filtros de relación de aspecto unitaria esto corresponde a tomar una máscara de radio 3σ . Dado que las máscaras que se adoptan son cuadradas se consideran máscaras de lado 6σ para cumplir las especificaciones planteadas.

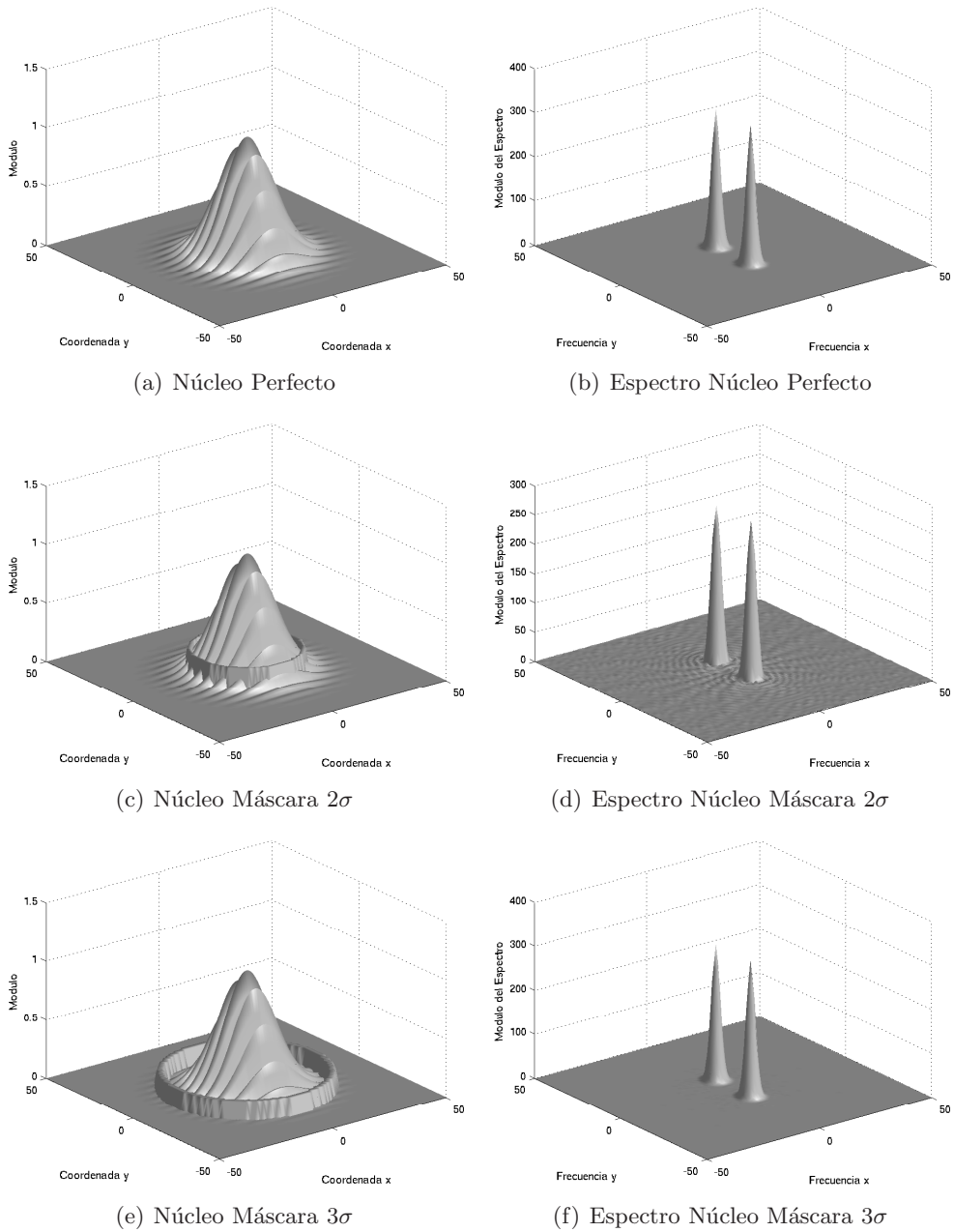


Figura 5.8: Diferentes tamaños de máscaras: ideal, radio 2σ , radio 3σ y su efecto en el espectro

Como se dijo anteriormente, el banco consta de un conjunto de filtros sintonizados mediante distintos parámetros. A modo de ejemplo se presenta un juego de parámetros desarrollados en [10]. El mismo consta de: 5 frecuencias, 8 orientaciones, 2 fases, siendo en total 80 filtros de Gabor:

Wavelets Wiskott

Parámetro	Sim	Valores
Orientación	θ	$\{0, \pi/8, 2\pi/8, 3\pi/8, 4\pi/8, 5\pi/8, 6\pi/8, 7\pi/8\}$
Período	λ	$\{4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\}$
Fase	ϕ	$\{0, \pi/2\}$
Radio	σ	$\sigma = \lambda$
R. Aspecto	γ	1

Además de los parámetros que aparecen en 5.2a y el tamaño de la máscara, si se quiere caracterizar completamente el funcionamiento del banco de filtros, es necesario especificar un parámetro adicional. Al mismo se le llama resolución y se define de la siguiente manera:

Resolución : Representa la definición del objeto de interés. Dado que en realidad no importa el tamaño absoluto de la imagen en píxeles, sino el tamaño de la porción de la imagen que contiene a la cara, se mide como la distancia en píxeles entre ambos ojos. Se representa con la letra d .

La elección del rango de frecuencias del banco de filtros va a estar condicionada a la resolución de la cara en la imagen. Por ejemplo, en [11] se plantea tomar imágenes de tamaño 128×128 , normalizando la cara de manera de obtener una distancia entre los ojos $d = 24$ píxeles. Si consideramos el banco de frecuencias de Wiskott, observamos lo siguiente:

$d = 24$ píxeles.	
$\lambda = \frac{1}{f}$	$\frac{d}{\lambda}$
4	6
$4\sqrt{2}$	4.24
8	3
$8\sqrt{2}$	2.12
16	1.5

La segunda columna de la tabla anterior muestra la cantidad de períodos que hay en una distancia igual a la distancia entre los ojos. Se observa que si se duplica la resolución de la imagen ($d = 48$), es decir se duplica la distancia en píxeles entre los ojos, la cantidad de períodos que entran entre los

ojos para cada frecuencia se verá duplicada, por lo que la respuesta a cada uno de los filtros será diferente. Intuitivamente si se duplica la resolución de la imagen deberemos de bajar una octava (dividir entre dos) el rango de frecuencias de los filtros con el que se está trabajando para así obtener respuestas similares al caso original. Este último razonamiento implica que si se tiene conocimiento del rango de frecuencias óptimo de un banco de filtros (por ejemplo el rango que maximiza determinado indicador de desempeño) y el tamaño de imagen utilizado, para la correcta implementación del mismo es necesario conocer también la resolución a la cual se encuentra el objeto a reconocer.

A modo de resumen, se observa que a los parámetros previstos para el banco de filtros se deberá adicionar la resolución del objeto a reconocer como un nuevo parámetro. Los juegos de parámetros que se utilizan a lo largo de este proyecto, incluyendo los antes definidos **Wavelets Wiskott**, son los siguientes:

Wavelets Bolme

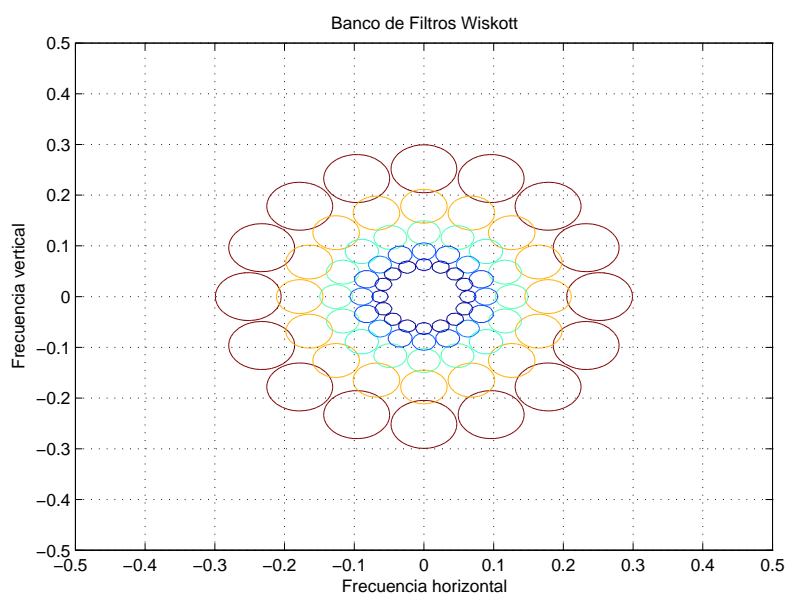
Parámetro	Sim	Valores
Orientación	θ	$\{0, \pi/8, 2\pi/8, 3\pi/8, 4\pi/8, 5\pi/8, 6\pi/8, 7\pi/8\}$
Período	λ	$\{4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\}$
Fase	ϕ	$\{-\pi/4, \pi/4\}$
Radio	σ	$\sigma = 3\lambda/4$
R. Aspecto	γ	1

Wavelets Nestares

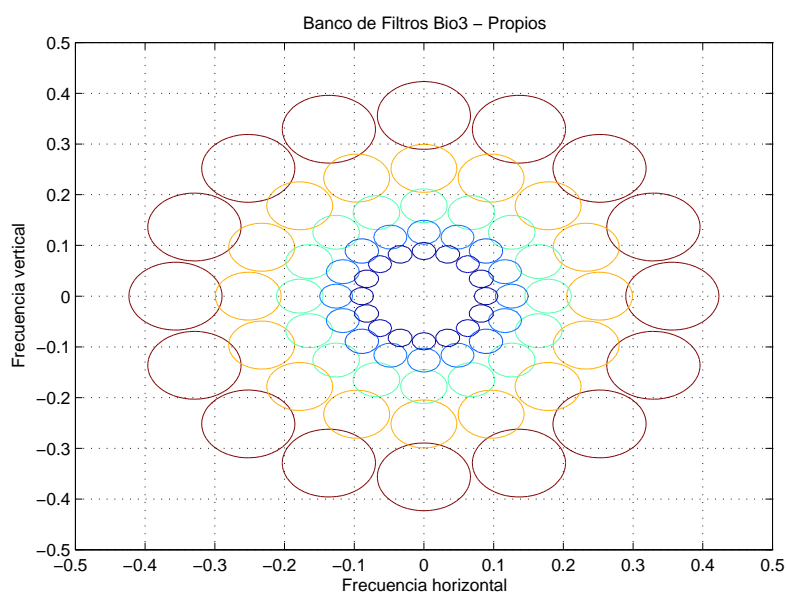
Parámetro	Sim	Valores
Orientación	θ	$\{0, 2\pi/8, 4\pi/8, 6\pi/8\}$
Período	λ	$\{4, 8, 16, 32\}$
Fase	ϕ	$\{0, \pi/2\}$
Radio	σ	$\sigma = \lambda$
R. Aspecto	γ	1

Wavelets Bio3 - Propios

Parámetro	Sim	Valores
Orientación	θ	$\{0, \pi/8, 2\pi/8, 3\pi/8, 4\pi/8, 5\pi/8, 6\pi/8, 7\pi/8\}$
Período	λ	$\{2\sqrt{2}, 4, 4\sqrt{2}, 8, 8\sqrt{2}\}$
Fase	ϕ	$\{0, \pi/2\}$
Radio	σ	$\sigma = 0,56\lambda$
R. Aspecto	γ	1



(a) Banco de filtros Wiskott



(b) Banco de filtros Bio3 - Propios

Figura 5.9: Cubrimiento del espectro para dos de los bancos de filtros utilizados.

En la figura 5.9 se puede observar los diferentes cubrimientos que se logra en el espacio de frecuencias según la configuración utilizada. Se observa que el banco de filtros propuesto por nosotros cubre de mejor manera las altas frecuencias, a costa de no cubrir tan bien las frecuencias más bajas.

5.3.2. Medidas de Similitud

Como medidas de similitud se adoptan las definidas en el algoritmo *Elastic Bunch Graph Matching* [44], algoritmo de reconocimiento de caras que se basa fuertemente en el uso de wavelets de Gabor. Una breve descripción del mismo se puede encontrar en el capítulo 3. En dicho algoritmo, los wavelets extraídos de un punto característico dado se denominan *jet*. Por coherencia con dicha nomenclatura se utiliza aquí este término. Se trabaja entonces con las siguientes estructuras:

Jet: $J = \{j^1, \dots, j^N\}$, siendo j^i números complejos y N la cantidad de coeficientes complejos resultantes de la extracción de los wavelets de Gabor. N queda determinado por la configuración del banco de filtros.

Patrón: $P = \{J_{p1}, \dots, J_{pM}\} = \{j_{p1}^1, \dots, j_{p1}^N; \dots; \dots; j_{pM}^1, \dots, j_{pM}^N\}$, siendo M la cantidad de puntos principales marcados en la cara. Cada patrón queda constituido por los M *jets* extraídos en la cara.

Se define varias medidas de similitud de *jets* considerando distintos aspectos de los wavelets de Gabor. Las mismas son:

- Similitud de fase S_ϕ
- Similitud de magnitud S_a
- Similitud combinada, de fase y magnitud S_D

A continuación se detalla cada una de éstas.

Similitud de Fase

Se calcula según la expresión:

$$S_\phi(J, J') = \frac{\sum_{j=1}^N a_j a'_j \cos(\phi_j - \phi'_j)}{\sqrt{\sum_{j=1}^N a_j^2 \sum_{j=1}^N a'_j^2}}$$

donde J y J' son los *jets* a comparar, a , a' , ϕ y ϕ' corresponden a módulo y fase de los números complejos de dichos *jets*, siendo N la cantidad de números complejos en cada *jet*.

Esta medida se basa en la magnitud de la respuesta al banco de filtros, siendo estos valores pesados por la similitud en la fase. Esto implica que se alcancen altos *scores* únicamente en el caso en que ambos, magnitud y fase, sean similares.

Similitud de Magnitud

Representada por la expresión,

$$S_a(J, J') = \frac{\sum_{j=1}^N a_j a'_j}{\sqrt{\sum_{j=1}^N a_j^2 \sum_{j=1}^N a'_j{}^2}}$$

esta medida considera únicamente la magnitud de la respuesta al banco de filtros. La información de fase no es utilizada.

La medida S_ϕ responde correctamente a la información de fase de la imagen. Dado que la información de fase cambia rápidamente con el desplazamiento, la medida S_ϕ tiene un valor bajo si los *jets* comparados provienen del mismo punto principal pero están desplazados una pequeña distancia. En dicho caso, la medida S_a tendrá un valor alto por no considerar la información de fase, pero producirá, por la misma razón, falsos positivos en otras oportunidades.

Similitud Combinada

La medida S_D pretende corregir errores asociados a pequeños desplazamientos que surgen en la medida de similitud de fase.

$$S_D(J, J', \vec{d}) = \frac{\sum_{j=0}^N a_j a'_j \cos(\phi_j - (\phi'_j + \vec{d} \cdot \vec{k}_j))}{\sqrt{\sum_{j=1}^N a_j^2 \sum_{j=1}^N a'_j{}^2}}$$

S_D estima la similitud entre J y J' como si el segundo fuese extraído a una distancia \vec{d} de su posición actual. Aplica una corrección en el valor de J' dada por el desplazamiento, no teniendo así que recalcular el *jet* en la nueva ubicación. Esta corrección se basa en dos hipótesis:

- El módulo de los coeficientes cambia muy poco con pequeños desplazamientos.
- La fase de los coeficientes cambia linealmente en la dirección de la senoide para pequeños desplazamientos.

Con esto se aproxima el valor del *jet* en el nuevo punto,

$$a' \approx a$$

$$\phi' \approx \phi + \vec{k} \cdot \vec{d}$$

siendo \vec{k} un vector según la dirección de la senoide.

El vector \vec{d} es desconocido y se estima de forma de maximizar S_D . De esta manera se evita dar un *score* bajo en casos en que los *jets* corresponden al mismo punto característico y a causa de un pequeño desplazamiento de la posición real el valor de S_ϕ es bajo.

Dentro de los métodos de estimación del vector de desplazamiento encontramos técnicas exhaustivas, que buscan el máximo de S_D en cada píxel de la imagen; técnicas que buscan maximizar S_D a través de aproximaciones de la misma, etc. En el capítulo 6 del presente documento se profundiza al respecto de los métodos implementados para la estimación del vector de desplazamiento.

Distancia entre patrones

Luego de definida la similitud de *jets*, se define la similitud de patrones como el valor medio de la similitud de todos los *jets* que los definen.

$$L(P, P') = \frac{1}{M} \sum_{i=1}^M S_x(J_i, J'_i)$$

donde M es la cantidad de puntos principales considerados, S_x es un método específico de medida de similitud de *jets*, J_i y J'_i son *jets* del i -ésimo punto principal, P y P' son los patrones a comparar.

Dado que la clasificación se realizará con técnicas de vecino más cercano se necesita una medida de distancia. Como medida de distancia entre patrones se considera el opuesto de la similitud de los mismos.

5.3.3. Ensayos y Resultados

En la etapa de evaluación se utiliza la base IIE. Las imágenes consideradas son preprocesadas según la normalización descrita en el capítulo anterior. No se aplica ecualización de histograma y máscara elíptica.

Para estos ensayos se toma un tamaño de imagen de 160×250 píxeles, por resultar de mayor comodidad en el marcado manual de puntos. Los ojos se llevan a una distancia $d = 80$ píxeles entre sí. Se consideran dos configuraciones diferentes para el banco de filtros utilizado en la extracción de características. Éstas son las antes descritas *Set de Wiskott* y *Set de Nestares*. Notar que en este caso $d = 80$, por lo que las características extraídas con dichos bancos de filtros no son las mismas que en [11] (donde $d = 24$), puesto que se trabaja a distinta resolución¹.

¹Ver definición de resolución en 5.3.1.

Para los ensayos se toma como set de entrenamiento las cuatro imágenes frontales de la primera toma (tomas a, d, j, g) y como set de evaluación dos imágenes de frente tomadas una semana después (tomas u, x) y cuatro imágenes de varias poses y problemas de iluminación. Estas últimas corresponden a las tomas de: perfil derecho (c), mirando arriba (k), mirando abajo (l) e iluminación no uniforme (t). Se marcan manualmente 15 puntos en cada una de las imágenes consideradas según la figura 5.10. La clasificación se realiza mediante K-NN con $k = 1$ y $k = 3$.

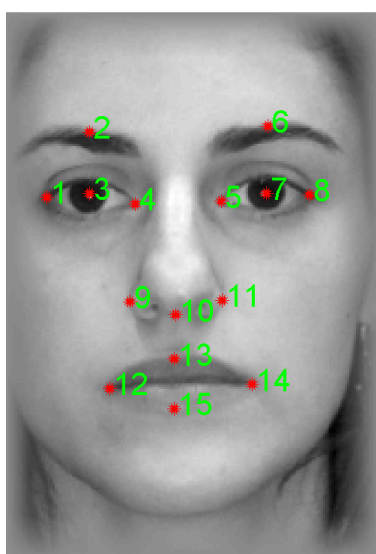


Figura 5.10: Ejemplo de marcado manual de puntos.

A los efectos de la evaluación se define dos ensayos:

- **Test I:** Conjuntos de imágenes antes definidos. 15 puntos marcados en cada imagen según figura 5.10. Filtros según *Set de Wiskott*.
- **Test II:** Conjuntos de imágenes antes definidos. 15 puntos marcados en cada imagen según figura 5.10. Filtros según *Set de Nestares*.

Las tablas siguientes despliegan los resultados de RR obtenidos para el Test I.

Test I	Tipo de Foto				
	u, x	c	k	l	t
$k = 1$					
Magnitud	98.8 %	91.3 %	93.5 %	65.2 %	76.1 %
Fase	100 %	91.3 %	89.1 %	72 %	80.4 %
Combinada	100 %	97.8 %	95.6 %	76.1 %	93.5 %

Test I $k = 3$	Tipo de Foto				
	u, x	c	k	l	t
Magnitud	98.8 %	91.3 %	93.5 %	67.4 %	76.1 %
Fase	100 %	91.3 %	91.3 %	73.9 %	82.6 %
Combinada	100 %	93.5 %	95.7 %	73.9 %	89. %

Se observa una buena performance de los descriptores, principalmente para las tomas frontales y las similitudes de fase y combinada. No aparece una diferencia significativa entre los resultados obtenidos para la clasificación 1-NN y 3-NN.

Seguidamente se aprecian los resultados de RR obtenidos en el Test II.

Test II $k = 1$	Tipo de Foto				
	u, x	c	k	l	t
Magnitud	96.5 %	73.9 %	97.8 %	50.0 %	26.1 %
Fase	97.7 %	91.3 %	97.8 %	78.3 %	60.9 %
Combinada	97.7 %	91.3 %	97.8 %	76.1 %	47.8 %

Test II $k = 3$	Tipo de Foto				
	u, x	c	k	l	t
Magnitud	96.5 %	76.1 %	95.6 %	47.8 %	28.3 %
Fase	98.8 %	89.1 %	97.8 %	78.3 %	56.5 %
Combinada	97.7 %	91.3 %	97.8 %	69.5 %	50.0 %

Se observa que la disminución en la dimensión de los patrones, dada por el cambio de configuración del *Set de Wiskott* al *Set de Nestares*, trae consigo un decaimiento en la performance de los descriptores.

5.3.4. Conclusiones

En primer lugar cabe señalar la excelente performance de los descriptores en la tarea de discriminar individuos a través de la cara. Los valores de RR obtenidos son realmente prometedores.

El *Set de Wiskott* genera patrones de dimensión 1200, mientras que el *Set de Nestares* utiliza 480 coeficientes. Observando las tablas de resultados se ve que este incremento de información implica distintas mejoras, dependiendo de la toma y medida consideradas. El caso más significativo aparece con la relación dimensionalidad - iluminación. Se verifica un importante incremento en los valores de RR, para las imágenes de mala iluminación, al aumentar la dimensión.

Se comparan los resultados obtenidos con los distintos métodos de clasificación, 1-NN y 3-NN, no observándose grandes diferencias. Con respecto a las distancias utilizadas, como era de esperarse los mejores resultados se dan para las distancias que consideran fase y combinación de fase y magnitud con estimación de desplazamiento. Estas distancias son las más complejas y las que contemplan más información. Cabe aclarar que la distancia que considera únicamente el módulo tiene la ventaja de requerir menos cálculo, por lo que es preferible en cuanto a tiempos. De todas maneras el incremento de tiempo que implican las distancias de fase y combinada no opaca su beneficio en performance.

5.4. Elección de Descriptores para el Sistema

Se decide en esta etapa, en base a los resultados y conclusiones expuestos a lo largo de este capítulo, utilizar descriptores locales en el sistema de reconocimiento de caras. Es importante aclarar que a lo largo de este capítulo solamente se estudian algunas de las técnicas existentes para el modelado de las caras. En realidad, existen una innumerable cantidad de descriptores locales y globales, de características variadas, que no se atacan en este proyecto.

En resumen, con los descriptores locales:

- Se obtienen los mejores porcentajes de identificación.
- Se observa mayor robustez frente a cambios, ya sea en la pose o la iluminación.
- El modelo utilizado para describir la imagen considera el hecho de estar trabajando con una cara. En particular, los puntos principales son impuestos como información *a priori*, tomando en general puntos relevantes de la cara como pueden ser bordes, esquinas de los ojos o la nariz.
- La performance es independiente del conjunto de entrenamiento.

Un aspecto importante a tener en cuenta es el hecho de que, al trabajar con descriptores locales, es necesario resolver la localización automática de los puntos principales en la cara. Esto se ilustra en la figura 5.11. El siguiente capítulo ataca este problema.

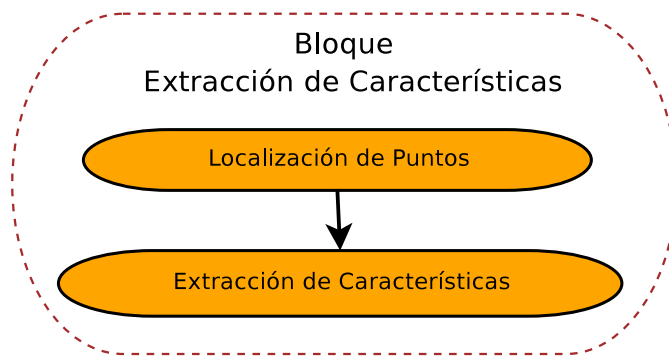


Figura 5.11: Diagrama del bloque de extracción de características.

5.5. Estudio del Banco de Filtros de Gabor

Como se introdujo anteriormente un banco de filtros de Gabor es un conjunto de filtros sintonizados en determinada orientación y en determinada frecuencia, que permite saber el contenido espectral en un entorno local. Se regulan mediante las expresiones (5.2a) y (5.2b). A la familia de filtros según (5.2a) la denominamos banco de filtros de gabor *base*. En la siguiente sección se estudian diferentes alternativas a dicha expresión para compensar particularidades presentes en las imágenes de caras.

5.5.1. Análisis de la Expresión base

Valor medio no nulo

En la expresión 5.2a se aprecia que los diferentes filtros no necesariamente garantizan un valor medio nulo². En particular, si se considera un banco de filtros con $\phi = \{0, \pi/2\}$, en el caso de $\phi = 0$ se introduce un término adicional de continua debido a que tanto la gaussiana como el coseno con fase nula son funciones pares. Con el objetivo de que los filtros sean lo más inmune posible al nivel de iluminación medio en una región particular, se busca eliminar dicho término. Para esto, a los núcleos de la expresión *base* se le debe restar un término adicional, obteniendo:

$$W(x, y, \theta, f, \phi, \sigma, \gamma) = e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} \left(\cos(2\pi f x') - \cos(\phi) e^{-2\pi^2 \frac{\sigma^2}{f^2}} \right)$$

Potencia

Otro punto importante a analizar es la potencia de cada uno de los filtros del banco. Cada uno de los núcleos anteriores no se encuentra normalizado en potencia, es decir la potencia de cada uno de los filtros depende de la frecuencia del mismo. Si se calcula la potencia de cada uno de los núcleos se ve que la misma es proporcional a $\frac{1}{f^2}$. Diferentes experimentos [59] muestran que las imágenes naturales presentan una densidad espectral de potencia que decae como f^2 a medida que aumenta la frecuencia. Por esta razón, los creadores de la técnica EBGGM [10] incluyen un término que compensa dicho decaimiento. Según observa Lades *et al.* en [60], la performance del algoritmo se ve seriamente afectada si no se logra independizar la potencia resultante al filtrar mediante cada uno de los núcleos de la frecuencia del núcleo en cuestión. En resumen, al introducir un factor para compensar el decaimiento de la potencia con la frecuencia en el caso de imágenes naturales se obtiene:

$$W(x, y, \theta, f, \phi, \sigma, \gamma) = f^2 e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} \left(\cos(2\pi f x') - \cos(\phi) e^{-2\pi^2 \frac{\sigma^2}{f^2}} \right)$$

²Cuando una señal filtrada no tiene término de continua a esta se la denomina comúnmente *DC-free*.

De esta manera la potencia encerrada en cada núcleo será proporcional a f^2 . Dicha compensación es correcta siempre y cuando las imágenes de interés sean imágenes naturales. A continuación se realiza un estudio del comportamiento estadístico del espectro de las imágenes de caras con intención de construir un modelo particular para las mismas.

5.5.2. Estudio Estadístico del Espectro de Imágenes de Caras

Diferentes experimentos estadísticos muestran que las imágenes naturales tienen un densidad espectral de potencia que decae como el cuadrado de la frecuencia. Se realizaron dos tipos de ensayos de manera de ver si las imágenes de caras siguen este mismo modelo.

Estimación de la densidad espectral de potencia

El primer ensayo realizado consiste en estimar la densidad de potencia de las imágenes de caras, tomando como base un conjunto particular de 25 imágenes. Las imágenes utilizadas son un pequeño subconjunto de imágenes de la base FERET. Para el cómputo de la densidad espectral de potencia se siguió un procedimiento basado en [61], que consiste en lo siguiente:

Estimación de la Densidad Espectral de Potencia

1. Se preprocesa la imagen.
2. Se multiplica la imagen por una ventana de Kaiser-Bessel de parámetro 2, con el fin de suavizar los efectos de borde.
3. Se calcula el módulo al cuadrado de la transformada de Fourier de la imagen enventanada. (Densidad Espectral de Potencia)
4. Para cada frecuencia se realiza un promedio en todas las direcciones posibles (promedio circular) de manera de independizarse de la dirección particular. A este vector unidimensional le llamamos $G(f)$: Densidad espectral promediada para la imagen en cuestión.
5. Se supone un modelo particular de la forma: $G(f) = Kf^\alpha$. Lo que interesa relevar es la constante de decaimiento α . En imágenes naturales $\alpha = -2$.
6. Se realiza un proceso de ajuste por mínimos cuadrados para relevar los parámetros del modelo supuesto para la imagen particular.
7. Se repite el procedimiento anterior para todas las imágenes del subconjunto, se promedia los valores relevados de α de cada modelo y se estima su desviación estándar.

En la figura 5.12 se puede observar el módulo del espectro de una imagen particular de una cara. Se observa que el decaimiento con la frecuencia es muy rápido. Además se observa que el espectro es simétrico en las distintas orientaciones. La figura 5.13 muestra el cuadrado del espectro (Potencia) circularmente promediado en las diferentes orientaciones. A continuación se muestra algunos resultados:

Ensayo I	α	σ_α
Sin Eq. Histograma	-3.05	0.12
Con Eq. Histograma	-2.98	0.16

De la tabla anterior se desprende que la densidad de potencia en imágenes de caras decae de manera más rápida que en las imágenes naturales. Por este motivo, el término de compensación de la potencia de cada uno de los núcleos de Gabor deberá ser mayor al previsto para imágenes naturales. Si a cada uno de los núcleos se lo multiplica por un término de la forma $f^{2+\beta}$ la potencia de cada núcleo será proporcional a: $f^{2+2\beta}$. Según este experimento si se quiere que la potencia resultante del filtrado con cada núcleo quede independiente de la frecuencia deberemos imponer: $2 + 2\beta = -\alpha$ o lo que es lo mismo: $\beta = -1 - \frac{\alpha}{2}$. En ambos casos se obtiene un valor cercano a $\beta = 0,5$.

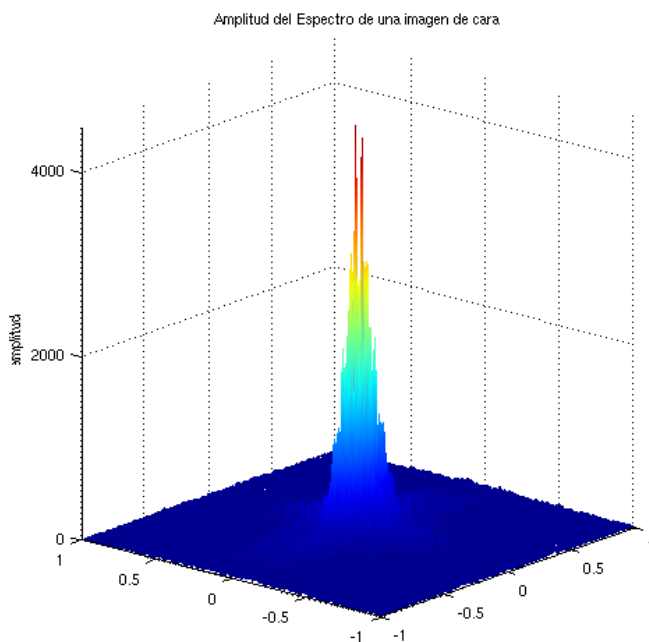


Figura 5.12: Módulo del Espectro de una imagen preprocesada.

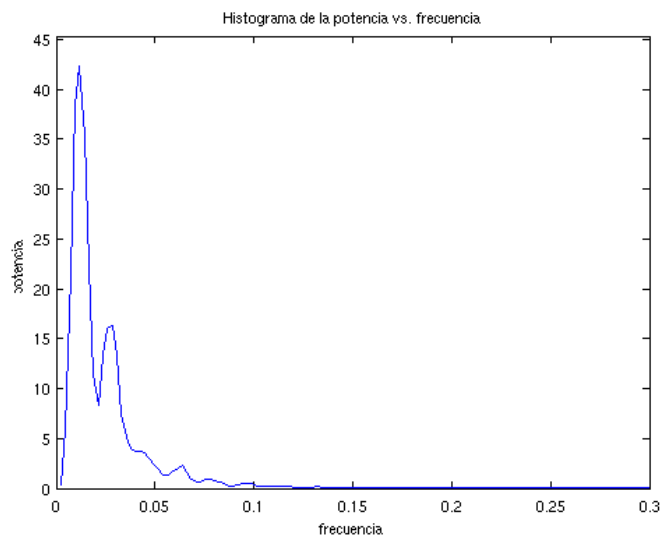


Figura 5.13: Módulo de la Potencia promediada en función de la frecuencia para una imagen preprocesada.

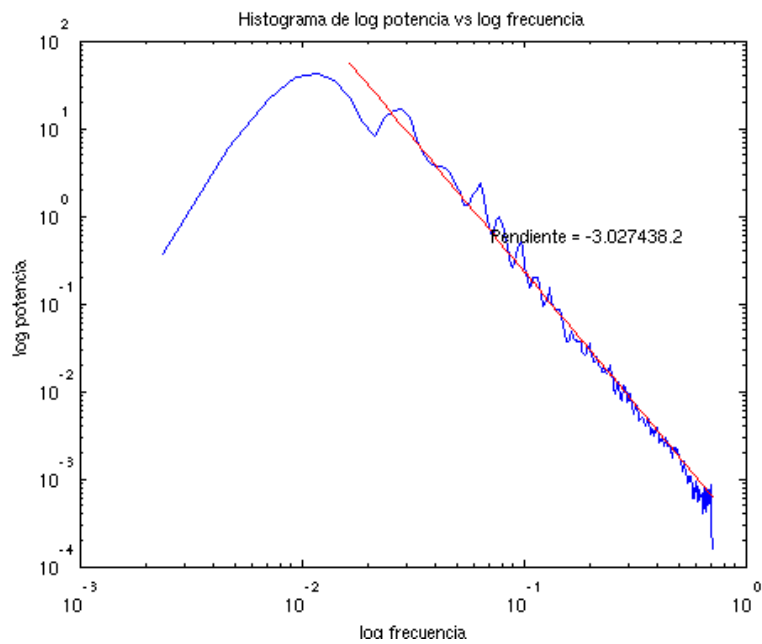


Figura 5.14: Módulo de la Potencia promediada en función de la frecuencia en escala logarítmica. Mejor pendiente, ajustada mediante mínimos cuadrados

En la figura 5.14 se muestra para una imagen de una cara preprocesada el espectro en escala logarítmica. Además se indica cuál fue el valor de

pendiente estimada mediante mínimos cuadrados. De manera de confirmar los resultados anteriores se realizó un experimento más directo para poder estimar la dependencia de la frecuencia con la potencia resultante de filtrar con cada uno de los núcleos.

Dependencia de la potencia del filtrado con la frecuencia

El segundo ensayo realizado consiste en estimar directamente la dependencia de la potencia resultante al filtrar imágenes de caras con cada uno de los núcleos. Para esto se supuso un factor de compensación de la forma: $f^{2+\beta}$, con $\beta \in [-1, 1]$. Para realizar este experimento se consideró un subconjunto de 10 imágenes de la base FERET y se discretizó el intervalo de β tomando $\beta_k = -1 + 0,05k$ con $k = 0, 1, \dots, 20$. Para cada β_k realizamos el siguiente procedimiento:

Dependencia de la potencia con la frecuencia

1. Se preprocesa cada una de las imágenes.
2. Se filtra cada imagen con el banco de filtros de Gabor **Wiskott** pero a cada uno de los filtros se lo ajusta por un factor: $f^{2+\beta_k}$. Se calcula la potencia obtenida al filtrar con cada uno de los núcleos.
3. Se promedian los resultados de filtrar mediante las distintas orientaciones para cada una de las escalas.
4. Se promedian los resultados obtenidos para cada imagen.
5. Finalmente se calcula la desviación estándar normalizada^a de los promedios calculados en el paso anterior para cada una de las escalas.

$$^a \text{stdn}(x) = \frac{\text{std}(x)}{\text{mean}(x)}$$

Se supone que si la compensación es buena la desviación estándar normalizada debe ser pequeña, es decir los resultados de filtrar con cada una de las escalas deberían de conservar todos más o menos la misma potencia.

En la figura 5.15 se puede ver la desviación estándar normalizada en función de β , la variable de ajuste. Se observa claramente que para los valores próximos a $\beta = 0,5$ existe un mínimo de la desviación estándar normalizada. También se observa que para el caso de las imágenes ecualizadas el mínimo se produce apenas antes que en el caso ordinario. En la tabla siguiente se resume los valores importantes del gráfico:

Ensayo II	β
Sin Eq. Histograma	0.50
Con Eq. Histograma	0.40

Como se puede observar el mínimo se da para valores próximos a $\beta = 0,5$, valor muy similar al encontrado en el Ensayo I. A modo de resumen, se ha logrado identificar el comportamiento espectral de imágenes de caras, logrando identificar la correcta compensación de potencia para estas imágenes particulares.

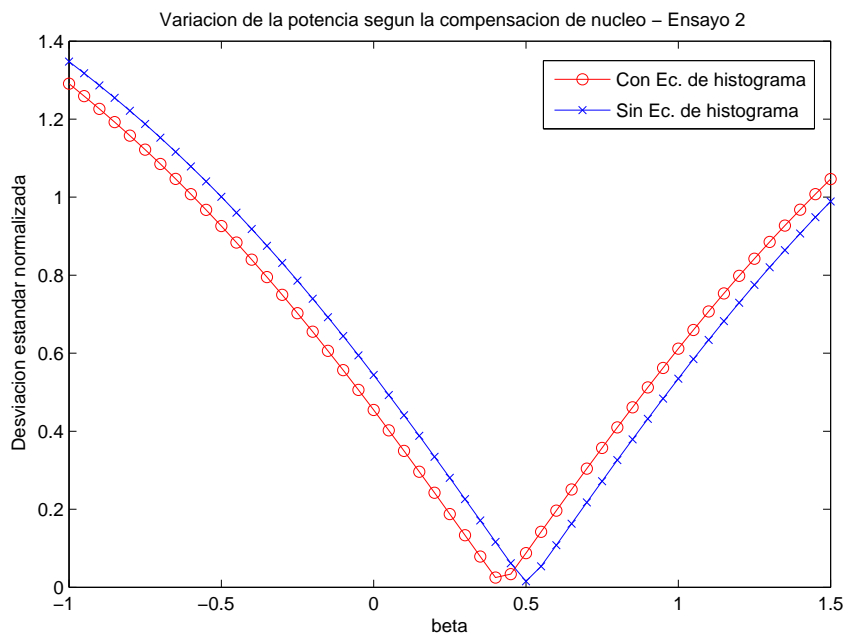


Figura 5.15: Resultados del segundo Ensayo realizado: Desviación estándar normalizada en función de β .

Por último la expresión de los filtros de Gabor con todas las consideraciones vistas queda:

$$W(x, y, \theta, f, \phi, \sigma, \gamma) = f^{2,5} e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} \left(\cos(2\pi f x') - \cos(\phi) e^{-2\pi^2 \frac{\sigma^2}{f^2}} \right)$$

En el capítulo 9 se observa el impacto de este estudio en el desempeño del sistema de reconocimiento de caras.

Capítulo 6

Localización de Puntos Principales

6.1. Introducción

El presente capítulo profundiza en los esquemas de localización de puntos principales estudiados para el sistema de reconocimiento. La elección de un descriptor orientado a características locales hace necesaria esta etapa en el bloque de extracción de características. Dentro de los algoritmos estudiados en el capítulo 3 se elige EBGM, Elastic Bunch Graph Matching y AAM, Active Appearance Models, como opciones a considerar. Se presenta a continuación un resumen teórico de estos dos algoritmos. Cada uno de ellos fue probado con implementaciones existentes, accesibles por trabajos realizados por grupos de investigación en el resto del mundo.

En abstracto, cada uno de estos algoritmos utiliza técnicas de ajuste basadas en criterios estadísticos de similitud entre estructuras (vectores de características). Cada componente de esta estructura es representante de ciertas características locales dentro de la cara. Este concepto se comprende ampliamente observando la esencia de ambos algoritmos que se describe en la primera parte de este capítulo.

6.2. Active Appearance Models

6.2.1. Introducción

El algoritmo Active Appearance Models, se introduce en 1998 por *Edwards et al.* [62]. Es una generalización del conocido Active Shape Models (ASM) [63], que utiliza toda la información de la imagen comprendida por el objeto de interés, en lugar de utilizar únicamente los bordes.

El método se basa en ajustar un modelo estadístico a una imagen nueva en busca de objetos pertenecientes a la clase representada por el modelo. Esto implica encontrar los parámetros del modelo que minimizan la diferencia entre la imagen original y la representación del objeto dada por el modelo. La gran cantidad de parámetros a sintonizar complejizan enormemente este problema.

6.2.2. Aplicación a Caras: Modelado de la Apariencia Facial

En el reconocimiento de caras queda claramente definido el objeto de interés, se debe generar un modelo de la apariencia facial. Este modelo se obtiene mediante la combinación de modelos de variación de forma y apariencia del rostro.

Modelado de la forma

Dada una imagen, se describe una forma marcando n puntos característicos (x_i, y_i) . Queda entonces definida la forma por el vector

$$x = (x_1, \dots, x_n, y_1, \dots, y_n)^T$$

Los puntos característicos elegidos varían según la aplicación y pueden ser marcados tanto de forma manual como semi-automática. Esta última implica generar un modelo de pocas imágenes marcadas manualmente y utilizar el mismo para ubicar puntos en nuevas imágenes. Luego se corrigen manualmente los puntos obtenidos y se enriquece el modelo. La figura 6.1 muestra un ejemplo de puntos característicos marcados a mano para el modelado de caras.

En primer lugar, antes de realizar un análisis estadístico de la forma, es importante que las imágenes a utilizar estén representadas en el mismo marco de coordenadas. Se desea eliminar así las variaciones atribuibles a transformaciones globales. Existen distintas propuestas para llevar las imágenes al mismo marco de coordenadas [64]. La idea básica es utilizar una transformación de similitud de forma de alinear los vectores con la forma media. Para ello se buscan los parámetros X_t, Y_t, s y θ de traslación, escalado y rotación, que minimicen la expresión:

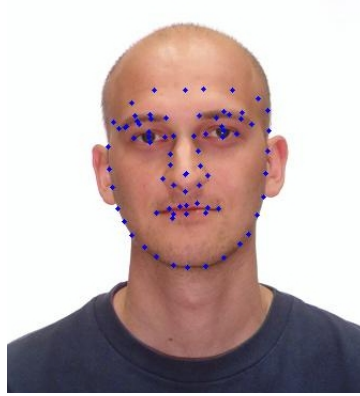


Figura 6.1: En la figura se observa un posible ejemplo de puntos de interés marcados manualmente para la creación del modelo de forma.

$$|T_{X_t, Y_t, s, \theta}(x) - \bar{x}|^2$$

siendo \bar{x} el valor medio de los vectores de forma considerados.

Sean $\{x_n^1, \dots, x_n^N\}$ un set de formas de entrada normalizadas al mismo marco de coordenadas. Se aplica PCA (Principal Component Analysis) a las mismas para lograr un subespacio representativo de la variación de la forma del objeto de interés. Se puede entonces aproximar cualquier elemento del set de entrenamiento, x_n , mediante la expresión

$$x_n \simeq \bar{x} + \Phi b_s$$

siendo Φ la matriz de vectores propios de la matriz de correlación de las muestras y b_s los coeficientes de proyección de x_n en el subespacio antes generado. El número de vectores propios considerados se elige de manera que el modelo represente cierta proporción de la variación total de los datos.

Se concluye entonces que un ejemplo del modelo de forma en una imagen, queda descrito por los parámetros b_s combinados con la transformación $T'_{X_t, Y_t, s, \theta}$, inversa de T , que lleva la imagen al marco de coordenadas original. Es decir, primero se sintetiza a través del modelo la forma normalizada y luego se transforma al espacio de coordenadas original.

$$x \simeq T'_{X_t, Y_t, s, \theta}(\bar{x} + \Phi b_s)$$

Modelado de la textura

Luego de generado el modelo estadístico de forma, se tiene la forma media. Utilizando la forma media se deforma cada muestra hasta la forma media

[64], obteniendo una imagen *shape-free*. El modelo estadístico de textura se construye en base a estas nuevas imágenes *shape-free*. Éstas contienen poca información de variaciones de textura debidas a expresiones exageradas, lo que se contempla básicamente en el modelo de forma.

Se muestrea la información de intensidad de la imagen *shape-free* sobre la región comprendida por la forma media, generando un vector de textura g . Para minimizar el efecto de variaciones de la textura debidas a la iluminación se aplica un escalado α y un offset β [64], obteniéndose

$$g_n = (g - \beta)/\alpha$$

Nuevamente recurrimos al análisis de componentes principales para generar un modelo que represente la variabilidad de la textura para el objeto cara. Sean $\{g_n^1, \dots, g_n^N\}$ un set de muestras de textura normalizadas según α y β . Se aplica PCA a las mismas y se obtiene el modelo de textura. Cada una de estas muestras puede entonces aproximarse según la expresión:

$$g_n \simeq \bar{g}_n + P_g b_g$$

siendo \bar{g}_n el valor medio de los datos normalizados, P_g un set de modos ortogonales de variación de textura y b_g los parámetros para este modelo. La cantidad de modos considerados se ve reflejada en la calidad de las texturas generadas.

La textura en el marco de la imagen queda dada por

$$g \simeq T_u(\bar{g}_n + P_g b_g)$$

donde T_u representa la transformación inversa al escalado y offset antes mencionados.

Combinación de modelos de forma y textura

Como se mencionara anteriormente el modelo de apariencia se forma combinando los modelos de forma y textura. La forma y la textura de cada muestra pueden ser representadas por los vectores de parámetros b_s y b_g . Considerando la correlación existente entre la variación de la forma y la textura, se aplica nuevamente PCA a la combinación de estos modelos para generar un único modelo de la apariencia facial.

Para cada muestra se genera el vector b igual a la concatenación de los vectores b_s y b_g . Se aplica PCA a dichos vectores obteniéndose el modelo

$$b = P_c c$$

donde P_c son los vectores propios y c es el vector de parámetros de apariencia, controlando ambos forma y textura.

Dada la naturaleza lineal del modelo se puede plantear

$$\begin{aligned}x &= \bar{x} + Q_s c \\g &= \bar{g} + Q_g c\end{aligned}\tag{6.1}$$

Una imagen puede ser sintetizada para un c dado generando la imagen *shape-free* de niveles de gris correspondientes al vector g y deformándola utilizando los puntos dados por x .

6.2.3. Búsqueda del Active Appearance Model

Dada una imagen nueva, un modelo de apariencia y una posición inicial aproximada se propone un algoritmo para ajustar los parámetros del modelo y sintetizar una cara que maximice la similitud con la presente en la imagen. El objetivo pasa a ser encontrar los parámetros que minimizan la diferencia entre la imagen original y la sintetizada a partir del modelo. Dado que se trabaja con gran cantidad de parámetros, del orden de 80, se utiliza información extra para realizar el ajuste. En particular se espera que la imagen error, diferencia entre valores de nivel de gris para la imagen original y la sintetizada, guarde información útil para el ajuste. Por ejemplo, si la máxima diferencia se alcanza en los costados de la cara, es necesario ajustar el parámetro que controla el ancho de la cara. Se vuelve crucial encontrar la relación entre la imagen error y el error en los parámetros. Planteos teóricos [64] que exceden el alcance de este material proponen una relación lineal entre la imagen error y el error en los parámetros.

Finalmente, luego de hallados los parámetros del modelo que mejor ajustan a la imagen de entrada, se puede obtener los puntos que determinan la forma de la cara. Se recuerda que el objetivo de esta etapa es localizar puntos característicos en la imagen. Se utiliza la expresión 6.1 para obtener en el vector x las coordenadas de los puntos buscados.

En la figura 6.2 se observa un ejemplo de resultados obtenidos al realizar la localización automática de puntos utilizando AAM. El modelo estadístico se construye con los puntos desplegados en la figura 6.1. Cabe aclarar que la imagen de muestra no fue utilizada en la construcción del modelo estadístico.

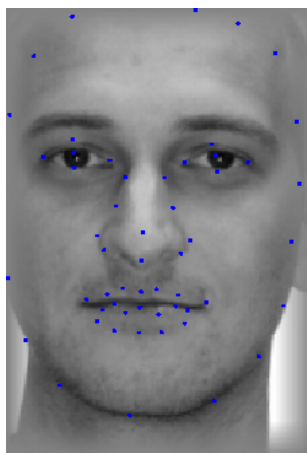


Figura 6.2: Resultado de la localización automática de puntos principales utilizando AAM.

6.3. Elastic Bunch Graph Matching

El algoritmo EBGM es una herramienta de reconocimiento de objetos, aplicada a caras en 1997 por *Wiskott et al.* [10], que implementa tres etapas básicas:

- Localización de puntos principales en el objeto de interés.
- Extracción de características en dichos puntos.
- Confrontación de estructuras de características para la validación del objeto.

Se utiliza aquí este algoritmo como herramienta de localización de puntos característicos para el reconocimiento de caras.

6.3.1. Aspectos fundamentales de la localización de puntos

Grafo de puntos característicos

Se define una estructura de grafo sobre el objeto de interés cuyos nodos son los puntos característicos elegidos. En el caso del reconocimiento de caras, ejemplos de dichos puntos principales son las pupilas, la punta de la nariz, los extremos de la boca, etc. Se eligen puntos fácilmente identificables de un individuo a otro. La figura 6.3 muestra un ejemplo de elección de dichos puntos.

Los descriptores utilizados son un banco de filtros de Gabor de diferentes frecuencias y orientaciones. En cada nodo del grafo se calculan dichos wavelets de Gabor, obteniéndose una estructura denominada *jet*. Cada nodo

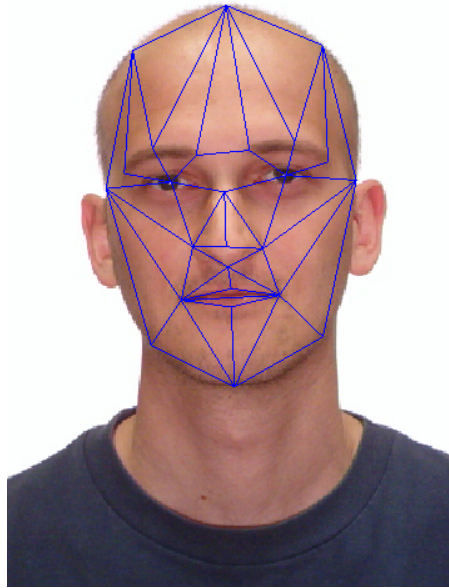


Figura 6.3: Ejemplo de grafo de puntos principales.

es entonces etiquetado con las coordenadas del punto característico y el *jet* asociado al mismo, mientras que las aristas definidas se identifican con la distancia entre nodos conectados.

Modelo estadístico de la cara: Face Bunch Graph

Para lograr la localización automática de puntos característicos en nuevas caras se necesita una descripción general del objeto cara. Dicha descripción debe cubrir una amplia gama de variaciones como ser diferentes formas de ojos, boca, presencia de barba, diferentes sexos, razas, edades, etc. Para ello se combina un grupo representativo de grafos modelo en una estructura del tipo stack denominada *Face Bunch Graph* (FBG). Cada grafo modelo tiene la misma forma y los nodos refieren a los mismos puntos principales. Cada nodo del FBG resulta etiquetado con un conjunto de *jets*, provenientes de los distintos grafos modelo, y el promedio de las coordenadas de los nodos participantes. Las aristas se etiquetan con la distancia promedio de los grafos modelo. La figura 6.4 muestra una representación intuitiva del modelo FBG.

El FBG representa las características que hacen a una cara y permite además contemplar variaciones existentes entre individuos. La amplitud y calidad del modelo dependen de la cantidad y variabilidad de las imágenes elegidas en su construcción. En la literatura dedicada a este algoritmo [44] se mencionan modelos de 70 imágenes, pudiéndose tomar esta cifra como referencia para el tamaño del mismo. Dependiendo de los requerimientos de



Figura 6.4: Representación intuitiva del FBG.

cada aplicación particular se define dicho valor. Como se comprenderá en breves instantes, el tamaño del FBG está íntimamente relacionado con el tiempo que toma la localización de nuevos puntos.

Por último cabe señalar que los grafos utilizados para la generación del FBG pueden ser marcados de forma manual o semiautomática. El marcado manual, aunque costoso, garantiza la correcta ubicación de los puntos. Por otro lado se proponen técnicas semiautomáticas, en las que se crea un FBG de pocas imágenes marcadas manualmente, que luego se utiliza para ajustar las imágenes siguientes. Se recomienda corregir los puntos obtenidos de forma automática e ir enriqueciendo el modelo.

Ajuste automático de grafos

El objetivo planteado para una imagen de entrada al sistema es encontrar los puntos principales obteniendo el grafo que maximice la similitud con el FBG considerado. La función de similitud utilizada considera similitud de *jets*.

Se proponen varios métodos al respecto [10, 44]. En general la idea básica tras éstos puede resumirse en los siguientes pasos:

1. Se obtiene la posición inicial de un nodo en la imagen. Se plantean varias técnicas para el cálculo de la misma.

2. Dada la estimación inicial del nodo, se extrae el *jet* correspondiente en la imagen de entrada. Se halla la similitud del mismo con cada uno de los *jets* del FBG para el nodo correspondiente. Se repite para posiciones cercanas a la inicial. Se retiene la posición del nodo en la imagen nueva que resulta en el máximo valor de similitud.
3. Se repite el paso anterior para cada nodo del grafo.

Los pasos anteriores son parte de un ciclo iterativo que mejora en cada etapa la posición de los nodos hasta cumplirse una condición de finalización. En la figura 6.5 se muestra un ejemplo de los puntos ajustados a la cara al final del proceso.

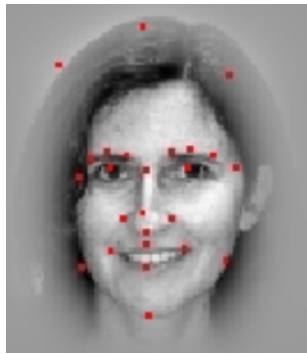


Figura 6.5: Puntos ajustados mediante EBGM.

Está claro que la anterior descripción es básica y presenta en la realidad muchas variantes. La misma pretende dar una idea de cómo se utiliza el FBG en el ajuste de grafo. En este punto se clarifica lo mencionado anteriormente sobre la relación entre el tamaño del FBG y el tiempo que tarda el ajuste. Cuantos más *jets* formen el FBG para cada nodo, más tiempo toma la evaluación de similitudes. De la misma manera, el incremento en el tamaño del FBG trae consigo mayor variabilidad para cada nodo.

6.3.2. Elección del Algoritmo de Localización

La performance obtenida en la localización de puntos característicos es buena para ambos algoritmos. Las figuras 6.2 y 6.5 muestran ejemplos de esto. En las mismas se observa cómo los puntos resultan correctamente ubicados, a menos de cierta tolerancia.

Analizando otros aspectos de interés para el sistema, se observa que los algoritmos utilizan para la localización un modelo estadístico de la cara independiente de las imágenes de trabajo. Es decir, ambos permiten tener un

modelo único y utilizarlo en distintas aplicaciones, no siendo necesario entrenar el sistema cada vez que ingresa un sujeto. Con respecto a la demora en el ajuste, tienen un desempeño similar.

La complejidad de los mismos se encuentra comparable, son algoritmos de compleja implementación. Los resultados desplegados se obtienen con implementaciones en código libre de ambos algoritmos. Para AAM se utiliza la versión desarrollada por uno de sus creadores, Tim Cootes [65], mientras que para EBGM se utiliza la versión desarrollada por David Bolme en [12]. No es objetivo de este proyecto codificar una nueva versión de ninguno de estos algoritmos, esto excede el alcance del mismo y no aporta a su esencia.

Considerando que tienen similar performance y no existiendo otro aspecto funcional que incline la decisión hacia uno de ellos, se opta por utilizar la localización de puntos característicos dada por EBGM. En esta decisión influyen distintos aspectos. En primer lugar, como se detalla en el capítulo anterior, se elige como descriptores de la cara los waveletes de Gabor definidos en este algoritmo, resultando natural mantener el ajuste propuesto en el mismo. Por otro lado, de las implementaciones antes mencionadas, únicamente la correspondiente a EBGM es realizada en código abierto. Una de las consignas de este proyecto es finalizar el trabajo dejando una implementación en código abierto del mismo, con lo cual se reafirma la decisión de utilizar EBGM.

6.4. EBGM: Una Implementación

En este proyecto se trabaja sobre una implementación base del algoritmo. La misma fue desarrollada en la Universidad Estatal de Colorado (CSU) [11] por David Bolme en su Tesis de Maestría en Computación. El código fuente de dicha implementación se encuentra disponible [12]. Esta tesis se basa en la publicación original realizada en BOCHUM/USC [66].

6.4.1. Estimación Inicial de la Posición

Considerando que en la etapa de detección se obtiene las coordenadas de los ojos con baja incertidumbre, se asume conocida la posición aproximada del resto de los nodos. Para la localización del i -ésimo nodo se utilizan las coordenadas de los $(i - 1)$ nodos previamente localizados. Se comienza con las coordenadas de los ojos conocidas y se procede a localizar desde el centro de la cara hacia afuera el resto de los nodos. Esto se realiza en forma radial y de a un nodo por vez.

La estimación inicial realiza las siguientes consideraciones geométricas:

- Por definición, cada nodo del FBG contiene las coordenadas promedio

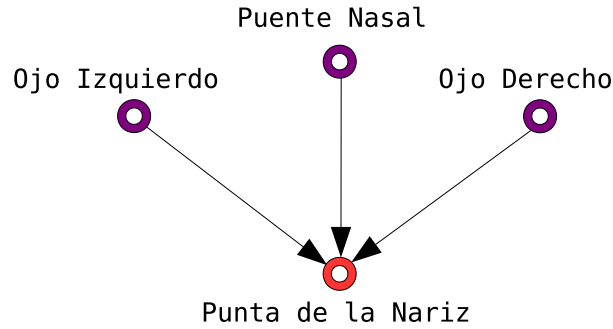


Figura 6.6: Estimación de la posición inicial de cada nodo.

de los puntos característicos de los grafos utilizados en el modelo. Por lo tanto para dos nodos cualesquiera m y n el vector diferencia entre las coordenadas correspondientes se puede representar en el plano mediante:

$$v_{mn}^{\vec{}} = \vec{p}_n - \vec{p}_m \quad (6.2)$$

- La idea de utilizar aquí los nodos promedio es que estos nos permiten estimar para cualquier nodo dado o_m :

$$p_{o_m}^{\vec{}} \approx \vec{p}_m + v_{mn}^{\vec{}} \quad (6.3)$$

siendo $v_{mn}^{\vec{}}$ la estimación de la diferencia utilizando el modelo estadístico FBG.

- Al conocer la ubicación de varios nodos promedio podemos mejorar la estimación sumando sobre los distintos vectores diferencia conocidos. La imagen 6.6 ilustra esta idea:

$$p_{o_m}^{\vec{}} \approx \frac{1}{M} \sum_{j=1}^M (\vec{p}_j + v_{jm}^{\vec{}}) \quad (6.4)$$

El último agregado que se realiza es el ponderar según la distancia de cada nodo ya estimado al nodo a estimar. De esta manera los nodos más cercanos tendrán mayor efecto que los nodos alejados, hecho que concuerda con nuestra intuición.

Tomando entonces:

$$d_{jm} = \|\vec{v}_{jm}\| \quad (6.5)$$

$$w_{jm} = e^{-d_{jm}} \quad (6.6)$$

se llega a la ecuación utilizada para estimar la posición inicial de cada nodo en base a los nodos previamente localizados:

$$\vec{p}_{o_m} \approx \frac{\sum_{j=1}^M (\vec{p}_j + \vec{v}_{jm}) w_{jm}}{\sum_{j=1}^M w_{jm}} \quad (6.7)$$

Luego de estimada la posición inicial de cada nodo se implementa una etapa de refinamiento que permite obtener las coordenadas del punto característico lo más ajustadas posibles. La siguiente sección describe dicho refinamiento.

6.4.2. Refinamiento de la Posición: Estimación de Desplazamiento

Con el objetivo de complementar el ajuste inicial de los puntos característicos a la imagen, surge el concepto de *Estimación de Desplazamiento* [11]. El mismo consiste en encontrar la ubicación real de un punto característico, a partir de la ubicación estimada del mismo. El vector diferencia entre la posición del punto inicial y la ubicación real de dicho punto es llamado *Desplazamiento*.

Para poder realizar esto se utiliza el modelo FBG, de manera de tener en cuenta cómo son los *jets* típicos de cada uno de los puntos característicos. La manera exhaustiva de encontrar la ubicación real del punto característico, es extraer un *jet* por cada píxel de la imagen, comparar ese *jet* contra los pertenecientes al FBG para ese mismo punto característico, y quedarse con el píxel que maximiza la similitud con el modelo. Lamentablemente realizar esto tiene un costo computacional extremadamente elevado. Es por esto que surgen otras técnicas alternativas para encontrar los puntos de la imagen que maximizan - en determinadas condiciones - dicha similitud.

Dado que las componentes de una cara guardan por lo general determinadas relaciones geométricas, localativas (es decir, la boca, la nariz, los ojos siempre están relativamente en la misma posición dentro de una cara), no es necesario realizar la anterior búsqueda en toda la imagen. Cada uno de los puntos característicos, con muy alta probabilidad, pertenecerá a una pequeña región fija en la imagen a ajustar. Como se viera en la sección 5.3.2, si se quiere encontrar la similitud con el modelo para un punto cercano al cual ya se extrajo el *jet*, no es necesario extraer el nuevo *jet* para re-calcular la similitud. Se puede utilizar la función de similitud aproximada S_D . La

idea central de las técnicas de Estimación de Desplazamiento es, a partir de un punto cercano al óptimo, maximizar la función S_D en un entorno de dicho punto. De esta manera se ahorra la re-extracción de *jets* en puntos muy próximos.

Si se plantea el problema de maximizar S_D de manera formal el objetivo es encontrar $\vec{d} = [d_x, d_y]$ tal que:

$$\vec{d} = \arg \max S_D(d, J_1, J_2)$$

La manera analítica de resolver este problema es resolver $\nabla_d S_D = 0$. Lamentablemente al tratarse S_D de una suma de funciones sinusoidales de diferente amplitud y frecuencia, este problema no tiene solución analítica obvia. Es por eso que surgen diferentes propuestas - no analíticas - de maximizar S_D .

Se puede separar las diferentes técnicas en subconjuntos:

- Basadas en Búsquedas: Este subconjunto de técnicas trata de encontrar el desplazamiento mediante búsquedas exhaustivas (en una pequeña región) o búsquedas *inteligentes* en direcciones particulares.
- Basadas en Aproximaciones: Este subconjunto busca encontrar el máximo, basándose en una aproximación de Taylor de segundo orden para la función de Similitud. Esta aproximación sí cuenta con una manera de ser maximizada analíticamente.
- Basadas en algoritmos iterativos genéricos de maximización.

A continuación se describen los estimadores de desplazamiento utilizados, tanto los implementados en el sistema CSU como los desarrollados en este proyecto (DEPredictiveBigScale, DEGradientDescent).

A. Técnicas basadas en búsquedas

A.1. **Sampleo de Grilla (DEGridSample)** Es la técnica más directa para encontrar el máximo de la función S_D . Consiste en evaluar la función en puntos de una grilla rectangular con centro en la estimación inicial. Para su funcionamiento es necesario especificar el tamaño de la grilla y la distancia entre sus puntos. A continuación se muestra un pseudo-código de dicho algoritmo:

```
J = LeerModelo(Modelo,Punto);
J' = ExtraerJet(Imagen,PuntoEstimado);
aux = 0;
paso = 0.1;
```

```

Dmax = 15;

for (dx=-Dmax; dx<Dmax; dx+=paso)
  for (dy=-Dmax; dy<Dmax; dy+=paso)
    d = [dx,dy]
    aux = Sd(J,J',d)
    if aux > max_parcial
      max_parcial = aux;
      d_parcial = d;
    end
  end
end

d_max = d_parcial;

```

Si bien este algoritmo es muy efectivo en encontrar el máximo de la función S_D , no es en absoluto eficiente.

A.2. Búsqueda Local Fija (DEFixedLocalSearch) Con el objetivo de hacer que la búsqueda sea más eficiente, surge la idea de no recorrer de manera exhaustiva la totalidad de la región. La búsqueda local fija, trata de encontrar el máximo moviéndose una distancia fija en una de cuatro direcciones (Norte, Este, Sur, Oeste). Para elegir en qué dirección moverse, se computa la similitud en los posibles cuatro desplazamientos y se decide en función de cuál haya sido el máximo. El proceso se repite, partiendo desde la nueva ubicación. Finaliza una vez que ninguna de las cuatro direcciones logre mejorar la ubicación actual o se haya superado un número determinado de iteraciones. Esta búsqueda es mucho más eficiente que el muestreo de grilla y logra resultados cercanos a la anterior.

A.3. Búsqueda Local con Reducción del Paso (DENarrowingLocalSearch) Esta búsqueda es una pequeña variante de la búsqueda local fija. La diferencia radica en que el paso de ajuste es reducido a la mitad en cada iteración, mientras que en la búsqueda anterior el paso se mantenía fijo.

B. Técnicas basadas en aproximaciones

Este conjunto de técnicas se basa en una aproximación de Taylor de la similitud S_D [10]. En vez de resolver el máximo para la función S_D , la cual no tiene una manera analítica obvia de ser resuelta, se busca resolver el máximo para la aproximación. La aproximación se basa en el desarrollo de Taylor de segundo orden para la función $\cos(\theta)$:

$$\cos(\theta) \approx 1 - \frac{1}{2}\theta^2$$

$$S_D(J, J', \vec{d}) \approx \frac{\sum_{j=1}^N a_j a'_j [1 - 0,5(\phi_j - \phi'_j - \vec{d} \cdot \vec{k}_j)^2]}{\sqrt{\sum_{j=1}^N a_j^2 \sum_{j=1}^N a'_j{}^2}}$$

A diferencia de la expresión original para S_D , la aproximación anterior sí cuenta con una manera analítica obvia de ser maximizada, para más información ver apéndice E. El vector de desplazamiento que maximiza S_D es:

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \frac{1}{\Gamma_{xx}\Gamma_{yy} - \Gamma_{xy}\Gamma_{yx}} \cdot \begin{pmatrix} \Gamma_{yy} & -\Gamma_{yx} \\ \Gamma_{xy} & \Gamma_{xx} \end{pmatrix} \begin{pmatrix} \Phi_x \\ \Phi_y \end{pmatrix}$$

donde

$$\Phi_x = \sum_j a_j a'_j k_{jx} (\phi_j - \phi'_j)$$

$$\Gamma_{xy} = \sum_j a_j a'_j k_{jx} k_{jy}$$

Se deberá tener en cuenta que los términos de fases deben ser re-escalados para cumplir con $-\pi < \phi_j - \phi'_j < \pi$.

Este método es muy rápido, pero solamente funciona bien en la región en la cual es *válida* la aproximación de Taylor, por lo que es muy importante partir de un punto inicial cercano al máximo. Existen diferentes técnicas que computan el desplazamiento anterior de diferentes maneras: paso único, iterativo o por escalas.

B.1. Paso Predictivo (DEPredictiveStep) Este método se basa en la aproximación anterior y calcula el vector de desplazamiento en un único paso. El inconveniente que presenta es no tener en cuenta que la similitud es una función periódica en el desplazamiento \vec{d} lo cual produce que la región de convergencia al máximo sea pequeña. Es imprescindible partir de un punto cercano al máximo. Se puede ver que en el peor caso, la región de convergencia es mayor o igual al período del wavelet de Gabor más rápido.

B.2. Paso Predictivo por Escalas (DEPredictiveBigScale) Este método es una sutil variante del método anterior. Busca relajar la condición que para una correcta convergencia al máximo buscado es necesario partir de un punto muy cercano al mismo. Con este objetivo, estima el desplazamiento primero utilizando las componentes más lentas del *jet* (bajas frecuencias, ajuste grueso) para luego ir refinando la estimación. El tiempo de procesamiento no se ve afectado significativamente y se logran mejores resultados.

B.3. Predicción Iterativa (DEPredictiveIter) Este método es una generalización del método `DEPredictiveStep`. Busca refinar el ajuste iterando el algoritmo `DEPredictiveStep` una cantidad determinada de veces o hasta que se cumpla una condición de parada. El algoritmo comienza con un desplazamiento nulo, computa el desarrollo de Taylor en dicho punto y resuelve las ecuaciones anteriores para hallar el máximo \vec{d}_1 de S_D . Luego se re-computa el desarrollo de Taylor en el punto inicial desplazado \vec{d}_1 y se procede de la misma manera. El procedimiento se repite, finalizando cuando no se encuentra un cambio significativo en el desplazamiento o cuando se supera un número establecido de iteraciones.

C. Técnicas basadas en algoritmos genéricos de maximización

C.1. Descenso por Gradiente (DEGradientDescent) Al ser la función S_D una función suave, esta permite aplicar técnicas genéricas para la maximización de funciones de varias variables. En este caso el objetivo es encontrar el máximo de la función de similitud aplicando la técnica de descenso por gradiente. A continuación se describe brevemente en qué consiste dicha técnica:

1. Se computa el gradiente $\nabla_d S_D(\vec{d}_i)$ en el punto actual.
2. Se calcula la nueva ubicación del punto según: $\vec{d}_{i+1} = \vec{d}_i + \mu \nabla_d S_D(\vec{d}_i)$
3. Se repite los pasos 1 y 2 hasta que $\|\vec{d}_{i+1} - \vec{d}_i\| < \epsilon$ o hasta que se produzca un número determinado de iteraciones.

Si bien esta técnica no tiene el problema de lidiar con la validez de la aproximación de Taylor, es decir la técnica siempre va a converger a un máximo de la función S_D , este máximo no necesariamente deberá ser un máximo global. Al ser la función S_D una suma de funciones del tipo sinusoidal, lo único que se puede garantizar es que la región de convergencia al máximo buscado va a ser mayor o igual al período del wavelet de Gabor de frecuencia más alta.

6.5. Alternativas en el Ajuste de Puntos de EBG

En la etapa de implementación y análisis del sistema se busca alternativas en la estimación inicial de los nodos del grafo y en el refinamiento de los mismos. Se presenta en las siguientes secciones los estudios y pruebas realizados.

6.5.1. Estimación Inicial de las Coordenadas del Grafo

Se trabaja fundamentalmente en tres ideas. La primera es no utilizar el ajuste de grafo y extraer las características en las coordenadas del *Face Bunch Graph*. La segunda consiste en trabajar en lo que *Wiskott et al.* describe como primer paso en su etapa de ajuste fino. En la misma se realiza una búsqueda local del candidato a nodo y se compara con un FBG “condensado”, grafo cuyos *jets* son el promedio de todos los *jets* del FBG. La tercera idea en la que se trabaja implica una modificación del algoritmo CSU en la ecuación 6.6, donde se utiliza la similitud de cada punto para ponderar su aporte al nodo que se está ajustando.

A. Uso de los nodos del Face Bunch Graph para la extracción de jets

La idea es utilizar los nodos del modelo estadístico como las coordenadas finales. Directamente en estas coordenadas es que se extraen los *jets* correspondientes a esta imagen. La motivación de esta prueba es poder cuantificar la performance de reconocimiento del sistema sin realizar esfuerzo alguno en el ajuste del grafo, sino que utilice las coordenadas promedio dadas por el modelo estadístico. En términos intuitivos, es de esperar que la performance del algoritmo sea menor en este caso respecto al caso en que se realice el ajuste.

B. Inicialización de nodos mediante búsqueda gruesa aproximada

Se quiere realizar una búsqueda gruesa como paso intermedio entre la estimación inicial y el refinamiento de cada nodo. En el sistema CSU dijimos que la estimación inicial se realiza con la información del *Face Bunch Graph*, por lo tanto no se utiliza información local de la imagen de entrada para la estimación inicial. En este sentido se implementó una búsqueda exhaustiva alrededor de cada nodo dado por el *Face Bunch Graph*. La comparación en este caso se realiza nodo a nodo contra un grafo que contiene los valores promedio del *Face Bunch Graph*. En la figura 6.7 se muestra el resultado de realizar una búsqueda exhaustiva en un lattice de 3 y 1 píxel de lado.

Se aprecia que la corrección gruesa planteada no es eficaz. Para el caso del lattice de 3 píxeles existe un corrimiento de los puntos característicos ya inaceptable. Debemos también considerar el problema del tiempo. Recordando que al ser una búsqueda exhaustiva, se realizan ocho extracciones

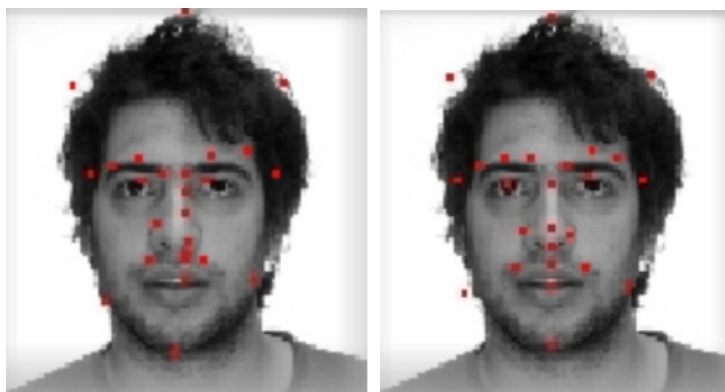


Figura 6.7: Resultado de la búsqueda exhaustiva para un lattice de 3 píxeles a la izquierda y para un lattice de 1 píxel a la derecha.

de *jets*, lo cual provoca un aumento excesivo de costo computacional que lo invalida directamente si estamos pensando en una aplicación de tiempo real.

C. Estimación inicial de los nodos ponderando por similitud

Como se explica en este capítulo el algoritmo de ajuste aplica a cada nodo a detectar dos etapas: la estimación inicial y el refinamiento. A su vez se vio que la estimación inicial se vale de una ponderación según la distancia de cada uno de los nodos previamente ajustados al nodo a ajustar actual. Cada nodo ya ajustado tiene una similitud definida con el *Face Bunch Graph*. Se puede definir una función de similitud de forma tal que pertenezca al intervalo $[0, 1]$. De esta manera podemos redefinir la ecuación 6.6 para este caso considerando ahora el peso s_{jm} definido como la similitud del nodo j ya ajustado.

Se llega a la ecuación utilizada para estimar la posición inicial de cada nodo ponderando por la similitud de cada uno de los nodos previamente localizados:

$$p_{o_m}^{\vec{}} \approx \frac{\sum_{j=1}^M (\vec{p}_j + v_{jm}^{\vec{}}) s_{jm}}{\sum_{j=1}^M s_{jm}} \quad (6.8)$$

Una variante de lo anterior que se puede obtener sin mucho esfuerzo es la combinación de los argumentos dados para la obtención de las ecuaciones 6.6 y 6.8:

$$p_{o_m}^{\rightarrow} \approx \frac{\sum_{j=1}^M (p_j^{\rightarrow} + v_{jm}^{\rightarrow}) w_{jm} s_{jm}}{\sum_{j=1}^M w_{jm} s_{jm}} \quad (6.9)$$

La última ecuación reflejará tanto la ponderación geométrica como la ponderación por similitud de los nodos previamente ajustados.

D. Ensayos realizados y Resultados

En los ensayos se mide el desempeño global del sistema para estudiar el impacto de las variantes presentadas. Se utiliza las dos bases de caras disponibles, la base IIE y la base FERET. El modelo estadístico (*Face Bunch Graph*) se genera con 68 imágenes de la base FERET. Es básicamente el que se utiliza en [11], el cual contiene 25 puntos definidos manualmente. Se fijan además las siguientes condiciones para realizar los experimentos: el banco de filtros de Gabor es el presentado por Wiskott y el método de estimación de desplazamiento es el de Predicción Iterativa.

Se descarta la segunda opción manejada, por lo que los ensayos realizados son los siguientes:

Ensayo 0 - Sistema CSU original (Est. inicial ponderando por distancia)

Ensayo 1 - Extracción de *jets* en los nodos del *Face Bunch Graph*

Ensayo 2 - Estimación inicial ponderando por similitud

Ensayo 3 - Estimación inicial ponderando por similitud y distancia

Se utiliza una misma herramienta de evaluación para las pruebas con la base IIE y la base FERET. En todos los casos se define un conjunto de m imágenes de referencia y n imágenes de prueba. Se realizan las confrontaciones positivas y negativas correspondientes y en base a esta información se obtiene la curva ROC, que permite ver el error que se comete en términos de falsos negativos y falsos positivos al ir variando el umbral de decisión.

D.1. Tests con la base IIE Para los tests sobre esta base se toma un conjunto de 47 identidades. Las imágenes de referencia corresponden al subconjunto *adjj* y *ad*. Estas imágenes corresponden a tomas de frente, en condiciones controladas de expresión y de iluminación. Las imágenes de prueba son el subconjunto *ux* que corresponde también con imágenes de frente obtenidas un período de tiempo posterior al subconjunto *adjj*. También se utiliza el subconjunto *m* que corresponde a la persona sonriendo.

En este caso se trabaja con conjuntos de imágenes pequeños por lo que los resultados tienen grandes incertidumbres asociadas¹. En las distintas pruebas no se observan diferencias considerables como para poder establecer conclusiones acerca de cuál es la opción de mejor desempeño.

D.2. Tests con la base FERET Se trabaja con un subconjunto de 275 imágenes de la base FERET pertenecientes a la galería fa y 259 imágenes del subconjunto de prueba fb. Todas ellas son imágenes de frente de complejidad relativa. Por ejemplo una cantidad considerable de rostros lleva anteojos.

Las mejoras relativas² respecto al Ensayo 0 son:

Ensayo 1: TER = 12,4 %

Ensayo 2: TER = 8,3 %

Ensayo 3: TER = 8,3 %

E. Conclusiones

Se implementa y evalúa una serie de alternativas a la estimación inicial de la posición de los nodos del sistema CSU. Algunas de las mismas surgen de revisar la idea original del algoritmo BOCHUM/USC, como por ejemplo la implementación de una búsqueda exhaustiva para un ajuste grueso previo a la estimación inicial. Se descarta esta alternativa por elevar el costo computacional a un nivel inaceptable.

Se trabaja con las dos bases de caras a disposición, la base IIE y la base FERET, y se utiliza el EER como índice de performance. En ambos casos, se padece del problema del tamaño de las muestras, lo que no permite extraer conclusiones categóricas.

En relación con la inicialización del grafo utilizando los nodos del *Face Bunch Graph* no se le descarta como una vía de muy bajo costo computacional y de una tasa de error aceptable. Este resultado en teoría indica que si el algoritmo de ajuste es efectivo, la performance del sistema debe estar acotada inferiormente por este error. Es decir que la inteligencia que agrega el algoritmo debe mejorar la performance del sistema. De todas maneras, pese a la incertidumbre de las medidas, en este caso no se llega a tal conclusión.

Para las pruebas que modifican los pesos que se le da a cada nodo previamente ajustado para la estimación inicial del nodo actual, se ve que el EER es del orden del EER del sistema original.

¹En el capítulo 9 se muestra cómo estimar los intervalos de confianza.

²Tasa de Error Relativa: $TER = \frac{EER_{BASE} - EER}{EER_{BASE}}$.

6.5.2. Refinamiento del Ajuste de Grafo

Para poder medir la performance de las diferentes técnicas de estimación de desplazamiento, se realizaron distintos ensayos.

Ensayo I

El primer ensayo realizado consiste en lo siguiente: Para un punto característico cualquiera (por ejemplo la punta de la nariz), una imagen cualquiera, se computan los diferentes algoritmos de estimación de desplazamiento en una ventana (horizontal de tamaño $1 \times W$), tomando como *jet* Modelo el *jet* extraído en el punto central de la ventana. Está claro que este ensayo recrea una situación ideal ficticia, pero igualmente nos permite independizarnos del modelo elegido, evaluando únicamente las ventajas y desventajas de cada uno de los estimadores.

Los puntos a evaluar son los siguientes:

Región de Convergencia: Es la región dentro de la cual el algoritmo logra estimar correctamente el desplazamiento buscado (centro de la ventana). Para cuantificar esta región, se consideró que el error máximo que se puede cometer para que un punto pertenezca a la región de convergencia X debe ser menor que X píxeles. La denotamos $RC_X = [a, b]$.

Error medio: Medida como la diferencia entre la estimación y el verdadero desplazamiento. Esto es calculable dado que en este ensayo conocemos el verdadero desplazamiento. Se promedia el error cometido en toda la Región de Convergencia RC_X .

Tiempo de estimación: Tiempo promedio que demora el algoritmo en estimar el desplazamiento.

En el gráfico 6.8 podemos apreciar la performance de los diferentes métodos de estimación. A continuación se presentan los resultados resumidos para el Ensayo I³.

Para facilitar la visualización se muestra los resultados separados por las categorías antes descritas. Se puede apreciar, que el predictor que tiene una región de convergencia más grande es el `DEGridSample`. Dentro de los estimadores por predicción el que mejor se desempeña es el predictor por escalas. En este gráfico se ve claramente el concepto de *Región de Convergencia*, observando que los estimadores tienen un muy buen desempeño en los píxeles cercanos al centro de la ventana.

³El tiempo se despliega de forma relativa donde una unidad es el tiempo que demora la estimación mediante `DEPredictiveStep`, $T_{DEPreStep} = 62\mu s$.

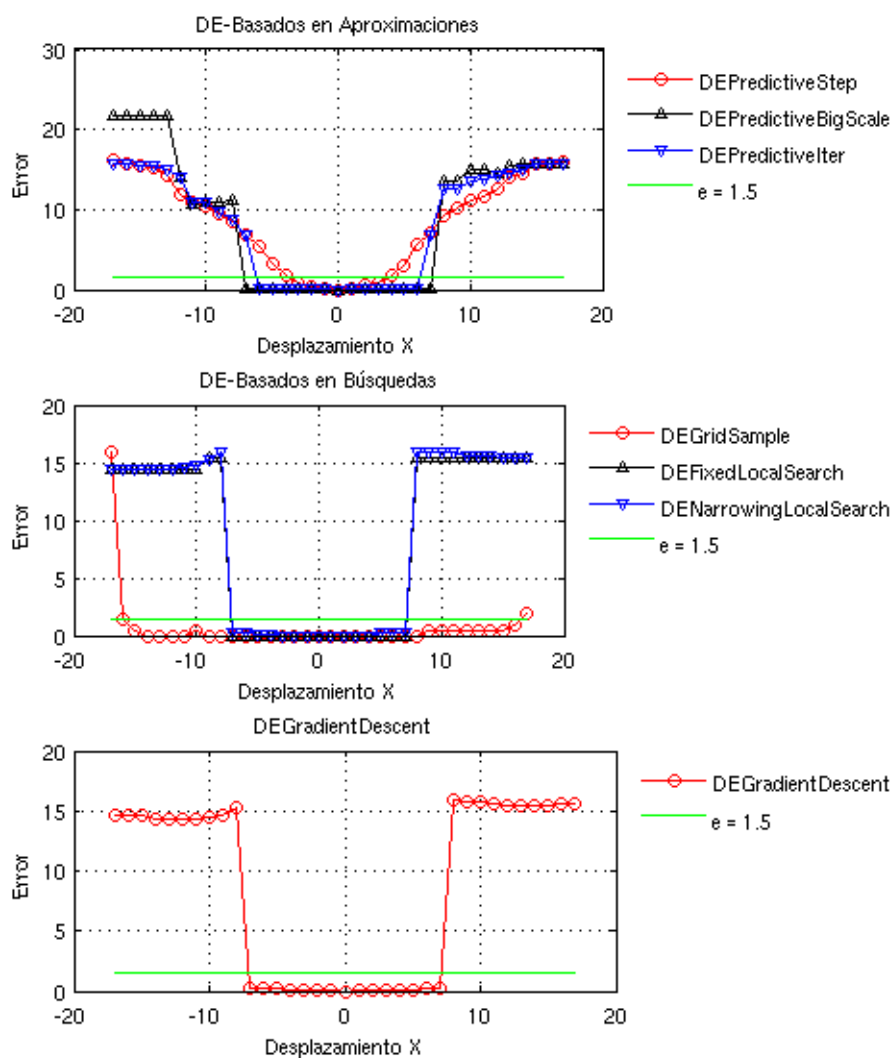


Figura 6.8: Resultados del Ensayo I. Estimación del desplazamiento para los diferentes estimadores.

<i>Ensayo-I</i>			
DE	RC-1.5 Px.	Error	Tiempo
DEPredictiveStep	[-5.5 , 6.0]	0.40 Px.	1.0
DEPredictiveBigScale	[-9.5 , 10.5]	0.33 Px.	1.3
DEPredictiveIter	[-9.5 , 10.5]	0.33 Px.	1.6
DEGridSample	[-16.0 , 16.0]	0.55 Px.	293.0
DEFixedLocalSearch	[-10.0 , 10.0]	0.34 Px.	4.9
DENarrowingLocalSearch	[-10.0 , 10.0]	0.34 Px.	5.1
DEGradientDescent	[-10.0 , 10.5]	0.38 Px.	7.4

Como se puede apreciar en la tabla anterior, el algoritmo que mejor computa el desplazamiento es la estimación por muestreo de grilla. Sin embargo el tiempo que demora en ejecutarla es inadmisiblemente superior al resto desde el punto de vista de una aplicación en tiempo real. El error cometido en promedio por este algoritmo, el cual resulta ser bastante más alto que el resto, se debe a que la grilla utilizada para la evaluación considera puntos cada 0.5 Px. El resto de las estimaciones tiene un desempeño similar en cuanto a Región de Convergencia, siendo las estimaciones basadas en aproximaciones (`DEPredictiveXXX`) las más rápidas. También se puede apreciar que las estimaciones basadas en búsquedas locales (`DEFixedLocalSearch` y `DENarrowingLocalSearch`) no obtienen mejores resultados que las basadas en Predicciones. La estimación mediante descenso por gradiente, no logra mejorar los resultados de las técnicas basadas en aproximaciones. Por último cabe mencionar que dentro de las técnicas basadas en aproximaciones la que mejor se desempeña es la `DEPredictiveBigScale`, que logra el mismo desempeño que la iterativa pero en un menor tiempo.

Es importante puntualizar que si bien los tiempos de ejecución de los algoritmos son muy bajos (salvo el `DEGridSample`, todos rondan las decenas de μs) esta etapa de estimación de similitud se va a realizar una gran cantidad de veces en el algoritmo, por lo que es importante seleccionar un estimador rápido.

Ensayo II

El Ensayo II es una generalización del Ensayo I. Se calculan los mismos indicadores de performance que en el caso anterior, mostrando los resultados promediados para 15 puntos característicos interiores a la cara. Al igual que en el caso anterior, los puntos característicos son seleccionados de manera manual. Mediante este ensayo se busca independizarse del punto particular presente en el Ensayo I.

<i>Ensayo-II</i>			
DE	RC-1.5 Px.	Error	Tiempo
DEPredictiveStep	[-4.0 , 4.0]	0.41 Px.	1.0
DEPredictiveBigScale	[-7.0 , 7.0]	0.27 Px.	1.2
DEPredictiveIter	[-7.0 , 7.0]	0.50 Px.	1.4
DEGridSample	[-15.0 , 15.0]	0.59 Px.	285.4
DEFixedLocalSearch	[-6.5 , 5.5]	0.56 Px.	4.5
DENarrowingLocalSearch	[-6.5 , 5.5]	0.28 Px.	5.6
DEGradientDescent	[-7.0 , 6.0]	0.38 Px.	18.3

En esta ocasión vemos que se obtienen resultados similares al Ensayo I. Quizás la diferencia principal con el primer caso esté en el aumento conside-

table del tiempo de ejecución del algoritmo de descenso por gradiente. En la figura 6.9 se puede observar una comparación de los tiempos de estimación promedio para cada algoritmo. Los tiempos se muestran en escala logarítmica respecto al tiempo de ejecución del `DEPredictiveStep`. Se observa que cualquiera de los algoritmos de predicción tiene tiempos bajos. Al igual que en el Ensayo anterior el algoritmo con mejor desempeño Error-Tiempo es el `DEPredictiveBigScale`.

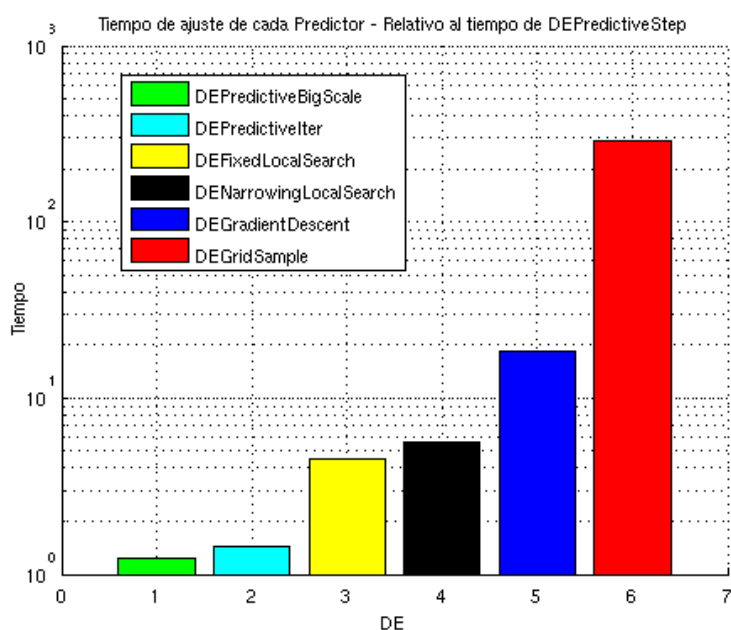


Figura 6.9: Resultados del Ensayo II. Tiempos de estimación promedios para cada algoritmo.

Ensayo III

El Ensayo III es un ensayo visual. Consiste en calcular los diferentes estimadores en una ventana rectangular con centro en uno de los puntos característicos. En la figura 6.10 se puede observar los diferentes resultados de cada una de las estimaciones tomando como centro de la ventana la punta de la nariz. En este gráfico se ve claramente la noción de región de convergencia llevada a dos dimensiones. Nuevamente el algoritmo que obtiene mejor región de convergencia es el muestreo por grilla. Los algoritmos de búsqueda obtienen resultados aceptables, mientras que los basados en aproximaciones por escalas e iterativo tienen un buen desempeño. Finalmente el algoritmo básico, el estimador por paso predictivo tiene un desempeño bastante pobre.

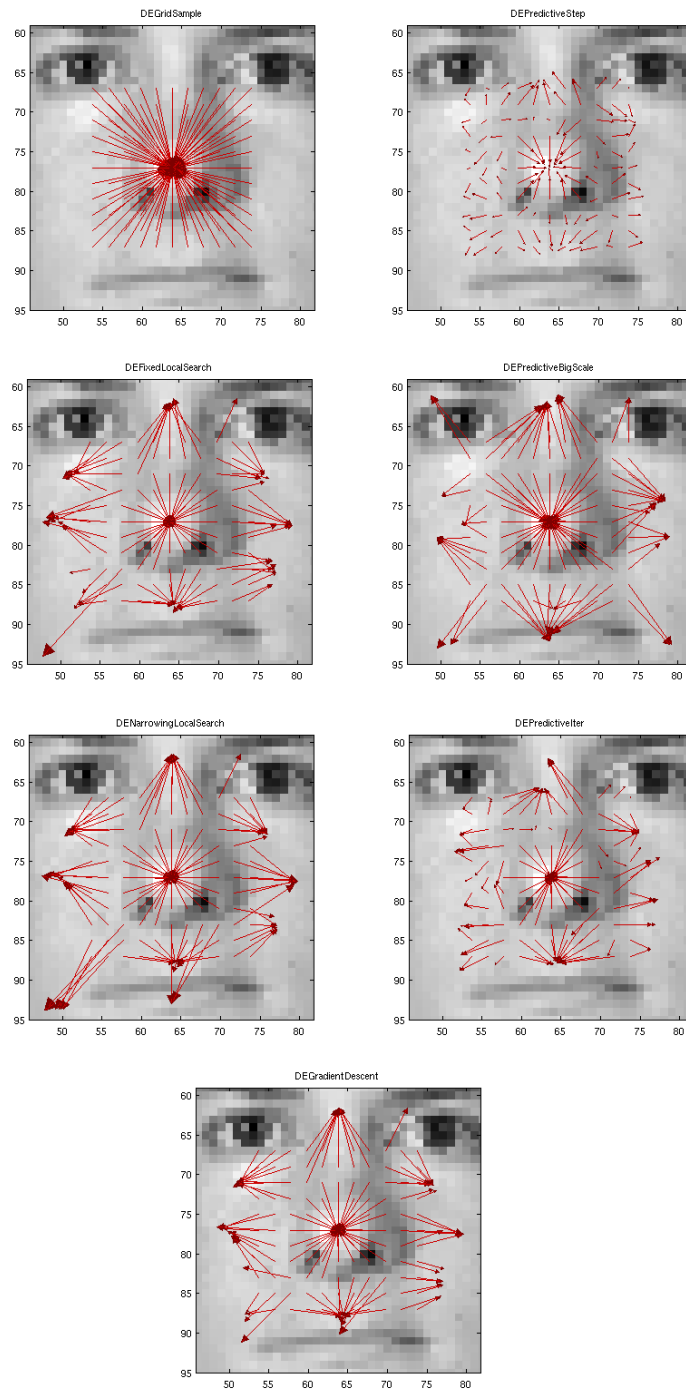


Figura 6.10: Resultados del Ensayo III. De izquierda a derecha de arriba a abajo: `DEGridSample`, `DEPredictiveStep`, `DEFixedLocalSearch`, `DEPredictiveBigScale`, `DENarrowingLocalSearch`, `DEPredictiveIter`, `DEGradientDescent`.

6.6. Generalización de la Localización de Puntos

Dentro de la amplia gama de dificultades que encuentra el reconocimiento de caras aparece la variabilidad en la pose del individuo a reconocer. La cara del sujeto puede adoptar una pose frontal o presentar diversos ángulos de rotación. Si bien el alcance de este proyecto incluye únicamente el caso controlado de imágenes frontales, se investiga en este aspecto buscando dar robustez al sistema ante caras con rotaciones mayores a 25° .

Como fuese detallado en secciones anteriores, la localización de puntos principales del algoritmo EBGM utiliza un modelo estadístico de la cara. En teoría, si este modelo es lo suficientemente variado e incluye imágenes con distintas poses, es posible ajustar correctamente puntos característicos en caras rotadas. Con la implementación utilizada se comprueba que agregando al modelo imágenes en distintas poses no se logra una buena localización de puntos.

Dado que la localización de puntos no alcanza una buena performance con la utilización de un único modelo estadístico, se propone utilizar distintos modelos dependiendo de la pose del individuo. Se generan tres modelos diferentes:

modelo frontal Formado por imágenes frontales e imágenes levemente rotadas.

modelo perfil derecho Formado por imágenes de perfil derecho con distintos ángulos de rotación.

modelo perfil izquierdo Formado por imágenes de perfil izquierdo con distintos ángulos de rotación

Se desprende de esta propuesta la necesidad de detectar la pose del individuo para luego aplicar el modelo correspondiente en la localización de puntos principales. A continuación se describen brevemente los métodos considerados.

6.6.1. Detección de la Pose

Ensayo I: Grafos asociados a poses

La primera idea propuesta para la detección de pose es utilizar tres grafos de puntos principales marcados manualmente; uno frontal, uno de perfil derecho y otro de perfil izquierdo. Dada una imagen nueva, los pasos a seguir para detectar la pose son:

1. Tomar una pose seleccionando uno de los grafos marcados manualmente.
2. Calcular los wavelets de Gabor de la imagen nueva en las coordenadas de los nodos dados por el grafo elegido.
3. Calcular la distancia entre los wavelets hallados para la imagen de entrada y los wavelets almacenados para el grafo marcado manualmente.
4. Luego de completados los pasos anteriores para los tres grafos marcados manualmente, seleccionar como pose de la cara la asociada al grafo que da la mínima distancia.

Se realizan algunas pruebas y se encuentra que los resultados obtenidos no son buenos. Si se grafican los grafos marcados manualmente sobre imágenes de testeo se encuentra que dichos puntos quedan ubicados en lugares no representativos de la cara. Esto implica que se comparen características de puntos principales de una cara rotada (los marcados manualmente) con características extraídas en puntos no significativos de la imagen de entrada.

Los grafos son marcados sobre caras de cierto tamaño y pose, con lo cual al fijar dichos puntos en una imagen nueva éstos no se corresponden con puntos característicos de la misma. En un principio se esperaba que el grafo de la pose correcta ubicado sobre la imagen tuviese una similitud mayor que el resto de los grafos. Debido a lo explicado anteriormente esto no se verifica. Resulta crítico ajustar el grafo como paso previo al cálculo de los wavelets en los nodos fijos del grafo marcado a mano.

Ensayo II: Modelos estadísticos diferenciados

Se propone a continuación un encare que incluye una etapa de ajuste de nodos. Se construyen tres modelos diferentes, cada uno de ellos conteniendo imágenes de una pose dada. El total de imágenes por modelo no puede ser alto, atendiendo al compromiso precisión - tiempo.

Se marca puntos principales en los ojos y la nariz para los modelos de prueba. Disminuir al mínimo la cantidad de puntos resulta crucial a la hora de considerar la velocidad de la implementación.

Teniendo entonces un modelo por pose, dada una imagen nueva el algoritmo realiza los siguientes pasos:

1. Tomar uno de los modelos.
2. Ajustar el modelo estadístico elegido a la imagen de entrada, lo cual resulta en una similitud asociada al ajuste.
3. Repetir los pasos anteriores para cada modelo.
4. Elegir como pose la asociada al modelo que devuelve la máxima similitud en el ajuste.

Evaluación

Para la evaluación de la implementación se utilizaron los siguientes modelos estadísticos, formados por imágenes de la base FERET marcadas en forma manual:

modelo frontal Formado por 5 imágenes frontales. Puntos según figura 6.11.

modelo perfil derecho Formado por 7 imágenes de perfil derecho rotadas 40 grados. Puntos según figura 6.12.

modelo perfil izquierdo Formado por 7 imágenes de perfil izquierdo rotadas 40 grados. Puntos según figura 6.12.

En las pruebas realizadas se toma como conjunto de evaluación imágenes pertenecientes a la base IIE. En el apéndice C se puede encontrar el significado de las letras utilizadas para nombrar las distintas imágenes de esta base.

Se definen los tests:

- **Test I:** Modelos definidos. `DEPredictiveIter` como estimador de desplazamiento. Imágenes de evaluación: frente (a, d, g, j, u, x), perfil derecho (c, w), perfil izquierdo (b, v).
- **Test II:** Modelos definidos. `DEPredictiveIter` como estimador de desplazamiento. Imágenes de evaluación: frente (a, d, g, j, u, x), perfil derecho (c, e, i, w), perfil izquierdo (b, f, h, v).

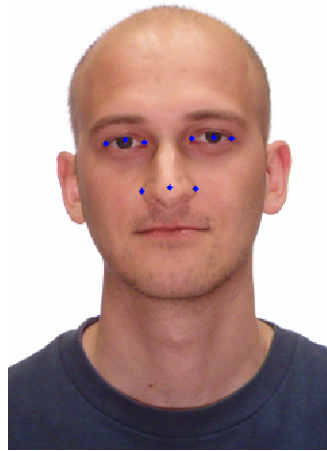


Figura 6.11: Ejemplo de puntos utilizados en el Ensayo II.



Figura 6.12: Ejemplo de puntos utilizados en el Ensayo II.

Se define una *clasificación correcta* cuando el algoritmo postula la pose correcta para el individuo en la imagen. En la tabla siguiente se aprecia los porcentajes de *clasificación correcta* obtenidos.

Test I	Normalización	
	Sin máscara elíptica	Con máscara elíptica
Frontal	82 %	82 %
Perfil Derecho	43 %	43 %
Perfil Izquierdo	53 %	51 %

Se encuentran nuevamente resultados no satisfactorios. No es suficiente el ajuste de los tres modelos de pose para la correcta clasificación de las mismas.

Se observa que la clasificación tiene mejor performance para la pose frontal. Recorriendo las distintas imágenes utilizadas para el testeo se encuentra que la pose frontal es bastante uniforme, es decir, las poses frontales de los distintos individuos no varían mucho. Por otro lado, las poses de perfil tienen gran variabilidad. Dependiendo del individuo, la pose de perfil es una rotación de 10, 20, 40 grados respecto a la pose frontal. Esto hace que algunos perfiles se acerquen mucho a una pose frontal mientras que otros sean similares a un perfil dado.

Se concluye que no es posible clasificar las distintas poses con los tres modelos utilizados.

Ensayo III: Gradualidad en los modelos de pose

Se propone agregar gradualidad a la estimación de pose. Esto se logra trabajando con cinco modelos de pose en lugar de tres. En la propuesta anterior se utilizaba un modelo frontal y dos correspondientes a los perfiles, ahora se trabaja con un modelo frontal y cuatro correspondientes a los perfiles, incrementando de uno a otro el ángulo de rotación de la cara.

Dados cinco modelos de pose y una imagen nueva el algoritmo completa los siguientes pasos:

1. Tomar uno de los modelos.
2. Ajustar el modelo estadístico elegido a la imagen de entrada, lo cual resulta en una similitud asociada al ajuste.
3. Repetir los pasos anteriores para cada modelo.
4. Si la máxima similitud en el ajuste se alcanza para la pose frontal o las poses de perfil extremas (en las pruebas realizadas, las poses rotadas 40 grados), dicha pose es elegida.
En caso que la máxima similitud se alcance para una de las poses de transición (en las pruebas realizadas, las poses rotadas 25 grados), se comparan los valores obtenidos para la pose frontal y el perfil correspondiente al de transición elegido, el máximo de estos es seleccionado.

Evaluación

Para generar los distintos modelos de pose se procede al marcado manual de imágenes de la base de caras FERET. Por razones de tiempo se generan modelos para poses frontal y perfil izquierdo, no perfil derecho. Las evaluaciones se realizan entonces para dichas poses, considerando por resultados anteriores, que es posible inferir conclusiones para el perfil derecho a partir del perfil izquierdo.

Los modelos estadísticos utilizados son:

`modelo frontal` Formado por 6 imágenes frontales. Puntos según figura 6.13.

`modelo perfil izquierdo leve` Formado por 6 imágenes de perfil izquierdo rotadas 25 grados. Puntos según figura 6.13.

`modelo perfil izquierdo` Formado por 6 imágenes de perfil izquierdo rotadas 40 grados. Puntos según figura 6.13.

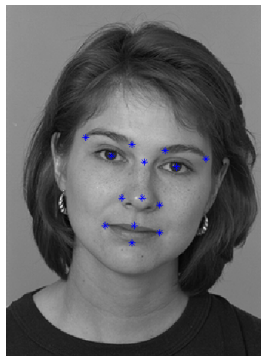


Figura 6.13: Ejemplo de puntos utilizados en el Ensayo III.

En las pruebas realizadas se toma como conjunto de evaluación imágenes pertenecientes a la base IIE.

Se define el test:

- **Test I:** Modelos definidos. Normalización sin máscara elíptica. `DEPredictiveIter` como estimador de desplazamiento. Imágenes de evaluación: frente (a, d, g, j, u, x), perfil izquierdo (b, f, h, v).

El porcentaje de *clasificación correcta* obtenido para pose frontal fue de 99% mientras que para perfil izquierdo se obtuvo 57%.

Los resultados obtenidos son buenos para las imágenes frontales aunque no lo son para las poses de perfil. Agregar gradualidad a los modelos permite clasificar correctamente más imágenes correspondientes a pose frontal. Cabe señalar el impacto que tiene en la clasificación el hecho mencionado anteriormente sobre la variabilidad de las poses de perfil. Ocurre en muchos casos que una imagen etiquetada como de perfil se acerca más a una pose frontal que a un verdadero perfil. Considerando que la detección de pose se implementa para la posterior selección del mejor modelo para el ajuste, es deseable que se asocie una pose frontal a una imagen etiquetada como perfil si realmente es más cercana a una pose frontal. En este sentido se encuentran positivos los resultados obtenidos dado que, en los pocos casos de imágenes presentando poses con perfiles marcados, se encuentra una correcta clasificación.

6.6.2. Localización de puntos considerando la pose

Utilizando la detección de pose propuesta en el Ensayo III se implementa la localización de puntos considerando la pose de la cara. Para ello se encuentra en primer lugar la pose del individuo y dependiendo de la misma se toma el modelo correspondiente y se localizan los puntos principales en la imagen de entrada.

Evaluación

Al igual que en los casos anteriores los modelos estadísticos utilizados son generados a partir de imágenes de la base FERET. Se utilizan dos sets de modelos, uno para la estimación de pose y otro para la localización de puntos principales. Los mismos son:

Set de detección de pose:

- **modelo frontal** Formado por 6 imágenes frontales. Puntos según figura 6.13.
- **modelo perfil izquierdo leve** Formado por 6 imágenes de perfil izquierdo rotadas 25 grados. Puntos según figura 6.13.
- **modelo perfil izquierdo** Formado por 6 imágenes de perfil izquierdo rotadas 40 grados. Puntos según figura 6.13.

Set de localización de puntos principales:

- **modelo frontal** Formado por 37 imágenes frontales. Puntos según figura 6.13.
- **modelo perfil izquierdo** Formado por 37 imágenes de perfil izquierdo rotadas 40 grados. Puntos según figura 6.13.

En todos los tests realizados se aplica normalización sin máscara elíptica. Se utiliza `DEPredictiveIter` como estimador de desplazamiento y en la etapa de confrontación se utiliza una medida de similitud considerando únicamente la magnitud (`fgSimMagnitude`).

Se definen los tests:

- **Test I:** Se utilizan sets de modelos antes definidos. Conjunto de referencia: 250 imágenes frontales pertenecientes a los sets `fa` y `ba` de la base FERET. Conjunto de evaluación: 250 imágenes de los sets `fb` y `bj` de la base FERET, correspondientes a las seleccionadas de los sets `fa` y `ba` para el conjunto de referencia. Por descripción sobre la base ver apéndice C.
- **Test II:** Se utilizan sets de modelos antes definidos. Conjunto de referencia: 250 imágenes frontales pertenecientes a los sets `fa` y `ba` de la base FERET. Conjunto de evaluación: 250 imágenes de los sets `bg` y `bh` de la base Feret (perfil izquierdo rotadas 25 y 40 grados).
- **Test III:** No se realiza estimación de pose, se utiliza únicamente el modelo frontal definido en el set de localización. Conjunto de referencia: 250 imágenes frontales pertenecientes a los sets `fa` y `ba` de la base FERET. Conjunto de evaluación: 250 imágenes de los sets `bg` y `bh` de la base FERET (perfil izquierdo rotadas 25 y 40 grados).
- **Test IV:** Se utilizan sets de modelos antes definidos. Conjunto de referencia: 196 imágenes pertenecientes al set `bg` (perfil izquierdo rotadas 25 grados) de la base FERET. Conjunto de evaluación: 196 imágenes pertenecientes al set `bh` (perfil izquierdo rotadas 40 grados) de la base FERET.

La elección de imágenes para el testeo se realiza de forma aleatoria dentro del total de imágenes de la base FERET, excluyendo las imágenes utilizadas en los modelos. La figura 6.14 muestra un set de imágenes con distintas tomas, asociadas a un individuo, pertenecientes a la base FERET. Las evaluaciones se realizan para las poses frontal y perfil izquierdo, considerando que por resultados anteriores, se pueden inferir datos para el perfil derecho a partir del perfil izquierdo.

Los resultados desplegados se expresan como mejoras relativas respecto al Test I.

Test II: TER = -250 %

Test III: TER = -270 %

Test IV: TER = -67 %



Figura 6.14: Ejemplo de las distintas poses utilizadas para un individuo. En orden se tiene: ba, bj, bg y bh.

6.6.3. Conclusiones

En primer lugar cabe señalar que la performance de todos los tests realizados es peor que la performance del test I. Los resultados obtenidos al clasificar caras de perfil utilizando caras frontales como referencia no son buenos. Tampoco es aceptable el resultado del test IV que implementa la clasificación de caras rotadas contra rostros de referencia levemente rotados.

Luego de este resultado poco alentador se plotean los grafos ajustados sobre varias imágenes de testeo. Se compara el ajuste con estimación de pose y el ajuste con un único modelo frontal. Se encuentra que el ajuste con estimación de pose es mucho más exacto que el ajuste con el modelo frontal. La figura 6.15 ilustra este hecho. En la misma se aprecia que la principal diferencia en la ubicación de los puntos característicos se da para el perfil rotado 40 grados. Observar que el modelo frontal no es capaz de ubicar los puntos correctamente.

Se concluye entonces que la información del grafo frontal no es suficiente para clasificar correctamente un grafo de perfil, independientemente de estar correctamente ajustado o no. Queda claro que las tomas frontales de distintos individuos son más cercanas entre sí que las tomas frontal y de perfil de un mismo individuo.

A partir de esta conclusión surge como propuesta de trabajo futuro la implementación de una técnica de compensación de *jets* esbozada en [67]. La idea es calcular los *jets* en el grafo ajustado al perfil y luego, estimando el ángulo de rotación de la cara, compensar los valores de los mismos hasta llevarlos a su equivalente en una pose frontal. Dicha implementación excede este proyecto pero se puede encontrar variada bibliografía al respecto. La combinación del correcto ajuste de perfil mediante la estimación de pose y la compensación de *jets* son un objetivo prometedor.

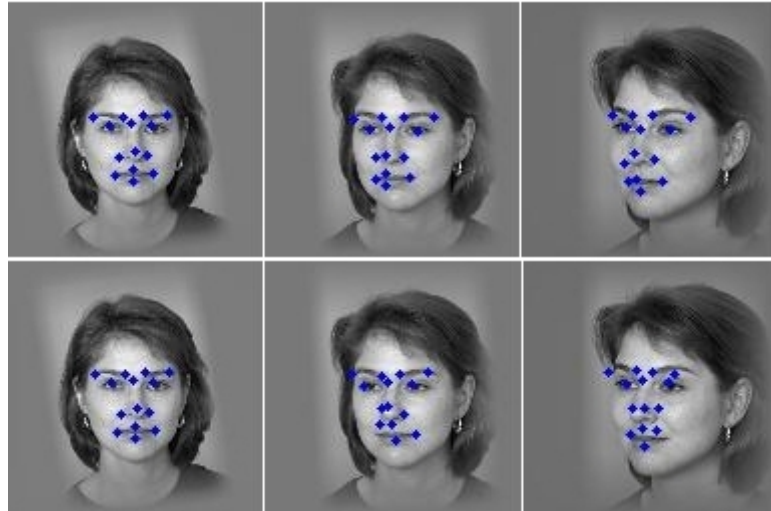


Figura 6.15: La fila superior muestra el ajuste de grafo con estimación de pose, mientras que la fila inferior muestra el ajuste con un único modelo frontal.

Capítulo 7

Medidas de Similitud

7.1. Introducción

En este capítulo se introducen las diferentes medidas de similitud utilizadas para la comparación de dos caras a los efectos de tomar la decisión de validar una identidad o rechazarla. Algunas de estas medidas de similitud derivan del sistema CSU, las cuales fueron presentadas en el capítulo de descriptores de caras, y otras surgen de estudios realizados a lo largo del proyecto.

Luego de tener ajustados los puntos principales elegidos en la cara, se define una estructura de grafo similar a la descrita en 6.3.1 para representar la misma. Cada imagen de una cara tiene su correspondiente estructura de grafo asociada. Dicho grafo contiene los *jets* correspondientes a los puntos principales incluyendo además puntos interpolados a partir de los anteriores. En regiones como las mejillas o la frente donde no hay puntos definidos, se extraen las características en el punto medio de cada arista definida en el grafo. En el sistema CSU se trabaja con un modelo de 25 puntos principales y otros 55 puntos interpolados que corresponden a cada una de las aristas definidas en el grafo. Se puede ver en la figura 7.1 un ejemplo donde se aprecian tanto los puntos principales como las aristas del grafo definidas por el modelo de cara utilizado.

7.2. Medidas de Similitud

La confrontación de dos imágenes, luego de hallados los grafos correspondientes, se reduce al cálculo de una medida de similitud entre ambos. La similitud puede ser calculada como función de los *jets*, de las ubicaciones de los puntos principales o de ambas. Se descarta en este trabajo el uso de la información de la ubicación de los puntos desalentados por los resultados obtenidos en [11].

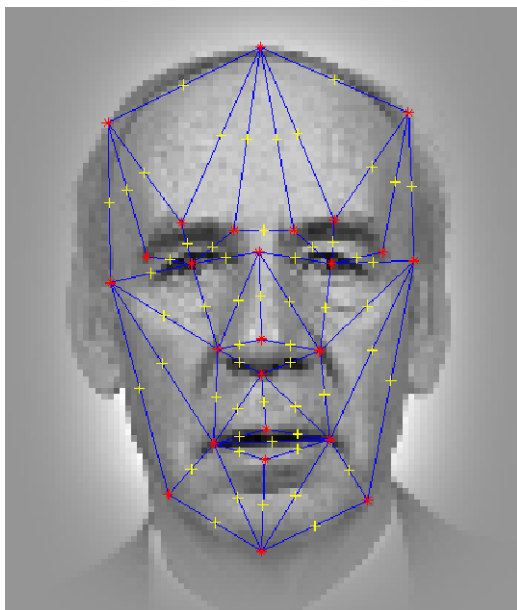


Figura 7.1: Estructura de grafo definida como modelo de cara.

7.2.1. Similitudes Promedio

Las similitudes promedio, teniendo definida una similitud entre *jets*, son aquellas que se calculan como el valor medio de la similitud en cada uno de los *jets*. La expresión de las mismas es la siguiente:

$$L(G, G') = \frac{1}{n} \sum_{i=1}^n S_x(J_i, J'_i)$$

donde n es la cantidad de *jets* definidos en el grafo, S_x es una medida de similitud entre *jets*, J_i y J'_i los i -ésimos *jets* de los grafos G y G' respectivamente.

Las distintas similitudes promedio corresponden a las distintas medidas de similitud entre *jets* utilizadas, las cuales se listan a continuación:

Magnitud Similitud que considera solo la magnitud (`fgSimMagnitude`).

Fase Similitud que considera magnitud y fase, sin corrección de la posición (`fgSimPhase`).

Fase con BS Similitud que considera magnitud y fase, con corrección de la posición utilizando el estimador de desplazamiento por paso predictivo por escalas (`fgSimPhasePredictiveBigScale`).

Fase con PS Similitud que considera magnitud y fase, con corrección de la posición utilizando el estimador de desplazamiento por paso predictivo

(fgSimPhasePredictiveStep).

Fase con PI Similitud que considera magnitud y fase, con corrección de la posición utilizando el estimador de desplazamiento por predicción iterativa (fgSimPhasePredictiveIter).

Las expresiones para dichas similitudes son las mismas que las definidas en el capítulo de descriptores. En los tres últimos casos, donde se realiza corrección de la posición de los puntos, la diferencia está en el estimador de desplazamiento que se utiliza.

7.2.2. Similitudes Ponderadas

Las similitudes ponderadas, como su nombre lo indica, se generan a partir de una combinación lineal de las similitudes en cada uno de los *jets*. Las similitudes promedio son un caso particular de estas similitudes, donde el peso es el mismo para cada *jet*. La idea es dar un mayor peso a aquellos *jets* que más aporten para clasificar a las personas.

En esta línea, varias opciones fueron estudiadas. En primer lugar se hace un estudio de la relevancia de cada punto, que verifica el hecho de que algunos son mejores que otros para la discriminación. Luego se introduce medidas de similitud con pesos 0 o 1, lo que corresponde a tener en cuenta o no cada punto para la confrontación. Por último se estudia cómo elegir los pesos como valores reales en el intervalo [0,1]. En todos los casos estudiados se utilizó como similitud entre *jets* la similitud de Magnitud.

A. Estudio de la relevancia de cada punto principal

El objetivo de este estudio es investigar la relevancia de cada uno de los puntos principales localizados en la cara para la identificación de los individuos. En este caso se analiza la performance del sistema de verificación tomando la información obtenida en un único punto, realizando esto para todos los puntos localizados. Se calculan las ROC y se comparan los valores de EER normalizados¹ correspondientes a cada punto de forma de observar cuáles discriminan mejor o peor a las personas de la base.

A.1. Ensayo realizado para puntos marcados manualmente La prueba se realiza para puntos localizados manualmente para evitar el error introducido en el ajuste automático de los mismos. En la figura 7.2 se indican los puntos seleccionados.

¹Los EER normalizados corresponden al EER dividido entre el EER mínimo que corresponde al punto que mejor discrimina.

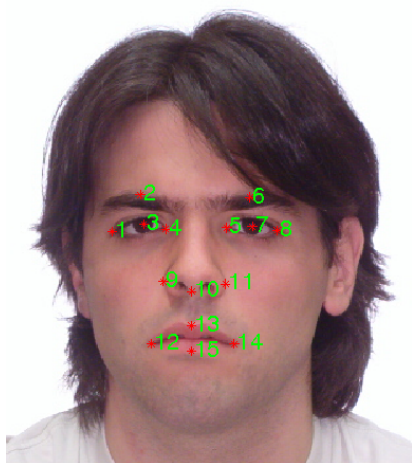


Figura 7.2: Puntos marcados manualmente.

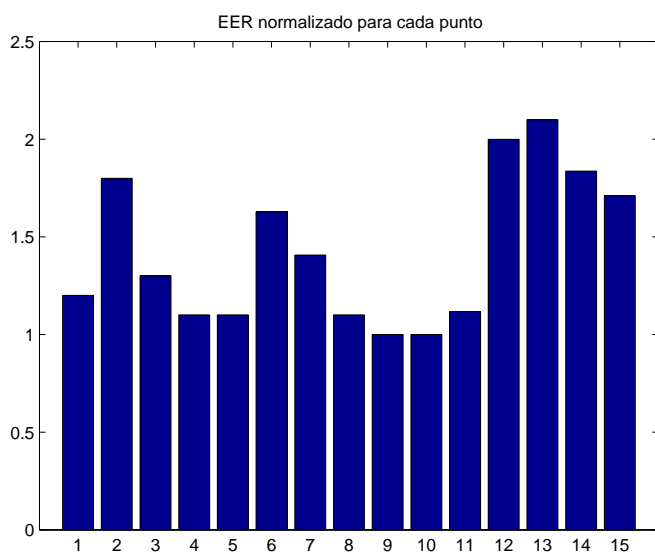


Figura 7.3: EER para cada punto de la cara.

Se trabaja con una base de 47 individuos con 2 imágenes de entrada para cada uno y 4 de referencia. En la figura 7.3 se muestran los resultados obtenidos.

Como se observa, los peores resultados son para los puntos $\{12,13,14,15\}$ que corresponden a la boca y $\{2,6\}$ que están en las cejas. Después tenemos una segunda franja con los puntos $\{3,7\}$ que están en el centro de cada ojo. Los mejores resultados fueron para los puntos $\{9,11\}$ en las esquinas de la

nariz, $\{1,4,5,8\}$ que corresponden a las esquinas de los ojos y $\{10\}$ el vértice inferior de la nariz.

A.2. Otros ensayos realizados En el caso en que se extiende este estudio agregando el ajuste automático de puntos, el análisis es más difícil porque los resultados no son tan claros y se debe considerar el error implícito del ajuste automático de los puntos. De todas formas se observan resultados interesantes acerca de qué puntos debemos tener en cuenta al utilizar distintos tipos de imágenes.

En particular se verifica que los puntos interpolados tienen relevancia comparable a la de los puntos principales y en general su desempeño es similar al de los puntos principales que se encuentran en zonas cercanas de la cara. Además se observa que según la base de imágenes con la que se hace la prueba, pueden ser de relevancia algunas zonas u otras, debido a que las zonas invariantes cambian según el tipo de imagen. Para que quede clara la idea se puede pensar en un caso donde las parejas de imágenes para un mismo individuo sean tomas con distinta expresión facial donde las variaciones más grandes se dan desde los ojos hacia abajo y en el otro caso se tenga imágenes con otro peinado o la presencia de lentes donde los cambios son desde los ojos hacia arriba.

Conclusiones

El análisis de relevancia para el caso de puntos marcados manualmente indica que existen puntos de mejor performance individual que otros. En este caso los mismos corresponden a la zona central de la cara, en la región donde se ubican los ojos y la nariz. Los puntos de la boca no arrojan buenos resultados, lo cual se explica por el hecho de que esta región es la más elástica de la cara. Esto implica una gran variación para imágenes de la misma persona y se refleja en un bajo poder de discriminación entre los individuos. Infiere en este hecho las características particulares de la base de imágenes utilizada para el ensayo, donde todas las tomas son de frente, con expresión neutra e iluminación controlada. Es muy difícil generalizar la idea de cuáles son los puntos más importantes para discriminar a las personas. Cabe destacar la importancia de los puntos interpolados que presentan resultados similares al de los puntos principales.

Un comentario final que se debe tener en cuenta, es el hecho de que este análisis de relevancia se realiza para cada punto sin tener en cuenta la información obtenida en los demás puntos. Para poder utilizar esto como argumento y decidir cuáles puntos mantener o eliminar debe ser válida la hipótesis de independencia entre los mismos. Este hecho no se cumple como veremos en el siguiente ítem. Hay dos razones principales que explican es-

to. La extracción de características considera entornos que se superponen entre sí por lo que hay información que se solapa. Además la cara tiene en general cierta simetría, lo que hace que puntos simétricos sean fuertemente dependientes entre sí.

A.3. Algoritmos de selección de características Se utiliza otro método para estudiar la relevancia de cada punto que consiste en aplicar algoritmos de selección de características, similar a lo que se realiza en [68]. Los tres métodos que se implementan se listan a continuación:

BIF² - Se combinan las N características de mejor desempeño individual.

SFS³ - Se agrega en cada paso la característica que sumada al subconjunto ya seleccionado obtiene el mejor desempeño.

SFFS⁴ - Es una mejora de SFS que en cada paso además de agregar la que más mejora, elimina aquella que al quitarla mejora la performance.

En nuestro caso cada característica utilizada corresponde a la información asociada a un determinado punto, es decir un *jet*.

Para el caso del BIF se debe elegir la cantidad de puntos con la que trabajar, por lo que se realiza las pruebas para N entre 1 y el total de puntos del modelo. En este caso se determina los puntos que mejor discriminan a los individuos. Se considera cada punto en forma separada como se hizo anteriormente y luego se seleccionan los N mejores para medir la performance considerando la información que aportan solamente los puntos seleccionados. Se realiza la prueba para los datos correspondientes a los 15 puntos localizados manualmente y utilizando la base IIE. El valor óptimo de N para este ensayo es 8.

Para los algoritmos SFS y SFFS los resultados son idénticos. Esto es debido a que en los ensayos nunca se observa que en las iteraciones alguno de los puntos se quite para depurar el conjunto de características. Se trabaja con los mismos conjuntos de datos que anteriormente. A continuación se listan los resultados obtenidos en cada caso.

BIF - Puntos seleccionados: {9,10,4,5,8,11,1,3}

SFFS - Puntos seleccionados: {4,10,5,2,11}

SFFS - Puntos seleccionados: {4,10,5,2,11}

²Best Individual Features

³Sequential Forward Selection

⁴Sequential Floating Forward Selection

Como se observa en el conjunto seleccionado con SFFS, los puntos no son los mismos que al utilizar BIF. Se muestra entonces que la información que contiene un *jet* no es independiente de la de los demás, puesto que sino los conjuntos deberían coincidir.

B. Variantes con pesos 0-1

El primer paso hacia una medida de similitud con pesos es utilizar valores 0 o 1 como pesos posibles lo que corresponde simplemente a utilizar o no la información obtenida en cada uno de los puntos. A continuación se detallan las variantes estudiadas. Para evaluar las distintas opciones es necesario determinar los pesos con un conjunto de entrenamiento por lo que para la validación debemos usar otros datos.

B.1. N puntos fijos Siguiendo la línea de los algoritmos de selección de características, la opción más sencilla para hacer la confrontación es tomar en cuenta solamente los puntos hallados con dichos algoritmos y promediar simplemente la similitud para estos *jets*.

Los resultados obtenidos para esta opción no fueron buenos, en gran medida debido a los argumentos manejados anteriormente de que según sea el caso, los puntos de mayor utilidad pueden ser unos u otros, pero es difícil obtener un conjunto de puntos que sirva para todas las situaciones. Además, en este caso no se solucionan problemas como el de las oclusiones debido a que los puntos seleccionados están fijos luego de ser determinados por el algoritmo. Lo anterior implica que si uno de estos puntos queda oculto por alguna razón, el sistema no va a funcionar.

B.2. N más semejantes Otro intento en busca de una medida de similitud que pese distinto la semejanza de cada *jet* con pesos 0 o 1 consiste en promediar solamente las N similitudes mayores de todas las obtenidas, descartando las restantes. A diferencia de la similitud anterior donde los puntos utilizados eran fijos, en este caso la información que se utiliza es la de los N *jets* más semejantes por lo que los puntos seleccionados varían para cada imagen en particular. Se analiza esta opción para 15 puntos marcados manualmente para determinar el N óptimo variando entre 1 y 15. Se obtiene un valor de N entre 7 y 9.

Una utilidad práctica de esta variante evaluada con la base IIE es el caso de imágenes con oclusiones, donde no todos los puntos del modelo están a la vista en la cara. Si se toma solamente los N más cercanos probablemente se descarte los que están ocultos. En la prueba realizada se trabaja con las imágenes con bufanda y las de lentes negros en modo verificación. Se espera que esta variante sea útil para solucionar el problema de las oclusiones. Los

resultados se muestran en el Capítulo 9.

C. Medida de similitud con pesos

La idea es trabajar con una medida de similitud que pese de manera diferente cada uno de los puntos de la cara, por lo que la medida de similitud entre dos imágenes queda de la siguiente manera:

$$Sim = \sum_i \beta_i SimJet_i$$

En todos los casos estudiados es necesaria una etapa de entrenamiento para hallar los valores de los pesos mediante la minimización de una cierta función de costo, por lo que se trabaja nuevamente con dos conjuntos de datos: uno para entrenamiento y otro para evaluación.

C.1. Pesos según el desempeño individual En [69] se describe un algoritmo para determinar los pesos a partir de ciertas condiciones impuestas a priori. La hipótesis más relevante es la de independencia que implica que para determinar el peso para el nodo k sólo debo mirar la performance obtenida utilizando únicamente la información de este nodo. Además se definen ciertas funciones paramétricas que llevan de la información hallada en cada jet a los valores de los pesos correspondientes. Esto hace que la optimización se reduzca a un problema de menos variables, puesto que la cantidad de parámetros de las funciones definidas es mucho menor que la cantidad de puntos.

En nuestro caso, siguiendo la idea propuesta en [69], se eligen los pesos proporcionales a la relevancia que tiene cada uno de los puntos para discriminar a las personas. Luego se aplica una función exponencial y lo que se determina es el exponente óptimo, por lo que el problema pasa a ser de una sola variable. El peso para el nodo k queda:

$$\beta_k = f(EEER_k) = C(1 - EEER_k)^\gamma$$

donde C es una constante para que los β_k queden normalizados.

Esta opción no descarta la información de ninguno de los puntos, por lo que es mejor que simplemente descartar algunos puntos. De todas maneras subsiste el problema de ajustarse a las imágenes con las que fueron entrenadas por lo que la utilidad está determinada por la aplicación posterior.

C.2. LDA y Redes neuronales Otras posibilidades manejadas son el uso de herramientas clásicas de reconocimiento de patrones para determinar los pesos de una función discriminante para un problema de dos clases. Se consideran las similitudes entre parejas de imágenes como patrones. En este

caso un vector cuya dimensión es la cantidad de puntos, donde la componente i -ésima corresponde a la similitud entre los *jets* i de cada imagen. De este modo, las clases a diferenciar son solamente dos: si las imágenes son de la misma persona serán de la clase A y de lo contrario serán de la clase B.

Durante este proyecto no se aborda el estudio de estas técnicas. Las mismas pueden servir para aprender la estadística de las similitudes en cada *jet* y de esta forma obtener mejores clasificadores para la discriminación de las personas. De todas formas se mantiene el hecho de que el desempeño está sujeto a que el entrenamiento logre captar la estadística que se tendrá en la aplicación real.

7.2.3. Alternativa aplicando LDA

En este caso se analiza el problema utilizando LDA, orientado a la identificación. Una vez que se extraen los *jets* lo que se tiene almacenado en cada grafo es un vector de características correspondiente a una clase particular, que representa la identidad de la persona de la cual fue extraída la información. Es así que surge el interés de aplicar una técnica de aprendizaje supervisado para clasificar datos que ya ha sido estudiada en este proyecto, conocida como LDA, Linear Discriminant Analysis ⁵. Este algoritmo se describe en el capítulo 3 y se profundiza en el capítulo 5 pero desde la perspectiva del uso de un algoritmo holístico de Reconocimiento de Caras. En esta parte del proyecto se lo estudia como herramienta de selección de características para la confrontación.

Un grafo en general está compuesto por N nodos, cada nodo está compuesto por G coeficientes complejos, resultado del filtrado de la imagen con un banco de filtros de Gabor. Por ejemplo si el banco de filtros comprende 5 orientaciones y 8 frecuencias, existen entonces 40 coeficientes complejos para cada nodo, que surgen de estas orientaciones y frecuencias. Por lo tanto, por grafo tendremos $N \times G$ coeficientes complejos, que en el ejemplo si agregamos que el número de nodos es 80, se tiene por cada grafo un vector de coeficientes complejos de dimensión 3200.

De lo anterior surge el interés de aplicar un algoritmo de extracción y selección de características directamente a la información que utiliza el algoritmo EBGm para representar las identidades. Lo que sigue es una breve introducción teórica del algoritmo LDA y sus variantes utilizadas, luego se explica los tests realizados y sus resultados y finalmente se extrae conclusiones del trabajo realizado.

⁵Existe una vasta bibliografía de este algoritmo incluso aplicado al problema del reconocimiento de caras

A. Motivación del algoritmo LDA

Se presenta una breve introducción a la idea que encierra el algoritmo utilizado para la extracción de características una vez que tenemos la representación de la cara dada por un *Face Bunch Graph*. Existen varias implementaciones de LDA, entre ellas se encuentran las dos utilizadas en esta etapa del proyecto, a saber: FISHER-LDA y *Direct Fractional-Step LDA* (DF-LDA).

B. Fisher LDA

Esta es una implementación clásica, que ha generado buenos resultados en el problema de la clasificación. Para obtener una idea teórica acerca de este algoritmo se puede revisar la sección 2.3 del capítulo 5. A continuación se retoman los conceptos necesarios para desarrollar las restantes variantes LDA.

Se definen las matrices de dispersión inter e intra clase S_B y S_W de la siguiente manera:

$$S_B = \sum_c N_c (\mu_c - \mu)(\mu_c - \mu)^T \quad (7.1)$$

$$S_W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T \quad (7.2)$$

siendo x_i los patrones etiquetados en las distintas clases, μ_c la media de cada clase, μ la media de todos los datos, N_c la cantidad de patrones de la clase c .

En esta implementación en que se trabaja con patrones del orden de miles de características se aplica primero una etapa de reducción de la dimensión de los datos mediante PCA. Luego, la solución del problema viene dada por maximizar la función objetivo:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \quad (7.3)$$

Con estas ideas de FISHER-LDA presentes mostramos las dos alternativas propuestas de LDA.

C. Direct Fractional-Step LDA

Las variantes que se incluyen aquí son el trabajo con D-LDA (Direct LDA) y con F-LDA (Fractional-Step LDA) [70]:

D-LDA Las implementaciones del tipo FISHER-LDA tienen que lidiar con el problema de "Small Sample Size" (SSS) que es el que se presenta cuando la dimensión de los patrones de entrada es mucho mayor que la cantidad de patrones. Esto provoca que las matrices de dispersión calculadas en el algoritmo degeneren o sean mal condicionadas. Una solución tradicional al problema de SSS ha sido la incorporación de PCA como paso de preprocesamiento⁶ de forma de reducir la dimensión de los datos descartando el espacio nulo de la matriz S_W . Sin embargo se ha demostrado [71] que al descartar el espacio nulo (dado que al proyectar sobre el espacio PCA se está sobre un subespacio generado por vectores propios con valores propios no nulos) se pierde información de discriminación entre clases relevante. Para prevenir esto es que se han presentado métodos Direct LDA en los cuales la información se procesa directamente en el espacio de alta dimensionalidad sin perder información discriminante.

Por lo tanto, la principal premisa de los métodos D-LDA que intentan resolver el problema SSS sin el paso PCA es que el espacio nulo de la matriz S_W presenta información significativa de discriminación entre clases si la proyección de S_B en esa dirección no es nula. A su vez asume que no se perderá información discriminatoria si se descarta el espacio nulo de S_B . Asumiendo que \mathcal{A} y \mathcal{B} representan el espacio nulo de S_B y S_W respectivamente, mientras que $\mathcal{A}' = \mathcal{R}^N - \mathcal{A}$ y $\mathcal{B}' = \mathcal{R}^N - \mathcal{B}$ son los espacios complementarios de \mathcal{A} y \mathcal{B} respectivamente, el subespacio óptimo pensado por D-LDA es el espacio intersección $(\mathcal{A}' \cap \mathcal{B})$. Para buscar la solución de 7.3 se debe diagonalizar la matriz S_B . El rango de S_B está determinado por $d = \text{ran}(S_B) = \min(N, C - 1)$, siendo N la dimensión de los datos de entrada y C el número de clases. Luego, \mathcal{A}' puede ser calculado resolviendo los vectores propios de una matriz $d \times d$. Finalmente $(\mathcal{A}' \cap \mathcal{B})$ se puede obtener como el espacio nulo de la proyección de S_W sobre \mathcal{A}' , matriz de dimensión $d \times d$.

El algoritmo que se utiliza aquí primero obtiene $(\mathcal{A}' \cap \mathcal{B})$ con D-LDA, que usualmente es de baja dimensión y luego aplica F-LDA en ese espacio de baja dimensión para derivar las características discriminantes óptimas del subespacio de intersección.

F-LDA El proceso de maximización que implica la resolución de la ecuación 7.3 no tiene conexión directa con el error de clasificación. Dado que el error de clasificación es un criterio de performance de los sistemas de reconocimiento las versiones F-LDA toman medidas directas sobre esto. Una solución es introducir funciones de clasificación con pesos en LDA, para pe-

⁶Esto no implica que PCA en sí mismo resuelva el problema de SSS sino que la resolución viene dada debido a la reducción drástica de la dimensionalidad obtenida.

nalizar aquellas clases que se encuentran juntas en el espacio de entrada del algoritmo y pueden llevar a una mala clasificación en el espacio de salida. Se introduce la matriz de dispersión interclases con pesos:

$$\hat{S}_B = \sum_{i=1}^C \phi_i \phi_i^T \quad (7.4)$$

en donde:

$$\phi_i = (L_i/L)^{\frac{1}{2}} \sum_{j=1}^C w(d_{ij})^{\frac{1}{2}} (\bar{\mu}_i - \bar{\mu}_j) \quad (7.5)$$

siendo $\bar{\mu}_i$ la media de la clase $\bar{\mathbf{Z}}_i$, L_i la cantidad de elementos en $\bar{\mathbf{Z}}_i$ y $d_{ij} = \|\bar{\mu}_i - \bar{\mu}_j\|$ la norma euclídea de la distancia entre las medias de las clases i y j . La función de pesos $w(d_{ij})$ es una función monótona decreciente con la distancia d_{ij} con la condición que la velocidad de decrecimiento de los pesos sea mayor que la velocidad con que crece la distancia euclídea entre las medias de las clases i y j . En base a esta matriz de dispersión con pesos se define un nuevo criterio de maximización que reemplaza a 7.3 y se reformula el algoritmo LDA.

Las ideas anteriores se pueden consultar en [72].

D. Evaluación y Resultados

En este caso se realizan pruebas sobre la base FERET.

Hay ciertas características particulares que se debe puntualizar:

- Se trabaja con los módulos de los coeficientes complejos de los grafos de las imágenes como los coeficientes característicos a la entrada del algoritmo LDA.
- Para la obtención del *score* entre dos vectores de características en el espacio LDA se utiliza un criterio de similitud particular. En las pruebas anteriores se utiliza la similitud que considera la magnitud de los *jets*. En este caso la comparación entre vectores de características se implementa computando la correlación de vectores y normalizando tal correlación entre 0 y 1. Por lo tanto el score de similitud conserva su espacio de salida.
- Se utilizan cerca de doscientas imágenes de la lista DUPI de la base FERET para poder enrollar en la base del sistema dos vectores de características por persona. Esto es necesario porque el algoritmo FISHER-LDA requiere para su etapa de entrenamiento al menos dos patrones por clase.

Para las pruebas realizadas se utilizan 187 personas, tomando por cada una dos imágenes como referencia y una para la confrontación posterior.

Los resultados obtenidos indican que las implementaciones LDA tienen una tasa relativa de error del orden de 30 % al 60 % menor que la implementación original.

E. Conclusiones

Se aplica LDA a los vectores de características de Gabor. La idea es llevar dichas características a un espacio que discrimine mejor entre las clases de entrada al sistema. Se utiliza dos implementaciones del algoritmo LDA, la clásica FISHER-LDA y la denominada DF-LDA que aplica argumentos algebraicos y de patrones para no tener que proyectar los datos de entrada sobre el espacio no nulo generado por vectores de la base PCA.

Los resultados obtenidos permiten establecer que tanto el algoritmo FISHER-LDA como el algoritmo DF-LDA superan en performance al sistema que confronta directamente con los vectores de características. Estos resultados son prometedores para que en un trabajo futuro se realicen esfuerzos para codificar en C++ una implementación y hacerla compatible con el resto del sistema.

Finalmente vale decir que el hecho de pasar a trabajar con vectores de características de mucho menor dimensión (del orden del número de identidades enroladas en la base) no mejora el algoritmo desde el punto de vista de su costo computacional. Recordar que la gran exigencia del algoritmo se encuentra en el ajuste de grafo y en la extracción de *jets* de Gabor. Se puede a su vez pensar que existe un ahorro desde el punto de vista del almacenamiento de la información, pero en la actualidad esto no es una restricción para la mayoría de los sistemas de almacenamiento. Se agrega una complejidad al sistema: cada vez que se ingrese una nueva persona al sistema se debe volver a entrenar la matriz y recalcular los representantes de cada identidad.

En síntesis, LDA aplicado a los vectores de características de Gabor puede ser una alternativa importante a la hora de discriminar entre las identidades enroladas en la base. También hay que tener presente que esta alternativa agrega un costo a la hora de enrolar a una nueva persona por lo expresado en el párrafo anterior.

7.3. Fusión de Referencias

Luego de establecidas las distintas opciones para confrontar dos imágenes es necesario definir un score para una imagen de entrada dada la identidad postulada de la persona y las distintas referencias almacenadas en la base

para ese individuo. A los efectos de encontrar la mejor forma de combinar las distancias a cada una de las referencias de la identidad postulada, se hallan las ROC para distintas opciones. Como similitud entre dos imágenes se toma en este caso simplemente el promedio de la similitud en cada *jet*, utilizando la similitud de Magnitud.

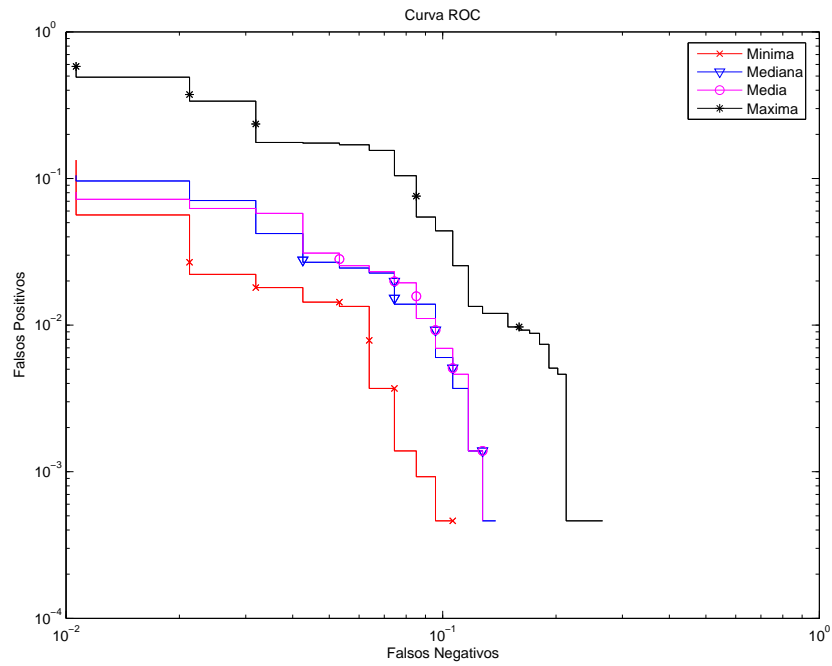
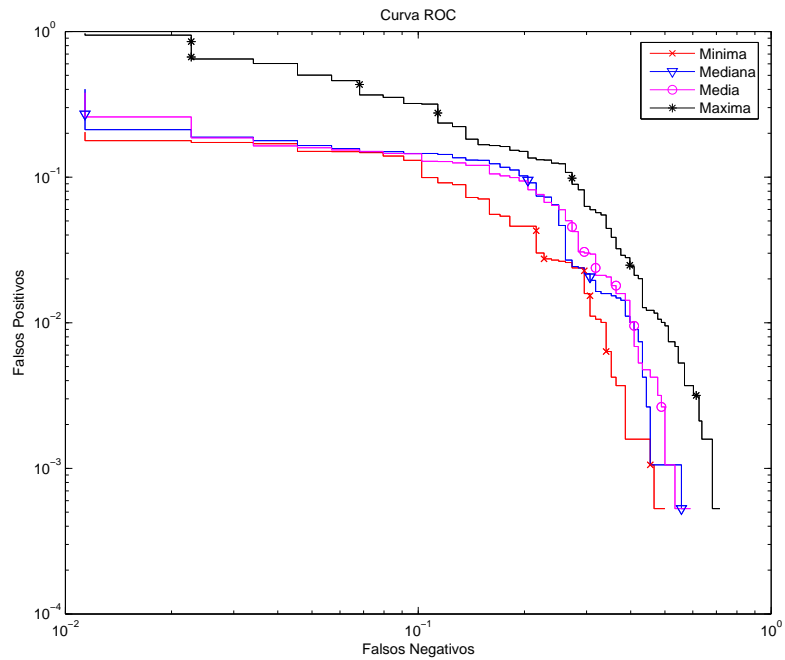
Las pruebas corresponden a los siguientes conjuntos de datos generados con la base IIE:

- **Ideal** - puntos marcados a mano.
Base de datos: 47 individuos, 4 referencias por persona.
- **Semi-automático** - ojos marcados a mano, ajuste de puntos automático.
Base de datos: 48 individuos, 4 referencias por persona.
- **Automático 2** - totalmente automático.
Base de datos: 50 individuos, 2 referencias por persona.
- **Automático 3** - totalmente automático.
Base de datos: 49 individuos, 3 referencias por persona.
- **Automático 4** - totalmente automático.
Base de datos: 44 individuos, 4 referencias por persona.

Las distintas maneras utilizadas para combinar las distancias a cada referencia fueron tomar la *mínima*, la *media*, la *mediana*, o la *máxima* de las distancias intra-clase.

En la decisión inter-clase el mejor resultado en todos los casos es para el score que considera la mínima distancia a las referencias, lo cual corresponde a tomar como score la similitud más alta de todas las obtenidas frente a cada una de las referencias. A continuación se muestran las ROC correspondientes a los conjuntos de datos **Ideal** y **Automático 4**, ambas en escala logarítmica.

Estos resultados muestran que la mejor performance del sistema de verificación se obtiene tomando la distancia con la referencia más cercana almacenada en la base de datos.

Figura 7.4: ROC en escala logarítmica - Resultados caso *Ideal*.Figura 7.5: ROC en escala logarítmica - Resultados caso *Automático 4*.

Parte III

Implementación y Evaluación

Capítulo 8

Sistema

8.1. Introducción

A lo largo de los cuatro capítulos anteriores se desarrolla las diferentes etapas del algoritmo de reconocimiento de caras implementado. En los mismos se detalla opciones y soluciones elegidas. Cada uno describe un bloque funcional que forma parte de una única implementación, el *sistema*.

Las secciones siguientes describen este *sistema* como un *todo*, detallando su constitución, aspectos funcionales y configuración.

8.2. Definición de Sistema

El sistema es la concatenación de los distintos bloques funcionales que resulta en la implementación de una herramienta de reconocimiento de caras. Incluye todos los aspectos que hacen a su funcionamiento autónomo, es decir, comprende el modelo estadístico utilizado en la localización de puntos y la base de datos de individuos. La figura 8.1 muestra un diagrama de bloques básico del sistema.

En la figura 8.2 se muestra un diagrama de bloques detallado del funcionamiento del sistema. El primer bloque implementa la adquisición de la imagen conteniendo el rostro de un individuo, el segundo bloque implementa la detección de la cara en la imagen, seguido por una etapa de normalización de la misma. Este bloque localiza las coordenadas de los ojos, en caso de que exista una única cara en la escena, y preprocesa la imagen para llevarla a un formato común para el ingreso a la siguiente etapa.

El siguiente bloque se encarga de la localización de los puntos característicos en la cara normalizada. Utiliza para esto el modelo estadístico, *Face Bunch Graph*. Posteriormente se extraen las características en los pun-

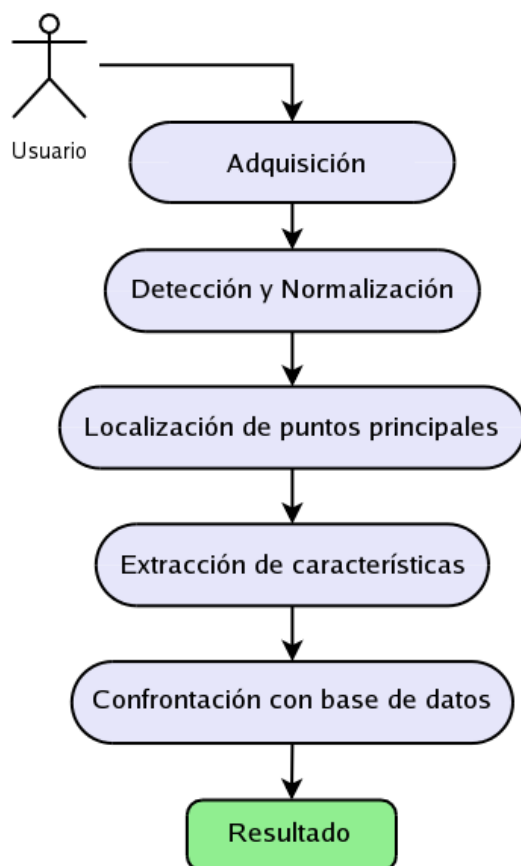


Figura 8.1: Diagrama de bloques del sistema.

tos principales mediante un banco de filtros de Gabor. Finalmente se tiene la etapa de confrontación que compara descriptores de caras para reconocimiento. Las tareas desarrolladas en esta etapa dependen del modo de funcionamiento del sistema como se explica en la siguiente sección.

Las funcionalidades de los distintos módulos se implementan a través de una serie de bibliotecas codificadas en C/C++, desarrolladas a lo largo de este proyecto. Por más información sobre las mismas referirse al Anexo A. El diseño modular llevado a la práctica por las distintas bibliotecas codificadas brinda un grado de flexibilidad que permite que las distintas etapas del sistema sean sustituibles por módulos análogos de idéntica funcionalidad. Por ejemplo, el módulo de ajuste de grafo puede ser intercambiado por cualquier otro módulo que reciba a la entrada una imagen con una cara detectada y normalizada y que devuelva las coordenadas de N puntos característicos. Estos puntos pasan al siguiente módulo, en el que se extraen

las características y se sigue avanzando en las etapas posteriores del sistema.

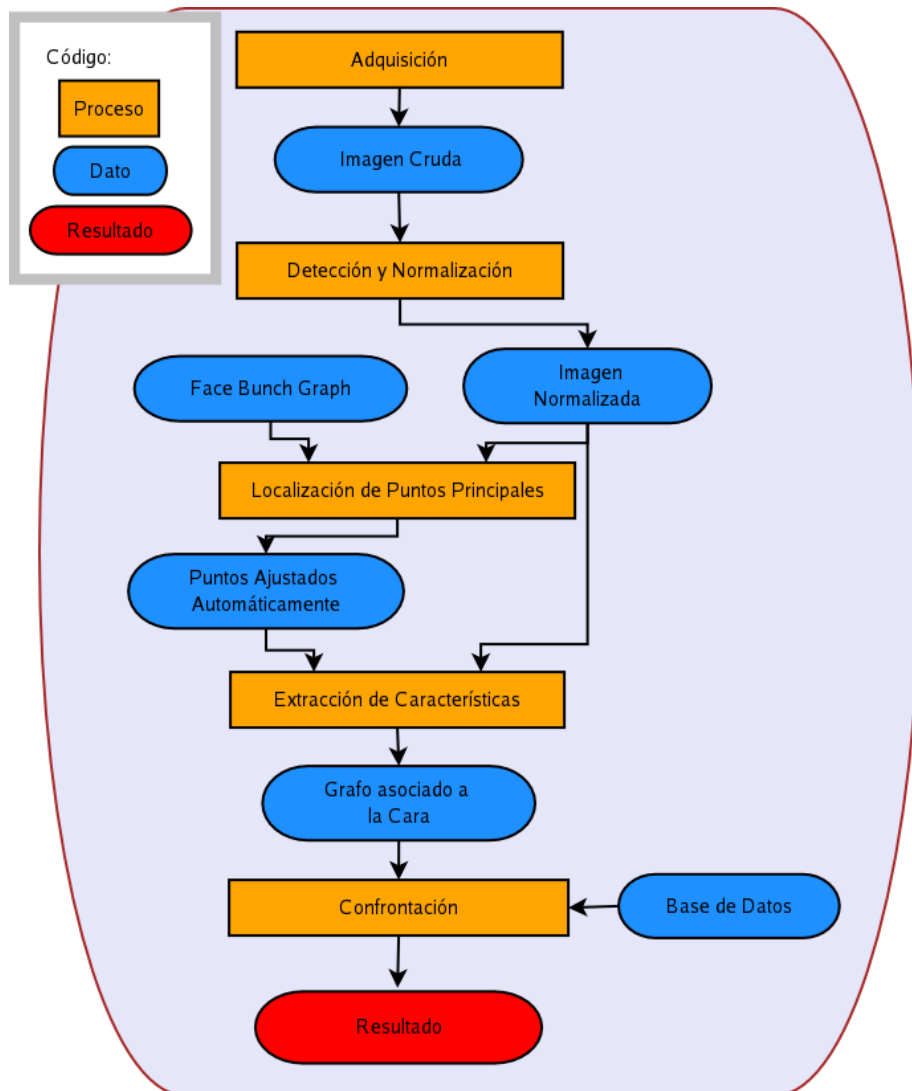


Figura 8.2: Diagrama de bloques del sistema que muestra los datos manejados en cada etapa.

8.3. Especificación Funcional

Existen dos modalidades básicas de utilización del sistema, la modalidad de registrado y la de confrontación. La primera consiste en ingresar un nuevo usuario al sistema. Para ello se adquiere una imagen tipo foto carnet del mismo y se procesa hasta obtener una descripción de la cara del individuo. Dicha descripción son los wavelets de Gabor extraídos en los puntos principales localizados en su cara. Esta información es almacenada para ser utilizada en posteriores ocasiones.

La segunda es la modalidad de confrontación, que presenta dos variantes: verificación e identificación. En la verificación, un usuario ya registrado se presenta al sistema a través de un identificador, por ejemplo un PIN. En base a una única imagen del individuo, el sistema confronta contra la información de la base de datos y valida su identidad. Por otro lado, en la identificación, el sistema recorre su registro evaluando la similitud de las identidades enroladas con respecto a la información del individuo y obtiene una identidad para el mismo.

La figura 8.3 presenta un diagrama de los distintos modos de funcionamiento del sistema.

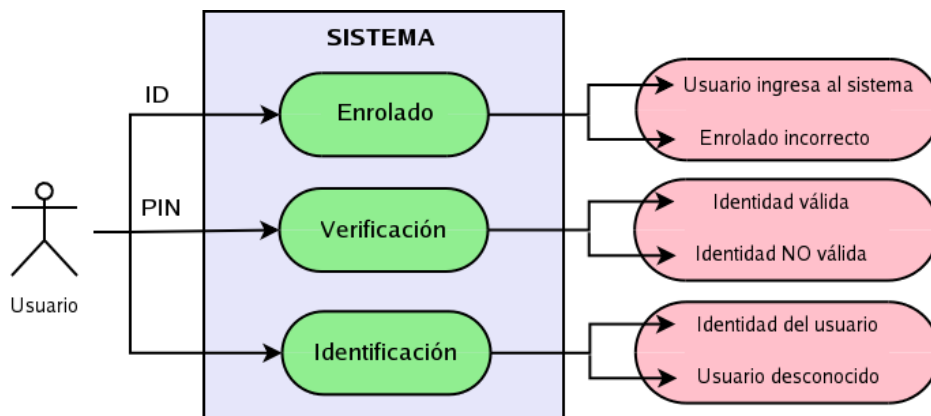


Figura 8.3: Diagrama representativo de los distintos modos de funcionamiento del sistema.

8.4. Configuración del Sistema

Los distintos bloques que componen el sistema presentan alternativas en su funcionamiento. Fijando el modo de trabajo de cada uno de éstos se define una configuración dada para el mismo. Algunos ejemplos de aspectos concernientes a la configuración son: los parámetros que definen el banco de filtros de Gabor, el modelo estadístico o *Face Bunch Graph* seleccionado, el

tipo de ajuste y la medida de similitud entre grafos utilizada. Una descripción detallada de las posibles alternativas de funcionamiento de cada bloque se encuentra en los capítulos anteriores. La figura 8.4 ilustra acerca de las mismas y dónde encontrarlas.

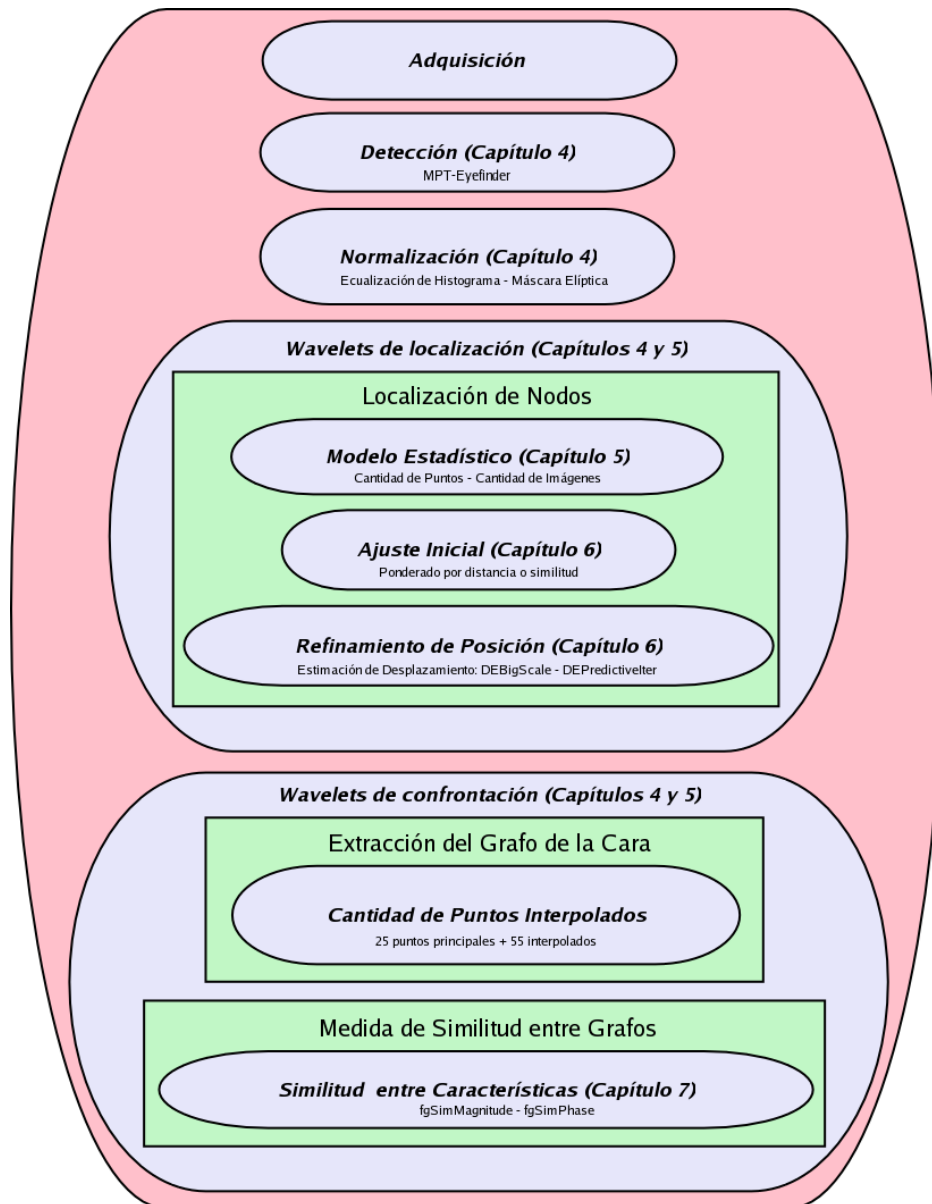


Figura 8.4: Alternativas en los distintos bloques del sistema y su distribución a lo largo de la presente documentación.

A partir del abanico de opciones presentado se conforma una *configuración base* del sistema. Cabe aclarar que no es esta una configuración óptima, sino que una configuración básica. Luego de completada la etapa de evaluación del sistema en el capítulo 9, se propone una configuración óptima del mismo. Para definirla es necesario elegir una serie de parámetros en cada uno de los módulos del sistema.

En la normalización se debe optar por ecualización de histograma, máscara elíptica o ninguna de ellas. Las demás etapas del proceso son las descritas en el capítulo 4. En la localización de puntos se debe elegir el banco de filtros de Gabor y la compensación de potencia. Además hay que definir la estimación inicial de la ubicación de los puntos y el estimador de desplazamiento para el refinamiento.

Es de suma importancia en este módulo el modelo estadístico *Face Bunch Graph*. Se analiza en este caso un importante trabajo, realizado en [11], en el que se utiliza un set de 150 imágenes de la base FERET, marcadas manualmente, para generar distintos modelos. En el mismo se testean modelos de 1, 2, 4, 8, 16, 32, 64 y 128 imágenes, formados a partir de las 150 imágenes antes mencionadas. Se encuentra que un compromiso razonable entre la cantidad de imágenes y la generalidad del modelo se alcanza para un *Face Bunch Graph* de 70 imágenes. Por esta razón la cantidad de imágenes del modelo se fija en 70 imágenes.

Cabe señalar que, más allá de la cantidad de imágenes que se tome, es de gran importancia cuáles son las imágenes elegidas. Esta selección es la que define la riqueza del *Face Bunch Graph* en cuanto a variabilidad de rasgos y características contempladas.

Pasando a la extracción de características para la posterior confrontación es necesario definir la estructura de grafo que representa la cara. En este caso se debe elegir los puntos principales y los interpolados, tanto la cantidad como la ubicación de los mismos. En la configuración base se trabaja con el modelo de 25 puntos + 55 puntos interpolados propuesto por Bolme. A lo largo de este proyecto se desarrolla una serie de modelos de menor cantidad de puntos, los cuales centran la atención en la región interior del rostro, tomando como hipótesis la no utilidad de los puntos del borde de la cara. Luego de avanzar en la evaluación del algoritmo se observa que, en algunos casos, la suposición anterior no es correcta. Por ejemplo, en caso de existir variaciones de expresión en el rostro, lo que permanece mayormente invariante en cada identidad es el contorno de la cara y no el interior. En dichos casos el algoritmo presenta un mejor desempeño considerando los puntos del borde. Una vez ajustados los puntos principales se puede interpolar y obtener una nueva serie de puntos en los cuales extraer características.

Aquí existe un compromiso entre el tiempo que toma el cálculo de las nuevas características y la riqueza que esto agrega a la hora de discriminar en la etapa de confrontación.

Se debe en este módulo elegir nuevamente con qué banco de filtros de Gabor trabajar. Es importante destacar que no tienen por qué ser los mismos que los elegidos para el ajuste, lo que resalta la modularidad del sistema. En particular el bloque de localización de puntos puede ser otro que no use wavelets, pero que sí sean estas las características definidas para el descriptor de la cara. Por último en el módulo de confrontación debemos definir cuál es la medida de similitud que se emplea para comparar los grafos correspondientes a cada cara.

Luego de repasar los parámetros que se debe seleccionar en el sistema, se define la *configuración base* del mismo:

Configuración Base del Sistema

Detección y Normalización - La detección de la cara se realiza de forma automática. Luego se trabaja con imágenes de 128x128 normalizadas geoméricamente según se explica en el capítulo correspondiente. No se utiliza máscara elíptica ni se realiza ecualización de histograma.

Localización - Se utiliza un modelo de 25 puntos formado por 70 imágenes de la base FERET marcadas de forma manual. El ajuste inicial se basa en ponderar por distancia. Los filtros de gabor son sintonizados mediante los parámetros *Wiskott* con la compensación para caras. El predictor utilizado en el ajuste es el de paso predictivo por escalas (*DEPredictiveBigScale*).

Extracción de características - Para la extracción se utiliza un grafo de 25 puntos + 55 puntos interpolados y el juego de parámetros *Wiskott* con la compensación para caras.

Confrontación - Como medida de similitud se utiliza la Similitud de Magnitud.

En el capítulo siguiente se realiza una evaluación completa del sistema, estudiando la elección de los parámetros mencionados anteriormente.

Capítulo 9

Evaluación

9.1. Introducción

El análisis de performance de los sistemas biométricos es un tema central en este tipo de proyectos, trabajos o publicaciones. Existen muchos artículos en donde los resultados indicados refieren a bases de datos que no son de dominio público, lo que impide compararse con los mismos. Incluso en casos donde se utilizan bases estándar, no se aclara de manera debida la mecánica utilizada para la estimación de desempeño. En particular, un aspecto que no es tenido en cuenta en gran parte de las publicaciones, pero que es de gran importancia para comparar la performance de los distintos sistemas, es el análisis de la incertidumbre en la estimación de desempeño. En esta sección se describe la metodología utilizada para la evaluación del sistema así como algunas variantes para el cálculo de intervalos de confianza tomadas de [73].

Desempeño en Verificación

Como se describe en la introducción a los sistemas biométricos en el capítulo 2, las medidas básicas de desempeño de un sistema de verificación o autenticación son el FAR (que corresponde al porcentaje de falsas aceptaciones) y el FRR (que corresponde al porcentaje de falsos rechazos). Para obtener el desempeño se releva los distintos puntos de funcionamiento del sistema desde $FAR = 0$ a $FRR = 0$ generando la curva ROC.

En la figura 9.1 se muestra un ejemplo de las distribuciones de *scores* obtenidas por el sistema para los usuarios genuinos y para los impostores. El punto donde se cortan ambas distribuciones corresponde al umbral que se debe utilizar para trabajar en el EER.

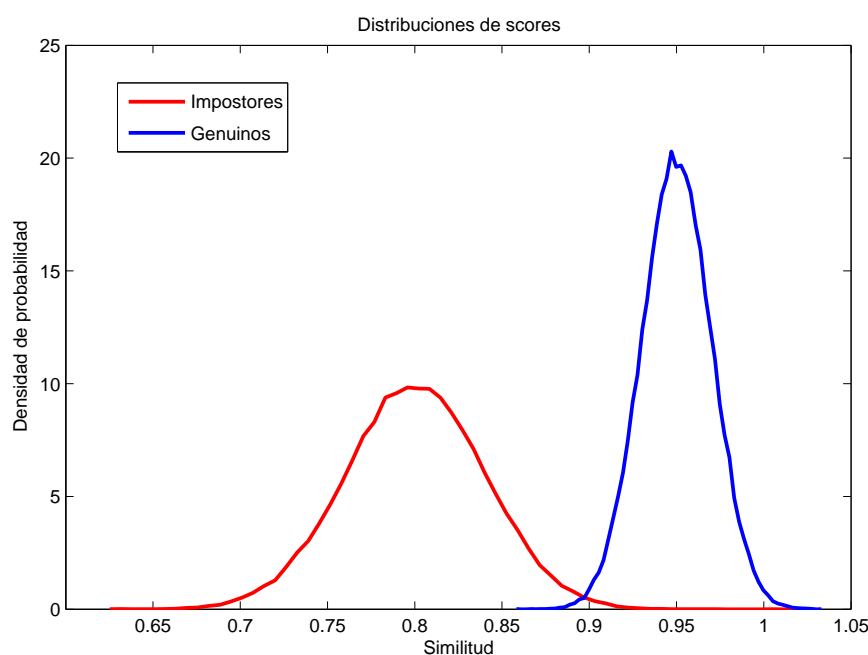


Figura 9.1: Ejemplo de la distribución de *scores* para usuarios genuinos e impostores.

Desempeño en Identificación

La medida básica de desempeño de un sistema de identificación es el RR, es decir la tasa de identificación correcta. Como medida del error se podría utilizar $ER = 1 - RR$, es decir la tasa de identificación incorrecta. Dado que ambas medidas de performance, el error o la tasa de identificación correcta, están relacionadas entre sí en su definición es posible utilizar cualquiera de ellas como indicador de desempeño.

9.2. Estimación de Intervalos de Confianza

9.2.1. Paramétrica

La estimación paramétrica asume ciertas hipótesis acerca de los datos disponibles y de la distribución de las muestras. En este caso se describe el procedimiento para hallar el FRR en verificación, siendo análogo el razonamiento para el FAR y el RR en el caso de identificación. Teniendo m *scores* correspondientes a parejas de imágenes de cada persona se supone que corresponden a muestras *i.i.d.* de una cierta distribución. Las muestras $X = \{X_1, \dots, X_m\}$ permiten fijando un cierto umbral T tomar $F\hat{R}R_T = \frac{1}{m} \sum 1(X_i \leq T)$ como estimador del FRR¹.

¹La función $1(\text{condición})$ es la indicatriz, vale 1 si la condición se cumple y 0 en caso contrario.

La hipótesis del test paramétrico es que $m \cdot FRR$ se considera una variable aleatoria con distribución binomial de parámetros m y p , con $0 \leq p \leq 1$. El valor esperado de esta variable es mp y una estimación de la media es $m \cdot F\hat{R}R$. Esto es razonable puesto que la binomial representa el número de éxitos en m intentos sucesivos, donde un éxito es $X_i \leq T$. Para estimar la varianza se utiliza la ley fuerte de los grandes números, para asumir que $F\hat{R}R$ tiene distribución normal de media FRR y varianza σ donde una estimación de σ está dada por $\hat{\sigma} = \sqrt{\frac{F\hat{R}R(1-F\hat{R}R)}{m}}$.

Luego de estimar la varianza se toma un intervalo de confianza al 95 % que corresponde a $\pm 2\sigma$. Para hallar un intervalo de confianza para el EER a partir de los intervalos hallados para el FAR y el FRR, se debe considerar sus respectivos intervalos y considerar cómo varía la ROC para hallar los peores casos. Esto se ilustra en la figura 9.2. Ver que el intervalo de confianza para el FAR es menor que para el FRR, puesto que en general se tiene más cantidad de *scores* correspondientes a parejas de personas distintas que de la misma persona.

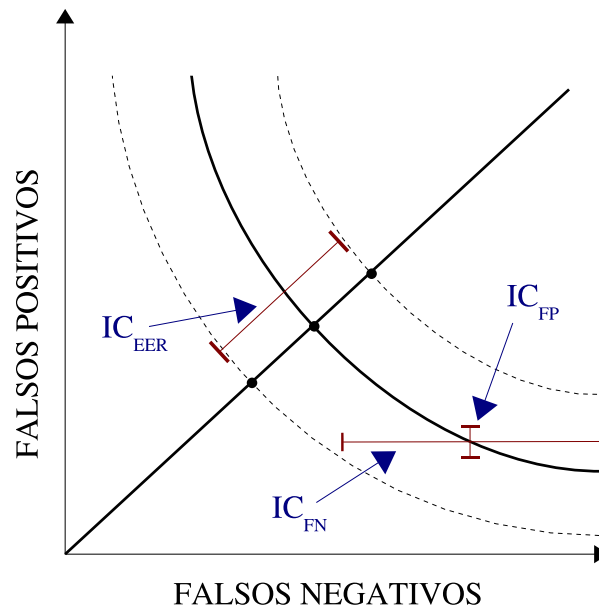


Figura 9.2: Intervalo de confianza para el EER con estimación paramétrica.

9.2.2. No Paramétrica

En este caso no se asume una distribución para las muestras de *scores*. El método utilizado se denomina *bootstrap* y emplea muestreo con repetición al conjunto X una gran cantidad B de veces, obteniendo los conjuntos $X_i^*, i = 1, \dots, B$ y calculando las estimaciones del FRR para cada conjunto FRR_i^* . Luego, la determinación de los intervalos de confianza se reduce a un problema de conteo. El procedimiento es el siguiente:

1. A partir de las muestras $X = \{X_1, \dots, X_m\}$ se estima el FRR para un cierto umbral T como $F\hat{R}R = \frac{1}{m} \sum 1(X_i \leq T)$, igual que en el caso paramétrico.
2. *Remuestreo*: Se crea el conjunto X_i^* a partir de m muestras aleatorias con repetición del conjunto X .
3. *Estimación*: Se estima el FRR calculando FRR_i^* a partir de X_i^* .
4. *Repetición*: Se repiten los pasos 2 y 3 B veces, obteniendo los valores $FRR_1^*, FRR_2^*, \dots, FRR_B^*$.

Para obtener el intervalo de confianza de $F\hat{R}R$, se ordenan las estimaciones en orden creciente $FRR_{(1)}^* \leq FRR_{(2)}^* \leq \dots \leq FRR_{(B)}^*$. El intervalo de confianza $(1-\alpha)100\%$ está dado por $FRR_{(q_1)}^*, FRR_{(q_2)}^*$, siendo $q_1 = \lfloor \frac{\alpha B}{2} \rfloor$ la parte entera de $\frac{\alpha B}{2}$ y $q_2 = B - q_1 + 1$.

En este caso para hallar el intervalo de confianza para el EER no se realiza lo mismo que en el caso paramétrico puesto que en este caso podemos realizar el procedimiento antes descrito directamente con el EER. Lo que se hace es hallar el EER_i para cada conjunto X_i y luego se procede análogamente a lo descrito para el FRR, hallando de esta forma el intervalo de confianza deseado.

9.2.3. Intervalos de Confianza en los Ensayos

En los ensayos realizados en la evaluación del sistema se utilizaron ambas técnicas descritas anteriormente para la estimación de los intervalos de confianza. En particular se toma en los dos casos intervalos de confianza al 95%. La nomenclatura utilizada en el resto del capítulo es la siguiente:

ICP - Estimación paramétrica del intervalo de confianza al 95 %.

ICB - Estimación no paramétrica del intervalo de confianza al 95 %.

Además se utilizan los subíndices R y E para identificación y verificación respectivamente.

9.3. Ensayos y Resultados

Además del análisis teórico de la incertidumbre existen otras condiciones que validan un sistema de evaluación. Para la correcta evaluación de un sistema de reconocimiento es importante utilizar una base de imágenes que cumpla con ciertos requerimientos básicos:

Estándar La base a utilizar debe estar bien aceptada por la comunidad científica que trabaja en el Reconocimiento de Caras. Este aspecto es importante para poder comparar el desempeño de distintos algoritmos de forma objetiva.

Tamaño La cantidad de imágenes de la base debe ser lo suficientemente grande como para poder calcular indicadores de performance con errores suficientemente bajos. Este punto es deseable sobre todo a la hora de comparar desempeños similares de diferentes algoritmos.

Variación Las imágenes de la base deben de ser variadas, es decir la base debe contener imágenes que al menos cubran los siguientes aspectos: pose variable, expresión variable, iluminación no uniforme, oclusiones y diferentes instantes de tiempo. Según sea la aplicación puede ser importante utilizar una base de mayor o menor variabilidad en este aspecto.

Dificultad Las imágenes de la base no deben de ser ni muy difíciles de clasificar ni muy fáciles. Este punto, un tanto subjetivo, es importante para evitar casos en los cuáles todos los algoritmos a evaluar tienen una performance extremadamente buena o extremadamente mala, en ambos casos se hace difícil la comparación de desempeño.

Además de estos puntos listados anteriormente existen otras consideraciones por ejemplo, el hecho de que la base sea gratuita y de uso libre para proyectos de investigación.

Siguiendo todas estas consideraciones se realizó una búsqueda de diferentes bases gratuitas disponibles. Se decidió utilizar la base FERET, una base ampliamente reconocida por la comunidad científica del reconocimiento de caras, con una gran cantidad de imágenes que cubren los aspectos citados anteriormente. A su vez, dado que el proyecto incluye el desarrollo de una aplicación orientada al control de acceso, se decidió desarrollar una base particular para dicha aplicación, la base IIE. La misma es una base que consta de imágenes provenientes de sesenta individuos, con una gran cantidad de imágenes por persona. Por más detalle acerca de ambas bases consultar el apéndice C.

9.3.1. Evaluación Modular del Sistema

En esta sección se muestra algunos de los resultados obtenidos al evaluar los diferentes bloques del sistema. Para esto se utiliza la configuración base definida en el capítulo 8, a excepción del bloque de detección de la cara. La normalización recibe como entrada además de la imagen, las coordenadas de los ojos en la misma. Este módulo se deja fuera del sistema a evaluar en esta etapa por dos razones fundamentales. La primera es el hecho de la influencia que tiene en el desempeño del resto de los bloques, lo cual afecta la evaluación correcta de los mismos. La segunda es poder compararse con otros resultados publicados con la base FERET que también usan la posición de los ojos como entrada al sistema.

El sistema base no es el sistema óptimo. El mismo se utiliza para evaluar los diferentes bloques de manera independiente y poder a partir de esos resultados conformar un sistema óptimo. En cada una de las etapas de evaluación, se modifica únicamente un parámetro del sistema, de manera de cambiar solamente lo que se desea evaluar.

La base FERET contiene un total de 14051 imágenes de más de 1000 individuos. Para realizar la evaluación se toma como referencia [74, 75] en donde se explica una metodología general para evaluar algoritmos de reconocimiento de caras utilizando la base FERET. Para someter al sistema a distintos niveles de dificultad, se trabaja con cinco grandes grupos de imágenes:

fa - Conjunto de imágenes base, forman la galería, cuentan con una imagen por sujeto. Total de imágenes: 1196.

fb - Conjunto de imágenes similar al **fa**. Difieren únicamente en la expresión del sujeto. Total de imágenes: 1195.

fc - Conjunto de imágenes tomado en la misma sesión que **fa** pero con diferente iluminación. Total de imágenes: 194.

duplicate I - Conjunto de imágenes tomadas un tiempo después que **fa**. Total de imágenes: 722.

duplicate II - Conjunto de imágenes tomadas al menos 18 meses después que el conjunto **fa**. Total de imágenes: 234.

A su vez, utilizando estos conjuntos de imágenes se definen cuatro tests: **FB**, **FC**, **DUP-I**, **DUP-II**. Cada uno de estos tests se realiza considerando al grupo **fa** como galería y al restante como conjunto de prueba. Es importante destacar que solamente se utiliza una única imagen por sujeto como referencia.

En esta primera etapa de evaluación, en la que se analiza el funcionamiento de cada bloque por separado, se realiza únicamente el test **FB** para la obtención de resultados.

A. Normalización

Las variantes que se evaluaron en la etapa de normalización se detallan a continuación:

Norm - Sin máscara ni ecualización de histograma.

Hist - Sin máscara, con ecualización de histograma.

Mask - Con máscara, sin ecualización de histograma.

H&M - Con máscara y con ecualización de histograma.

Los resultados obtenidos se muestran en la tabla adjunta. La figura 9.3 corresponde a las curvas de rango obtenidas para las variantes evaluadas.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
Norm	94.7	[93.4, 96.0]	2.26	[1.57, 2.87]	[1.59, 2.85]
Hist	95.8	[94.6, 97.0]	1.92	[1.59, 2.40]	[1.59, 2.34]
Mask	94.3	[93.0, 95.6]	2.07	[1.61, 2.66]	[1.59, 2.59]
H&M	95.1	[93.9, 96.3]	1.76	[1.24, 2.16]	[1.26, 2.12]

En la figura 9.3 se nota que la ecualización de histograma mejora la performance y que al aplicar la máscara empeora. La utilidad de la máscara es para imágenes con fondo complejo donde es deseable que el mismo no influya en la performance del sistema. Como en este caso la base de imágenes es con fondo controlado, lo que se quiere es que la máscara no empeore el desempeño del sistema. En la tabla vemos que el EER casi no varía y en ambos casos disminuye. Aplicando la máscara y la ecualización de histograma combinadamente se obtiene una curva de rango que presenta buen desempeño y el EER obtenido es mejor que para el resto de los casos.

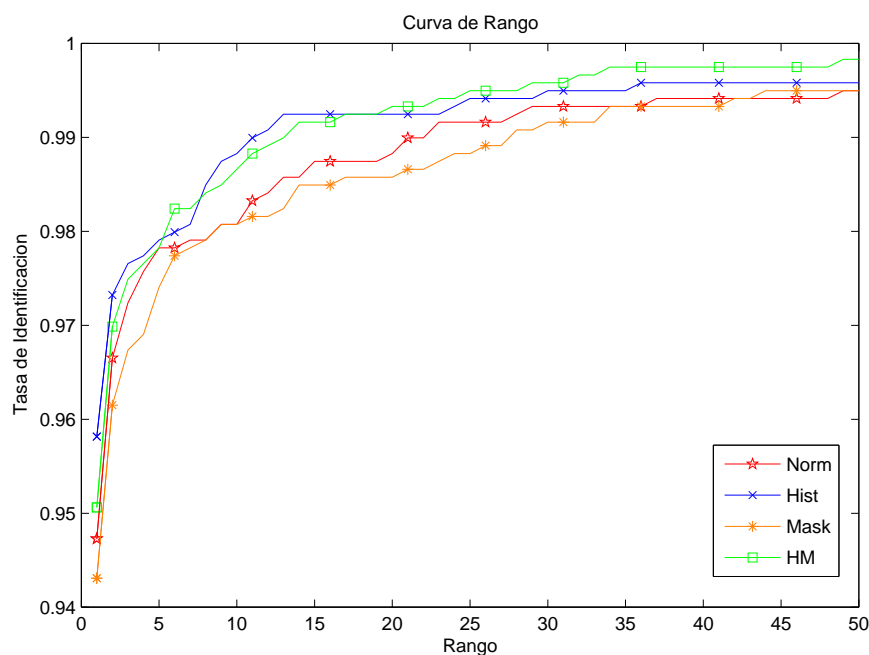


Figura 9.3: Curvas de Rango (CMC) para las variantes en la Normalización.

B. Localización de puntos

Dado que la localización de puntos depende de varios parámetros se evalúa la performance al cambiar cada uno de ellos: estimación inicial, estimador de desplazamiento y wavelets.

B.1. Estimación Inicial En este punto se evalúa utilizar las diferentes técnicas de estimación inicial vistas en el proyecto: por distancia, por similitud o combinada. Para el resto de los parámetros se utiliza la configuración base. Las técnicas evaluadas son las siguientes:

AjuDist - Estimación inicial utilizada en el sistema base. Se basa en ponderar las diferentes estimaciones respecto de cada nodo, según la distancia al nodo a ubicar.

AjuSim - Estimación inicial basada en ponderar las diferentes estimaciones según la similitud obtenida en cada uno de los nodos.

AjuComb - Estimación inicial que combina ambas técnicas anteriores. Se pondera la estimación de cada nodo por la distancia al nodo a ubicar y la similitud obtenida al ajustar el nodo base.

Los resultados obtenidos se muestran en la tabla a continuación.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICB_E	ICP_E
AjuDist	94.7	[93.4, 96.0]	2.26	[1.59, 2.92]	[1.57, 2.87]
AjuSim	95.0	[93.7, 96.3]	2.19	[1.75, 2.83]	[1.73, 2.86]
AjuComb	94.8	[93.5, 96.1]	2.34	[1.67, 2.92]	[1.63, 2.97]

De la tabla anterior se desprende que todas las técnicas de estimación inicial exploradas tienen un desempeño similar. Dado el tamaño de los intervalos de confianza no se puede concluir que exista una mejor técnica de estimación inicial. En cuanto a tiempo de procesamiento, todas las técnicas insumen tiempos similares, siendo una porción despreciable dentro del tiempo total de la localización insumido por el algoritmo.

B.2. Estimadores de Desplazamiento En este caso se compara la performance utilizando los distintos estimadores de desplazamiento para la etapa de refinamiento de la posición de los puntos en la cara. Las pruebas efectuadas son las siguientes:

BScale - Refinamiento utilizando estimación de desplazamiento por paso predictivo por escalas (DEPredictiveBigScale).

PStep - Refinamiento utilizando estimación de desplazamiento por paso predictivo (DEPredictiveStep).

NoFit - No se realiza refinamiento en la localización de puntos.

GraD - Refinamiento utilizando estimación de desplazamiento por descenso por gradiente (DEGradientDescent).

PIter - Refinamiento utilizando estimación de desplazamiento por predicción iterativa (DEPredictiveIter).

Se resume en la tabla a continuación los resultados obtenidos.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
BScale	94.7	[93.4, 96.0]	2.26	[1.57, 2.87]	[1.59, 2.85]
PStep	94.6	[93.3, 95.9]	2.18	[1.50, 2.71]	[1.57, 2.69]
NoFit	94.9	[93.6, 96.2]	2.26	[1.68, 2.84]	[1.67, 2.81]
GraD	90.5	[88.8, 92.2]	4.44	[3.66, 5.37]	[3.61, 5.32]
PIter	94.6	[93.3, 95.9]	2.18	[1.67, 2.63]	[1.67, 2.60]

Como podemos apreciar no existen diferencias en la mayoría de los casos. La única excepción corresponde al caso del estimador que utiliza descenso por gradiente, el cual presenta un desempeño peor al de los demás estimadores. Se destaca el buen desempeño obtenido, al no refinar el ajuste inicial a la cara (NoFit). Si bien su performance es similar al del resto, es claro que su costo computacional es mucho menor.

B.3. Wavelets Localización La última etapa por evaluar en la localización son los filtros de Gabor utilizados. Se cambia el juego de parámetros de los filtros por los vistos en el capítulo 4. En este caso los filtros de Gabor utilizados en la etapa de confrontación siguen siendo los del sistema base. Los bancos de filtros evaluados son los siguientes:

Wiskott - Juego de parámetros del sistema base. Es el juego de parámetros de la implementación del sistema BOCHUM/USC original.

Bolme - Juego de parámetros del sistema implementado por Bolme.

Bio3 - Juego de parámetros propuesto en este proyecto.

En la tabla adjunta se presenta los resultados obtenidos.

Test	Identificación		Verificación		Tiempo medio ²
	RR(%)	ICP_R	EER(%)	ICP_E	
Wiskott	94.7	[93.4, 96.0]	2.26	[1.57, 2.87]	1.69
Bolme	94.8	[93.5, 96.1]	2.18	[1.67, 2.85]	1.31
Bio3	94.6	[93.3, 95.9]	2.22	[1.74, 2.94]	1

Dado que los EER y los RR de todos los sistemas dan muy similares, sus intervalos de confianza se solapan casi en su totalidad. Por lo tanto, no se logra destacar ninguno con un desempeño mejor que el resto. Sin embargo, se observa que el juego de parámetros Bio3 tiene un tiempo medio de ajuste significativamente menor al de Wiskott. Este último aspecto resulta extremadamente importante a la hora de desarrollar una aplicación del tipo control de acceso.

C. Confrontación

C.1. Potencia de los Filtros de Gabor En esta sección se estudia el impacto que tiene la potencia de los filtros de Gabor en el desempeño global del sistema. Como se vio en el capítulo 4 las imágenes de caras presentan aspectos particulares que las diferencian del conjunto de imágenes naturales. A continuación se muestran los resultados obtenidos al utilizar el sistema base, modificando la compensación de potencia de los filtros de Gabor. Las compensaciones estudiadas son:

Bolme_f0 - Potencia de los filtros utilizada en la implementación Bolme. La potencia es constante en todo el espectro.

Wiskott_f2 - Potencia de los filtros utilizada en el sistema original BOCHUM/USC. Crece con el cuadrado de la frecuencia.

Caras_f3 - Potencia de los filtros propuesta en este proyecto. Crece con el cubo de la frecuencia, según los ensayos en este trabajo se adaptada a imágenes de caras.

Se resume en la siguiente tabla los resultados obtenidos.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICB_E	ICP_E
Bolme_f0	87.9	[86.0, 89.8]	4.22	[3.51, 4.94]	[3.44, 5.01]
Wiskott_f2	92.7	[91.2, 94.2]	2.59	[2.09, 3.28]	[2.05, 3.35]
Caras_f3	94.7	[93.4, 96.0]	2.26	[1.59, 2.92]	[1.57, 2.87]

Observando los resultados anteriores se destaca la importancia de la compensación de potencia en los filtros de Gabor. La compensación propuesta por nosotros implica una mejora significativa respecto a las anteriores. En el gráfico 9.4 se puede apreciar la mejora obtenida en la curva de rango.

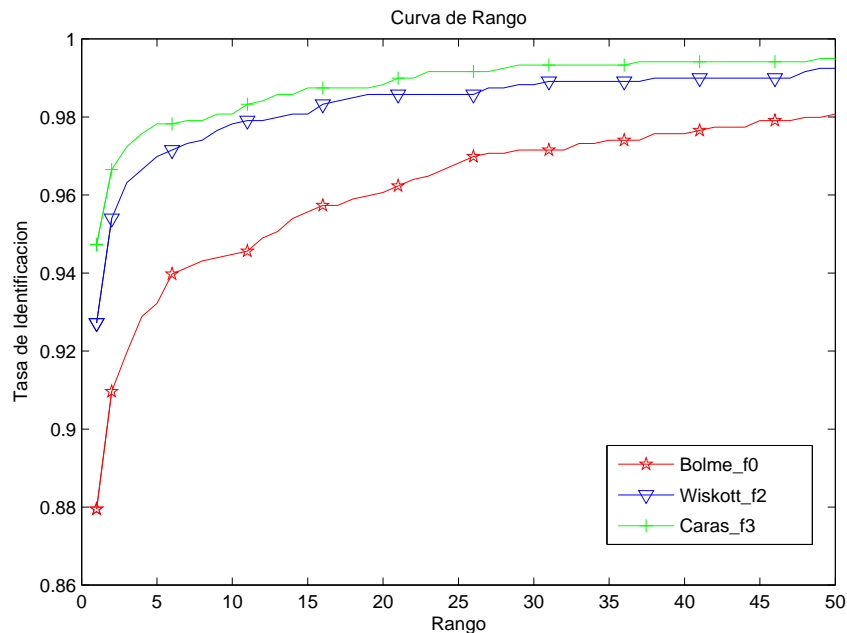


Figura 9.4: Curvas de Rango (CMC) para diferentes compensaciones de potencia en los núcleos de Gabor.

C.2. Wavelets Confrontación Otro punto a evaluar es la sensibilidad del sistema frente a la elección de diferentes wavelets de Gabor. A continuación se muestran los resultados obtenidos al utilizar los diferentes juegos de parámetros de wavelets vistos a lo largo del proyecto. Los diferentes sistemas a evaluar son:

Wiskott - Juego de parámetros del sistema base. Es el juego de parámetros de la implementación del sistema BOCHUM/USC original.

Bolme - Juego de parámetros del sistema implementado por Bolme.

Bio3 - Juego de parámetros propuesto en este proyecto.

Se presenta a continuación la tabla con los resultados obtenidos. En la figura 9.5 se muestra las distintas curvas de rango para cada alternativa.

Test	Identificación		Verificación		Tiempo medio
	RR(%)	ICP_R	EER(%)	ICP_E	
Wiskott	94.7	[93.4, 96.0]	2.26	[1.57, 2.87]	8.85
Bolme	96.2	[95.1, 97.3]	2.09	[1.52, 2.84]	4.66
Bio3	96.7	[95.7, 97.7]	1.76	[1.33, 2.27]	1

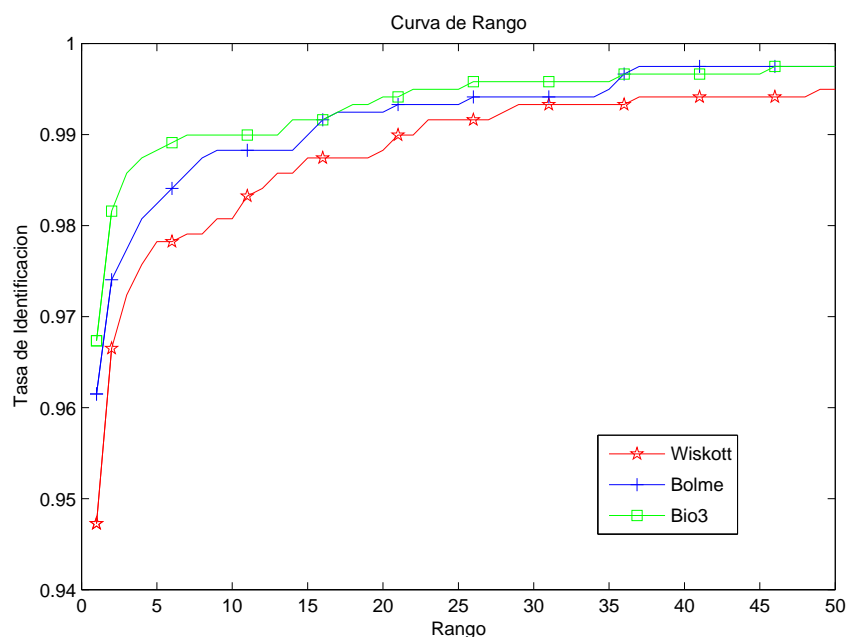


Figura 9.5: Curvas de Rango (CMC) para los diferentes wavelets analizados en la confrontación.

En la tabla anterior se logra apreciar que los diferentes wavelets tienen desempeños diferentes. A pesar de la incertidumbre en la medida de los

indicadores, se logra observar que el juego de parámetros Bio3 tiene una mejora respecto a los anteriores. Es importante aclarar que si bien puede parecer una mejora pequeña, en términos relativos³, vemos que es del orden del 20%. Dicha mejora no es solo en términos de reconocimiento, sino también en términos de ejecución. El tiempo medio⁴ de procesamiento para la extracción de dichos wavelets es sustantivamente menor al resto. Esto último implica una baja considerable del tiempo total que demora en correr el algoritmo.

C.3. Similitudes Promedio Se realiza la confrontación utilizando distintas medidas de similitud derivadas del sistema CSU de Bolme. Las pruebas realizadas corresponden a:

simMag - Similitud que considera solo la magnitud (`fgSimMagnitude`).

simFase - Similitud que considera magnitud y fase, sin corrección de la posición (`fgSimPhase`).

simBS - Similitud que considera magnitud y fase, con corrección de la posición utilizando el estimador de desplazamiento por paso predictivo por escalas (`fgSimPhasePredictiveBigScale`).

simPS - Similitud que considera magnitud y fase, con corrección de la posición utilizando el estimador de desplazamiento por paso predictivo (`fgSimPhasePredictiveStep`).

simPI - Similitud que considera magnitud y fase, con corrección de la posición utilizando el estimador de desplazamiento por predicción iterativa (`fgSimPhasePredictiveIter`).

En la tabla se presentan los resultados obtenidos. La figura 9.6 muestra las ROC obtenidas en escala logarítmica para las similitudes evaluadas.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
simMag	94.7	[93.4, 96.0]	2.26	[1.57, 2.87]	[1.59, 2.85]
simFase	84.9	[82.8, 87.0]	5.61	[4.69, 6.41]	[4.76, 6.23]
simBS	92.3	[90.8, 93.8]	2.93	[2.29, 3.55]	[2.26, 3.51]
simPS	90.2	[88.5, 91.9]	4.23	[3.48, 4.76]	[3.51, 4.69]
simPI	91.2	[89.6, 92.8]	3.60	[2.77, 4.19]	[2.77, 4.10]

³Mejora de Error Relativa: $MER = \frac{EER - EER_{BASE}}{EER_{BASE}}$.

⁴Para la comparación de los tiempos de extracción se definió la una unidad temporal como el tiempo que demora en extraer los wavelets el juego Bio3, $t_{B3} = 0,32$ s.

En este caso los resultados son peores para todas las similitudes que consideran la fase. De todas ellas la que presenta el mejor desempeño es la que utiliza el mismo predictor que se utiliza para ajustar los grafos, en este caso el de escalas. La similitud que considera la fase sin realizar corrección de la posición es la que muestra los peores resultados.

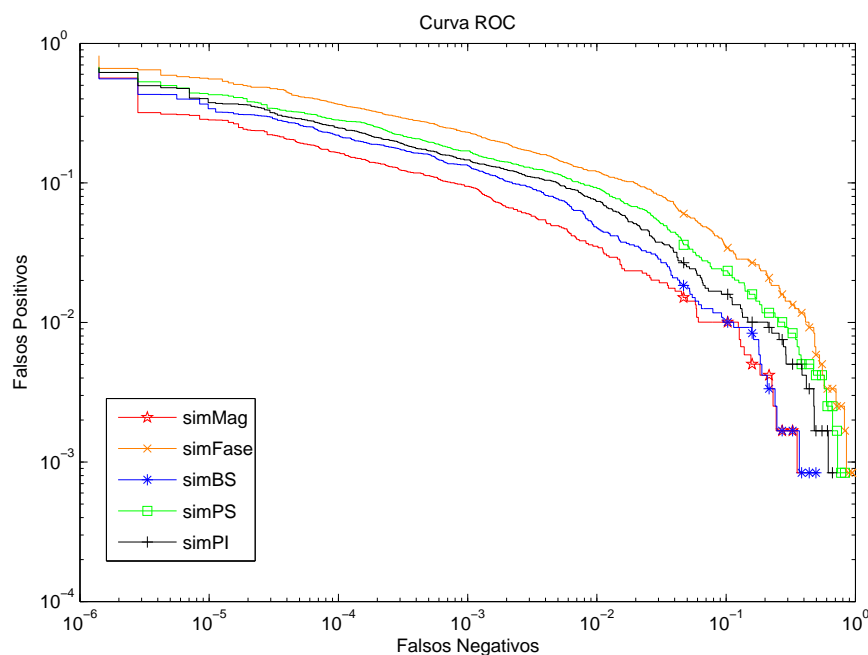


Figura 9.6: Curvas ROC para las similitudes evaluadas.

C.4. Similitudes Ponderadas Aquí se prueban algunas medidas de similitud que pesan de manera diferente la similitud en cada *jet*. Las mismas se detallan a continuación:

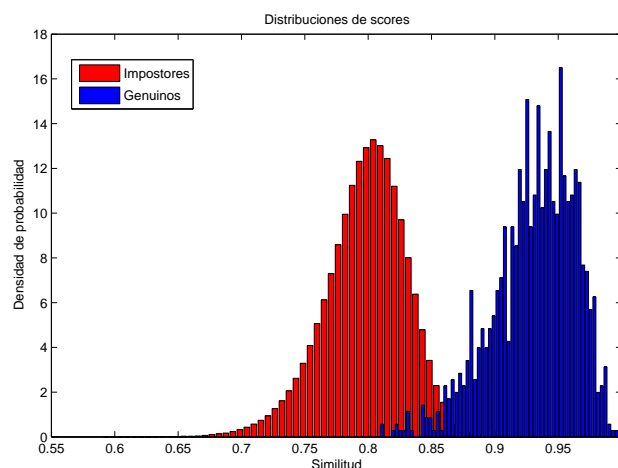
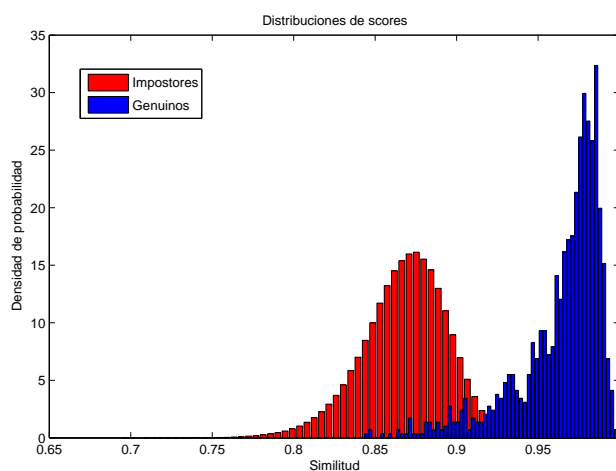
simMag - Similitud que considera el promedio de la similitud en cada punto (`fgSimMagnitude`).

simOrd - Similitud que considera el promedio de la similitud en cada punto, pero considerando solo los N (en este caso 30) puntos con similitud mayor (`fgSimMagOrd`).

simPeso - Similitud que considera el promedio ponderado de la similitud en cada punto, tomando pesos proporcionales al desempeño individual de cada punto (`fgSimMagPeso`).

En la tabla adjunta se presentan los resultados obtenidos.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
simMag	94.7	[93.4, 96.0]	2.26	[1.57, 2.87]	[1.59, 2.85]
simOrd	95.5	[94.3, 96.7]	5.52	[4.45, 6.42]	[4.46, 6.36]
simPeso	94.2	[92.8, 95.6]	2.76	[2.12, 3.46]	[2.13, 3.51]

(a) Similitud $fgSimMagnitude$.(b) Similitud $fgSimMagOrd$.Figura 9.7: Distribución de *scores* para ambas similitudes.

No se obtienen mejoras significativas para las variantes estudiadas en la similitud respecto al resultado que se obtiene al promediar la similitud de todos los puntos. Un detalle a resaltar es que para el caso en que se toma solo los puntos con mayor similitud se obtiene un buen resultado para la

identificación, como se observa en la tabla. Sin embargo, para verificación el desempeño no es bueno, como se aprecia en el EER.

En la figura 9.7 se muestra las distribuciones de *scores* para los ensayos `simMag` y `simOrd`. Podemos ver que en ambos casos la distribución correspondiente a impostores es más suave que la que corresponde a los genuinos por tener mayor cantidad de datos.

Como se observa, para el caso de tomar los puntos de mayor similitud aumenta la media de ambas distribuciones, tanto genuinos como impostores. Notar que en este caso los puntos no tienen por qué ser los mismos para cada pareja de imágenes. Esto se refleja en una distribución de similitudes más compleja. La misma se puede pensar como la combinación de las distribuciones correspondientes a cada selección de puntos posible. La caída del desempeño se explica claramente observando que las clases quedan más solapadas entre sí.

9.3.2. Elección del Sistema Óptimo

Observando el desempeño de cada uno de los bloques en la sección anterior, no se logra definir un único sistema óptimo. Dado que las diferencias son muy ajustadas, y por lo tanto mayores a la incertidumbre de cada prueba, se proponen dos sistemas a evaluar con todos los sets de la base FERET. Los mismos consisten en las siguientes configuraciones:

Configuración BS

Normalización - Idem al sistema base.

Localización - Idem al del sistema base, aunque la familia de wavelets utilizados es la Bio3 con la compensación para caras.

Extracción de características - Se utiliza el mismo grafo del sistema base y el juego de parámetros Bio3 con la compensación para caras.

Confrontación - Como medida de similitud se utiliza la misma que en el sistema base: Similitud de Magnitud.

Configuración NoFit

Normalización - Idem al sistema base.

Localización - El modelo es el mismo que el del sistema base. No se realiza ajuste, la ubicación de los puntos principales es tomada del modelo. La familia de wavelets utilizados es la Bio3 con la compensación para caras.

Extracción de características - Se utiliza el mismo grafo del sistema base y el juego de parámetros Bio3 con la compensación para caras.

Confrontación - Como medida de similitud se utiliza la misma que en el sistema base: Similitud de Magnitud.

En la figura 9.8 se observa que en los tests **FB** y **FC** el RR del sistema con ajuste es mejor al RR del sistema sin ajuste. En cambio, en los otros dos tests **DUP I** y **DUP II** ocurre lo opuesto. Cabe aclarar, que al ser las diferencias tan pequeñas, mayores a la incertidumbres de cada prueba, no se logra obtener una conclusión genérica de qué sistema es mejor. Sin embargo, se destaca que el sistema sin refinamiento es menos costoso computacionalmente y por lo tanto más rápido de ejecutar que el sistema con refinamiento.

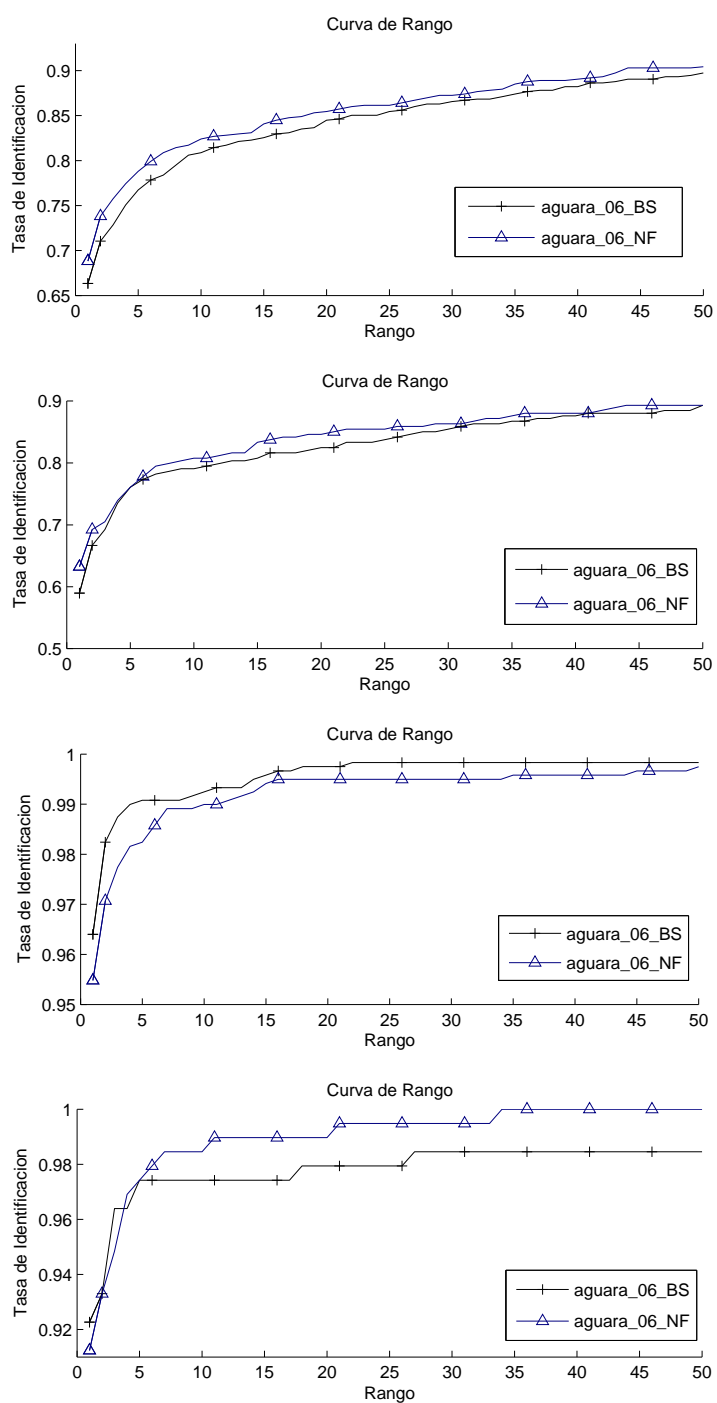


Figura 9.8: Curva de Rango comparando el desempeño de los dos sistemas propuestos en los sets **dup I**, **dup II**, **fb** y **fc** de la FERET.

9.3.3. Comparación con otros algoritmos

Con el objetivo de realizar una comparación de los resultados obtenidos en este trabajo, se presentan los mismos junto con el desempeño de otros algoritmos. Entre ellos los concursantes en el *FERET Test*⁵ [74, 75] y otros más recientes. Se comparan los siguientes algoritmos:

arl_cor - Army Research Laboratory - Baseline Correlation.

arl_ef - Army Research Laboratory - Baseline Eigenface [76], [9].

bayesian_03 - Bayesian Map (Teixeira 2003) [77]

bolme_03 - EBGM CSU-Bolme [11].

excalibur - Excalibur Inc.

gabor_02 - Gabor features - (Kepenekci et al.,2002) [78].

log-G_PCA_06 - Log-Gabor PCA - (V. Perlibakas) [79]

mit_sep_96 - MIT Media Lab [80].

aguará_06_BS - Configuración BS del sistema.

aguará_06_NF - Configuración NoFit del sistema.

umd_mar_97 - University of Maryland [81] [29].

usc_mar_97 - University of Southern California [10].

Cabe aclarar que no se encontraron los resultados completos de todos los algoritmos comparados en 9.3.3, lo cual impidió presentar sus resultados en forma gráfica.

En el cuadro se observa que los algoritmos propuestos tienen un buen desempeño en comparación al resto de los algoritmos. Su performance es mejor que la de *bolme_03* y *usc_mar_97*, ambas implementaciones de EBGM. La diferencia más importante se produce en el test **FC** donde el desempeño en identificación es superior al resto de los algoritmos. Además, los algoritmos propuestos tienen un desempeño aceptable en el set más duro, *dup-II*.

En las figuras 9.9, 9.10, 9.11 y 9.12 se muestra las curvas de rango (CMC) de los distintos algoritmos para las pruebas antes definidas.

⁵Desarrollado entre Setiembre de 1996 y Marzo de 1997.

Algoritmo	RR				EER			
	d-I	d-II	FB	FC	d-I	d-II	FB	FC
arl_cor	36.3	17.1	82.7	5.2	18.7	19.2	4.1	24.8
arl_uf	41.0	22.2	79.7	18.6	17.4	19.0	6.7	17.6
bayesian_03	52.4	31.6	81.9	37.1	-	-	-	-
bolme_03	46.3	24.4	89.8	41.8	-	-	-	-
excalibur	41.4	19.7	79.4	21.6	15.4	18.4	4.8	14.5
gabor_02	58.3	47.7	96.3	69.6	-	-	-	-
log-G_PCA	72.4	65.8	98.0	90.2	3.6	4.7	1.0	0.3
mit_sep_96	57.6	34.2	94.8	32.0	17.7	21.2	4.9	18.0
aguará_06_BS	66.3	59.0	96.4	92.3	11.2	12.4	1.70	3.6
aguará_06_NF	68.8	63.2	95.5	91.2	9.0	10.0	1.6	2.7
umd_mar_97	47.2	20.9	96.2	58.8	12.6	13.4	1.2	10.0
usc_mar_97	59.1	52.1	95.0	82.0	13.3	14.2	2.5	5.1

Tabla 9.1: Comparación del desempeño de diferentes algoritmos frente a los cuatro tests **DUP-I**, **DUP-II**, **FB** y **FC** presentados para la base FERET.

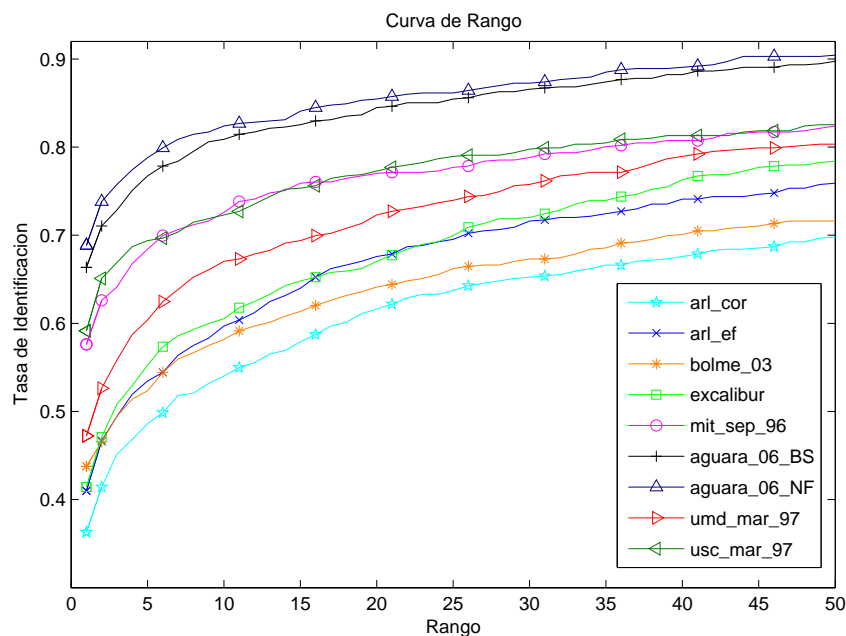


Figura 9.9: Curva de Rango comparando el desempeño de varios algoritmos en el set **dup-I**.

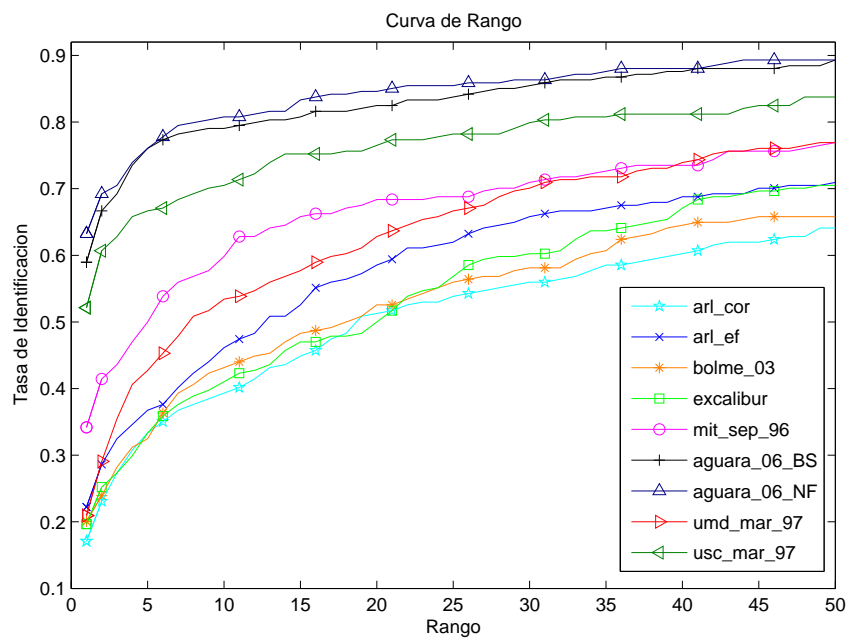


Figura 9.10: Curva de Rango comparando el desempeño de varios algoritmos en el set **dup-II**.

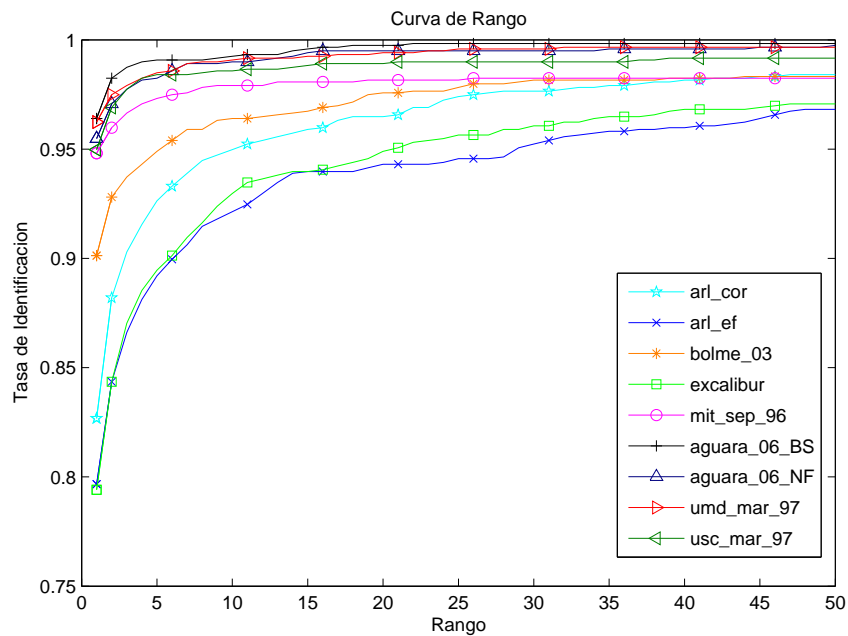


Figura 9.11: Curva de Rango comparando el desempeño de varios algoritmos en el set **FB**.

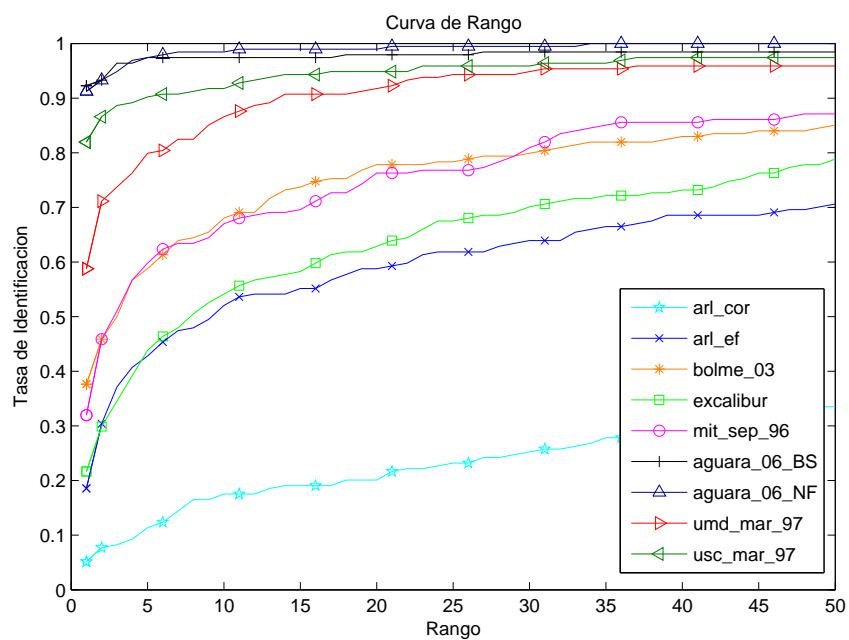


Figura 9.12: Curva de Rango comparando el desempeño de varios algoritmos en el set **FC**.

9.3.4. Evaluación del Sistema Automático

Luego de efectuar las distintas pruebas para el caso semi-automático, se realizan ensayos considerando el sistema punta a punta, con el bloque de detección automática de la cara. Para estas pruebas solamente se considera la configuración **BS** del sistema.

En este caso tendremos dos efectos superpuestos que implican que el desempeño sea peor. Por un lado la precisión con la que se ubican los ojos que afecta tanto la normalización como el ajuste de puntos posterior. Por otro lado existen *outliers* que corresponden a caras mal detectadas donde se obtiene similitudes muy bajas para usuarios genuinos que distorsionan las distribuciones de *scores*. No se consideran las imágenes donde no se detecta una cara puesto que en el capítulo 4 correspondiente al módulo de detección ya se evaluó el desempeño del detector de caras.

A. Base FERET

Se realiza la evaluación del sistema definido para el set **fb** de la base FERET. Los resultados se muestran en la tabla a continuación.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
Sistema	88.6	[86.8, 90.4]	5.50	[4.36, 6.63]	[4.42, 6.58]

Como podemos ver, el desempeño disminuye de manera considerable. En este caso se observó que la influencia de los *outliers* es mucho mayor que la del error cometido en la ubicación de los ojos.

B. Base IIE

En este caso uno de los objetivos era el de comprobar la generalidad del modelo estadístico. La idea era poder localizar puntos en imágenes de la base IIE con un *Face Bunch Graph* generado a partir de imágenes de la base FERET. Se observa que el ajuste tiene un desempeño similar al obtenido con el modelo generado con imágenes de la misma base. Este punto es importante puesto que avala la utilización de un modelo estadístico genérico que se adapta a cualquier base de prueba.

El otro objetivo era aprovechar la riqueza de la base adquirida en el IIE para evaluar el desempeño del sistema frente a distintas circunstancias. En la lista a continuación se resumen los cinco estudios efectuados:

Frente Se utilizan las imágenes de frente para evaluar el desempeño en condiciones óptimas.

Perfil Se utilizan las imágenes de perfil para evaluar la robustez frente a caras levemente rotadas.

Oclusión Se utilizan las imágenes con oclusiones para evaluar la robustez frente a caras parcialmente cubiertas.

Expresión Se utilizan las imágenes sonriente, sorprendido, enojado y con guiñada para evaluar la robustez frente a distintas expresiones faciales.

Luz Se utilizan las imágenes con luz lateral para evaluar la robustez frente a iluminación no uniforme.

B.1. Frente En un caso se utiliza como referencia una sola imagen de cada individuo y en el otro cuatro por persona. Se busca ver la influencia de disponer de más de una referencia para cada individuo. Las pruebas son las siguientes:

Frente_1 - Referencias: a . Prueba: u .

Frente_2 - Referencias: a . Prueba: x .

Frente_3 - Referencias: adj . Prueba: u .

Frente_4 - Referencias: adj . Prueba: x .

En la tabla se resume los resultados para el caso de frente.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
Frente_1	97.9	[93.8, 100]	4.35	[2.55, 8.08]	[2.13, 6.57]
Frente_2	90.9	[82.8, 99.0]	6.82	[1.06, 12.1]	[2.27, 11.1]
Frente_3	100	[99.0, 100]	1.20	[0.17, 3.18]	[0.19, 1.85]
Frente_4	100	[99.0, 100]	4.54	[0.00, 8.97]	[0.46, 8.44]

Con la base IIE los resultados son superiores a los que se obtienen con la FERET pero en este caso los intervalos de confianza son mayores debido a la menor cantidad de imágenes disponibles. También se observa que al disponer de mayor cantidad de referencias para cada individuo los resultados mejoran, llegando a clasificar correctamente el total de las imágenes.

B.2. Perfil Para las pruebas con imágenes de perfil se estudian dos casos: el primero con tomas de perfil obtenidas en la misma sesión fotográfica que las referencias y el segundo con imágenes tomadas una semana después de las referencias. En esta prueba se trabaja en ambos casos con cuatro referencias por individuo.

Perfil_1 - Referencias: *adjj*. Prueba: *bc*.

Perfil_2 - Referencias: *adjj*. Prueba: *vw*.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
Perfil_1	89.4	[83.2, 95.6]	22.8	[13.3, 29.3]	[14.9, 29.8]
Perfil_2	63.8	[50.2, 77.4]	25.5	[17.1, 36.1]	[18.9, 36.2]

En este caso se evidencia la falta de robustez del sistema frente a imágenes de perfil. Vale aclarar que la base del IIE presenta imágenes de perfil con rotaciones mayores a 10° , situación poco probable en un escenario real de una aplicación de control de acceso.

B.3. Oclusión También se realizan pruebas para las tomas con bufanda y anteojos negros como ejemplos de oclusiones posibles. En este caso se utilizan dos medidas de similitud de las estudiadas: el promedio de las similitudes tomando todos los puntos y por otro lado tomando solo los N puntos de mayor similitud. Nuevamente se tiene en este caso cuatro referencias por individuo.

Oclus_1 - Referencias: *adjj*. Prueba: *qs*.

Oclus_2 - Referencias: *adjj*. Prueba: *q*.

Oclus_3 - Referencias: *adjj*. Prueba: *s*.

simMag	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
Oclus_1	93.3	[88.3, 98.3]	11.1	[4.38, 18.3]	[5.93, 17.8]
Oclus_2	92.3	[84.8, 99.8]	23.2	[11.3, 30.6]	[13.0, 28.2]
Oclus_3	93.0	[85.9, 100]	10.4	[4.46, 16.3]	[4.65, 16.3]

simOrd	Identificación		Verificación		
	Test	RR(%)	ICP_R	EER(%)	ICP_E
Oclus_1	97.8	[94.9, 100]	6.67	[1.45, 11.9]	[2.22, 11.1]
Oclus_2	100	[99.0, 100]	1.40	[0.00, 4.57]	[0.00, 2.07]
Oclus_3	97.7	[93.4, 100]	6.98	[1.53, 12.4]	[2.32, 11.6]

Los resultados obtenidos son mejores que para el caso de perfil, lo cual a priori podría no ser lo que se espera. Sin embargo, al analizar lo que ocurre, se observa que en este caso los puntos que no están ocultos se encuentran en similares condiciones que para el caso de las imágenes de frente sin oclusiones. Por lo tanto todos los puntos que no están cubiertos presentan similitudes altas. Para el caso de imágenes de perfil todos los puntos sufren una transformación que implica que la similitud con las referencias de frente disminuya. Lo anterior se reafirma al observar los resultados de la prueba que considera solamente los puntos de mayor similitud.

B.4. Expresión Se estudia el desempeño del sistema frente a tomas con distintas expresiones faciales. En este caso se estudia la robustez frente a las imágenes con sonrisa, sorpresa, enojo y guiñada.

Expres_1 - Referencias: *adjj*. Prueba: *m*.

Expres_2 - Referencias: *adjj*. Prueba: *n*.

Expres_3 - Referencias: *adjj*. Prueba: *o*.

Expres_4 - Referencias: *adjj*. Prueba: *p*.

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
Expres_1	100	[98.9, 100]	2.50	[0.00, 6.88]	[0.65, 6.67]
Expres_2	93.5	[86.2, 100]	7.12	[1.11, 14.1]	[2.17, 13.0]
Expres_3	95.7	[89.7, 100]	8.60	[0.51, 14.8]	[2.17, 13.0]
Expres_4	95.1	[88.4, 100]	10.3	[2.69, 16.8]	[4.60, 17.0]

Si bien las tasas de identificación (RR) obtenidas son buenas, se observa que el desempeño en verificación empeora. Esto se debe nuevamente a la variación que se produce en los *scores*, lo que hace que las clases genuinos e impostores estén más mezcladas. La mayor robustez del sistema se observa para caras sonrientes.

B.5. Luz Por último se realizan pruebas para la toma con luz lateral para estudiar el efecto de la iluminación no uniforme.

Luz - Referencias: *adj.* Prueba: *t.*

Test	Identificación		Verificación		
	RR(%)	ICP_R	EER(%)	ICP_E	ICB_E
Luz	91.3	[83.0, 99.6]	6.52	[1.41, 10.0]	[2.17, 8.70]

Se observa un desempeño aceptable teniendo en cuenta la dificultad de la prueba realizada.

En todos los casos en los que se enriquece el registro de las identidades con más de una imagen por persona en la base de datos se observa un mejor desempeño del algoritmo. Para aplicaciones reales esto podrá ser explotado siempre y cuando el aumento del costo de procesamiento permita alcanzar los requerimientos particulares de tiempos de estos sistemas.

Capítulo 10

Conclusiones y Trabajo Futuro

10.1. Conclusiones

En este proyecto se presentó un estudio del área del Reconocimiento de Caras. Se realizó una importante búsqueda bibliográfica sobre las variadas técnicas existentes, intentando además introducirse en problemas relacionados tales como la Detección de Caras, y la Clasificación de Patrones. Además, se profundizó en el estudio del algoritmo EBGM el cual utiliza descriptores de Gabor locales [10]. Para ello se partió de una implementación en lenguaje de programación C/C++ existente [11]. Se logró mejorar el desempeño de dicha implementación, así como también, superar el desempeño del algoritmo original [10].

El estudio del área de interés se realizó con la intención de recopilar información sintética y concreta sobre los principales métodos existentes para la detección y el reconocimiento de caras.

Se realizó estudios sobre distintas formas de mejorar la implementación del algoritmo considerado. Entre ellas, se destaca un estudio estadístico para compensar la potencia de los descriptores locales de Gabor con el objetivo de aplicarlos al reconocimiento de caras. Los buenos resultados obtenidos, remarcan la importancia de las conclusiones de dicho análisis.

Por otro lado, se desarrolló un Sistema de Reconocimiento de Caras que resuelve la adquisición de la imagen, la detección de la cara, la extracción de características y finalmente el reconocimiento del individuo. El mismo, se utilizó como base para el desarrollo de un prototipo de aplicación de reconocimiento de personas. La aplicación desarrollada en C/C++ con código libre, permite ejecutar las principales operaciones de un sistema de estas características. El diseño modular implementado a través de bibliotecas, permite sustituir cualquiera de los bloques funcionales, en caso de así necesitarlo.

El desarrollo de una base de imágenes propia, la base IIE, así como también la obtención de la base de imágenes estándar FERET permitió la evaluación del sistema desarrollado. Los buenos resultados obtenidos mediante la evaluación con la base estándar, colocan este sistema cerca de los mejores resultados existentes. Además, el desarrollo de la base propia permitió la evaluación de la aplicación simulando un sistema de control de acceso. En todos los casos, se realizó un estudio metodológico, haciendo especial énfasis en las incertidumbres de las distintas pruebas.

10.2. Trabajo a Futuro

Desde el punto de vista de la mejora del algoritmo EBGM han surgido importantes elementos a tratar y alternativas a evaluar que no han sido consideradas en el presente trabajo. A continuación se mencionan las más importantes.

Es importante que para que el algoritmo presente mayor robustez se continúe trabajando en las técnicas que contemplan caras rotadas. En ese sentido se puede seguir explorando la fusión de modelos con imágenes en distintas poses y se puede trabajar en la compensación de jets para confrontar caras con diferentes rotaciones.

En la extracción de jets es importante estudiar cuantitativamente el aporte para el reconocimiento de las distintas frecuencias y orientaciones de los filtros de Gabor, con el objetivo de dar diferente peso según su poder de discriminación.

En la etapa de confrontación se plantea la necesidad de seguir trabajando con herramientas de reconocimiento de patrones para procesar los vectores de características de muy alta dimensión que devuelve el algoritmo.

También existe la posibilidad de explorar técnicas de preprocesado de las imágenes que resalten las características de Gabor como descriptores locales.

Finalmente, respecto al desarrollo de una aplicación orientada al control de acceso, se observa que los tiempos de ejecución y las tasas de reconocimiento del sistema implementado permiten una aplicación en tiempo real, con un nivel de confiabilidad aceptable para situaciones de ambiente controlado. Sin embargo existe una brecha entre el prototipo desarrollado y lo que pudiese ser una aplicación del tipo comercial. De todas maneras se considera que el mismo es una buena base para el desarrollo de un producto de estas características.

Bibliografía

- [1] Biometric Watch. Journal for the Biometric Industry, última actualización: 18 de Setiembre de 2006. <http://www.biometricwatch.com/index.html>.
- [2] Wikipedia. Biometría — Wikipedia, la enciclopedia libre, 2006. [Internet; descargado 3-octubre-2006].
- [3] G. Lovell. Face recognition. *Cognitive Psychology Course, Tutorial Handouts*, 1998.
- [4] Julio Cazabán, Adrián Fernández, and José Mazzeo. Deciclas: Clasificación de fichas decidactilares. Facultad de Ingeniería, Universidad de la República, Uruguay, 2002.
- [5] Gonzalo Vallarino, José Luis Barattini, and Gustavo Gianarelli. Índice de calidad y confrontación de huellas dactilares. Facultad de Ingeniería, Universidad de la República, Uruguay, 2004.
- [6] National Institute of Standards & Technology (NIST). The Facial Recognition Technology (FERET) Database, 1993-1997. http://www.itl.nist.gov/iad/humanid/feret/feret_master.html.
- [7] Andrew Senior, Rein-Lien Hsu, Mohamed Abdel Mottaleb, and Anil K. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.
- [8] I. Fasel, R. Dahl, J. Hershey, B. Fortenberry, J. Susskind, and J. R. Movellan. The Machine Perception Toolbox, última actualización: 19 de Enero de 2005. <http://mplab.ucsd.edu/grants/project1/free-software/mptwebsite/introduction.html>.
- [9] Matthew Turk and Alex Paul Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [10] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. In Gerald Sommer, Kostas Daniilidis, and Josef Pauli, editors, *Proceedings*

- of the 7th International Conference on Computer Analysis of Images and Patterns, CAIP '97, Kiel*, number 1296, pages 456–463, Heidelberg, Germany, 1997. Springer-Verlag.
- [11] David S. Bolme. Elastic Bunch Graph Matching. Master's thesis, Colorado State University, Fort Collins, Colorado, USA, 2003.
- [12] D.S. Bolme, J.R. Beveridge, M. Teixeira, and B.A. Draper. Evaluation of face recognition algorithms, última actualización: 20 de Abril de 2004. <http://www.cs.colostate.edu/evalfacerec/algorithms5.html>.
- [13] Anil K. Jain and David Maltoni. *Handbook of Fingerprint Recognition*, chapter 1. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
- [14] Anil K. Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.
- [15] Wikipedia. Juan Vucetich — Wikipedia, la enciclopedia libre, 2006. [Internet; descargado 3-octubre-2006].
- [16] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.
- [17] BioID HumanScan. About FAR, FRR and EER, última actualización: 3 de Marzo de 2004. http://www.bioid.com/sdk/docs/About_EER.htm.
- [18] Biometric System Laboratory University of Bologna. Fingerprint Verification Competition (FVC), 2006. <http://bias.csr.unibo.it/fvc2006/>.
- [19] National Institute of Standards & Technology (NIST). Face Recognition Vendor Test (FRVT), 2006. <http://www.frvt.org/FRVT2006/>.
- [20] I. Biederman and P. Kalocsai. Neural and psychophysical analysis of object and face recognition. In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogeman Soulié, and T. S. Huang, editors, *Face Recognition: From Theory to Applications*, pages 3–25. Springer-Verlag, 1998.
- [21] P. Thompson. Margaret Thatcher: a new illusion. *Perception*, 9(4):483–484, 1980.
- [22] A. J. Goldstein, L. D. Harmon, and A. B. Lesk. Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760, 1971.
- [23] T. Kohonen. *Self-organization and associative memory*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.
- [24] M. Kirby and L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

- [25] M-H. Yang, D.J Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, January 2002.
- [26] Erik Hjelmas and Boon Kee Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3):236–274, 2001.
- [27] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.
- [28] C. Havran, L. Hupet, J. Czyz, J. Lee, L. Vandendorpe, and M. Verleysen. Independent component analysis for face authentication. In *Proceedings of the 6th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, KES '02*, September 2002.
- [29] W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *Proceedings of the 3rd International Conference on Automatic Face and Gesture Recognition, FG '98*, pages 336–341, Nara, Japan, April 14-16 1998.
- [30] K. Delac, M. Grgic, and S. Grgic. Independent comparative study of PCA, ICA, and LDA on the FERET data set. *International Journal of Imaging Systems and Technology*, 15(5):252–260, 2005.
- [31] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
- [32] Ming-Hour Yang. Face recognition using kernel methods. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Proceedings of the Neural Information Processing Systems Conference, NIPS'01*, volume 14, pages 1457–1464, Vancouver, British Columbia, Canada, December 3-8 2001. MIT Press.
- [33] Jianke Zhu, Mang I Vai, and Peng Un Mak. Face recognition, a kernel PCA approach. In *Proceedings of the Chinese Conference on Medicine and Biology, CMBE '03*, Wuxi, P. R. China, October 24-26 2003.
- [34] Pablo Navarrete and Javier Ruiz del Solar. Eigenspace-based recognition of faces: Comparisons and a new approach. In *Proceedings of the 11th International Conference on Image Analysis and Processing, ICIAP '01*, page 42, Palermo, Italy, September 26-28 2001.
- [35] Chengjun Liu and Harry Wechsler. Evolutionary pursuit and its application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):570–582, 2000.

- [36] Baback Moghaddam, Tony Jebara, and Alex Pentland. Bayesian face recognition. *Pattern Recognition*, 33(11):1771–1782, 2000.
- [37] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [38] G. Guo, S. Li, and K. Chan. Face recognition by support vector machines. In *Proceedings of the International Conferences on Automatic Face and Gesture Recognition, FG '00*, pages 196–201, 2000.
- [39] K. Jonsson, J. Matas, J. Kittler, and Y. Li. Learning support vectors for face verification and recognition. In *Proceedings of the International Conferences on Automatic Face and Gesture Recognition, FG '00*, pages 208–213, 2000.
- [40] Gareth J. Edwards, Timothy F. Cootes, and Christopher J. Taylor. Face recognition using active appearance models. In *Proceedings of the 5th European Conference on Computer Vision, ECCV '98*, volume 2, pages 581–595, Freiburg, Germany, June 2-6 1998. Springer-Verlag.
- [41] T. F. Cootes, K. Walker, and C. J. Taylor. View-based active appearance models. In *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, FG '00*, pages 227–232, Grenoble, France, 2000.
- [42] R. Hsu and A. Jain. Face modeling for recognition. In *IEEE International Conference on Image Processing, ICIP '01*, volume 2, pages 693–696, February 2001.
- [43] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.
- [44] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. In L. C. Jain, U. Halici, I. Hayashi, and S. B. Lee, editors, *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, chapter 11, pages 355–396. CRC Press, 1999.
- [45] A. Nefian. *A hidden Markov model-based approach for face detection and recognition*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 1999.
- [46] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, chapter 6, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

- [47] Topi Mäenpää and Matti Pietikäinen. Texture analysis with local binary patterns. In *Image Analysis*, chapter 1. Springer Berlin, Heidelberg, Germany, 2004.
- [48] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution grayscale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.
- [49] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *Proceedings of the 8th European Conference on Computer Vision, ECCV '04*, pages 469–481, Prague, Czech Republic, May 11-14 2004.
- [50] Paul T. Jackway and Mohamed Deriche. Scale-space properties of the multiscale morphological dilation-erosion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):38–51, 1996.
- [51] Alumni Association University of Nebraska. The Champion Database, última visita: 23 de Mayo de 2005. http://www.libfind.unl.edu/alumni/events/breakfast_for_champions.htm.
- [52] I. Fasel, B. Fortenberry, and J. Movellan. A generative framework for real time object detection and classification. *Computer Vision and Image Understanding*, (98):182–210, 2005.
- [53] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 2(57):137–154, 2004.
- [54] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(20):23–28, 1998.
- [55] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Department of Statistics, Stanford University, Technical Report*, 1998.
- [56] Laboratory of Computer and University of Helsinki Finland Information Science, Neural Networks Research Centre. FASTICA (GPL) package for Matlab, última actualización: 19 de Octubre de 2005. <http://www.cis.hut.fi/projects/ica/fastica/>.
- [57] Bruce A. Draper, Kyungim Baek, Marian Stewart Bartlett, and J. Ross Beveridge. Recognizing faces with PCA and ICA. *Computer Vision and Image Understanding*, 91(1-2):115–137, 2003.
- [58] M. Welling. Fisher linear discriminant analysis. *Department of Computer Science, University of Toronto, Technical Report*.

- [59] David J. Field. Relations between the statistics of natural images and the response profiles of cortical cells. In *Journal of the Optical Society of America.*, volume 4, pages 2379–2394, 1987.
- [60] Martin Lades, Jan C. Vorbrüggen, Joachim Buhmann, J. Lange, Christoph von der Malsburg, Rolf P Würtz, and Wolfgang Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311, 1993.
- [61] Arjen van der Schaaf. Modelling the power spectra of natural images: Statistics and information. In *Natural image statistics and visual processing*, chapter 2. Graduate School for Behavioural and Cognitive Neurosciences, University of Groningen.
- [62] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition, FG '98*, page 300, Nara, Japan, April 14-16 1998.
- [63] T. Cootes and C. Taylor. Active shape models - smart snakes. In *Proceedings of the British Machine Vision Conference*, page 266. Springer-Verlag, 1992.
- [64] T. Cootes and C. Taylor. Statistical models of appearance for computer vision. *Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering, University of Manchester, Technical Report*, March 8 2004.
- [65] T. Cootes. Active Appearance Models - Modelling and Search Software, última actualización: 9 de noviembre de 2004. http://www.isbe.man.ac.uk/~bim/software/am_tools_doc/index.html.
- [66] Kazunori Okada, Johannes Steffens, Thomas Maurer, Hai Hong, Egor Elagin, Hartmut Neven, and Christoph von der Malsburg. The BOCHUM/USC Face Recognition System and how it fared in the FERET Phase III test. In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogeman Soulié, and T. S. Huang, editors, *Face Recognition: From Theory to Applications*, pages 186–205. Springer-Verlag, 1998.
- [67] Thomas Maurer and Christoph von der Malsburg. Single-view based recognition of faces rotated in depth. In *Proceedings of the International Workshop on Automatic Face and Gesture Recognition, FG '95*, pages 248–253, Zurich, Switzerland, June 1995.
- [68] B. Gokberk, M.O. Irfanoglu, L. Akarun, and E. Alpaydin. Optimal gabor kernel location selection for face recognition. In *Proceedings of the International Conference on Image Processing, ICIP '03*, volume 1, pages 677–680, Barcelona, Spain, 2003.

- [69] N. Kruger. An algorithm for the learning of weights in discrimination functions using a priori constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):764–768, July 1997.
- [70] Juwei Lu, Kostantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Face recognition using LDA-based algorithms. *IEEE Transactions on Neural Networks*, 14(1):195–200, 2003.
- [71] F. Chen, H.Y. Mark Liao, M.T. Ko, J.C. Lin, and G.J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33:1713–1726, 2000.
- [72] R. Lotlikar and R. Kothari. Fractional-step dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):623–627, 2000.
- [73] Ruud M. Bolle, Sharath Pankanti, and Nalini K. Ratha. Evaluation techniques for biometrics-based authentication systems (FRR). In *Proceedings of the 15th International Conference on Pattern Recognition, ICPR '00*, volume 02, page 2831, Barcelona, Spain, September 03-08 2000.
- [74] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.
- [75] S. Rizvi, P. Phillips, and H. Moon. The FERET verification testing protocol for face recognition algorithms. *Image and Vision Computing Journal*, 1999.
- [76] H. Moon and P.J. Phillips. Analysis of PCA-based face recognition algorithms. In *Workshop on Empirical Evaluation Methods in Computer Vision*, Santa Barbara, California, June 1998.
- [77] Marcio Luis Teixeira. The bayesian intrapersonal/extrapersonal classifier. Master's thesis, Colorado State University, Fort Collins, Colorado, USA, 2003.
- [78] Burcu Kepenekci, F. Boray Tek, and Gozde Bozdagi Akar. Occluded face recognition by using gabor features. In *3rd COST 276 Workshop on Information and Knowledge Management for Integrated Media Communication*, Budapest, October 2002.
- [79] Vytautas Perlibakas. Face recognition using principal component analysis and log-gabor filters. *Image Processing and Analysis Laboratory, Computational Technologies Centre*, 2006.

- [80] Baback Moghaddam, Chahab Nastar, and Alex Pentland. Bayesian face recognition using deformable intensity surfaces. In *Proceedings of the 15th Computer Vision and Pattern Recognition Conference, ICVPR '96*, 1996.
- [81] Kamran Etemad and Rama Chellappa. Discriminant analysis for recognition of human face images (invited paper). In *Proceedings of the 1st International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA '97*, pages 127–142, London, UK, 1997. Springer-Verlag.
- [82] Ralph Gross. Face databases. In S.Li and A.Jain, editors, *Handbook of Face Recognition*. Springer, New York, February 2005.
- [83] National Institute of Standards & Technology (NIST). Best practice recommendation for the capture of mugshots, version 2.0, september 23, 1997, última actualización: 9 de Abril de 2003. http://www.itl.nist.gov/iad/894.03/face/bpr_mug3.html.
- [84] J. Black, M. Gargesha, K. Kahol, P. Kuchi, and S. Panchanathan. A framework for performance evaluation of face recognition algorithms. In *Proceedings of the ITCOM Conference on Internet Multimedia Systems II*, July 2002.
- [85] National Institute of Standards & Technology (NIST). FERET Evaluation, última actualización: 7 de Junio de 2001. <http://www.itl.nist.gov/iad/humanid/feret/perf/eval.html>.

Parte IV

Apéndices

Apéndice A

Biblioteca Desarrollada

Se presenta las interfases de la biblioteca que implementa el sistema de reconocimiento de caras. Esta biblioteca surge del modelo conceptual del sistema y en base al análisis del código disponible. A partir de las funcionalidades que debe cumplir cada uno de los bloques, se define los métodos que implementan los distintos módulos del sistema. La interfaz fue concebida como un API¹ para el desarrollo de aplicaciones de reconocimiento de caras.

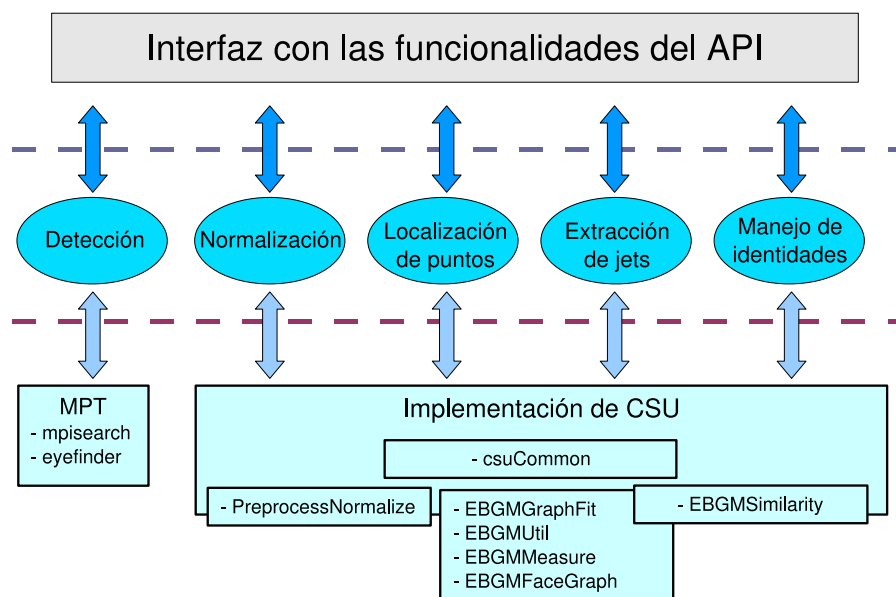


Figura A.1: Representación en capas de la biblioteca implementada.

¹Application Programming Interface - Interfaz para Programación de Aplicaciones.

En la figura A.1 se muestra el diseño realizado, así como el código disponible reutilizado. Se hace especial énfasis en la abstracción del código que implementa la funcionalidad de cada bloque, quedando éste como aspecto interno a la biblioteca.

En base al análisis realizado y tomando como base el código existente, se trabajó en un nuevo código hasta lograr una implementación orientada a nuestras necesidades. La biblioteca finalmente desarrollada consta de los siguientes módulos:

- `libdetect`
- `libPreprocessNormalize`
- `libcsuFaceGraph`
- `libdidentidad`
- `libcsu2magick`
- `libcsuCommon`

Se presenta cada uno de los módulos con el siguiente esquema:

1. Breve descripción de la funcionalidad de la biblioteca.
2. Enumeración de las estructuras de datos y clases que utiliza o define.
3. Enumeración de los métodos que define la biblioteca.
4. Entradas y salidas de cada método.

A.1. Detección: `libdetect`

Este módulo contiene los métodos que implementan la detección de una cara en una imagen y la ubicación de los ojos en la misma. Utiliza los módulos auxiliares `libmpisearch` y `libeyefinder`, heredados del *Machine Perception Toolbox* [8] con pequeñas modificaciones. A su vez estos módulos utilizan la biblioteca `ImageMagick` para el tratamiento de las imágenes.

Estructuras de datos:

```
struct coordenada {
    float x;
    float y;
};
```

```

struct ojos {
    coordenada izq;
    coordenada der;
};

struct cara {
    coordenada izqsup;
    coordenada derinf;
};

```

Métodos:

```

[cara] detectarcara(Image image);

--> image : imagen de la clase Image de ImageMagick.
<-- [cara] : estructura definida en esta biblioteca.

[ojos] detectarojos(Image image);

--> image : imagen de la clase Image de ImageMagick.
<-- [ojos] : estructura definida en esta biblioteca.

```

A.2. Preprocesado y Normalización:

libPreprocessNormalize

Este módulo contiene un método que normaliza una imagen a partir de las coordenadas de los ojos en la misma. La implementación de este módulo se toma del sistema CSU [12] con modificaciones menores y un módulo de CSU para el tratamiento de las imágenes.

Estructuras de datos:

```

struct TwoPoints {
    double x1, y1, x2, y2;
};

struct NormaParam {
    int histType;
    int pixelNorm;
    int preNormType;
    int maskType;
};

```

Métodos:

```
[Image] Normaliza_Imagen(Image imagen, TwoPoints eyes, ...
..., NormaParam* param);
```

--> imagen : imagen de la estructura Image de csuCommon.

--> eyes : coordenadas de los ojos.

--> param : parámetros de normalización.

←-[Image] : imagen normalizada de la estructura Image de csuCommon.

A.3. Localización y Extracción de Jets:**libcsuFaceGraph**

El módulo de localización y extracción de jets se encarga de la obtención de los grafos a partir de las imágenes normalizadas. Permite además generar el modelo estadístico a partir de un conjunto de imágenes y las marcas de los puntos principales. Los métodos y las estructuras de datos se heredan de la implementación CSU [12], con modificaciones realizadas para adaptarlo a nuestras necesidades.

Estructuras de datos:

```
struct Edge {
    int vert1, vert2;
};

struct Vert {
    double x, y;
};

struct graph_discription {
    int numVert; /*número de nodos del grafo*/
    int numEdge; /*número de aristas del grafo*/
    Vert* verts;
    Edge* edges;
    char** vertLabels;
    JetBunch* bunch;
};

typedef graph_discription *GraphDiscription;

struct jet_masks {
```



```

    int size;
    Image *masks;
    GaborJetParams params;
};

typedef jet_masks* JetMasks;

struct face_graph {
    int geosize; /*número de puntos localizados*/
    int totalsize; /*número de puntos localizados e interpolados*/
    GaborJetParams params;
    GaborJet* jets;
};

typedef face_graph* FaceGraph;

```

Métodos:

```

[void] generoModeloEstadistico (char* masksFile, ...
char* modelFile, char* graphDir, char* imageDir, ...
..., char* bunchFile);

```

--> masksFile : puntero apuntando a archivo conteniendo las máscaras de convolución.

--> modelFile : puntero apuntando a archivo conteniendo la lista de imágenes modelo.

--> graphDir : puntero apuntando a directorio donde se guardará el archivo de salida.

--> imageDir : puntero apuntando al directorio donde se encuentran las imágenes modelo.

--> bunchFile : puntero apuntando a archivo que contiene el modelo estadístico generado.

←-- [void]: vacío.

```

[GraphDiscription] AjusteGrafo(Image novelImage, ...,
char* bunchFile, GraphDiscription graphTemplate, ...
..., JetMasks masks, JetDisplacementEstimator dispEst);

```

--> novelImage : imagen a la que se le ajustará el grafo.

--> bunchFile : puntero apuntando a archivo que contiene el modelo estadístico generado.

--> graphTemplate : estructura conteniendo el modelo de grafo utilizado para el ajuste.

--> masks : estructura conteniendo las máscaras de convolución.

```
--> dispEst : alias de función, que estima la similitud entre dos gabor jets.
<-- [GraphDiscription] : estructura conteniendo el grafo ajustado.
```

```
[FaceGraph] ExtraerJets(Image novelImage, ...
..., GraphDiscription gd, JetMasks masks);
```

```
--> novelImage : imagen a la cual le quiero calcular los jets.
--> gd : estructura conteniendo el grafo ajustado a esa imagen.
--> masks : estructura conteniendo las máscaras de convolución.
<--[FaceGraph] : estructura conteniendo el FaceGraph de la imagen de
entrada (jets).
```

```
[double] CalculoDistancia(FaceGraphSimilarity distMeasure, ...
..., FaceGraph f1, FaceGraph f2);
```

```
--> distMeasure : alias de función que calcula la distancia entre dos grafos.
--> f1 : grafo de entrada 1.
--> f2 : grafo de entrada 2.
<-- [double] : número real de doble precisión (distancia).
```

A.4. Manejo de Identidades: libidentidad

El módulo `libidentidad` implementa lo relacionado con la verificación e identificación así como el enrolado de las personas en la base de datos. La solución que se encontró razonable para este proyecto fue trabajar con un sistema de archivos y directorios a modo de base de datos. Este módulo se implementó en su totalidad en este proyecto.

Estructuras de datos:

```
typedef FaceGraph grafo;
```

Métodos:

```
[bool] verificar(int id, grafo grafoin, const char *rutabase);
```

```
--> id : número de cuatro cifras asociado a una identidad de la base.
--> grafoin : grafo obtenido de la imagen que ingresa al sistema.
--> rutabase : apuntador al directorio de la base de datos.
<-- [bool] : true si verifica, false si no verifica.
```

```
[int] identificar(grafo grafoin, const char *rutabase);
```

```
--> grafoin : grafo obtenido de la imagen que ingresa al sistema.
--> rutabase : apuntador al directorio de la base de datos.
<-- [int] : devuelve la identidad si hubo identificación.
```

```
[void] enrollar(int id, grafo grafoin, const char *rutabase);
```

--> id : número de cuatro cifras asociado a una identidad de la base.

--> grafoin : grafo obtenido de la imagen que ingresa al sistema.

--> rutabase : apuntador al directorio de la base de datos.

<-- [void] : vacío.

```
[void] borrar(int id, const char *rutabase);
```

--> id : número de cuatro cifras asociado a una identidad de la base.

--> rutabase : apuntador al directorio de la base de datos.

<-- [void] : vacío.

A.5. csu2magick

Este módulo se encarga de manejar la conversión entre la clase imagen de la biblioteca `ImageMagick` utilizada en la etapa de detección y la definida en el sistema CSU utilizada en el resto de los módulos. Se escribió un módulo que implementa los métodos para pasar de una definición a otra y viceversa, para así compatibilizar el uso de las dos definiciones.

Métodos:

```
[Magick::Image] csu2magick(Image csuIm);
```

--> csuImage : imagen de la estructura `Image` de `csuCommon`.

<-- [Magick::Image] : imagen de la clase `Image` de `ImageMagick`.

```
[Image] magick2csu(Magick::Image MagickIm);
```

--> `Magick::Image` : imagen de la clase `Image` de `ImageMagick`.

<-- [Image] : imagen de la estructura `Image` de `csuCommon`.

A.6. csuCommon

Incluye las funcionalidades que no están en ninguno de los módulos anteriores pero que son utilizadas por uno o varios de los mismos. En este caso no se describen los métodos y tipos de datos dado que no tienen una interfaz como API a nivel de usuario como el resto de los módulos. Dicho de otra manera, los usuarios de este módulo son el resto de los módulos descritos anteriormente.

Apéndice B

Interfaz de Usuario

Se explica brevemente el funcionamiento del prototipo desarrollado en el marco del Proyecto. Se presenta las dos versiones disponibles. Una versión de línea de comando y una gráfica que cuenta con un manual de usuario.

B.1. Versión de Línea de Comando

El programa tiene cuatro funcionalidades principales:

- **Enrolar** : Se ingresa un ID de usuario (número entero de hasta cuatro cifras) seguido por la ruta a una imagen en algún formato estándar (por ejemplo: JPG, PGM, TIFF). El programa enrola esa imagen asociándola con la ID de usuario ingresada.
- **Eliminar** : Se ingresa un ID de usuario (número entero de hasta cuatro cifras), el cual se elimina de la base de personas enroladas. Se elimina la totalidad de registros de ese usuario.
- **Verificar**: Se ingresa un ID de usuario (número entero de hasta cuatro cifras) seguido por la ruta a una imagen en algún formato estándar (por ejemplo: JPG, PGM, TIFF). El programa verificará esa imagen asociándola con la ID de usuario ingresada. El sistema decide si el usuario corresponde con el registro de ID y valida o no el acceso.
- **Identificar**: Se ingresa la ruta a una imagen en algún formato (por ejemplo: JPG, PGM, TIFF). El programa identifica esa imagen pudiendo asociarla a alguna de las identidades enroladas en la base de datos del sistema.

A continuación se muestra la forma en que deben ser ingresados los comandos.

Ayuda - Aguará Consola v1.0

Uso: ./aguara_consola [opciones]

Las opciones válidas son:

```
--verif id archivoImagen, Para verificar el usuario id.  
  
--ident archivoImagen, Para identificar un usuario.  
  
--cerca archivoImagen, Para listar las identidades más  
cercanas a un usuario.  
  
--enrol id archivoImagen coords, Para enrolar un grafo  
del usuario id.  
  
--elim id, Para eliminar el usuario id.  
  
--help, muestra esta ayuda.
```

Para el correcto funcionamiento debe existir el archivo `aguara.cfg` con la configuración del programa.

B.2. Interfaz de Usuario: Manual Básico

Con el objetivo de un uso amigable del sistema se ha desarrollado una interfaz gráfica, de ahora en más denominada IGU (Interfaz gráfica de Usuario). Este programa soporta las funcionalidades del sistema básico que corre desde consola, permitiendo el enrolamiento, la verificación, la identificación y el borrado de individuos de la base de datos del sistema. Son dos los modos de funcionamiento previstos: Modo Captura y Modo Lectura. El usuario debe elegir el modo de funcionamiento con el cual quiere trabajar.

El Modo Captura habilita a trabajar con la adquisición de imágenes en tiempo real a través de un dispositivo tipo webcam. El Modo Lectura habilita a trabajar con archivos de imágenes guardados en alguna dirección de

almacenamiento del PC en el que corre la aplicación.

Se describe a continuación los módulos que componen la interfaz y los modos de uso para los cuales ha sido diseñada.

B.2.1. Elementos que Componen la Interfaz Gráfica

La ventana de la interfaz se encuentra dividida en seis sectores o cuadros diferenciados, tres de ellos en la parte superior y los otros tres en la parte inferior:

- Menú Principal
- Despliegue de Video
- Captura de Imagen
- Modos y botón de Captura
- Ajustes de Señal de Video
- Funcionalidades del Sistema
- Botones de Uso General

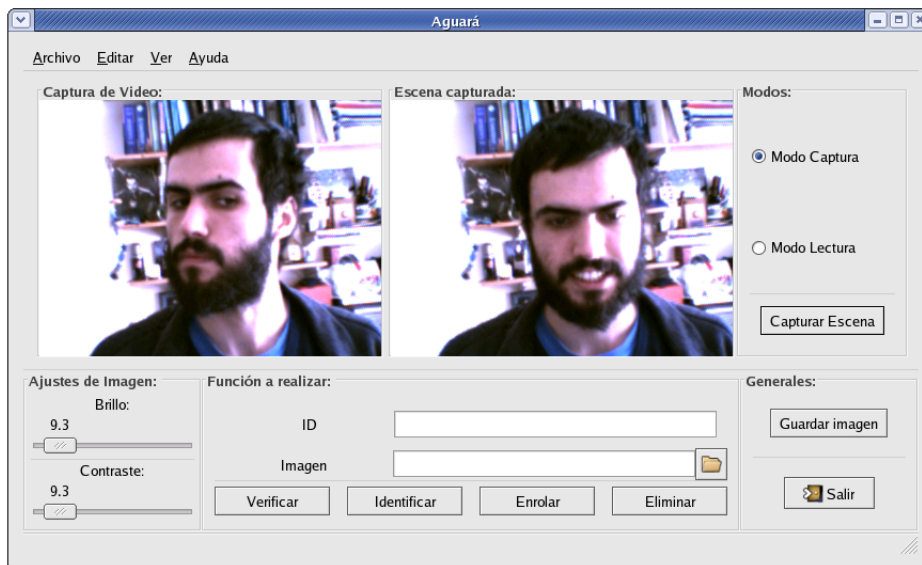


Figura B.1: Foto de ejemplo de la interfaz gráfica.

Menú Principal

El menú incluye las funcionalidades básicas de guardado en archivo de imágenes, salida del programa, información general y ayuda.

Despliegue de Video

Este módulo se encarga de desplegar en tiempo real la captura de video de un dispositivo webcam USB conectado al PC. Particularmente, además del software necesario para el funcionamiento del sistema se deberá contar con un dispositivo hardware tipo cámara *web*.

Captura de Imagen

Este módulo despliega la imagen capturada por el usuario en cierto instante de tiempo.

Modos y Botón de Captura

Se tiene dos partes:

1. Selección del Modo de funcionamiento que permite elegir entre el *Modo Captura* y el *Modo Lectura*.
2. Botón de Captura de la imagen que ordena el despliegue inmediato de la imagen que estamos observando en el cuadro de Despliegue de Video en el cuadro de Captura de Imagen.

Ajuste de Señal de Video

Este sector permite al usuario setear directamente con botones de ajuste la señal digital de video las siguientes características:

- Nivel de Brillo
- Nivel de Contraste

Funcionalidades del Sistema

Este sector consta de dos entradas de texto y cuatro botones con las funcionalidades básicas del sistema de reconocimiento: Verificación, Identificación, Enrolado y Eliminado de la Base de Datos del sistema. Las entradas de texto permiten el ingreso de la identidad y la ruta a un archivo imagen en cada caso correspondiente, como se observa en la figura B.1

Botones de Uso General

Aquí se encuentra el botón de salida y un botón que permite guardar una imagen capturada.

B.2.2. Ejemplos de Uso de la Interfaz

Ejemplo 1: Enrolado y Verificación en Modo Captura

Para trabajar en Modo Captura se debe chequear que este haya sido elegido en el sector correspondiente. Se desea enrolar a una persona con el identificador válido ID = 1234. Como primer paso la persona a enrolar debe enfrentarse a la cámara y se realiza los ajustes de la señal y de la toma correspondientes. Cuando la escena está pronta se debe capturar la imagen deseada con el botón *Capturar Escena*. A continuación se ingresa en la entrada de texto con etiqueta *ID* el identificador de la persona, en este caso es el 1234. Luego se presiona el botón *Enrolar*. A continuación el sistema procesa la información y despliega un diálogo que indica si se pudo realizar el enrolado con éxito.

Como segundo paso se quiere aplicar la verificación al individuo 1234. Para ello se repiten los pasos anteriores a excepción del último, en el que se presiona el botón *Verificar*. El sistema devuelve un cartel de diálogo mostrando el resultado del proceso de validación. En la figura B.2 se muestra el resultado de la verificación de la persona 1234.



Figura B.2: Ejemplo del resultado de verificación en Modo Captura.

Ejemplo 2: Identificación en Modo Lectura

En primer término se debe seleccionar el *Modo Lectura* en el sector de

modos. Siguiendo con el ejemplo anterior se quiere identificar al individuo 1234 en base a una imagen de archivo de este individuo. En la entrada de texto *Imagen* se selecciona el archivo imagen y a continuación se presiona el botón *Identificar*. En la figura de ejemplo B.3 se muestra el resultado de identificación que despliega el sistema finalmente.

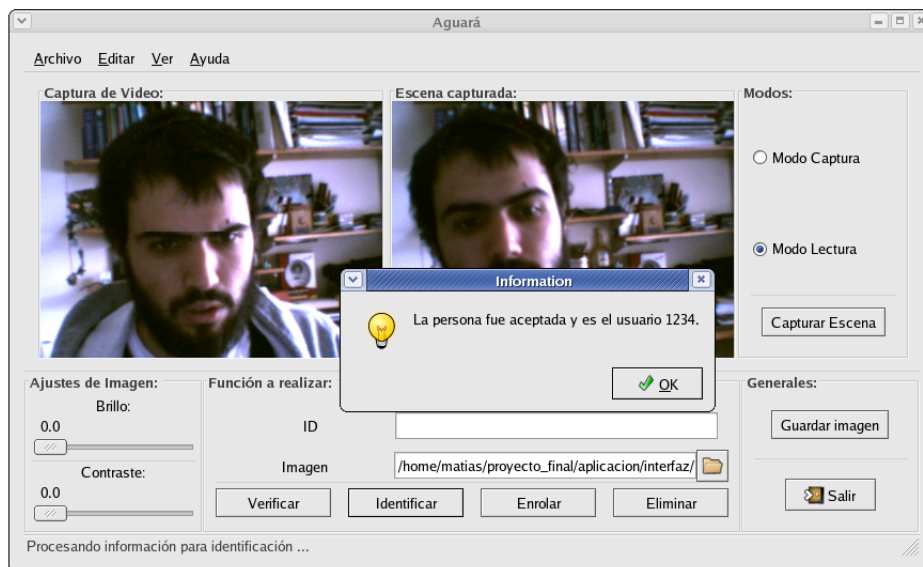


Figura B.3: Ejemplo del resultado de identificación en Modo Lectura.

B.2.3. Requerimientos del Sistema

Además de los requerimientos incluidos para la ejecución del sistema desde línea de comando se necesita para la versión gráfica los siguientes elementos:

- Biblioteca de Desarrollo Gráfico GTK v2.x.
- Biblioteca de Adquisición de video v4L (Video For Linux).
- Cámara *web* USB.

La biblioteca GTK es indispensable para que corra la aplicación en el modo que sea. Los siguientes dos requerimientos son necesarios en caso de querer trabajar en tiempo real con la adquisición de video.

Apéndice C

Bases de Caras

En este apéndice se describen las dos bases de caras principales utilizadas durante el proyecto: la Base IIE y la Base FERET.

C.1. Base IIE

Con el objetivo de sistematizar la evaluación de los algoritmos de reconocimiento de caras a implementar, se realizó una base de imágenes de caras propia. La base fue adquirida en condiciones controladas, utilizando una cámara digital de uso familiar. La adquisición fue realizada siguiendo una serie de procedimientos y recomendaciones que se detallan a continuación. Para ello se tomó como referencia [82, 83, 84].

Descripción general

- Cantidad de sujetos: 50
- Poses: frontal, medio perfil derecho e izquierdo (aproximadamente 15 grados), todos en una toma con *tilt* y en otra sin *tilt*. Los casos con *tilt* comprenden desviación de la cabeza hacia arriba y hacia abajo. En total 9 poses por sujeto. De la pose frontal se adquiere 4 imágenes, lo que hace un total de 12 fotos por sujeto.
- Iluminación: foco de luz a la derecha o izquierda de la cabeza. En total un caso de iluminación no uniforme por sujeto.
- Expresiones: feliz, sorprendido, enojado, guiñada (la expresión neutra se considera en el caso de pose frontal). En total cuatro expresiones por sujeto.
- Oclusión: utilizando lentes (comunes y de sol) o bufanda. Total de 3 oclusiones por persona.

- Tiempo: luego de una semana de la primera sesión de fotos se toman 2 imágenes frontales más y 2 correspondientes a medio perfil.
- Total imágenes por sujeto: 24
- Total imágenes de la base: 1200
- Se utilizan imágenes color.

Recomendaciones adoptadas

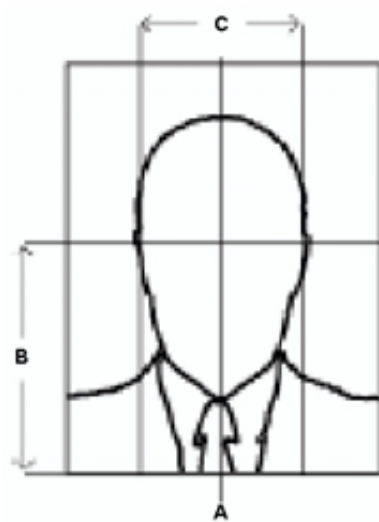


Figura C.1: Diagrama de ubicación de la cara en la imagen.

Profundidad de campo La imagen de la cara debe siempre estar en foco desde la nariz hasta las orejas. Esto puede resultar en que el fondo quede fuera de foco, pero de todas maneras lo fundamental es que la cara quede bien enfocada.

Centrado Se deben cumplir las siguientes condiciones:

- El punto medio de la boca y la punta de la nariz deben pasar por una línea vertical atravesando el centro de la imagen.
- La línea de los ojos debe pasar por una línea horizontal ubicada a una distancia de la base inferior de la imagen, equivalente al 55 % del alto de la imagen.
- El ancho de la cara del sujeto debe ocupar aproximadamente el 50 % del ancho de la imagen. Se considera el ancho de la cara como la distancia entre los dos lóbulos más alejados de las orejas.

La figura C.1 ilustra la anterior descripción.

Iluminación Las luces deben ser ubicadas de manera de reducir las sombras y las zonas de demasiada iluminación (mejillas, frente).

Especificación de la cámara utilizada

- **Cámara** HP *photosmart 735xi*
- **Compresión** JPEG *Baseline (Standard CCITT 1992)* - Calidad Media 100-110 kB
- **Tamaño** 640 x 480 píxeles
- **Color** 24 bpp (bits por píxel)

Nombre de los archivos

El nombre de los archivos está compuesto por un número (ID de usuario) y una letra que indica el tipo de foto. Por ejemplo *01a.jpg* es la foto *a* de la persona *01*. A continuación se indica la codificación utilizada:

a :	centro	m :	sonriente
b :	izquierda	n :	sorprendido
c :	derecha	o :	enojado
d :	centro	p :	guiñada
e :	arriba derecha	q :	bufanda
f :	arriba izquierda	r :	anteojos normales
g :	centro	s :	anteojos negros
h :	abajo izquierda	t :	iluminación
i :	abajo derecha	u :	2da toma centro
j :	centro	v :	2da toma izquierda
k :	arriba centro	w :	2da toma derecha
l :	abajo centro	x :	2da toma centro

Ejemplo de secuencia adquirida

Tratando de seguir las recomendaciones descritas anteriormente y teniendo en cuenta las restricciones materiales y locativas se realizó la adquisición de la base en el período Octubre-Noviembre de 2005 en el IIE. La figura C.2 muestra una de las secuencias adquiridas.

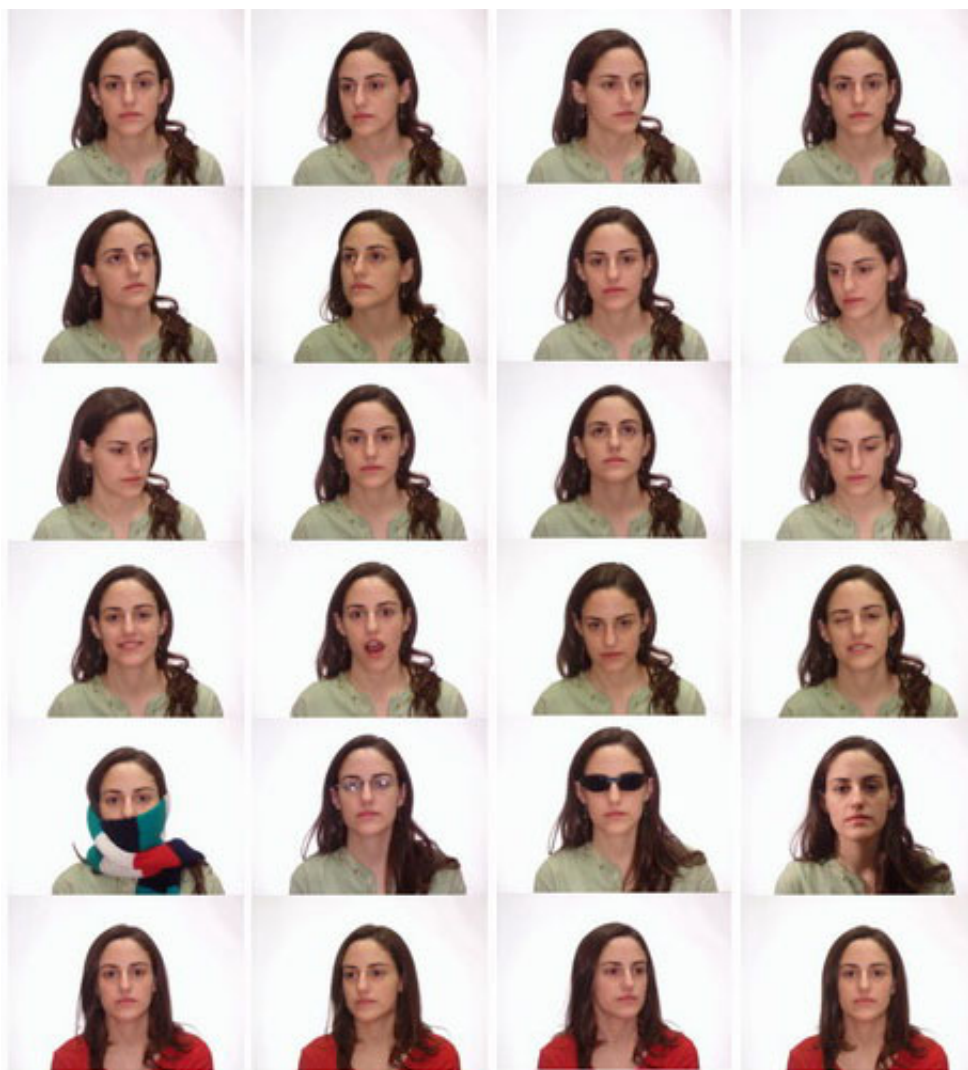


Figura C.2: Ejemplo de secuencia adquirida.

C.2. Base FERET

A los efectos de realizar pruebas con bases de mayor cantidad de individuos que la Base IIE se solicitó una copia gratuita de la base estándar FERET. Las imágenes utilizadas corresponden a la base de imágenes en escala de grises, existiendo una nueva versión de la misma con imágenes color. Por más información acerca de esta base y de cómo obtener un ejemplar de la misma ver <http://www.itl.nist.gov/iad/humanid/colorferet/home.html> [85].

El programa FERET¹ se desarrolló entre 1993 y 1997. Su misión principal era desarrollar tecnologías de reconocimiento automático de caras para ser utilizadas en aplicaciones diversas de inteligencia y seguridad.

Durante el programa se adquirió una base de imágenes para colaborar con el gobierno en tareas de evaluación de distintos algoritmos usando procedimientos estándar. El conjunto obtenido consta de 14051 imágenes en escala de gris de 8 bits de caras humanas con distintas poses, desde frontales a perfiles derechos e izquierdos, así como también distinta iluminación, expresión y diferentes momentos de adquisición. Se encuentran disponibles los resultados de las evaluaciones realizadas tanto en internet [85] como en diferentes publicaciones. [75, 74].

Es importante resaltar que esta base de caras ha sido ampliamente aceptada en la comunidad de investigadores y desarrolladores en el área. Se la puede calificar correctamente como una base estándar de facto para el análisis y la comparación de performance entre los sistemas de reconocimiento de caras.

Los distintos conjuntos de imágenes permiten evaluar un algoritmo y su robustez frente a diferentes situaciones. A continuación se listan los conjuntos de imágenes utilizados con la nomenclatura correspondiente:

fa Conjunto de base que forma la galería de imágenes. Cuenta con una imagen por sujeto. Contiene 1196 imágenes.

fb Expresión facial: conjunto de imágenes similar a **fa** tomadas en el mismo momento con diferente expresión. Contiene 1195 imágenes.

fc Iluminación: conjunto de imágenes tomadas en el mismo momento que **fa** con diferente iluminación. Contiene 194 imágenes.

Dup I Paso del tiempo: conjunto de imágenes tomadas en distinto momento que **fa**, entre un minuto a 1031 días después. Contiene 722 imágenes.

¹Facial Recognition Technology.

Dup II Paso del tiempo: corresponde a un subconjunto más difícil que DUP I, con diferencia entre tomas de un mismo individuo de más de 18 meses. Contiene 234 imágenes.

A partir de estos conjuntos se definen cuatro tests que son los que se realizan en la evaluación estándar definida con la base FERET. Cada uno de ellos consiste en utilizar el conjunto **fa** como galería, es decir las referencias que se tienen de cada individuo, y los otros conjuntos como conjunto de prueba, es decir las imágenes a clasificar.

A modo de ejemplo se presenta la imagen C.3 con las distintas parejas correspondientes a cada uno de los tests. Arriba a la izquierda tenemos el ejemplo de **fafb**, arriba a la derecha el de Dup I, abajo a la izquierda el de **fafc** y abajo a la derecha el de Dup II.



Figura C.3: Ejemplos de la base FERET.

Además de los conjuntos detallados que corresponden a diferente expresión facial, iluminación y momento de adquisición, la base FERET contiene imágenes de diferentes poses. Existen conjuntos de imágenes con rotaciones de diferentes ángulos (15° , 25° , 40° , 60°) tanto hacia la izquierda como hacia la derecha. Se especifican también conjuntos de imágenes con caras rotadas un cuarto de perfil ($\pm 22.5^\circ$), medio de perfil ($\pm 67.5^\circ$) y totalmente de perfil ($\pm 90^\circ$). Estas imágenes fueron utilizadas para el trabajo realizado

en detección de pose. En la figura C.4 se muestra un ejemplo de las imágenes utilizadas.



Figura C.4: Imágenes con rotaciones de 0°, 25°, 40° y 60° respectivamente.

C.3. Bases Estándar

Existe una serie de bases que son utilizadas comúnmente en los proyectos de reconocimiento de caras. Se compara a continuación las bases utilizadas en este proyecto con algunas de las más utilizadas en general de modo de tener una referencia de las dimensiones de las distintas bases.

La tabla C.1 muestra información asociada a diferentes bases utilizadas para la evaluación de algoritmos de reconocimiento. En particular se detalla la cantidad de individuos en la base, el número de poses, iluminaciones y expresiones faciales diferentes. La columna *Tiempo* indica la cantidad de momentos distintos en los que se adquieren imágenes de un individuo. En los casos en que no se dispone de la información necesaria se indica “++”.

Como se observa existe una amplia gama de bases, todas con características particulares enfocadas en el tipo de evaluación para la que fueron generadas. En [82] se tiene un resumen de más de 20 bases de caras estándar, detallando la utilidad de cada una de ellas.

Base de caras	Nº Indiv.	Pose	Ilum.	Expr.	Tiempo
AR	116	1	4	4	2
BANCA	208	1	++	1	12
CAS-PEAL	66-1040	21	9-15	6	2
CMU Hyper	54	1	4	1	1-5
CMU PIE	68	13	43	3	1
Equinox IR	91	1	3	3	1
FERET	1199	9-20	2	2	2
Harvard RL	10	1	77-84	1	1
IIE	60	9	2	5-8	2
KFDB	1000	7	16	5	1
MIT	15	3	3	1	1
MPI	200	3	3	1	1
ND HID	300+	1	3	2	10/13
NIST MID	1573	2	1	++	1
ORL	10	1	++	++	++
UMIST	20	++	1	++	1
U. Texas	284	++	1	++	1
U. Oulu	125	1	16	1	1
XM2VTS	295	++	1	++	4
Yale	15	1	3	6	1
Yale B	10	9	64	1	1

Tabla C.1: Información de distintas bases de caras.

Apéndice D

Contenido del CD Adjunto

Bibliografía

En la carpeta bibliografía se encuentran en versión electrónica la mayoría de las publicaciones citadas en esta documentación, así como otras publicaciones utilizadas como material adicional clasificadas por tema.

Software

Se adjuntan las herramientas utilizadas como base para el desarrollo de nuestro sistema:

- MPT [8] - *Machine Perception Toolbox*
- CSUeval [12] - *Colorado State University Face Recognition*

Biblioteca - API

La biblioteca generada puede ser utilizada como un API para desarrollar nuevas aplicaciones.

Base IIE

Base de imágenes de caras adquirida a lo largo del proyecto, junto con las coordenadas de los ojos marcadas manualmente.

Aplicación

Se encuentran disponibles dos versiones de la aplicación desarrollada: una para consola: `aguara_consola` y otra con interfaz gráfica: `aguara_gui`.

Apéndice E

Cálculos Estimación del Desplazamiento

Como se introdujo en el capítulo 6 existe un subconjunto de técnicas para la estimación del desplazamiento que se basan en una aproximación de Taylor de segundo orden de la similitud,

$$S_D(J, J', \vec{d}) \approx \frac{\sum_{j=1}^N a_j a'_j [1 - 0,5(\phi_j - \phi'_j - \vec{d} \cdot \vec{k}_j)^2]}{\sqrt{\sum_{j=1}^N a_j^2 \sum_{j=1}^N a_j'^2}}$$

Para maximizar S_D , se debe resolver $\nabla_d S_D = 0$, con lo cual se tiene:

$$\frac{\partial S_D}{\partial d_x} = \frac{\sum_{j=1}^N a_j a'_j k_{jx} (\phi_j - \phi'_j - \vec{d} \cdot \vec{k}_j)}{\sqrt{\sum_{j=1}^N a_j^2 \sum_{j=1}^N a_j'^2}} = 0$$

o lo que es lo mismo,

$$\sum_{j=1}^N a_j a'_j k_{jx} (\phi_j - \phi'_j) = \sum_{j=1}^N a_j a'_j k_{jx} (d_x k_{jx} + d_y k_{jy}) \quad (\text{E.1})$$

Mediante un razonamiento análogo pero planteando $\frac{\partial S_D}{\partial d_y} = 0$ se obtiene:

$$\sum_{j=1}^N a_j a'_j k_{jy} (\phi_j - \phi'_j) = \sum_{j=1}^N a_j a'_j k_{jy} (d_x k_{jx} - d_y k_{jy}) \quad (\text{E.2})$$

Se define:

$$\Phi_x = \sum_{j=1}^N a_j a'_j k_{jx} (\phi_j - \phi'_j)$$

$$\Gamma_{xy} = \sum_{j=1}^N a_j a'_j k_{jx} k_{jy}$$

y Φ_y , Γ_{xx} , Γ_{yx} , Γ_{yy} definidos de manera análoga.

Sustituyendo las definiciones en las ecuaciones (E.1) y (E.2) se plantea el siguiente sistema:

$$\begin{cases} \Phi_x = \Gamma_{xx}d_x + \Gamma_{yx}d_y \\ \Phi_y = \Gamma_{xy}d_x + \Gamma_{yy}d_y \end{cases}$$

Resolviendo el sistema anterior se obtiene el desplazamiento buscado,

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \frac{1}{\Gamma_{xx}\Gamma_{yy} - \Gamma_{xy}\Gamma_{yx}} \cdot \begin{pmatrix} \Gamma_{yy} & -\Gamma_{yx} \\ \Gamma_{xy} & \Gamma_{xx} \end{pmatrix} \begin{pmatrix} \Phi_x \\ \Phi_y \end{pmatrix} \quad (\text{E.3})$$

Apéndice F

Dedicación

En todo proyecto de ingeniería de magnitud considerable es necesario establecer cronogramas de plazos y planificar tareas para cumplir con los objetivos intermedios y finales del mismo. Este proyecto fue desarrollado en un lapso de 14 meses, durante los cuales lo planificado hacia comienzos del mismo fue rediscutido y repensando. En términos generales, los objetivos centrales del proyecto no se modificaron radicalmente. Sí lo hicieron los medios para lograr tales objetivos.

En las siguientes secciones se muestra la tabla original planificada en el curso de Gestión de Proyecto ¹ y la tabla de ejecución real del mismo. Se observa que tanto la duración de las tareas como los nombres de algunas de ellas han cambiado.

¹Curso de gestión dictado en el IIE.

F.1. Planificación

<i>Nº</i>	<i>Tarea</i>	<i>Duración</i>	<i>Horas</i>	<i>Fecha ini.</i>	<i>Fecha fin</i>
1	Recopilación y estudio de bibliografía	42 d	240	08/08/05	18/09/05
2	Base de imágenes: Preparación Base IIE	14 d	40	19/09/05	02/10/05
3	Base de imágenes: FERET	7 d	10	19/09/05	25/09/05
4	Base de imágenes: Adquisición Base IIE	21 d	60	03/10/05	23/10/05
5	Modelado: Búsqueda Inicial	7 d	40	24/10/05	30/10/05
6	Modelado: Estudio del Material encontrado	14 d	80	31/10/05	13/11/05
7	Modelado: Confrontación dos mejores opciones	7 d	40	14/11/05	20/11/05
8	Módulo II - Extracción de Características	60 d	360	21/11/05	02/02/06
9	Módulo III - Confrontación	14 d	200	03/02/06	16/02/06
10	Estudio de C++ y Librerías de imágenes	21 d	300	27/02/06	19/03/06
11	Recodificación en C++ de módulos II y III	14 d	240	20/03/06	02/04/06
12	Módulo I - Detección de la Cara	14 d	200	03/04/06	16/04/06
13	Ensamblado: Primera Versión punta a punta	7 d	100	17/04/06	23/04/06
14	Ajustes y Validación del Software	21 d	300	24/04/06	14/05/06
15	Depurado y Mejoras: Versión Final	21 d	300	29/05/06	18/06/06
16	Evaluación y Análisis de resultados	28 d	300	19/06/06	16/07/06
17	Interfaz gráfica y preparación Demo	14 d	100	12/07/06	23/07/06
18	Entrega final: Documentación	7 d	100	24/07/06	30/07/06

F.2. Ejecución Real

<i>Nº</i>	<i>Tarea</i>	<i>Duración</i>	<i>Horas</i>	<i>Fecha ini.</i>	<i>Fecha fin</i>
1	Recopilación y estudio de bibliografía	50 d	300	08/08/05	23/09/05
2	Gestión del Proyecto	40 d	120	01/09/05	20/10/05
3	Base de imágenes: Preparación Base IIE	7 d	50	01/09/05	07/09/05
4	Base de imágenes: Adquisición Base IIE	20 d	150	20/09/05	10/10/05
5	Base de imágenes: Carta petición FERET	7 d	10	25/09/05	31/09/05
6	Modelado: Búsqueda Inicial	10 d	40	20/10/05	01/11/05
7	Modelado: Estudio Material Encontrado	14 d	80	01/11/05	10/11/05
8	Modelado: Confrontación de Opciones	12 d	68	10/11/05	22/11/05
9	Reconocimiento de Caras: Características Globales	56 d	360	01/12/05	27/02/06
10	Reconocimiento de Caras: Características Locales	56 d	360	01/12/05	27/02/06
11	Incorporación de Implementación CSU	10 d	70	28/02/06	07/03/06
12	Módulo de Detección de Caras	14 d	100	07/03/06	21/03/06
13	Recodificación: Diseño de Librerías	30 d	430	01/04/06	01/05/06
14	Recodificación: Primera Versión Punta a Punta	14 d	100	02/05/06	16/05/06
15	Búsqueda de Alternativas de Implementación	45 d	650	02/05/06	28/06/06
16	Interfaz de Usuario + Adquisición	45 d	300	01/07/06	14/08/06
17	Evaluación Final del Sistema	31 d	220	01/08/06	01/09/06
18	Documentación Final	60 d	400	01/08/06	30/09/06

F.3. Evaluación de la Planificación y Conclusiones

En las primeras etapas que engloban la recopilación y estudio del tema Reconocimiento de Caras, la obtención de las bases de caras y el estudio del modelado, los tiempos de ejecución se respetan.

Ya en la etapa de los módulos de extracción de características y confrontación el proyecto toma otros carriles que los proyectados. Motivados por el proyecto final del curso Reconocimiento de Patrones ² es que se estudian algoritmos de extracción de características locales y globales.

Luego se incorpora la implementación CSU que permite focalizar el estudio en los distintos algoritmos y no enfocarse únicamente en la implementación. Cuando se define un sistema punta a punta se abre una etapa de recodificación en la cual se agrega un módulo de detección de caras tal como se había previsto.

Luego de la obtención de una API con librerías recodificadas se abre una etapa de profundización en la implementación punta a punta y en la búsqueda de alternativas. Aquí el objetivo pasa de hacer una implementación a optimizar la implementación obtenida.

Finalmente se trabaja fuertemente en dos aspectos: la implementación de la interfaz de usuario junto con la resolución de la adquisición por un lado y la evaluación del sistema final y sus alternativas por otro. Repasando las tareas planificadas 14, 15, 16 y 17 se entiende que en su momento no se tenía los suficientes elementos para estimar el tiempo que podía tomar cada etapa e inclusive qué significarían cada una de ellas. Por ejemplo, con la incorporación de la implementación CSU las tareas 14 y 15 pasaron a tener una importancia fundamental y se les dedicó mucho mayor trabajo. Otro ejemplo es la dedicación estimada para la codificación de la interfaz gráfica. En la práctica ésta tomó mucho más tiempo que el estimado dado que se resolvió la adquisición en tiempo real que en la planificación inicial no se tuvo en cuenta.

Finalmente, el proceso de ensamblado de la documentación final fue tal vez la actividad que fue más subestimada a la hora de la planificación. Si bien se cumplió con la política de llevar las documentaciones al día a medida que se avanzaba en las etapas del proyecto, la realimentación y la obtención del aval por parte del grupo tomó un tiempo muy considerable. Se cree igualmente que fue un tiempo bien aprovechado dado que enriquece mucho la puesta en común de los criterios y los distintos enfoques.

²Curso dictado en el IIE.

Es difícil extrapolar la planificación a la ejecución en un proyecto que fue cambiando su enfoque y modelando el problema a medida que se implementaba. De todas maneras se cree que los plazos generales se fueron cumpliendo con una tolerancia adecuada.

